

Universidade Federal do Rio de Janeiro
Instituto de Matemática
Núcleo de Computação Eletrônica

Leonardo Cunha de Miranda

RoboFácil: Especificação e Implementação de Artefatos de Hardware e Software de Baixo Custo para um Kit de Robótica Educacional

Rio de Janeiro
2006

Leonardo Cunha de Miranda

ROBOFÁCIL: ESPECIFICAÇÃO E IMPLEMENTAÇÃO DE ARTEFATOS DE
HARDWARE E SOFTWARE DE BAIXO CUSTO PARA UM KIT DE ROBÓTICA
EDUCACIONAL

Dissertação de Mestrado apresentada ao
Programa de Pós-Graduação em Informática,
Instituto de Matemática, Núcleo de Computação
Eletrônica, Universidade Federal do Rio de
Janeiro, como parte dos requisitos necessários à
obtenção do título de Mestre em Informática.

Orientador: Fábio Ferrentini Sampaio, Ph.D.

Rio de Janeiro

2006

M672

Miranda, Leonardo Cunha de
RoboFácil : especificação e implementação de artefatos de hardware e software de baixo custo para um kit de robótica educacional / Leonardo Cunha de Miranda. Rio de Janeiro : UFRJ/IM/NCE, 2006.

124 f.: il.

Dissertação (Mestrado em Informática) – Universidade Federal do Rio de Janeiro. Instituto de Matemática. Núcleo de Computação Eletrônica, 2006.

Orientador: Fábio Ferrentini Sampaio

1. Robótica educacional – Teses. 2. Hardware & Software – Teses. 3. Informática na educação – Teses. I. Sampaio, Fábio Ferrentini. II. Universidade Federal do Rio de Janeiro. Instituto de Matemática. Núcleo de Computação Eletrônica. III. Título.

CDD 371.334

Leonardo Cunha de Miranda

ROBOFÁCIL: ESPECIFICAÇÃO E IMPLEMENTAÇÃO DE ARTEFATOS DE
HARDWARE E SOFTWARE DE BAIXO CUSTO PARA UM KIT DE ROBÓTICA
EDUCACIONAL

Rio de Janeiro, 25 de abril de 2006.

Aprovada por:

Prof. Fábio Ferrentini Sampaio, Ph.D., UFRJ/NCE – Orientador

Prof. João Vilhete Viegas d'Abreu, D.Sc., UNICAMP/NIED

Prof. Francisco Artur Braun Chaves, Ph.D., UFRJ/IF

Prof. Marcos da Fonseca Elia, Ph.D., UFRJ/NCE

Dedicatória

Dedico esse trabalho aos professores que se esmeram em investigar e aplicar novas alternativas didáticas para apoiar a prática do dia-a-dia de sala de aula e romper com alguns dogmas do tradicional processo educacional em voga nas escolas de nosso País. Ofereço, também, aos estudantes ávidos por conhecimento, que conduzem seus estudos, sistematicamente ou não, com afino e dedicação.

Agradecimentos

Muitas são as pessoas que contribuíram de forma direta e/ou indireta para a realização desse trabalho.

Primeiramente, gostaria de registrar meus agradecimentos aos professores que nortearam a minha trajetória acadêmica durante o mestrado, e especialmente pelo direcionamento constante e sempre presente dessa pesquisa, os meus orientadores, Prof. Ph.D. Fábio Ferrentini Sampaio e Prof. Doutorando José Antonio dos Santos Borges.

Ao M.Sc. Eng. Diogo Fujio Takano, pelas dicas sobre a arquitetura eletrônica do *hardware* concebido.

Ao Prof. Ph.D. Carlo Emmanuel Tolla de Oliveira pela didática, ao apresentar os conceitos e as aplicabilidades, que fomentou a Orientação a Objetos em minha mente.

Aos meus alunos que, por muitas vezes durante o período do mestrado, exteriorizaram e expressaram suas idéias e opiniões acerca desse trabalho, contribuindo com diferentes tópicos indagados nessa pesquisa.

Agradeço, também, ao meu *little baby* pelo apoio realizado durante essa pesquisa, pois sem ele muitas das tarefas do mestrado não teriam sido realizadas.

Ao meu Gerente na Prefeitura da Cidade do Rio de Janeiro (PCRJ), o Sr. Antonio Fernando Lucas Barbosa pelas inumeráveis demonstrações de colaboração à pesquisa e atos inerentes à mesma.

Ao meu parceiro de trabalho na PCRJ, o M.Sc. Luiz Antonio Alves de Oliveira, que por muitas vezes contribuiu com questões pertinentes a execução dessa pesquisa.

Ao ex-Diretor da Diretoria de Recursos Humanos, o Sr. Sérgio Henrique Rodrigues Pereira, o ex-Assessor de Gestão e Qualidade Interna, o Sr. José Mariano, a Diretora de Recursos Humanos, a Sra. Genilse Pinheiro de Souza e a Chefe do Setor de Pessoal Efetivo, a Sra. Maricélia Silva Alves, todos da Fundação de Apoio a Escola Técnica do Estado do Rio de Janeiro (FAETEC) pela flexibilização das atividades pedagógicas durante a realização do mestrado.

Ao Prof. M.Sc. César Augusto Rangel Bastos e a direção do Colégio Notre Dame (Rio de Janeiro – RJ) pela colaboração para com essa pesquisa ao conceder visita ao laboratório de robótica desse Colégio, tornando factível uma experiência prática do ensino mediado pela robótica.

A Profa. M.Sc. Maria do Rosário Paim de Santana, Prof. Mario Sergio Mattos de Almeida e a direção do Colégio Nobel (Salvador – BA) que permitiram a minha visita ao laboratório de robótica do referido Colégio, tornando possível a realização de observações e interação direta com o público alvo dessa pesquisa.

A Rafael Cunha de Miranda pelas colaborações lingüísticas acerca dessa dissertação.

E, por último, a Erica Esteves Cunha pelo apoio cotidiano que, substancialmente, influenciou a minha chegada, com êxito, ao final desse percurso da minha vida acadêmica.

“O intelecto, deixado a si mesmo, na mente sóbria, paciente e grave, sobretudo se não está impedida pelas doutrinas recebidas, tenta algo na outra via, na verdadeira, mas com escasso proveito. Porque o intelecto não regulado e sem apoio é irregular e de todo inábil para superar a obscuridade das coisas.” (BACON, 2005, p. 36)

Resumo

MIRANDA, L.C. **RoboFácil: Especificação e Implementação de Artefatos de Hardware e Software de Baixo Custo para um Kit de Robótica Educacional**. Rio de Janeiro – RJ. 2006. 124 f. Dissertação (Mestrado em Informática) – Instituto de Matemática, Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro, Rio de Janeiro – RJ, 2006.

O presente trabalho tem como objetivo a proposição e implementação de um kit de robótica educacional de baixo custo, a fim de facilitar uma maior utilização de tal ferramental nas escolas brasileiras. Para alcançar tais objetivos, foram realizados estudos comparativos entre distintos kits comercializados no mercado nacional. Tais estudos focaram, principalmente, o custo de obtenção e seus recursos de hardware e software, proporcionando um maior entendimento das necessidades de projeto e o seu aproveitamento em sala de aula, viabilizando a concepção e o desenvolvimento do kit de robótica educacional denominado RoboFácil.

Abstract

MIRANDA, L.C. **RoboFácil: Especificação e Implementação de Artefatos de Hardware e Software de Baixo Custo para um Kit de Robótica Educacional.** Rio de Janeiro – RJ. 2006. 124 f. Dissertação (Mestrado em Informática) – Instituto de Matemática, Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro, Rio de Janeiro – RJ, 2006.

This work aims the proposal and the implementation of a low cost educational robotics kit in order to facilitate the use of this tool in brazilian schools. To reach such objectives, comparative studies between different kits, commercialized in the national market, had been carried out. Such studies focused mainly on the acquisition cost and resources of the hardware and software. These studies gave us important insight into the specification and implementation of the educational robotics kit called RoboFácil.

Lista de Tabelas

Tabela 1 – Comparação entre quatro kits de robótica educacional.....	40
Tabela 2 – Rótulo, propriedade, valores e testes lógicos dos objetos de hardware de saída da janela Meu Mundo	50
Tabela 3 – Rótulo e testes lógicos dos objetos de hardware de entrada da janela Meu Mundo	50
Tabela 4 – Rótulo e ações dos objetos de hardware de saída da janela Meu Programa	52
Tabela 5 – Rótulo e finalidade dos objetos de programação da janela Meu Programa	52
Tabela 6 – Macro-códigos assembly do objeto Led (códigos iniciados com 1).....	58
Tabela 7 – Macro-códigos assembly do objeto Display (códigos iniciados com 2)	58
Tabela 8 – Macro-códigos assembly do objeto Motor (códigos iniciados com 3)	59
Tabela 9 – Macro-códigos assembly do objeto Temporizador (códigos iniciados com 4)	59
Tabela 10 – Macro-códigos assembly do objeto Sensor de Luminosidade (códigos iniciados com 5).....	59
Tabela 11 – Macro-códigos assembly do objeto Sensor de Temperatura (códigos iniciados com 6).....	59
Tabela 12 – Macro-códigos assembly dos códigos de controle de execução (códigos iniciados com 0).....	59
Tabela 13 – Instruções interpretadas pelo firmware.....	74
Tabela 14 – Sequência de instruções mínimas para se criar um looping finito	75
Tabela 15 – Sequência de instruções mínimas para se criar um looping infinito	76
Tabela 16 – Sequência de instruções mínimas para se criar uma estrutura condicional Se	76
Tabela 17 – Valores limites dos recursos utilizados no assembly virtual do hardware RoboFácil.....	76
Tabela 18 – Mapa de endereços de memória.....	91
Tabela 19 – Mapa de endereços de I/O.....	92
Tabela 20 – Custo detalhado de um hardware RoboFácil.....	94

Lista de Figuras

Figura 1 – Kit Robótica Fácil. (1) Placa principal; e (2) Placa de relés.....	34
Figura 2 – Kit Super Robby. (1) Interface Super Robby; (2) Fonte de alimentação; e (3) Motor.....	35
Figura 3 – GoGo Board.....	36
Figura 4 – Lego MindStorms. (1) RCX; (2) Transmissor infravermelho para RCX; (3) Caixa com peças Lego; e (4) Revista com material pedagógico.....	38
Figura 5 – RCX, motores e sensores. (1) RCX; (2) Motores; (3) Sensores de toque; e (4) Sensor de luz.....	38
Figura 6 – Objeto Comentário (7) e Objetos de Hardware disponíveis na barra de ferramenta da janela Meu Mundo. (1) Led; (2) Display; (3) Lâmpada; (4) Motor; (5) Sensor de Luminosidade; e (6) Sensor de Temperatura.....	49
Figura 7 – Objeto Comentário (12), Objetos de Hardware de Saída – (1) Led; (2) Display; (3) Lâmpada; e (4) Motor – e Objetos de Programação – (5) Temporizador; (6) Estrutura de Controle Se; (7) Início de Looping; (8) Fim de Looping; (9) Linha de Programação; (10) Início de Programa; e (11) Fim de Programa – disponíveis na barra de ferramenta da janela Meu Programa.....	51
Figura 8 – Exemplo de um modelo de comportamento no ProgameFácil em simulação. Na área (1) é apresentado o “Meu Mundo” e na área (2) o “Meu Programa”.....	55
Figura 9 – Botões de controle do simulador. (1) Iniciar a simulação do Meu Mundo/Continuar com a simulação no Meu Mundo; (2) Pausar a simulação do Meu Mundo; (3) Realizar próximo passo da simulação do Meu Mundo; e (4) Interromper a simulação do Meu Mundo.....	56
Figura 10 – Representação binária das instruções do assembly virtual do hardware RoboFácil.....	58
Figura 11 – Botão que ativa o processo de compilação.....	60
Figura 12 – Compilação do exemplo de modelo de comportamento da Figura 8. (1) Assembly amigável com comentários; (2) Assembly que deverá ser interpretado pelo hardware RoboFácil; (3) Informações adicionais de compilação; e (4) Botão que inicia a transferência do assembly virtual para o hardware RoboFácil.....	61
Figura 13 – Botão de transferência do firmware do RoboFácil presente na ferramenta ProgameFácil.....	69
Figura 14 – Camadas do firmware do RoboFácil.....	70
Figura 15 – Níveis funcionais entre o software ProgameFácil e o firmware do hardware RoboFácil.....	72
Figura 16 – Interface textual fornecida pelo firmware.....	77
Figura 17 – Descrição dos pinos do microcontrolador Intel 8051.....	82
Figura 18 – Display e Teclado. (1) Botão que comanda o início da execução.....	84

Figura 19 – Plugin de controle dos motores de passo. (1) Placa de circuito do plugin; e (2) Motores de passo	88
Figura 20 – Plugin de controle dos leds. (1) Placa de circuito do plugin; e (2) Oito leds	89
Figura 21 – Plugin de controle do sensor de luminosidade. (1) Placa de circuito do plugin; e (2) LDR	90
Figura 22 – Plugin de controle do sensor de temperatura. (1) Placa de circuito do plugin; e (2) NTC	91
Figura 23 – Hardware RoboFácil atual	102

Lista de Abreviaturas

ASCII – American Standard Code for Information Interchange

BIOS – Basic Input-Output System

CPU – Central Processing Unit

CR – Carriage Return

DC – Direct Current

EPROM – Erasable Programmable Read-Only Memory

GINAPE – Grupo de INformática APlicada à Educação

HI – High

HMI – Human-Machine Interaction

IDE – Integrated Development Environment

IM – Instituto de Matemática

LCD – Liquid Crystal Display

LDR – Light Dependent Resistor

LED – Light-Emitting Diode

LO – Low

MB – Megabyte

MCU – Micro Controller Unit

MDI – Multiple Document Interface

MIT – Massachusetts Institute of Technology

NCE – Núcleo de Computação Eletrônica

NI – National Instruments

NTC – Negative Temperature Coefficient

PC – Personal Computer

RAD – Rapid Application Development

RAM – Random Access Memory

RCX – Robotic Command eXplorer

ROM – Read-Only Memory

TIC – Tecnologia da Informação e da Comunicação

UFRJ – Universidade Federal do Rio de Janeiro

V – Volts

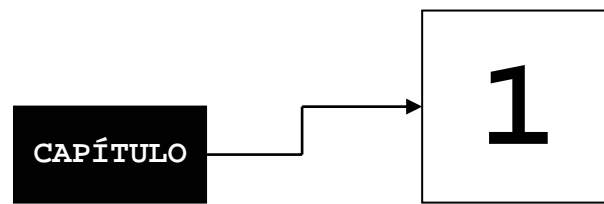
Sumário

SEÇÃO I	17
1 INTRODUÇÃO	18
1.1 Origem do Trabalho	19
1.2 Objetivo da Pesquisa	22
1.3 Relevância do Estudo e Contribuições Esperadas.....	22
1.4 Estrutura do Trabalho	23
2 REVISÃO DA LITERATURA	25
2.1 Novas Tecnologias na Educação	26
2.2 Robótica Educacional.....	27
2.3 Robótica na Sala de Aula	29
2.4 Kits de Robótica Educacional	32
2.4.1 Robótica Fácil	34
2.4.2 Super Robby.....	35
2.4.3 GoGo Board.....	36
2.4.4 Lego MindStorms.....	37
2.4.5 Análise Comparativa	39
2.5 Dificuldades Impostas	40
SEÇÃO II	43
3 PROGRAMEFÁCIL	44
3.1 Motivações e Justificativas.....	45
3.2 ProgrameFácil.....	47
3.2.1 Definição	47
3.2.2 Objetos	48
3.2.3 Meu Mundo.....	49
3.2.4 Meu Programa	50
3.2.5 Modelo de Comportamento (Modelo).....	52
3.2.6 Programa do Modelo (Programa)	53
3.2.7 Simulador.....	55
3.2.8 Compilador	57
3.2.9 Hardware Perfeito	61
3.2.10 Interface	62
3.2.11 Direitos Autorais/Custo.....	64
4 FIRMWARE	66
4.1 Motivações e Justificativas.....	67
4.2 Firmware	69
4.2.1 Loader (Carregador).....	73
4.2.2 Interpretador.....	73
4.2.3 Interface.....	77
4.2.4 Direitos Autorais/Custo	78
5 ROBOFÁCIL	79

5.1	Motivações e Justificativas.....	80
5.2	RoboFácil.....	81
5.2.1	Arquitetura Geral	81
5.2.2	Unidade de Controle e Sistema de Memória.....	82
5.2.3	Subsistemas Internos	83
5.2.3.1	Comunicações	83
5.2.3.2	Teclado (Botoeira).....	84
5.2.3.3	Mostrador (Display).....	85
5.2.3.4	Conversor Digital-Analógico (D/A)	85
5.2.3.5	Conversor Analógico-Digital (A/D)	86
5.2.4	Interfaces Genéricas de Plugins	86
5.2.4.1	Técnicas para Conexão de Interfaces Plugins.....	86
5.2.4.2	Plugin de Controle dos Motores de Passo	87
5.2.4.3	Plugin de Controle de Motores DC.....	89
5.2.4.4	Plugin de Controle de Lâmpadas, Relés ou Leds	89
5.2.4.5	Plugin de Controle do Sensor de Luminosidade.....	90
5.2.4.6	Plugin de Controle do Sensor de Temperatura	90
5.2.5	Endereçamentos	91
5.2.6	Alimentação Elétrica.....	92
5.2.7	Conector Externo.....	93
5.2.8	Interface.....	93
5.2.9	Custo	94
	SEÇÃO III.....	97
6	CONCLUSÃO	98
6.1	Considerações Finais	99
6.2	Peculiaridades dos Produtos Resultantes.....	101
6.3	Trabalhos Futuros.....	102
6.3.1	Projeto RoboFácil	103
6.3.2	Software ProgrameFácil	103
6.3.3	Firmware.....	104
6.3.4	Hardware RoboFácil.....	105
	REFERÊNCIAS BIBLIOGRÁFICAS.....	107
	APÊNDICES.....	114
	Apêndice A – Esquemas dos Circuitos Eletrônicos.....	115
	Apêndice A.1 – Circuito Eletrônico Principal.....	115
	Apêndice A.2 – Display e Portas Paralelas.....	116
	Apêndice A.3 – Porta Serial e Teclado	117
	Apêndice A.4 – Conversores DAC e ADC	118
	Apêndice A.5 – Plugin de Controle dos Motores de Passos e Sensores	119
	Apêndice A.6 – Plugin de Controle dos Leds e Sensores.....	120
	Apêndice A.7 – Cabo Serial.....	121
	ANEXOS	122
	Anexo A – Tabela ASCII Standard.....	123
	Anexo B – Mapa dos registradores da RAM interna do microcontrolador 8051	124

SEÇÃO
I

A primeira seção desse trabalho reúne os objetivos gerais e específicos que motivaram a realização dessa pesquisa em sua totalidade, os pilares bibliográficos onde essa se fundamentou, bem como o seu detalhamento, permitindo a identificação de seu escopo, relevância e contribuições à sociedade de um modo geral.



INTRODUÇÃO

O Capítulo 1 traz em seu corpo a apresentação da pesquisa desenvolvida, seus objetivos gerais e específicos, seu escopo, sua relevância e as contribuições esperadas. Apresenta, ainda, a organização textual dessa dissertação.

1.1 Origem do Trabalho

A utilização de artefatos tecnológicos e a implementação de metodologias apropriadas para esses artefatos – muitos desses constituídos da fusão das mais variadas tecnologias da engenharia mecatrônica com a ciência da computação – na educação podem viabilizar a criação de ambientes de ensino diferenciados e inovadores, possibilitando ao discente a reflexão sobre a própria aprendizagem.

Em especial, nos últimos anos, presencia-se um aumento da utilização da robótica educacional como ferramenta pedagógica em sala de aula de algumas escolas brasileiras. Tal constatação é verificável, por exemplo, pelo aumento do número de artigos publicados em congressos e periódicos (*journals*), dissertações de mestrado e teses de doutorado que exploram essa temática em toda sua amplitude, a partir da segunda metade dos anos 90 e dos primeiros anos do século XXI.

A real extensão do emprego da robótica aplicada à educação no Brasil é dada ao observar que diversas universidades brasileiras constituem projetos nessa área (ROBOTICANDO, 2004; SIROS, 2004) e empresas nacionais se esmeram para desenvolver produtos (SUPER ROBBY, 2004; CYBERBOX, 2004). Empresas estrangeiras, que possuem interesses mercadológicos, buscam constituir representantes no Brasil para distribuir seus produtos, tal como a EDACom Tecnologia em Sistemas de Informática Ltda. (EDACOM, 2005), representante exclusiva no Brasil da LEGO Educational Division (LEGO, 2005a), divisão educacional do grupo LEGO (LEGO, 2005b), que comercializa o LEGO MindStorms (LEGO, 2005c).

Tendo em vista o relato e as inúmeras vantagens do emprego da robótica pedagógica no ambiente escolar já assinaladas por diferentes autores (CHELLA, 2002b; STEFFEN, 2002; ZILLI, 2004), acredita-se que – pelo menos – dois problemas ainda impedem a expansão do emprego da robótica educacional por professores do ensino médio nas escolas brasileiras: (1) o

custo de aquisição de kits de robótica educacional; e (2) a limitação tecnológica de alguns produtos.

Fazendo uso do conhecimento prévio desses dois problemas, foram delineadas duas hipóteses como alicerce para essa pesquisa: (1) É possível desenvolver um kit de robótica educacional a um baixo custo, e que não seja limitado nos seus recursos de *hardware* e *software* (funcionamento autônomo¹ e (re)programado por *software*, respectivamente); e (2) Um *software* para robótica educacional que simule na tela do monitor o funcionamento de um *hardware* de robótica facilitará o emprego dessa ferramenta pedagógica nas escolas brasileiras.

Assim sendo, pode-se dizer que a presente dissertação é a consolidação dos resultados das explorações que vêm sendo realizadas desde o ano de 2001 pelo Grupo de Informática Aplicada à Educação (UFRJ/IM-NCE/GINAPE) no sentido de solucionar os problemas acima citados.

As pesquisas nessa área despontaram quando o então mestrando José Henrique Gandra e o doutorando José Antonio dos Santos Borges projetaram e implementaram, com o apoio do M.Sc. Eng. Diogo Fujio Takano, um *hardware* genérico que possuía uma arquitetura eletrônica voltada ao desenvolvimento de aplicações pedagógicas.

A interação entre o usuário e esse artefato tecnológico era realizada através da escrita de programas, utilizando um editor de texto padrão ASCII, na linguagem *assembly* adotada pelo microcontrolador (MCU) empregado no referido *hardware*². Os programas desenvolvidos eram convertidos da linguagem *assembly* para o formato Intel HEX (INTEL, 1988) através do uso de três diferentes *softwares*³ da Intel Corporation (INTEL, 2005a). Depois de concluída a conversão, o programa era transferido para o *hardware* através de uma interface serial do microcomputador,

¹ Entende-se, nessa pesquisa, funcionamento autônomo como a característica que um hardware pode ter de não necessitar ficar conectado ao microcomputador durante sua execução, mas apenas no momento da transferência de dados entre esses dispositivos.

² O microcontrolador adotado no *hardware* é o Intel 8031AH. O mesmo será tratado com detalhes no Capítulo 5.

³ Os programas desenvolvidos na linguagem *assembly* eram convertidos para código-objeto através do software MCS-51 Macro Assembler versão 2.2. Posteriormente, o código-objeto era linkeditado pelo programa MCS-51 Relocator and Linker versão 3.0, para finalmente, gerar o programa no formato Intel HEX, utilizando o software OH versão 1.0.

fazendo uso do *software* HyperTerminal⁴ (HILGRAEVE, 2005). Logo após o recebimento do “binário” de um programa pelo *hardware*, ainda era necessário o envio de uma instrução – “/8000” (endereço inicial, em hexadecimal, de execução da memória RAM) – para que o *hardware* iniciasse a execução do programa previamente transferido.

Devido à carência de uma linguagem mnemônica textual e/ou visual, na língua portuguesa, como artefato facilitador de interação entre o usuário e o *hardware* em questão, leva-se a crer que o seu emprego pelos estudantes adolescentes do ensino médio, público alvo dessa pesquisa, não seria trivial; e, para uma parcela significativa, impossível de ser realizada, pois existiria a necessidade dos alunos conhecerem previamente a arquitetura eletrônica adotada no projeto desse *hardware*. Tal conhecimento é fundamental para realização de acesso direto aos endereços de memória do microcontrolador, permitindo o manuseio dos diversos recursos eletrônicos, bem como a programação do equipamento através do uso da linguagem *assembly*.

Essa forma quase que rudimentar de interação com o usuário, nos dias atuais, é antagônica às linhas de pensamento contemporâneo sobre interfaces amigáveis de *software* (friendly environment), onde o fator humano é um dos mais importantes atributos que deve ser considerado durante o projeto de uma interface (QUEIROZ, L.R. et al., 1998).

Sabendo-se também do porquê e das potencialidades do emprego da robótica na educação já assinaladas por díspares autores e que, segundo Chella (2002b, p. 25),

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.

⁴ HyperTerminal é um programa produzido pela Hilgraeve, Inc. que permite a comunicação através das interfaces seriais do microcomputador. Tal software está nativamente incorporado aos sistemas operacionais para *desktop* da Microsoft, desde o Windows 95.

Nesse contexto, essa dissertação inicia-se com a necessidade de aprimorar essa primeira solução – *hardware* genérico, no sentido de ampliar as suas possibilidades de uso nas escolas brasileiras.

1.2 Objetivo da Pesquisa

Desenvolver um kit de robótica educacional, visando o baixo custo, que deverá atender às peculiaridades do emprego dessa ferramenta pedagógica nas escolas brasileiras, e que tem como seu público alvo os adolescentes das instituições educacionais de ensino médio da iniciativa pública e privada.

Visando alcançar o desenvolvimento pretendido foram traçados os seguintes passos intermediários que serão realizados na ordem abaixo:

- 1) Analisar os custos envolvidos na aquisição de kits de robótica educacional por escolas brasileiras (públicas e privadas);
- 2) Verificar os recursos técnicos de *hardware* e *software* existentes, com o intuito de apontar a limitação de cada kit em diferentes exemplos de projetos pedagógicos que façam uso da robótica; e
- 3) Especificar e implementar os *hardwares* ainda necessários e os *softwares* que são imprescindíveis à implementação do kit de robótica educacional que se tenciona produzir.

1.3 Relevância do Estudo e Contribuições Esperadas

Devido à característica multidisciplinar do emprego da robótica na educação, diferentes autores de variadas áreas da ciência, tais como engenharia, informática, comunicação, pedagogia e educação, têm proposto trabalhos buscando a geração de soluções que explorem aplicações tanto

na vertente tecnológica como na pedagógica, visando subsidiar a expansão da utilização da robótica educativa pelos estudantes brasileiros.

No que se refere especificamente a kits de robótica educacional, poucos são os produtos que, disponíveis para comercialização no Brasil, possuem uma razoável relação “recursos X custo”. O uso desses kits é dificultado pelo fato dos modelos importados terem, normalmente, um elevado custo para os padrões nacionais e os fabricados no Brasil possuem limitações de *hardware* e/ou *software* para algumas aplicações, tais como a impossibilidade de permanecerem desconectados do microcomputador durante sua execução e de não terem seu comportamento alterado – (re)programado – por *software*.

A relevância dessa pesquisa está na fusão de diferentes conhecimentos teóricos das distintas áreas da engenharia e ciência da computação que, integradas, proporcionarão a implementação de um kit nacional composto de *hardware* e *software* aplicado à robótica educacional. A descrição formal dos conceitos abstraídos e aplicados no decorrer desse trabalho representa contribuição significativa dessa pesquisa, pois serão esses elementos o alicerce do desenvolvimento de uma arquitetura escalonável de *hardware* e modular de *software*, traduzidos no Kit RoboFácil.

O RoboFácil e suas características o fazem diferente da maioria das soluções fabricadas no mercado brasileiro. O Kit visa atender a um público carente de alternativas nessa área – alunos do ensino médio. Não descartando, todavia, outros segmentos do ensino diferentes do público alvo dessa pesquisa, tais são as particularidades que serão expostas em detalhes no decorrer dessa dissertação.

1.4 Estrutura do Trabalho

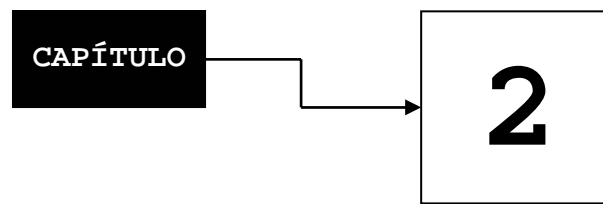
Esse trabalho está estruturado em três seções. A primeira seção está composta pelos Capítulos 1 e 2. No Capítulo 1 foi apresentado o objetivo deste trabalho, como também um

detalhamento da pesquisa, de forma a permitir identificar seu escopo, relevância e contribuições. No Capítulo 2 será realizada uma revisão da literatura da área, discutindo aspectos relevantes de educação e robótica.

A segunda seção desse trabalho será composto pelos Capítulos 3, 4 e 5. No Capítulo 3 será explorada a ferramenta educacional implementada denominada ProgrameFácil. No Capítulo 4 será exposto o *firmware* do qual o *hardware* RoboFácil está fazendo uso. E, finalmente, no Capítulo 5 serão apresentadas as questões relacionadas ao desenvolvimento, propriamente dito, do *hardware* do kit de robótica.

A última seção dessa dissertação é composta pelo Capítulo 6, onde serão tecidas as considerações finais e apresentadas algumas propostas de trabalhos futuros relevantes à continuidade do projeto de ação pedagógica vinculado a essa pesquisa.

Encontram-se ainda presentes nessa dissertação 64 referências bibliográficas que formaram o alicerce para a realização dessa pesquisa em toda sua extensão, além de um apêndice e dois anexos que detalham em maior profundidade alguns tópicos expostos no decorrer dos capítulos.



REVISÃO DA LITERATURA

Esse capítulo relata uma visão geral relacionada ao estado da arte da robótica educacional no Brasil. Com esse fim, pesquisas bibliográficas foram realizadas possibilitando a avaliação comparativa entre distintas ferramentas – kits de robótica educacional – disponíveis no mercado nacional.

2.1 Novas Tecnologias na Educação

Há algumas décadas, diferentes teóricos discutem a inserção das mais diversificadas Tecnologias da Informação e da Comunicação (TICs) no ambiente educacional. Busca-se estabelecer, com essas novas tecnologias, uma melhor qualidade do ensino e ambientes de aprendizagem mais ricos e motivadores para os discentes. Como exemplo de aplicação, d'Abreu e Chella (2003, p. 3) sinalizam que “[...] a utilização das ferramentas de hardware e software têm possibilitado desenvolver metodologias de ensino-aprendizagem baseadas no uso do computador [...]”.

Dentre o amplo espectro de idéias apresentado por esses autores, no que se refere aos artefatos computacionais, é notório observar que as soluções apresentadas, em sua maior parte, exploram apenas a vertente do *software*. No entanto, a demanda por novos aparatos tecnológicos de *hardware* na educação vem crescendo, sendo evidenciada, sobretudo, pelos esforços da comunidade acadêmica em propor a inserção da robótica com fins pedagógicos.

Segundo Schons, Primaz e Wirth (2004, p. 3),

[...] sabemos que os tempos mudaram e se faz necessário uma reflexão sobre essa prática de ensino tradicional. O conhecimento adquirido deve continuar sendo passado a posterioridade, uma vez que o progresso não se conquista sem conhecimentos anteriores às descobertas atuais e futuras. Porém, esse mesmo progresso só se efetiva com a flexibilização do conhecimento, que implica no raciocínio lógico do educando, sobre o que é ministrado. É é nisto que caracteriza a robótica educativa, que visa ser um elemento motivador no desenvolvimento cognitivo do aluno, procurando estabelecer um ambiente de trabalho escolar agradável, no qual se simula uma série de acontecimentos, muitas vezes da vida real, para assim obter os fins específicos desejados.

Portanto, deduz-se que a aplicação de novas tecnologias na educação não deve ser realizada de forma autocrática sobre os diferentes atores do processo educacional – docentes e discentes – mas, principalmente, de maneira reflexiva e coerente com a epistemologia da educação. Cogita-se, assim, o uso dessas tecnologias como instrumento de apoio à construção do conhecimento e desenvolvimento de habilidades e competências específicas por parte dos aprendizes.

Alves, Blikstein e Lopes (2005, p. 2.596), afirmam: “[...] o fundamental, cada vez mais, é saber usar as novas tecnologias como ferramentas de construção de conhecimento – e não simplesmente de coleta e resenha de informações, como tem sido a prática comum nas escolas.”

Para Oliveira Netto (2004, p. 36),

É necessária, também, uma cuidadosa reflexão por parte de todos que compõem a comunidade escolar, para que a tecnologia possa de fato contribuir para a formação de indivíduos competentes, críticos, conscientes e preparados para a realidade em que vivem. Necessariamente, o uso das novas tecnologias de informação e comunicação na escola está vinculado a uma concepção de ser humano e mundo, de educação e seu papel na sociedade moderna.

Dessa forma propõem-se, como alternativa viável, aulas teóricas e práticas balanceadas objetivando a solução dos desafios propostos pelos educadores e sendo o aluno apoiado por um ambiente de robótica educacional.

2.2 Robótica Educacional

Adotou-se, no âmbito desse trabalho, o conceito de robótica educacional⁵ definido por d’Abreu (1999, p. 1). Segundo esse autor,

[...] a robótica pedagógica pode ser definida como a utilização da robótica industrial num contexto onde as atividades de construção e controle de dispositivos, usando kits de montar ou outros materiais, propicia o manuseio de conceitos de ciências em geral num ambiente de sala de aula, nos diferentes níveis de ensino.

Ocupa-se de uma atividade desafiadora e lúdica, que utiliza o esforço do educando na criação de soluções, sejam essas compostas por *hardware* e/ou *software*, visando a resolução de um problema proposto – podendo o mesmo ser real. O aluno, nesse processo de construção, tem nos momentos de introspecção do modelo a ser criado para solucionar o problema, a oportunidade de participar de uma aprendizagem mais efetiva e desenvolver uma percepção mais acurada dos fenômenos científicos.

⁵ Nesse trabalho, o significado dos termos “robótica educativa” e “robótica pedagógica” são considerados sinônimos de “robótica educacional”.

A construção das soluções fazendo uso da robótica pedagógica pode envolver a utilização de: (1) kits de robótica educacional pré-fabricados; e (2) dispositivos eletrônicos/eletromecânicos que, usados de maneira integrada e contextualizada, formam uma proposta de solução para um problema. Porém, nesse trabalho, explora-se apenas as soluções que fazem uso de kits de robótica pré-fabricados com fins educativos, visto que a maior parte dos projetos desse gênero, no Brasil, fazem uso desses artefatos.

Podem-se citar algumas vantagens com a adoção de kits de robótica educacional existentes no mercado em geral: (1) *hardware* e *software* construído direcionado a atender propósitos educacionais específicos; (2) flexibilidade na forma de utilização em distintas aplicações; (3) existência de documentação técnica e manual do usuário dos produtos, incluindo, em alguns casos, material pedagógico de apoio às atividades do professor em sala de aula; e (4) maior facilidade para aquisição e operação por usuários não familiarizados/especializados com as tecnologias da eletrônica e/ou informática. Cita-se, algumas desvantagens da utilização de kits de robótica pré-fabricados: (1) alto custo de aquisição de alguns produtos; e (2) limitação tecnológica de alguns kits.

Uma boa parte dos projetos educacionais que utilizam a robótica pedagógica, fazem uso da teoria construtivista de Jean Piaget, e segundo Schons, Primaz e Wirth (2004, p. 5), “[...] constitui nova ferramenta que se encontra à disposição do professor, por meio da qual é possível demonstrar na prática muitos dos conceitos teóricos, às vezes de difícil compreensão, motivando tanto o professor como principalmente o aluno.”.

Sendo assim, na robótica pedagógica, segundo Schons, Primaz e Wirth (2004, p. 6), “[...] o educador deve deixar por completo a idéia das escolas dogmáticas behaviorista (Skinner) e passar a usar uma teoria construtivista, sendo um facilitador.”.

A robótica educacional pode desenvolver as seguintes competências, segundo Zilli (2004, p. 40):

- ❑ raciocínio lógico;
- ❑ habilidades manuais e estéticas;
- ❑ relações interpessoais e intrapessoais;
- ❑ utilização de conceitos aprendidos em diversas áreas do conhecimento para o desenvolvimento de projetos;
- ❑ investigação e compreensão;
- ❑ representação e comunicação;
- ❑ trabalho com pesquisa;
- ❑ resolução de problemas por meio de erros e acertos;
- ❑ aplicação das teorias formuladas a atividades concretas;
- ❑ utilização da criatividade em diferentes situações; e
- ❑ capacidade crítica.

Santos e Menezes (2005, p. 2.752), durante as considerações finais sobre uma pesquisa de aplicação da robótica educacional no aprendizado de Física para alunos do ensino fundamental, enfatizam: “[...] Ficou clara a facilidade de aprendizagem através da construção. Os alunos puderam presenciar no concreto alguns conceitos de difícil entendimento devido a dificuldade que encontram em abstrair.”.

2.3 Robótica na Sala de Aula

Diferentes estudiosos realizaram projetos em distintas atmosferas educativas do território nacional, expondo resultados favoráveis da aplicação da robótica com fins educativos interdisciplinares nesses ambientes (D’ABREU, J.V.V., 1998; CHELLA, 2002a; D’ABREU, J.V.V., GONÇALVES, L.M.G., GARCIA, M.F. & GARCIA, L.T.S., 2002; SCHONS, C.,

PRIMAZ, E. & WIRTH, G.A.P., 2004; SANTOS, C.F. & MENEZES, C.S., 2005; ALVES, A.C., BLIKSTEIN, P. & LOPES, R.D., 2005).

A despeito dessas conclusões positivas, a fim de construir uma opinião mais embasada da robótica aplicada à educação, este autor participou de um mini-curso no XIV Simpósio Brasileiro de Informática na Educação (SBIE) e de uma oficina no XXIV Congresso da Sociedade Brasileira de Computação (CSBC) relacionados com as áreas de abrangência dessa pesquisa. Em seqüência, já instrumentado, foram realizadas visitas a dois colégios – Colégio Notre Dame (Rio de Janeiro – RJ) e Colégio Nobel (Salvador – BA) – que utilizam a robótica educacional como ferramental de apoio à exploração de conceitos referenciados na prática do dia-a-dia em sala de aula.

Observou-se nos ambientes escolares que a robótica proporciona ao aluno a possibilidade de “aprender fazendo”, através do aprendizado empírico – baseado na experimentação prática. Uma clássica demonstração que, segundo Papert (ALVES, A.C.; BLIKSTEIN, P.; LOPES, R.D., 2005, p. 2.597, apud Papert, 1999), “[...] aprendemos melhor fazendo, porém aprendemos melhor ainda, se combinamos nossas atividades falando e pensando sobre o que fizemos.” e que, segundo d’Abreu e Chella (2001, p. 1), “[...] Aprender envolve dar sentido a experiência, do pensamento, ou do fenômeno no contexto”.

Para Freire (ALVES, A.C.; BLIKSTEIN, P.; LOPES, R.D., 2005, p. 2.596, apud Freire, 1921),

[...] toda a ação educativa deve ser acompanhada de um momento de reflexão sobre o homem e seu habitat. Para que a mudança sobre a concepção do uso das tecnologias em sala de aula ocorra, é necessário que as tecnologias tenham significado, estando presentes no cotidiano do professor e do aluno. O uso dessas tecnologias deve ser justificado pela necessidade da tecnologia para a realização de uma atividade e não pela realização de atividades desconexas que o justifiquem.

Concluiu-se, ao observar o caráter multidisciplinar do uso da robótica na educação, que a sua utilização no ambiente escolar pode promover uma integração harmoniosa e expressiva entre diferentes disciplinas.

Assim sendo, pensa-se que os mentores de projetos amparados pela robótica educacional devem olhar para esses projetos não apenas sobre o ângulo de visão eminentemente tecnicista, mas, sobretudo com propostas pedagógicas vinculadas as atividades. A robótica aplicada na educação traz mais um material rico para apoiar o raciocínio lógico explorando os potenciais cognitivos dos adolescentes, contribuindo também com uma possível fusão entre os conhecimentos teóricos e a experimentação prática.

A exploração de um problema em sala de aula – igualmente fazendo uso da robótica com fins pedagógicos – pode ser realizada em quatro momentos distintos:

- 1) A problematização – O aluno entra em contato com as vertentes do problema;
- 2) Exploração de potenciais soluções – O aprendiz explora possíveis soluções, compartilhando e discutindo com outros discentes as formas pensadas para solucionar o problema;
- 3) Desenvolvimento das soluções teóricas – Nessa etapa é permitido o desenvolvimento das soluções teóricas pelo discente, podendo ser utilizada fórmulas matemáticas ou outras representações; e
- 4) Análise dos resultados – É feita a verificação, nesse momento, se os resultados gerados resolvem concretamente o problema, permitindo que o aluno – sozinho ou coletivamente com a turma – possa refletir sobre o mesmo, e caso seja necessário retornando as atividades do segundo momento para uma nova interação.

A diferenciação com a utilização da robótica pode ocorrer, por exemplo, numa oficina de robótica pedagógica, no terceiro momento, onde o aluno pode romper o plano teórico – abstrações – utilizando os artefatos tecnológicos de um kit de robótica educacional para propor soluções experimentais – concretas – através da realização de montagens físicas e programação no microcomputador das possíveis soluções para resolver um problema, podendo assim integrar

e complementar teoria com prática. Nesse formato, surge um momento intermediário antes da “análise dos resultados”, que tem como objetivo realizar testes nas implementações e coletar resultados parciais.

Na sala de aula, o professor-facilitador e até mesmo os discentes, tem a possibilidade de propor projetos com situações-problema para serem resolvidos/explorados. Mediante a apresentação de um problema um(a) aluno/grupo/turma buscará uma solução, utilizando-se do ferramental disponibilizado por um kit de robótica e outros materiais – podendo ser utilizado, inclusive, sucata e materiais alternativos. A concretização da solução é efetuada ao idealizar um modelo para solucionar o problema. Tal solução pode passar pela programação e personalização do kit de robótica através da inclusão/remoção de dispositivos eletrônicos nesse artefato, e ainda, se o kit assim permitir, simular através do *software*, e também executá-lo em *hardware* para validação do modelo em sua amplitude e no atendimento das necessidades impostas inicialmente.

Para d’Abreu (1999, p. 3),

[...] O enfoque da metodologia a ser desenvolvida [e que pode ser aplicada à robótica educacional em sala de aula] deve centrar-se na implementação de ambientes de aprendizagens rico em situações que permitem ao aluno construir o seu conhecimento, através do uso do computador e dos dispositivos robóticos. A metodologia deve propiciar subsídios para uma diversificação, diferenciação e expansão na forma de aquisição e manuseio de conceitos.

Na visão de Santos, Magalhães Netto e Menezes (2004, p. 585),

A robótica educacional tem como metodologia a aprendizagem baseada em problemas (PBL), ou seja, o problema é utilizado como estímulo à aquisição de conhecimento e compreensão de conceitos. A robótica proporciona uma aprendizagem cooperativa e colaborativa, promovendo desafios que levam ao desequilíbrio da situação, fazendo com que o aluno busque informações em diferentes fontes. Com isso os alunos dão formas as suas idéias, levantam hipóteses e fazem testes até alcançarem seus objetivos de forma envolvente e divertida.

2.4 Kits de Robótica Educacional

Segundo o Dicionário Houaiss da Língua Portuguesa a acepção do verbete kit é: “[...] jogo de elementos que atendem juntos a um mesmo propósito ou utilidade [...]” (HOUAISS, A.,

2001, p. 1.702). Utilizando-se como referencial o significado dessa palavra conceitua-se, nesse trabalho, a expressão “kit de robótica educacional” como o “conjunto de um ou mais artefatos de *hardware* e *software* que, integrados, tenham como proposta apoiar o processo de ensino-aprendizagem do ambiente escolar”.

Baseadas nessa definição, as investigações feitas constataram que, atualmente, existem várias empresas que desenvolvem e comercializam kits de robótica educacional para serem utilizados em sala de aula. Desses kits de robótica disponíveis no mercado foram selecionados, para a realização dessa pesquisa, 04 (quatro) kits: 02 (dois) projetados e desenvolvidos no Brasil (Robótica Fácil e Super Robby), e outros 02 (dois) no exterior (GoGo Board e Lego MindStorms). O critério para a escolha dos pares fundamentou-se na maior quantidade numérica de referências encontradas para cada kit durante a realização da primeira fase dessa pesquisa – o levantamento bibliográfico.

Adquiriu-se, através do levantamento bibliográfico e da utilização concreta desses produtos, conhecimento das principais características de cada kit, culminando em um quadro comparativo dos mesmos, incluindo custo (Tabela 1), o que pode colaborar na escolha de determinado kit para cada contexto de utilização da robótica educativa.

Vale mencionar que kits de robótica educacional por si só são meros artefatos tecnológicos. É necessário ter um propósito educacional ao utilizá-lo para que o kit seja útil no processo de ensino-aprendizagem.

2.4.1 Robótica Fácil

Os kits de robótica educativa concebidos e desenvolvidos no Brasil apresentam algumas limitações, principalmente nos recursos da linguagem de controle⁶ que acompanha cada produto, fato que se deve principalmente à ausência de recursos eletrônicos mais sofisticados nesses kits.

O Kit Robótica Fácil (ROBÓTICA FÁCIL, 2003) é composto por uma placa principal que contém relés, cabo paralelo, cabo *flat* de 30 vias, fonte de 12V e um *software* de controle que já vem com o kit – não necessitando sua aquisição à parte. Esse produto permite o controle de dispositivos eletrônicos tais como: aparelhos eletrônicos, lâmpadas, relês, motores, entre outros.

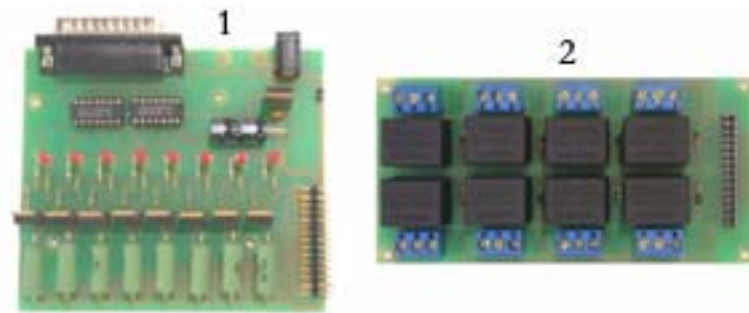


Figura 1 – Kit Robótica Fácil. (1) Placa principal; e (2) Placa de relês

O *software* utilizado para a criação do comportamento é baseado na interface de linha de comando, apesar de funcionar no ambiente Windows. O programa que acompanha esse kit foi o que apresentou uma quantidade menor de recursos, apenas possibilitando ligar ou desligar algum dispositivo eletrônico, tais como um motor ou uma lâmpada, e esperar alguma quantidade de tempo, em segundos. Constatou-se que essa limitação da linguagem pode ser atribuída à arquitetura eletrônica trivial desse produto.

⁶ A linguagem de controle também pode ser limitada devido a falta de uma visão educacional de quem projeta esses ambientes.

2.4.2 Super Robby

O Kit Super Robby (SUPER ROBBY, 2004) comercializado pela ARS Consult é composto pela interface Super Robby, 01 (uma) fonte de alimentação, 01 (um) motor contínuo, 03 (três) lâmpadas, 06 (seis) metros de fio e 01 (um) disquete de 3½ que contém o *software* que controla o Kit. Durante a análise desse produto foi utilizado para teste o *software* Super Robby v.1.0.



Figura 2 – Kit Super Robby. (1) Interface Super Robby; (2) Fonte de alimentação; e (3) Motor

Esse programa, apesar de possuir uma interface visual, possui limitações no desenvolvimento de modelos. Essa restrição de funcionalidades parece ser reflexo dos poucos recursos de *hardware* disponibilizados nesse kit.

O Super Robby também pode utilizar um “Kit de Acessórios para o Super Robby”, que é vendido separadamente, e contém: 02 (dois) motores de passo, 03 (três) motores de corrente contínua, 01 (uma) chave de fenda, 08 (oito) *leds*, 10 (dez) lâmpadas, 01 (um) sonorizador, 02 (dois) relés, 02 (dois) potenciômetros, 02 (dois) sensores de luz, 02 (dois) sensores de temperatura, 04 (quatro) sensores de toque, 10 (dez) metros de fio e 01 (um) disquete de 1,44 MB, contendo os programas Guia de Projeto e Kit de Acessórios.

Esse Kit, identicamente ao Kit Robótica Fácil, também possui uma arquitetura eletrônica simplificada. Contudo, a empresa desenvolvedora desse produto também disponibiliza, para

compra, um kit de acessórios que expande as possibilidades de sua utilização. Uma vantagem desse Kit, em relação ao outro produto nacional, está na possibilidade de utilizar outras linguagens com foco educacional, tais como o MegaLogo (MEGALOGO, 2005) – esse *software* deixou de ser comercializado no Brasil – e o Imagine (IMAGINE, 2005), para a criação de modelos com esse produto. A utilização dos programas supracitados pode ampliar o leque de aplicações para esse produto.

2.4.3 GoGo Board

O Kit GoGo Board (GOGO BOARD, 2004) é uma coleção de dispositivos eletrônicos que possui sua arquitetura eletrônica e *softwares* publicados/disponibilizados na Internet, o que não significa que o uso do mesmo seja facilitado para os professores. Um diferencial desse Kit em relação aos outros três avaliados, reside nessa característica, possibilitando modificações no seu *hardware* e até mesmo no *software*, para atender a necessidades específicas. Não obstante, a impossibilidade de compra do kit “pronto para uso”, impossibilitou a apresentação de seu custo na tabela comparativa entre os kits de robótica analisados (Tabela 1).



Figura 3 – GoGo Board

Diversos *softwares* que podem controlar esse kit estão disponíveis, inclusive no *site* do projeto. Bibliotecas prontas também estão publicadas, permitindo utilizar esse Kit com

linguagens de programação comerciais, tais como Microsoft Visual C++ (VISUAL C++, 2006) e Microsoft Visual Basic (VISUAL BASIC, 2006), além de outros ambientes com foco educacional, como o Imagine (IMAGINE, 2005) e o Microworlds (MICROWORLDS, 2005).

Como a maioria dos outros kits analisados nesse trabalho, a placa GoGo também foi projetada com componentes e superfícies de soldagem grandes. Não foram usados componentes ultra-miniaturizados que requerem experiência técnica ou equipamentos especiais para sua montagem, mesmo assim necessita de conhecimentos de eletrônica para ser montado/utilizado.

A característica de destaque desse produto está na concepção dos idealizadores desse kit, de não comercializá-lo, mas sim disponibilizar todas as ferramentas necessárias para sua montagem. Durante o levantamento dos componentes necessários para a montagem do kit, foi constatada a dificuldade de se realizar a compra de todo material que compõem esse Kit e a montagem do mesmo por usuários não familiarizados com eletrônica.

2.4.4 Lego MindStorms

Para a realização do estudo sobre o kit de robótica da Lego (LEGO, 2005c), foi utilizado o modelo MindStorms for Schools 9793 (composto pelos conjuntos 9790 e 2000687), que possui 828 peças, e o *software* RoboLab v.2.01 (ROBOLAB, 2004). Salienta-se que esse foi o kit que apresentou o maior número de referências bibliográficas em projetos nacionais relacionados ao emprego da robótica educacional.

O principal componente desse Kit é o Robotic Command Explorer (RCX), que é um bloco programável. Para a realização da programação do RCX utilizou-se o RoboLab que é uma linguagem de programação visual. Um dos grandes diferenciais desse kit, além da parte mecânica do mesmo, é possuir uma linguagem de programação icônica intuitiva e comunicação do microcomputador com o RCX através de infravermelho.



Figura 4 – Lego MindStorms. (1) RCX; (2) Transmissor infravermelho para RCX; (3) Caixa com peças Lego; e (4) Revista com material pedagógico

O kit comercializado pela Lego está sendo aperfeiçoado há mais de uma década em parceria com o Massachusetts Institute of Technology (MIT) (MIT, 2006) – para desenvolver o RCX, e com a National Instruments (NI) (NI, 2006) e Tufts University (TUFTS, 2006) – para implementar o RoboLab. No caso da Lego, o RCX que possui um microprocessador da série Hitachi H8/3292 é o principal componente, podendo ser considerado o “cérebro” do kit. No Kit de Robótica Educacional RoboFácil – produto dessa dissertação, essa função é executada pelo microcontrolador Intel 8031AH, conforme será visto no Capítulo 5.

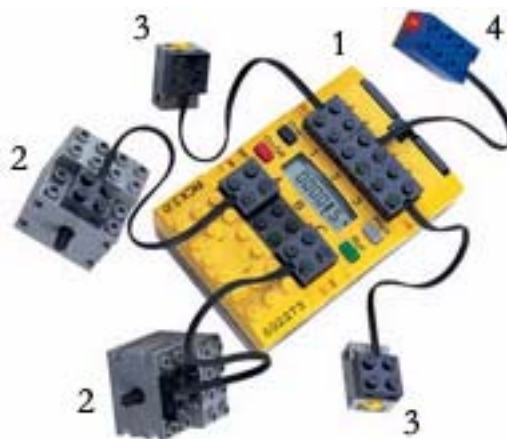


Figura 5 – RCX, motores e sensores. (1) RCX; (2) Motores; (3) Sensores de toque; e (4) Sensor de luz

Pode-se citar pelo menos duas vantagens peculiares na utilização dessas CPUs nos kits de robótica supracitados. A primeira delas é a possibilidade dos mesmos serem controlados e

reprogramados através do microcomputador. Dessa forma, os alunos podem produzir uma gama variada de comportamentos para os dispositivos eletrônicos acoplados a um kit, possibilitando sua utilização em projetos interdisciplinares.

A segunda vantagem é a característica dos kits poderem ser desconectados do microcomputador após terem seu comportamento estabelecido – programado – pelos alunos. Essa propriedade torna esses kits versáteis para serem utilizados em projetos de cunho científico e didático.

Segundo Tavares, Antunes e Gonçalves (2004, p. 80),

[...] devido a sua complexidade e seu potencial muito alto [Lego MindStorms] para a faixa etária a qual foi destinado e também ao aparecimento ou desenvolvimento de várias ferramentas de software para controle em nível um pouco mais baixo (usando linguagem 'C' ao invés da linguagem visual provida pelo fabricante), os kits passaram a ser mais usado nos ambientes de pesquisa e no ensino superior.

Foram constatados, também durante a realização dos testes desse kit, outros motivos pelo qual esse é um dos kits mais utilizado no Brasil: (1) a mesma estrutura mecânica dos brinquedos da Lego (LEGO, 2005b), o que pode tornar sua utilização intuitiva por crianças e adolescentes; (2) seu manuseio é facilitado com a utilização dos blocos de montar; (3) a existência de suporte e materiais técnicos/pedagógicos desse produto no País (EDACOM, 2005); e (4) a possibilidade de uso de outras linguagens com foco educacional para programação do Kit, além do já mencionado *software* RoboLab (ROBOLAB, 2004) – adquirido separadamente, tais como, o Microworlds (MICROWORLDS, 2005) e o SuperLogo (SUPERLOGO, 2005).

Muito embora possua características desejadas, seu custo de aquisição pode inviabilizar a utilização desse produto – R\$ 1.844,00 – maior valor entre os kits analisados.

2.4.5 Análise Comparativa

Após apresentar os 04 (quatro) kits de robótica educacional que foram objetos de estudo dessa pesquisa, apresenta-se uma tabela comparativa entre esses produtos, enfatizando alguns

itens relevantes quando da escolha de um kit de robótica para ser empregado em projetos educacionais.

Tabela 1 – Comparação entre quatro kits de robótica educacional

Kits Características	Robótica Fácil	Super Robby	GoGo Board	Lego MindStorms
Custo (R\$) ⁷	300,00	450,00	Não foi possível mensurar (vide item 2.3.3)	1.644,00 (kit) + 200,00 (software)
Arquitetura eletrônica pública	Não	Não	Sim	Não
Código fonte do Software público	Não	Não	Sim	Não
Interface com o microcomputador	Cabo de dados via interface paralela	Cabo de dados via interface paralela	Cabo de dados via interface serial	Infra-vermelho via interface serial
Software para Windows	Sim	Sim	Sim	Sim
Programação por linha de comando	Sim	Não	Sim	Sim
Programação icônica	Não	Sim	Sim	Sim
Estruturas condicionais na ferramenta padrão de controle do <i>hardware</i> do kit	Não	Não	Sim	Sim
Funcionamento autônomo	Não	Não	Sim	Sim

2.5 Dificuldades Impostas

Não obstante as inúmeras vantagens do emprego da robótica pedagógica no ambiente escolar, não foram encontrados registros – documentados – que relatassem, no aspecto quantitativo, a utilização da robótica educacional nos 21.980 estabelecimentos de ensino médio

⁷ O levantamento de preço dos kits de robótica educacional foram realizados na primeira quinzena de novembro de 2004.

presentes em todo o território nacional (INEP, 2004). Uma possível justificativa para tal fato poderia ser a baixa utilização da robótica educativa como ferramental de apoio ao processo de ensino-aprendizagem nas escolas brasileiras, principalmente em laboratórios de ensino de ciências. Uma outra possibilidade seria a falta de tradição no Brasil, do ensino baseado na experimentação.

Na academia, poucas pesquisas são desenvolvidas com o intuito de clarificar os motivos pelos quais a robótica educacional pode estar sendo pouco utilizada nas salas de aula. Os trabalhos realizados, em sua maioria, limitam-se a propor novas arquiteturas de *hardware* e/ou *software*, desconsiderando as particularidades desses ambientes.

Esse possível baixo uso da robótica aplicada à educação pode ter origem também nos fatores relatados a seguir: (1) gestores educacionais não possuem conhecimento dessa ferramenta pedagógica; (2) professores não estão capacitados para trabalhar de forma metodológica com essa tecnologia, pois os cursos de Graduação para formação de professores – Licenciatura – não estão explorando esse ferramental em sua grade curricular; (3) kits de robótica comercializados no Brasil possuem elevado custo para o padrão da maioria das escolas públicas ou particulares; e (4) limitação tecnológica – *hardware* e *software* – de alguns produtos.

Buscando um entendimento mais apurado, pensa-se que essa possível baixa utilização pode ser impulsionada pelo emprego de kits de robótica educacional desenvolvidos no exterior. Esses kits geralmente possuem um elevado custo para os padrões da maior parte das escolas do ensino fundamental e médio, sejam elas públicas ou particulares, podendo assim inviabilizar o emprego dessa ferramenta pedagógica. Um outro fato averiguado que pode também estar contribuindo para o aparente baixo uso da robótica é que parte dos kits desenvolvidos no Brasil carece de alguns recursos técnicos de *hardware* e *software* para sua utilização em projetos educacionais, tais como (re)programação por *software* e funcionamento autônomo.

Esse capítulo explorou a revisão da literatura que embasou a realização desse trabalho. Demonstrou-se também que há, atualmente no mercado, diferentes kits de robótica com fins educacionais, e dentre esses foram selecionados 04 (quatro) – os que obtiveram maior quantidade numérica de referências encontradas durante o levantamento bibliográfico dessa pesquisa – para serem analisados individualmente e comparados entre si a partir de diferentes aspectos.

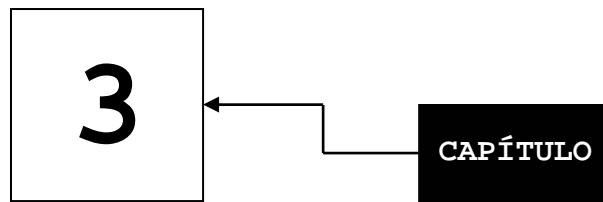
Essa análise teve como objetivo buscar uma alternativa aos kits de robótica estudados nesse capítulo e conjugar vantagens/desvantagens desses em relação ao kit desenvolvido durante essa pesquisa: Kit de Robótica Educacional RoboFácil. Esse Kit é constituído pelos artefatos de *hardware* e *software* apresentados na Seção II dessa dissertação.

Particularmente, no próximo capítulo, é apresentada a ferramenta de ensino⁸ desenvolvida para controlar/programar o Kit RoboFácil.

⁸ De acordo com Loureiro (1998, p. 72), “A caracterização de um software enquanto ferramenta de ensino é bastante evidente se percebemos que neste caso o aluno é que ensina o computador [ou um hardware de um kit de robótica educacional], diferindo dicotomicamente da primeira classificação [software enquanto máquina de ensinar] onde o software ensina o aluno.”

SEÇÃO
II

A Seção II desse trabalho aborda as principais tecnologias de *hardware* e *software*, analisando-as em seus principais conceitos. Em seguida descreve os produtos desenvolvidos nessa pesquisa, qual sejam, o *software* ProgrameFácil, o *hardware* RoboFácil e o seu *firmware*.



PROGRAMEFÁCIL

O Capítulo 3 apresenta as motivações e justificativas que permearam a concepção e implementação do *software* educacional ProgrameFácil, seguido da descrição detalhada dos diferentes conceitos e elementos presentes nesse ambiente de programação visual, bem como sua interface.

3.1 Motivações e Justificativas

A opção pelo desenvolvimento do *software* ProgrameFácil foi impulsionada pela necessidade de um ambiente visual que tivesse uma interface amigável, possibilitando aos estudantes do ensino médio – público alvo dessa pesquisa – a programação do comportamento do *hardware* RoboFácil de forma intuitiva, e sem conhecimento prévio da arquitetura eletrônica adotada no projeto do mesmo, visto que anteriormente era necessário realizar acessos diretamente aos endereços de memória do microcontrolador, proporcionando o manuseio dos diversos recursos eletrônicos, e da linguagem *assembly*.

Sabendo-se que o desenvolvimento de programas é um processo que cresce em complexidade em função de sua envergadura (BITTENCOURT, J.V., 2004, p. 51) e que a especificação de requisitos é um dos principais problemas que compromete a criação de *softwares* (GIRAFFA, L.; MARCZAK, S.; PRIKLADNICKI, R., 2005), foi proposto pelos orientadores dividir o desenvolvimento da ferramenta em duas etapas para melhor definir os requisitos. Dessa forma, a primeira etapa foi constituída de estudos sobre funcionalidades e aspectos de interface para *software* educacional com foco em robótica, e na materialização da proposta de um ambiente de programação para kits de robótica em folhas de cartolina e pedaços recortados de papel, empregados na discussão entre os membros da equipe.

A implementação computacional foi realizada durante a segunda etapa – fazendo uso do método Top-Down de programação, utilizando um dos *softwares* disponíveis no mercado, o ambiente integrado de desenvolvimento (IDE) Borland Delphi versão 6.0 (DELPHI, 2005), que proporciona o desenvolvimento rápido de aplicações (RAD). Tal ferramenta possui nativamente, funcionalidades que garantem ao desenvolvedor condições simplificadas de codificação e depuração de aplicações.

De acordo com d’Abreu (1999, p. 2), “[...] o programa [que controla dispositivos robóticos] deve apresentar características pedagógicas que permitam ao aluno descrever, de forma

simples e rápida, para o computador o funcionamento de um certo objeto [dispositivo robótico] [...]”. Assim sendo, as fases de concepção, projeto e implementação da linguagem computacional visual ProgameFácil levaram sempre em conta a necessidade de criação de um ambiente intuitivo, de forma a torná-lo agradável de usar e eficiente para o controle dos recursos eletrônicos do *hardware* do Kit de Robótica Educacional RoboFácil.

A fim de garantir flexibilidade e qualidade na forma de interação entre o ProgameFácil e o *hardware* RoboFácil, dividiu-se a solução em duas partes distintas: (1) compilador; e (2) interpretador. A primeira solução contempla recursos de compilação que foram agregados ao referido *software*. Esse compilador tem como função converter os elementos que compõem o programa em macrocódigos *assembly*⁹.

A segunda solução visa criar um ambiente, transparente ao usuário, através da definição de macrocódigos *assembly*, para interpretar os programas dos modelos compilados no *software* ProgameFácil. Esse interpretador, presente no *firmware*¹⁰ do RoboFácil – explorado no próximo capítulo – permite converter programas escritos nos macrocódigos *assembly*, gerados ou não automaticamente pelo ProgameFácil, em instruções que podem ser executadas no *hardware* propriamente dito, tais como as atividades de acender ou apagar *led*, escrever mensagem no *display*, entre outras.

Visando colaborar com a solução dos problemas e subsidiar a demonstração de uma das hipóteses identificadas para essa pesquisa, foi construído e integrado a essa ferramenta um ambiente de simulação. Esse ambiente possibilita, na tela do monitor, a visualização de um modelo de comportamento de um *hardware* sendo executado/simulado. Essa característica faz com que, entre outras vantagens, não seja necessária a compra da mesma quantidade de *hardware* e *software* para uso da robótica em ambiente escolar.

⁹ Emprega-se, nessa pesquisa, o termo “macrocódigos assembly” para referenciar os códigos hexadecimais definidos que representam os pseudocódigos de máquina desse hardware (assembly virtual do hardware RoboFácil).

¹⁰ Firmware, segundo o Dicionário Houaiss da Língua Portuguesa são: “programas de computador mantidos na memória permanente, como dispositivos operacionais de controle do hardware.” (HOUAISS, A., 2001, p. 1.348)

As características já mencionadas o qualificam e o diferenciam dos *softwares* educacionais que tem como objetivo programar e controlar kits de robótica com fins pedagógicos.

Apesar da experiência prática ser uma característica relevante da educação (QUEIROZ, L.R. et al., 1998), a possibilidade de se reduzir o número de *hardwares* de robótica, necessários quando do seu emprego em sala de aula, pode contribuir para o aumento do uso dessa ferramenta pedagógica nas escolas brasileiras, pois será possível economizar recursos financeiros.

3.2 ProgrameFácil

3.2.1 Definição

O ProgrameFácil é uma linguagem computacional icônica¹¹ que permite programar dispositivos eletrônicos e/ou eletromecânicos que estejam acoplados a um *hardware* de um kit de robótica educacional, tais como lâmpadas, *displays*, *leds*, motores de passo, sensores de luminosidade e sensores de temperatura, fazendo uso das tradicionais construções de programação estruturada, tais como: condicional e repetição (MIRANDA, L.C.; SAMPAIO, F.F.; BORGES, J.A.S., 2005).

Essa linguagem foi concebida sobre dois pilares: o primeiro referencia a metáfora de um mundo hipotético – disponível no ProgrameFácil através da janela “Meu Mundo”. Nessa janela especifica-se a configuração desejada do *hardware* por meio da inclusão de *leds*, motores, sensores, entre outros componentes eletrônicos na referida janela e visualiza-se seu funcionamento com a simulação. O outro pilar refere-se à metáfora de um programa que controla esse mundo, visualizado no ProgrameFácil pela janela “Meu Programa”, onde é permitido explicitar a programação do *hardware* detalhado no “Meu Mundo”.

¹¹ Baseada na manipulação de ícones gráficos.

Esses pilares viabilizam a criação, por parte do usuário-aluno, de modelos de comportamento – em tempo de *design* – e a observação de seu comportamento – em tempo de execução/simulação. A adoção de etapas distintas e explícitas para “desenhar” e executar/simular modelos têm como objetivo facilitar a averiguação da lógica empregada em cada programa.

3.2.2 Objetos

Segundo Boratti (2001, p. 13) um objeto pode ser definido como: “[...] sendo a abstração de uma entidade do mundo real, que apresenta sua própria existência, identificação, características de composição e tem alguma utilidade, [...]”. Fazendo uso parcial dessas idéias, definiu-se objeto, na esfera do ProgameFácil, como sendo uma representação gráfica – semelhante a um ícone – que pode ser manipulado tanto na janela do “Meu Mundo” como do “Meu Programa”. Os objetos no ProgameFácil foram divididos em três categorias para melhor identificar seu propósito: (1) Objetos de Hardware; (2) Objetos de Programação; e (3) Objetos de Apoio.

Os “Objetos de Hardware” representam dispositivos eletrônicos e foram divididos em duas sub-categorias: (1) Objetos de Hardware de Entrada; e (2) Objetos de Hardware de Saída. Os “Objetos de Programação” fazem referência a estruturas usadas em linguagens de programação. Por sua vez, os “Objetos de Apoio” visam unicamente proporcionar facilidades e recursos operacionais aos usuários da ferramenta.

Os “Objetos de Hardware de Saída” são nomeados igualmente em ambas as janelas, todavia possuindo distintas características, por exemplo: o objeto Led do “Meu Mundo” possui a particularidade de “Cor” e do “Meu Programa” a de “Acender”.

Após a inclusão de um “Objeto de Hardware” na ferramenta é possível, na Janela “Meu Mundo”, alterar as propriedades desse objeto. Na janela “Meu Programa” os “Objetos de Hardware de Saída” permitem a alteração de suas ações, e através da utilização associada do “Objeto de Programação Estrutura de Controle Se” com os “Objetos de Hardware de Entrada”

ou “Objetos de Hardware de Saída” – com exceção do “Objeto Display” – é possível estabelecer testes condicionais lógicos sobre esses objetos.

3.2.3 Meu Mundo

O “Meu Mundo” é a janela onde o usuário poderá especificar o mundo hipotético que representa uma configuração de *hardware* de um kit de robótica¹². Os objetos concretos que representam os elementos eletrônicos disponíveis para escolha pelo usuário são apresentados em um menu gráfico – barra de ferramentas – dessa janela, com exceção do “Objeto de Apoio Comentário”, que visa unicamente permitir a inserção de texto no corpo de um modelo.

Na Figura 6 é apresentada uma parte da barra de ferramentas da janela “Meu Mundo”.



Figura 6 – Objeto Comentário (7) e Objetos de Hardware disponíveis na barra de ferramenta da janela Meu Mundo. (1) Led; (2) Display; (3) Lâmpada; (4) Motor; (5) Sensor de Luminosidade; e (6) Sensor de Temperatura

Os objetos presentes nessa janela foram abstraídos da vida real, logo, para associá-los a um *hardware*, faz-se necessário conhecer suas características físicas e ações permitidas na realidade. Nessa janela, apenas “Objetos de Hardware” podem ser manipulados.

Para clarificar o entendimento desses objetos, serão apresentados nas Tabelas 2 e 3 os atributos pertencentes aos objetos dessa categoria.

¹² Lembra-se que, todo kit de robótica possui limitação quantitativa dos seus recursos. Portanto, buscando maior proximidade com a realidade, essa limitação também foi contemplada na ferramenta.

Tabela 2 – Rótulo, propriedade, valores e testes lógicos dos objetos de hardware de saída da janela Meu Mundo

Objeto	Rótulo	Propriedade	Valores	Testes Lógicos
Led	Ex	Cor	Verde, Amarelo e Vermelho	Aceso e Apagado
Display	Dx	Cor de Fundo	Verde e Azul	–
Lâmpada	Lx	Cor	Verde, Amarelo e Vermelho	Acesa e Apagada
Motor	Mx	–	–	Ligado e Desligado

NOTA: x representa o identificador numérico de cada objeto.

Tabela 3 – Rótulo e testes lógicos dos objetos de hardware de entrada da janela Meu Mundo

Objeto	Rótulo	Testes Lógicos
Sensor de Luminosidade	SLx	Recebendo Luz e Não está Recebendo Luz ¹³
Sensor de Temperatura	STx	Recebendo Calor e Não está Recebendo Calor ¹⁴

NOTA: x representa o identificador numérico de cada objeto.

3.2.4 Meu Programa

O “Meu Programa” pode ser definido como o local onde o usuário especifica ações associadas aos objetos gráficos inclusos na janela “Meu Mundo”, podendo fazer uso das tradicionais estruturas de programação condicional e repetição. A exceção se dá com o objeto “Comentário” que possui finalidade igual à do objeto de mesmo nome da janela “Meu Mundo”.

Apresenta-se, na Figura 7, um segmento da barra de ferramentas da janela “Meu Programa”, que possui os recursos agrupados logicamente – por categoria de objeto – visando aumentar a eficiência¹⁵ do diálogo entre o usuário e a ferramenta.

¹³ O objeto “Sensor de Luminosidade” foi implementado na ferramenta ProgrameFácil seguindo as formas de funcionamento desse tipo de sensor no hardware RoboFácil (Capítulo 5). Portanto, como a obtenção dos dados do sensor de luminosidade é realizada de forma binária, a ferramenta explorou esse formato de leitura. O resultado dos testes lógicos “Recebendo Luz” e “Não está Recebendo Luz” – verdadeiro ou falso – dependerá do valor de referência configurado nesse hardware.

¹⁴ O objeto “Sensor de Temperatura” também foi desenvolvido no ProgrameFácil, seguindo os princípios de funcionamento desse tipo de sensor no hardware RoboFácil. Sendo assim, como a coleta de dados desse sensor de temperatura é realizada de forma binária, a ferramenta explorou esse formato de leitura. O resultado dos testes lógicos “Recebendo Calor” e “Não está Recebendo Calor” – verdadeiro ou falso – dependerá do valor de referência configurado nesse hardware.

¹⁵ Rocha e Baranauskas (2003, p. 31), comentando os atributos de usabilidade definidos por Nielsen (1993), descrevem que “O sistema precisa ser eficiente no uso, de forma que uma vez aprendido o usuário tenha um elevado nível de produtividade.”.



Figura 7 – Objeto Comentário (12), Objetos de Hardware de Saída – (1) Led; (2) Display; (3) Lâmpada; e (4) Motor – e Objetos de Programação – (5) Temporizador; (6) Estrutura de Controle Se; (7) Início de Looping; (8) Fim de Looping; (9) Linha de Programação; (10) Início de Programa; e (11) Fim de Programa – disponíveis na barra de ferramenta da janela Meu Programa

O objetivo é tornar possível a construção de uma estrutura lógica de programação entre os objetos inseridos na janela “Meu Programa”, formando assim o que é definido no contexto do *software* ProgameFácil como o “programa do modelo”. Para alcançar esse propósito a linguagem computacional foi alicerçada em conformação com cinco regras de formação:

Primeira regra: Cada objeto tem um ou nenhum sucessor na estrutura lógica de programação;

Segunda regra: O objeto Sinal Vermelho – que representa fim de programa – não poderá ter sucessores;

Terceira regra: O objeto SE – que representa a estrutura condicional Se (If) – terá até dois sucessores;

Quarta regra: Cada objeto pode ter um ou mais predecessores na estrutura lógica de programação; e

Quinta regra: O objeto Sinal Verde – que representa início de programa – não poderá ter predecessores.

A inclusão de “Objetos de Hardware de Saída” na janela “Meu Programa” está atrelada a sua existência no “Meu Mundo”. Observe também que, nessa janela, tanto “Objetos de Hardware” como “Objetos de Programação” podem ser utilizados.

Nas Tabelas 4 e 5 serão detalhados os elementos presentes no menu gráfico – barra de ferramentas – que podem ser usados na formulação do programa de um modelo.

Tabela 4 – Rótulo e ações dos objetos de hardware de saída da janela Meu Programa

Objeto de Hardware de Saída	Rótulo	Ações
Led	Ex	Acender e Apagar
Display	Dx	Escrever Mensagem e Limpar
Lâmpada	Lx	Acender e Apagar
Motor	Mx	Ligar e Desligar

NOTA: x representa o identificador numérico de cada objeto.

Tabela 5 – Rótulo e finalidade dos objetos de programação da janela Meu Programa

Objeto de Programação	Rótulo	Finalidade
Temporizador	Tx	Temporizar – Esperar uma quantidade de tempo em segundos
Estrutura de Controle Se	SEx	Desvia para fluxos de execução diferenciados dependendo do resultado da condição de teste ser verdadeira ou falsa. As condições de testes são caracterizadas como testes lógicos sobre um dos possíveis status dos Objetos de Hardware do Meu Mundo – Led, Lâmpada, Motor, Sensor de Luminosidade e Sensor de Temperatura
Início de Looping	LPx	Define o início de uma estrutura de repetição finita ou infinita
Fim de Looping	LPx	Marca a posição final de um bloco de comandos dentro de uma estrutura de repetição
Linha de Programação	–	Estabelece o fluxo de execução atemporal entre os objetos
Início de Programa	IP	Define o início de um programa
Fim de Programa	FPx	Estabelece a posição final de execução/simulação de um programa

NOTA: x representa o identificador numérico de cada objeto.

3.2.5 Modelo de Comportamento (Modelo)

Geralmente, modelos são criados com um cunho investigativo na tentativa de isolar ou melhor observar, sob condições pré-definidas ou conhecidas, um fato ou uma teoria do mundo real que tem intrinsecamente a complexidade como característica.

Dessa forma, definiu-se no ProgrameFácil o termo modelo de comportamento, ou simplesmente modelo, como a configuração de um *hardware* de robótica hipotético ou real, com o seu respectivo programa de controle. A materialização dessas idéias na ferramenta ProgrameFácil foi alcançada através do conjunto de peças/componentes eletrônicos – Objetos de Hardware – do “Meu Mundo” e dos “Objetos de Programação” do “Meu Programa” com o seu respectivo “Programa do Modelo”.

O modelo criado pelo usuário-aluno na ferramenta ProgrameFácil poderá retratar ou planejar arquiteturas, casos, atividades, comportamentos, falhas existentes ou não, contribuindo assim para a estruturação do pensamento e posterior simulação acerca dessa modelagem que permitirá infinitas combinações, passíveis de averiguação no que tange à lógica definida.

Optou-se, na ferramenta ProgrameFácil, por não permitir que o *hardware* alvo – *hardware* que se pretende programar/controlar – seja alterado durante o desenvolvimento de um modelo pelo usuário. Dessa forma, almejam-se garantir a integridade do modelo que está sendo concebido com os recursos eletrônicos realmente existentes em determinado *hardware* de um kit de robótica.

Na prática, para se criar um modelo na linguagem computacional ProgrameFácil, é necessário realizar três etapas distintas: (1) especificar os “Objetos de Hardware” que serão utilizados na janela “Meu Mundo”; (2) incluir na janela “Meu Programa” os “Objetos de Hardware” e os “Objetos de Programação” que serão usados; e (3) estabelecer na janela “Meu Programa” o fluxo de execução através do objeto “Linha de Programação”.

Os manejos da interface a fim de realizar os passos descritos acima estão detalhados na ajuda presente no *software*.

3.2.6 Programa do Modelo (Programa)

O “Programa do Modelo”, ou simplesmente “Programa”, estabelece a forma de comportamento de uma configuração de um *hardware* de robótica e pode ser compreendido como o conjunto de objetos do “Meu Programa”, com as suas respectivas ações definidas, interligados por um fluxo de execução atemporal – fluxo constituído sem nenhuma referência temporal para execução, estabelecido pelo objeto “Linha de Programação”. O Programa inicia com o Objeto de Programação “Início de Programa” (sinal verde) e termina no Objeto de Programação “Fim de

Programa” (sinal vermelho), quando da sua simulação – mundo virtual – no *software* ProgrameFácil, ou execução – mundo real (concreto) – no *hardware* RoboFácil.

Demonstra-se, na Figura 8, um exemplo de modelo de comportamento construído no ProgrameFácil. Esse modelo possui dois *leds* (E1 e E2), um motor (M1), um sensor de temperatura (ST1), um sensor de luminosidade (SL1) e um *display* alfanumérico (D1). Na área 1 é apresentado o “Meu Mundo” com esses “Objetos de Hardware” e na área 2 o “Meu Programa” com os respectivos “Objetos de Hardware” e “Objetos de Programação” que compõem o “Programa do Modelo”.

O modelo exemplificado tem como objetivo acender o *led* E2, caso o sensor de luminosidade (SL1) esteja sobre incidência de luz. Quando da execução desse modelo no *hardware* RoboFácil ou da simulação desse modelo no *software* ProgrameFácil, se o referido sensor não estiver sob incidência de luz, a mensagem “Sem Luz” deverá ser apresentada no *display* (D1).

Observa-se, na Figura 8, que os semáforos que indicam o status do modelo estão com a luz inferior acesa (luz verde), indicando que o “Meu Mundo” está sendo simulado e que o “Meu Programa” está sendo executado.

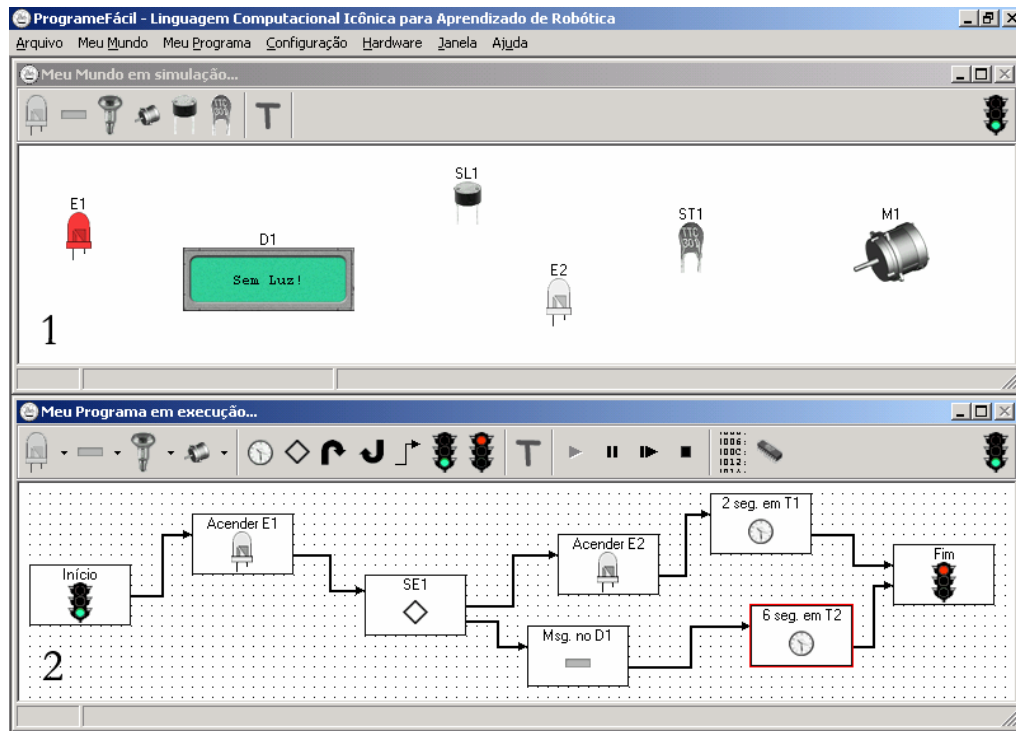


Figura 8 – Exemplo de um modelo de comportamento no ProgameFácil em simulação. Na área (1) é apresentado o “Meu Mundo” e na área (2) o “Meu Programa”

3.2.7 Simulador

O simulador foi desenvolvido e anexado ao *software* a partir da necessidade de tornar exequível – virtualmente – o modelo anteriormente criado na tela do monitor, antes de realizar a transferência do programa criado para o *hardware* RoboFácil. Essa funcionalidade, entre outras vantagens, visa proporcionar um momento intermediário de reflexão do programa criado entre os alunos, permitindo treinamento, organização e avaliação, e desenvolvendo provavelmente o senso crítico através da observação.

As características do simulador permitem o funcionamento de forma automática, após solicitação de início pelo usuário (Figura 9 – botão 1), ou interativamente – de forma pausada: passo-a-passo (Figura 9 – botão 3).



Figura 9 – Botões de controle do simulador. (1) Iniciar a simulação do Meu Mundo/Continuar com a simulação no Meu Mundo; (2) Pausar a simulação do Meu Mundo; (3) Realizar próximo passo da simulação do Meu Mundo; e (4) Interromper a simulação do Meu Mundo

Assim, há dois semáforos agregados ao ambiente – um na janela “Meu Mundo” e outro na janela “Meu Programa”, sendo esses posicionados ao lado direito, na barra de ferramenta de cada janela, que funcionam em conjunto, e representam sempre o mesmo estado. Sua função é indicar o *status* do modelo de comportamento: em desenvolvimento (sinal vermelho) em ambas as janelas, em pausa (sinal amarelo) em ambas as janelas, em simulação (sinal verde) na janela “Meu Mundo” e em execução (sinal verde) na janela “Meu Programa”.

Na janela “Meu Mundo” os objetos do *hardware* hipotético ou real, quando em simulação, vão alterando suas propriedades baseadas no programa do modelo desenvolvido. Conforme a simulação vai prosseguindo, na janela “Meu Programa”, um “Ponteiro de Execução” – retângulo vermelho que contorna os “Objetos de Hardware” e “Objetos de Programação” da janela “Meu Programa” (se configurado sua visualização pelo usuário) identifica o ponto atual de execução do programa.

O recurso de simulação incorporado à ferramenta, entre outras vantagens, minimiza a necessidade de se possuir um *hardware* para cada grupo de alunos. Tal característica contribui, entre outros, com a redução de custos na montagem de laboratórios de robótica. Entretanto, é importante ressaltar que a escola deve ter um número de kits suficientes para que todos os alunos participem do processo de construção/montagem e controle do *hardware* de robótica educacional.

3.2.8 Compilador

Segundo Machado e Maia (1999, p. 21), o compilador é: “[...] o utilitário responsável por gerar, a partir de um programa escrito [programa de um modelo] em uma linguagem de alto nível [linguagem icônica do ProgrameFácil], um programa em linguagem de máquina não executável (módulo-objeto). [*assembly* virtual do *hardware* RoboFácil]”.

Por conseguinte, para se converter o programa de um modelo desenvolvido na linguagem computacional visual ProgrameFácil nos macrocódigos *assembly* concebidos para o *hardware* do kit em questão (*assembly* virtual do *hardware* RoboFácil), integrou-se um compilador a essa ferramenta. A tradução é engendrada através da correspondência de macrocódigos *assembly* para cada “Objeto de Hardware” e/ou “Objeto de Programação” presentes no programa de um modelo, que em seguida será entendida pelo interpretador, componente do *firmware* do RoboFácil.

Os macrocódigos definidos são formados por dois blocos básicos: operador e operando(s). O operador é descrito como a junção do objeto com a ação planejada sobre esse, e o(s) operando(s) é caracterizado como uma propriedade do objeto. Por exemplo: qual motor deve ser ligado, qual *led* deve ser aceso, entre outros.

Tendo em vista o projeto atual, e prevendo futuras expansões, os macrocódigos *assembly* foram concebidos a partir das seguintes regras:

Primeira regra: cada instrução deverá ser composta de, no mínimo, 8 bits e, no máximo, 32 bits;

Segunda regra: os oito primeiros bits, obrigatórios, representam objeto e ação, sendo os quatro bits mais significativos a representação do objeto e os quatro bits menos significativos a ação exercida sobre o mesmo;

Terceira regra: As instruções de controle são definidas quando os quatro bits mais significativos possuírem valor zero;

Quarta regra: Cada instrução pode não ter operando ou até três operandos; e

Quinta regra: Para cada valor adicionado a um operando, a instrução deverá ser acrescida de 8 bits.

A Figura 10 ilustra mais nitidamente as regras definidas que especificam a formação de cada instrução *assembly* virtual compreendidas pelo *hardware* RoboFácil.

Instrução (máximo de 32 bits)																															
Operador (8 bits)								Grupos de Operandos (máximo de 24 bits)																							
Objeto (4 bits)				Ação (4 bits)				1.º Operando (8 bits)								2.º Operando (8 bits)								3.º Operando (8 bits)							
00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

Figura 10 – Representação binária das instruções do assembly virtual do hardware RoboFácil

Seguindo essas premissas, de modo a atender às exigências no que se refere a técnicas de desenvolvimento e depuração de *firmware*, foi convertida da base binário ou decimal para base hexadecimal (Anexo A) a codificação de todos os valores referentes aos objetos dessa linguagem.

As Tabelas 6, 7, 8, 9, 10, 11 e 12 apresentam os macrocódigos *assembly*, em hexadecimal, definidos.

Tabela 6 – Macrocódigos assembly do objeto Led (códigos iniciados com 1)

Código	Ação	1.º Operando	2.º Operando	3.º Operando	Instrução	
1	0	Apagar	Id. do Led	–	–	16 bits
1	1	Acender	Id. do Led	–	–	16 bits
1	F	Testar	Id. do Led	–	–	16 bits

NOTA: Id. representa a identificação numérica.

Tabela 7 – Macrocódigos assembly do objeto Display (códigos iniciados com 2)

Código	Ação	1.º Operando	2.º Operando	3.º Operando	Instrução	
2	0	Escrever Mensagem	End. Inicial (HI)	End. Inicial (LO)	Qnt. de Endereços	32 bits
2	1	Limpar	–	–	–	8 bits

NOTA: End. representa o endereço de memória; HI denota os oito bits de mais alta ordem; e LO os oito bits de mais baixa ordem; Qnt. representa a quantidade numérica.

Tabela 8 – Macro-códigos assembly do objeto Motor (códigos iniciados com 3)

Código	Ação	1.º Operando	2.º Operando	3.º Operando	Instrução	
3	0	Desligar	Id. do Motor	–	–	16 bits
3	1	Ligar	Id. do Motor	–	–	16 bits
3	F	Testar	Id. do Motor	–	–	16 bits

NOTA: Id. representa a identificação numérica.

Tabela 9 – Macro-códigos assembly do objeto Temporizador (códigos iniciados com 4)

Código	Ação	1.º Operando	2.º Operando	3.º Operando	Instrução	
4	0	Temporizar	Qnt. de Tempo (HI)	Qnt. de Tempo (LO)	–	24 bits

NOTA: Qnt. representa a quantidade numérica; HI denota os oito bits mais significativos; e LO os oito bits menos significativos.

Tabela 10 – Macro-códigos assembly do objeto Sensor de Luminosidade (códigos iniciados com 5)

Código	Ação	1.º Operando	2.º Operando	3.º Operando	Instrução	
5	F	Testar	–	–	–	8 bits

Tabela 11 – Macro-códigos assembly do objeto Sensor de Temperatura (códigos iniciados com 6)

Código	Ação	1.º Operando	2.º Operando	3.º Operando	Instrução	
6	F	Testar	–	–	–	8 bits

Tabela 12 – Macro-códigos assembly dos códigos de controle de execução (códigos iniciados com 0)

Código	Ação	1.º Operando	2.º Operando	3.º Operando	Instruções	
0	0	Parar Programa	–	–	–	8 bits
0	1	Desviar Para	End. (HI)	End. (LO)	–	24 bits
0	2	Carregar Contador	Id. do Contador	Valor (HI)	Valor (LO)	32 bits
0	3	Decrementar Contador	Id. do Contador	–	–	16 bits
0	4	Desviar Se Flag	End. (HI)	End. (LO)	–	24 bits
0	5	Desviar Se Não Flag	End. (HI)	End. (LO)	–	24 bits

NOTA: End. representa o endereço de memória; Id. representa a identificação numérica; HI denota os oito bits mais significativos; e LO os oito bits menos significativos.

Evidencia-se a preocupação, na interação com o usuário, em fornecer qualidade e conforto na utilização da ferramenta, minimizando pré-requisitos como conhecimentos

avançados de programação. Assim, o processo de compilação do programa de um modelo é realizado apenas com um único *click* do mouse sobre o botão correspondente a essa funcionalidade (Figura 11). Depois de concluído esse processo o compilador pode apresentar na interface uma janela (Figura 12) – configurável pelo usuário – informando o resultado obtido: compilação realizada com sucesso ou compilação realizada com erro(s).



Figura 11 – Botão que ativa o processo de compilação

Visando a mesma proposta de transparência da ferramenta, as informações adicionais apresentam comentários – com cores distintas por função – detalhando a conversão dos objetos em macrocódigos *assembly* e os motivos pelos quais erros podem ter sido introduzidos no programa desenvolvido, sinalizando as correções necessárias a serem realizadas.

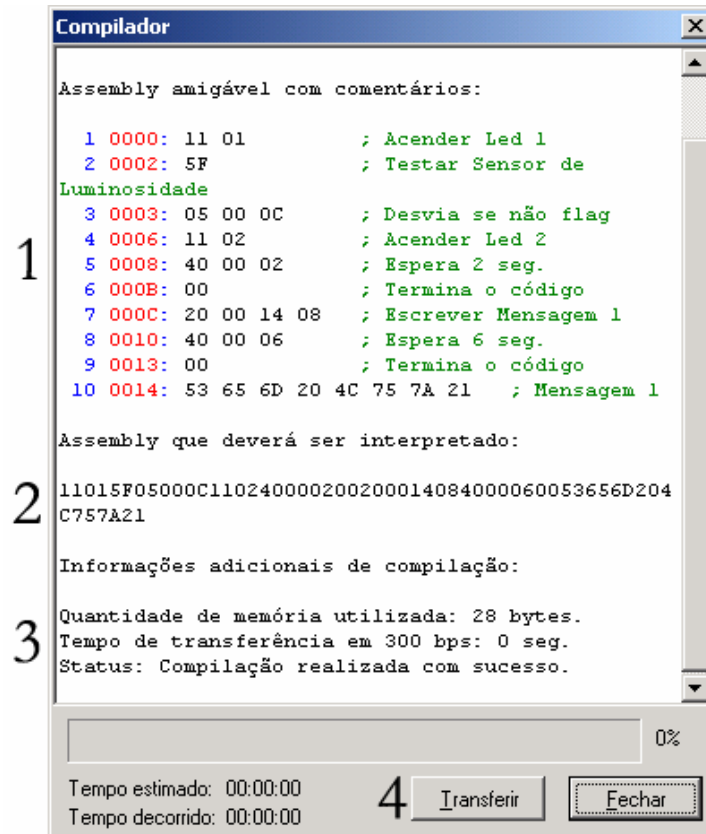


Figura 12 – Compilação do exemplo de modelo de comportamento da Figura 8. (1) Assembly amigável com comentários; (2) Assembly que deverá ser interpretado pelo hardware RoboFácil; (3) Informações adicionais de compilação; e (4) Botão que inicia a transferência do assembly virtual para o hardware RoboFácil

Pensando na forma de interação entre o *software* ProgrameFácil e o *hardware* RoboFácil, implementou-se funcionalidades (Figura 12 – item 4) que permitem a transferência do *assembly* virtual gerado pelo compilador da ferramenta ProgrameFácil, diretamente para o *hardware*, via interface serial do microcomputador. Portanto, não é necessário utilizar *software* adicional para realizar a comunicação serial entre o microcomputador e o *hardware* RoboFácil.

3.2.9 Hardware Perfeito

Estruturalmente, o desenvolvimento do *software* ProgrameFácil foi pautado de forma a permitir modularidade aos recursos inseridos, tornando simples estender novas funcionalidades ao referido *software* e sua futura personalização para ser utilizado com outros kits de robótica.

Portanto, faz-se indispensável, para que essa ferramenta possa concretizar a programação e o controle de outros *hardwares* de robótica, ampliar suas regras de compilação.

Objetivando também promover liberdade quantitativa de componentes eletrônicos de um *hardware* hipotético/ideal – esses recursos são inseridos na janela “Meu Mundo” – durante a especificação de modelos, ofereceu-se a possibilidade de seleção do *hardware* alvo de utilização, nesse caso denominado Hardware Perfeito. Esse recurso tornou o uso do *software* ProgameFácil independente de qualquer *hardware*, permitindo a criação de modelos de comportamento sem limitação na quantidade dos “Objetos de Hardware” que um usuário possa querer fazer uso durante a especificação de um *hardware* supostamente “ideal”.

3.2.10 Interface

Métodos e diretrizes para projetos de interfaces da área de Interação Homem-Máquina (HMI) são aqui utilizados com o propósito de tornar o *software* ProgameFácil uma ferramenta útil e facilmente manipulada pelo público alvo dessa pesquisa.

Estudos evidenciam que a eficiência do usuário na manipulação de um programa está diretamente relacionada com a influência da sua interface sobre o indivíduo (QUEIROZ, L.R. et al., 1998). Sendo essa constatação uma das grandes preocupações no que concerne a definição da interface do ProgameFácil, foi definido para o mesmo o “design centrado no aprendiz”, para que a concretização do objetivo educacional pudesse ser alcançado pelos usuários da robótica na educação.

Dessa forma, e em termos gerais, foram explorados elementos gráficos propositalmente direcionados ao público alvo dessa dissertação – estudantes do ensino médio das escolas brasileiras. Empregaram-se características fundamentais e diferenciais a esse projeto, como o idioma – Língua Portuguesa, as imagens coloridas e atuais, as mesmas metáforas empregadas em aplicativos de uso comum, os erros *default* gerados pelo Delphi – esses foram interpretados e

personalizados ao contexto da aplicação – e a inclusão de caixas de diálogo que visam interagir com o usuário-aluno.

A interface de múltiplos documentos (MDI) da ferramenta em conjunto com seus dois principais MDI *Child* – janelas “Meu Mundo” e “Meu Programa” – proporcionam a criação de um ambiente amigável por sua simplicidade e perceptibilidade. Objetivando tornar o ambiente intuitivo, optou-se por disponibilizar a janela “Meu Mundo”, normalmente, na parte superior e a janela “Meu Programa” na parte inferior da ferramenta. Cada janela possui uma barra de ferramenta com botões, contendo representações gráficas (figuras), que visam facilitar o entendimento das funcionalidades específicas de cada botão, além dos mesmos estarem agrupados logicamente nessa barra, que, supõe-se, aumentará a eficiência da comunicação explicitada pelo usuário-aluno nesse ambiente.

Valorizando também as experiências previamente adquiridas pelos usuários em outros ambientes, por exemplo, o sistema operacional Windows, escolhas foram feitas no que tange a definição da interface, como o emprego de menus “drop-down” e a possibilidade de movimentar os objetos apenas com o arrastar e soltar do mouse.

Sabendo da importância de visualização das características dos objetos – como a cor de um *led*, seu estado: ligado ou desligado, foram incluídas uma barra de *status* e um semáforo – indicativo do *status* do modelo: em desenvolvimento, em pausa ou em execução/simulação – em cada janela – Meu Mundo e Meu Programa.

Salienta-se que os objetos presentes na janela “Meu Mundo” não podem ser interligados, ou seja, conectados com outros objetos. O mesmo não acontece no “Meu Programa”, visto ser necessária a ligação através do objeto “Linha de Programação” para o estabelecimento do fluxo de execução do programa implementado no modelo.

A ferramenta também possui recursos de interface que permitem ao usuário receber um *feedback* dinâmico do *software*, com sinalizações de diversos *status* dos recursos disponíveis no ambiente, incluindo mensagens informativas nos tempos de *design*, compilação e simulação. Essas

concepções e características permearam a implementação dessa forma de interação com a interface, que teve como objetivo proporcionar um ambiente de fácil aprendizado (*learnability*) e com facilidades para ser lembrado (*memorability*) tanto pelos professores como pelos alunos – usuários.

3.2.11 Direitos Autorais/Custo

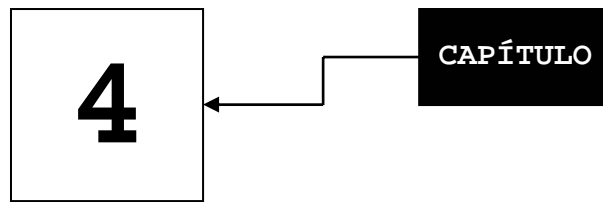
O ProgameFácil, sendo um *software* desenvolvido dentro de um programa de Pós-Graduação em Informática de uma Universidade pública e tendo o referido Curso financiamento parcial do Governo para sua realização, garante à Universidade os direitos autorais sobre o referido programa, tornando presumível a sua livre distribuição, sem custos adicionais no que tange ao licenciamento de *software*, para o mais diversificado público usuário – provavelmente professores e alunos dos diversos níveis do ensino (ROBOFÁCIL, 2005).

Após apresentar as motivações e justificativas que resultaram na idealização e implementação do *software* educacional nomeado de ProgameFácil, foram descritos, em detalhes, os elementos presentes nesse ambiente de programação visual.

Essa ferramenta pode ser resumida como a forma mais intuitiva e simplificada de programar, atualmente, o *hardware* RoboFácil pelo público alvo definido para essa pesquisa. Todavia, se existir interesse em empregar esse kit de robótica com alunos de outros níveis do ensino, integrando outros tópicos do currículo escolar à prática do ensino mediada pela robótica, o docente pode sugerir aos seus discentes utilizar outros ambientes de programação já disponíveis no mercado para programar o *hardware* RoboFácil, podendo fazer uso das linguagens de programação: Assembly, C, Basic, entre outras.

Para que os modelos de comportamento especificados pelo usuário na ferramenta ProgameFácil pudessem ser efetivamente executados no *hardware* RoboFácil, foi necessário

desenvolver um conjunto de rotinas básicas que passaram a compor o *firmware* do referido *hardware*. Tais rotinas são discutidas no capítulo a seguir.



FIRMWARE

O Capítulo 4 apresenta a descrição e a justificativa que impulsionaram a criação de um conjunto de rotinas, que integrado ao *hardware* RoboFácil, permita facilidades e velocidade na transferência dos modelos de comportamento desenvolvidos na ferramenta ProgrameFácil, ou diretamente via sua interface textual com o usuário.

4.1 Motivações e Justificativas

Tendo como horizonte o público alvo dessa pesquisa, observou-se que a dificuldade de utilização – programação – do Kit de Robótica Educacional RoboFácil poderia ser elevada, visto que, inicialmente, novas formas de comportamento – programas de um modelo – eram codificadas apenas utilizando a linguagem *assembly* do MCU adotado no projeto desse *hardware*. Um outro aspecto que poderia comprometer o projeto como um todo era a reduzida velocidade de transferência dos programas desenvolvidos entre o microcomputador e o referido *hardware*.

Para contornar ou sanar os problemas descritos e associar outras facilidades ao Kit, optou-se pelo desenvolvimento de um novo *firmware* em duas etapas, conforme proposto pelos orientadores. A primeira foi composta de um estudo aprofundado sobre as ferramentas disponíveis no mercado que permitissem gerar código binário no formato Intel HEX (INTEL, 1988) para o MCU utilizado no *hardware* RoboFácil, fazendo uso da linguagem *assembly* ou C.

Essa primeira etapa também foi constituída pelo desenvolvimento das bibliotecas básicas para controlar todos os recursos eletrônicos disponíveis no citado *hardware* – descrito em detalhes no próximo capítulo, tais como *leds*, *display*, motores de passo, sensor de luminosidade e sensor de temperatura, fazendo uso da linguagem C. As mesmas vantagens que impulsionaram essa escolha para implementar as bibliotecas foram, outrora, comentadas por Pech (2002, p. 5), “[...] nesta nova base de desenvolvimento [de software para microcontroladores] pode-se fazer uso de linguagens mais estruturadas, mais estáveis e de manutenção mais fácil que o *assembly*, como as linguagens C, C++, e ADA, por exemplo.”

O IDE escolhido para implementar o código-fonte do *firmware* na linguagem C foi o Keil C51 Professional Developers Kit versão 7.50a (KEIL, 2005). Esse ambiente de desenvolvimento é constituído pela integração de uma diversidade de ferramentas – compiler, debug, breakpoint, performance analyzer, memory map, inline assembly, disassembly windows, entre outras –

notadamente úteis aos desenvolvedores de bibliotecas básicas para *firmwares* das mais diversificadas marcas de microcontroladores disponíveis no mercado.

Segundo Cardoso de Sá (2005, p. 35),

A Keil desenvolveu para a linha Intel 8051 um dos mais eficientes conjuntos de ferramentas para essa família de microcontroladores.

Por meio do μ Vision2 IDE podemos executar todos os passos necessários para a criação, gerenciamento do projeto, teste (debug) e finalização de um sistema, com a geração do arquivo HEX necessário para a gravação dos programas nos chips derivados do 8051.

A segunda etapa teve como objetivo a implementação do interpretador e sua integração com as bibliotecas básicas, de modo a permitir que o usuário não precise mais programar o citado *hardware* diretamente pelos comandos *assembly* entendidos pelo seu microcontrolador, mas que por meio desse recurso – interpretador, possa fazer uso das instruções definidas no *assembly* virtual do *hardware* RoboFácil.

A programação do conjunto de rotinas que compõem o *firmware* do *hardware* RoboFácil se assemelha ao desenvolvimento de *software* embarcado (*embedded software*), pois temos no Kit de Robótica Educacional todos os elementos que caracterizam esse tipo de sistema.

Um sistema embarcado pode ser entendido como a combinação de *hardware* e *software* de computadores, e dispositivos mecanismos projetados para desempenhar uma função específica (BARR, M., 1999, p. 1). Gandour, no prefácio do livro “Software Embarcado: A nova onda da Informática” (TAURION, C., 2005) expressa algumas considerações sobre *software* embarcado.

[...] em geral, ao software embarcado é reservado um espaço de memória exíguo e, portanto, ele precisa ter um [uma] estrutura de programação minimalista. [...] É software para gente criativa e econômica. É software para quem sabe conviver na escassez. De espaço de memória, de meios de armazenamento e de capacidade de CPU.

Portanto, para implementar esse *firmware*, utilizou-se alguns conceitos e técnicas normalmente empregados no desenvolvimento de *software* embarcado, onde questões nitidamente relacionadas com as limitações de processamento, memória, interface, entre outras, que estão intrinsecamente ligados a esse *hardware*, tiveram que ser explicitamente colocadas em pauta.

4.2 Firmware

A criação dessa camada intermediária de *software* – *firmware* – entre o *hardware* RoboFácil e os modelos a serem criados garante facilidades para o usuário na utilização do referido *hardware*, pois extrai do usuário a responsabilidade de conhecer questões pontuais do *hardware* RoboFácil e sua programação em linguagem de baixo nível, tornando transparente o controle de todos os dispositivos eletrônicos/eletromecânicos presentes nesse equipamento.

Todas as funcionalidades implementadas para essa solução foram agrupadas permitindo, assim, sua utilização de forma simplificada pelo usuário, generalizado para um único *click* do mouse. Ou seja, circunstancialmente, através do *software* ProgrameFácil é realizada a transferência do novo *firmware* pelo pressionamento do botão (Figura 13) que representa a ação de transmissão, exclusivamente, desse recurso ao *hardware*. Depois de concluída a transferência, automaticamente o interpretador é executado, passando a assumir o controle do *hardware*, e aguardando o envio de programas para serem interpretados. É importante ressaltar que esse *firmware* só precisa ser transferido uma única vez, assim que o *hardware* RoboFácil for ligado.



Figura 13 – Botão de transferência do firmware do RoboFácil presente na ferramenta ProgrameFácil

A metodologia adotada para dar origem ao novo *firmware* do RoboFácil baseou-se no paradigma dos sistemas operacionais em camadas. Nessa classe de sistemas uma camada sobrepõe a outra, com a característica de que cada uma possui módulos para proporcionar um conjunto de funções a serem utilizadas apenas por outros módulos das camadas inferiores (MACHADO & MAIA, 1999, p. 62). Sendo assim, o novo *firmware* foi implementado seguindo semelhante estratégia, com o objetivo de isolar funções e facilitar futuras depurações e/ou

incorporações de novas funcionalidades para controle/integração entre os artefatos resultantes dessa pesquisa. Na Figura 14 demonstra-se as camadas implementadas no projeto desse novo *firmware*.

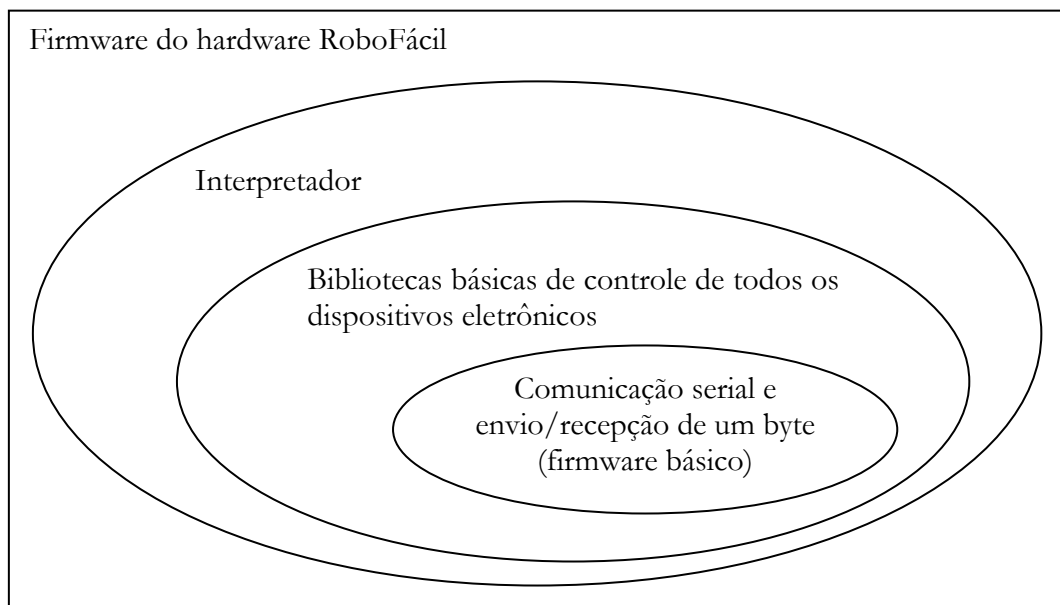


Figura 14 – Camadas do firmware do RoboFácil

A camada mais interna (primeira) do *firmware* – Comunicação serial e envio/recepção de um byte – (*firmware* básico) compõem o *bootstrap*¹⁶ do *hardware* RoboFácil. Essas rotinas foram escritas na linguagem *assembly*, durante a primeira fase desse projeto, e estão gravadas numa EPROM¹⁷. Posteriormente, diversas funcionalidades presentes nessa camada foram reescritas na linguagem C objetivando criar facilidades de cunho operacional quando da sua integração com as outras camadas.

A segunda camada – bibliotecas básicas de controle de todos os dispositivos eletrônicos – e a terceira camada – interpretador – foram escritas diretamente na linguagem C. Sendo assim, as

¹⁶ Bootstrap, segundo o Dicionário Houaiss da Língua Portuguesa é a: “operação que dá início ao funcionamento de um computador.” (HOUAISS, A., 2001, p. 488)

¹⁷ Para Azevedo Júnior (1984, p. 369), “As memórias EPROM são as mais populares das ROM’s. Elas podem ser programas pelo usuário e, se necessário, apagadas e novamente programadas, sendo possível repetir o processo um número ilimitado de vezes. Esta característica torna a utilização das memórias EPROM especialmente indicada durante o desenvolvimento de um projeto.”

três camadas que compõem o *firmware* são transferidas – carregadas – na memória do *hardware* através do acionamento de uma funcionalidade específica do *software* ProgrameFácil. Esse procedimento deve ser realizado após o *hardware* RoboFácil ser ligado.

A gravação na EPROM do novo *firmware* não foi realizada visando proporcionar facilidades para adicionar novas funcionalidades ao *software* ProgrameFácil e *hardware* RoboFácil. Portanto, o *hardware* RoboFácil ainda está fazendo uso da EPROM – firmware básico – gravado na primeira versão desse *hardware*, que após sua inicialização permite que o RoboFácil receba novos programas via interface serial, inclusive esse novo *firmware*.

Dessa forma, é mantida a flexibilidade para realizar futuras atualizações no ProgrameFácil, pois a inclusão de alguns “Objetos de Hardware” e/ou “Objetos de Programação”, nessa ferramenta, poderá provocar a necessidade de novas alterações no *firmware*.

O fato do novo *firmware* ser transferido do microcomputador para a memória RAM de programa do *hardware* RoboFácil, através do *software* ProgrameFácil, para sua posterior inicialização, possibilitou o abandono de *software* de comunicação serial, tal como o HyperTerminal (HILGRAEVE, 2005), que permitia a transferência do *firmware* e programas para o *hardware* RoboFácil.

Salienta-se que se o processo de carregamento de um novo *firmware* não fosse realizado via *software*, com a alteração nas rotinas, haveria a necessidade de regravar a EPROM, e essa ROM ainda deveria ser instalada, fisicamente, no *hardware* do Kit. Essas tarefas, para a maioria dos usuários – alunos e professores – que utilizarão o Kit RoboFácil, não é intuitivo e nem trivial, necessitando de algumas habilidades específicas e conhecimentos de eletrônica para sua concretização.

A camada referente ao interpretador foi idealizada para satisfazer a transcrição do *assembly* virtual do *hardware* RoboFácil, permitindo assim a execução de instruções que garantam o “entendimento” pelo referido *hardware*. Dessa forma, as tradicionais “palavras reservadas da linguagem” presentes na maioria dos *assembly* igualmente serão processadas, tais como as

instruções designadas: Desviar Para (01h), Carregar Contador (02h), Decrementar Contador (03h), Desviar Se Flag (04h), Desvir Se Não Flag (05h), entre outras.

Apresenta-se na arquitetura da Figura 15, para maior entendimento, os níveis e fluxos dos processos realizados quando da utilização integrada do *hardware* RoboFácil com seu ambiente visual de programação – *software* ProgrameFácil.

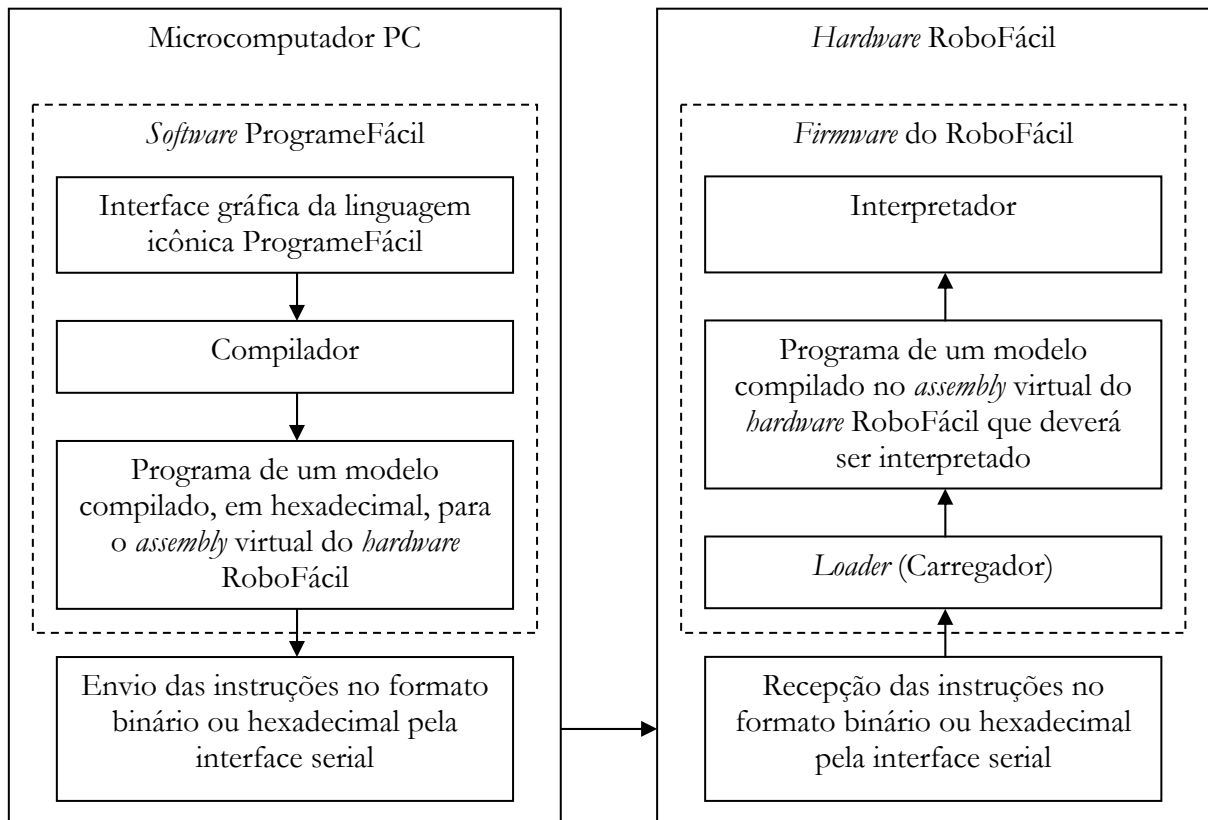


Figura 15 – Níveis funcionais entre o software ProgrameFácil e o firmware do hardware RoboFácil

O lado referente ao *software* ProgrameFácil foi explorado no Capítulo 3 e a partir da próxima seção estará sendo descrito o *firmware* do RoboFácil.

4.2.1 Loader (Carregador)

Segundo Machado e Maia (1999, p. 23), o *loader*, também chamado de carregador é: “[...] o utilitário responsável por colocar fisicamente na memória um programa [programa de um modelo convertido para o *assembly* virtual do *hardware* RoboFácil] para execução.”.

Por conseguinte, para colocação das instruções em formato hexadecimal do seu *assembly* virtual na memória RAM de dados do *hardware* RoboFácil – transferidos via interface serial do microcomputador para o mesmo – foi incorporado ao *firmware* um novo *loader*. Assim, após a alocação de um programa na memória do referido *hardware* para execução, o *hardware* fica aguardando um comando do usuário – pressionamento de um dos botões do teclado do mencionado *hardware* – para iniciar a execução do programa previamente carregado na sua memória.

4.2.2 Interpretador

Segundo Machado e Maia (1999, p. 21), o interpretador é: “[...] considerado um tradutor que não gera código-objeto. A partir de um programa-fonte, escrito em linguagem de alto nível [programa de um modelo compilado pela ferramenta ProgrameFácil], o interpretador, no momento da execução do programa, traduz cada instrução e a executa em seguida.”.

Sendo essa a definição base para que o *hardware* RoboFácil tivesse a capacidade de entender e executar cada modelo de comportamento especificado pelo usuário, foi necessário agregar um interpretador ao *firmware* desse artefato, tornando a execução de programas desenvolvidos pelo usuário, através do ambiente gráfico de programação (ProgrameFácil), uma realidade exequível.

A execução de um programa, instrução a instrução, só é possível a partir da prévia alocação de espaço na memória RAM de dados. Em seguida, o interpretador entenderá os

macrocódigos enviados pelo ProgameFácil quando da compilação de modelos, permitindo que o *hardware* RoboFácil entenda um fluxo lógico de execução.

O *assembly* virtual do *hardware* RoboFácil possui dezessete (17) instruções definidas, sendo quatro (04) instruções de 8 bits, sete (07) instruções de 16 bits, quatro (04) instruções de 24 bits e duas (02) instruções de 32 bits. Esse conjunto de instruções com seus respectivos parâmetros e função desempenhada podem ser visualizadas na Tabela 13.

Tabela 13 – Instruções interpretadas pelo firmware

Instrução						
Código (1.º byte)	1.º Operando (2.º byte)	2.º Operando (3.º byte)	3.º Operando (4.º byte)	Tamanho	Função	
1	0	Id. do Led	–	–	16 bits	Apagar Led
1	1	Id. do Led	–	–	16 bits	Acender Led
1	F	Id. do Led	–	–	16 bits	Testar Led
2	0	End. Inicial (HI)	End. Inicial (LO)	Qnt. de Endereços	32 bits	Escrever Mensagem no Display
2	1	–	–	–	8 bits	Limpar Display
3	0	Id. do Motor	–	–	16 bits	Desligar Motor
3	1	Id. do Motor	–	–	16 bits	Ligar Motor
3	F	Id. do Motor	–	–	16 bits	Testar Motor
4	0	Qnt. de Tempo (HI)	Qnt. de Tempo (LO)	–	24 bits	Temporizar
5	F	–	–	–	8 bits	Testar Sensor de Luminosidade
6	F	–	–	–	8 bits	Testar Sensor de Temperatura
0	0	–	–	–	8 bits	Parar Programa
0	1	End. (HI)	End. (LO)	–	24 bits	Desviar Para
0	2	Id. do Contador	Valor (HI)	Valor (LO)	32 bits	Carregar Contador
0	3	Id. do Contador	–	–	16 bits	Decrementar Contador
0	4	End. (HI)	End. (LO)	–	24 bits	Desviar Se Flag
0	5	End. (HI)	End. (LO)	–	24 bits	Desviar Se Não Flag

NOTA: Id. representa a identificação numérica; End. representa o endereço de memória; Qnt. representa a quantidade numérica; HI denota os oito bits mais significativos; e LO os oito bits menos significativos.

É importante mencionar que esse interpretador possui algumas variáveis que são de suma importância para seu funcionamento, tal como a variável “flag”, utilizada para armazenar o

resultado de um teste lógico, permitindo identificar o caminho que deve ser seguido pelo “ponteiro” do interpretador no programa previamente estabelecido. Nas convenções adotadas na criação do ProgrameFácil (vide 3.2.8), os objetos que possuem seus quatro bits de ação com o valor um – Fh – apresenta a capacidade de realizar a ação “Testar”, que quando interpretada, carrega o estado verdadeiro ou falso nessa variável, dependendo do caso.

Como exposto na Tabela 13, as instruções de controle são aquelas onde os quatro bits mais significativos possuem valor zero (0h). Para exercer sua função o interpretador segue a seguinte regra: o interpretador após identificar o código da instrução verifica se o mesmo possui parâmetro(s). Caso possua, selecionada o(s) parâmetro(s) e executa. Não existindo parâmetro(s), executa diretamente a instrução.

As estruturas apresentadas nas Tabelas 14, 15 e 16 demonstram a utilização das instruções de controle de execução do *assembly* virtual do *hardware* RoboFácil para contemplar a criação de estruturas condicional e de repetição na formulação de um programa de um modelo.

Será descrita na Tabela 14 a seqüência de instruções mínimas necessárias para se criar um looping finito.

Tabela 14 – Seqüência de instruções mínimas para se criar um looping finito

Seqüência de Instruções	Descrição da Instrução
...	Instruções antes do looping finito
02 XX YY YY	Carrega Contador XX com valor YYYY
03 XX	Decrementa um (1) ao Contador X; e atualiza flag se necessário
04 ZZ ZZ	Desvia para o endereço de memória ZZZZ (endereço inicial das instruções após o looping finito) se flag igual a verdadeiro (Desvia se flag)
...	Bloco de instruções que será executado YYYY vezes
01 WW WW	Desvia para o endereço de memória WWW (endereço inicial da instrução que decrementa um (1) ao Contador X
...	Instruções após o looping finito

NOTA: X, Y, Z e W são valores na base hexadecimal.

Segue na Tabela 15 a seqüência de instruções mínimas necessárias para se criar um looping infinito.

Tabela 15 – Seqüência de instruções mínimas para se criar um looping infinito

Seqüência de Instruções	Descrição da Instrução
...	Instruções antes do looping infinito
...	Bloco de instruções que será executado infinitas vezes
01 WW WW	Desvia para o endereço de memória WWWW (endereço inicial do bloco de instruções que será executado infinitas vezes)
...	Instruções após o looping infinito

NOTA: W são valores na base hexadecimal.

Segue na Tabela 16 a seqüência de instruções mínimas necessárias para se criar uma estrutura condicional Se.

Tabela 16 – Seqüência de instruções mínimas para se criar uma estrutura condicional Se

Seqüência de Instruções	Descrição da Instrução
...	Instruções antes da estrutura condicional Se
XF YY	Testa o objeto X, se for o caso, com a identificação YY
05 ZZ ZZ	Desvia para o endereço de memória ZZZZ (endereço inicial das instruções após o looping finito) se flag igual a falso (Desvia se não flag)
...	Bloco de instruções que será executado se teste do flag for igual a verdadeiro
00	Termina a execução
...	Bloco de instruções que será executado se teste do flag for igual a falso
00	Termina a execução

NOTA: X, Y e Z são valores na base hexadecimal.

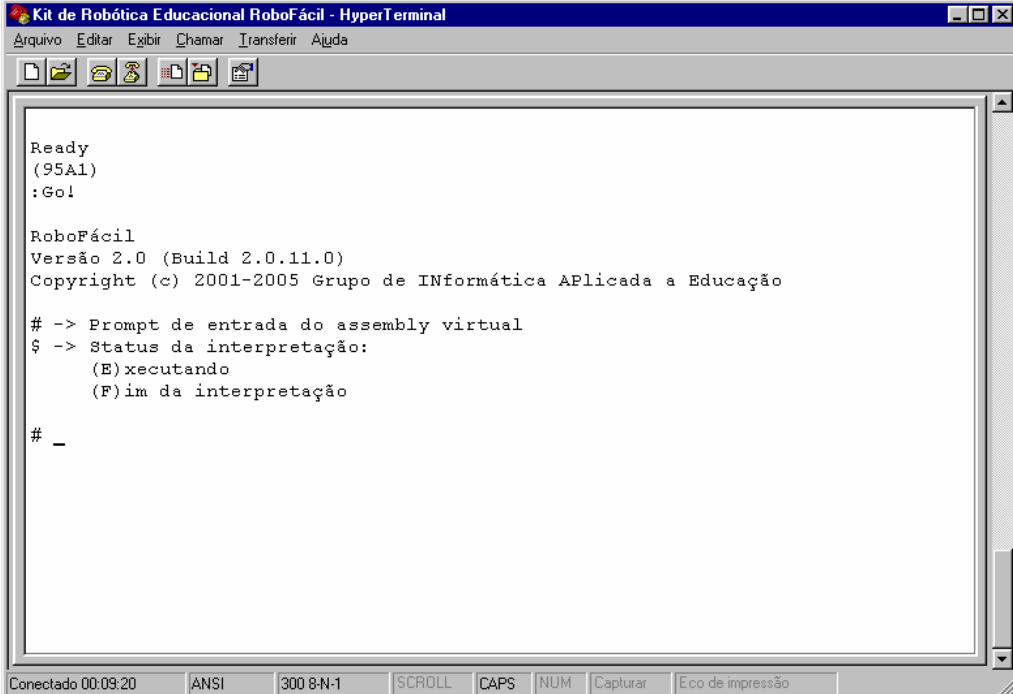
Conforme exposto, o *assembly* virtual especificado para o *hardware* RoboFácil possui algumas limitações. A Tabela 17 apresenta algumas dessas restrições.

Tabela 17 – Valores limites dos recursos utilizados no assembly virtual do hardware RoboFácil

Recurso	Valor Limite
Contadores	255 (FFh)
Leds	255 (FFh)
Motores	255 (FFh)
Quantidade de caracteres do Display	255 (FFh)
Valor de um Contador	65.535 (FFFFh)
Valor de um Temporizador (em segundos)	65.535 (FFFFh)

4.2.3 Interface

Como uma alternativa ao ambiente visual de programação do *hardware* RoboFácil (ProgrameFácil), implementou-se nesse *firmware* uma interface baseada em texto para controlar/programar esse *hardware*. Sendo assim, buscou-se desenvolver uma interface textual clara e objetiva, inclusive, com sinalização dos diferentes *status* possíveis do equipamento, quando da interação do usuário com o *hardware* RoboFácil por meio de um *software* de comunicação serial.



```

Ready
(95A1)
:Go!

RoboFácil
Versão 2.0 (Build 2.0.11.0)
Copyright (c) 2001-2005 Grupo de INformática Aplicada a Educação

# -> Prompt de entrada do assembly virtual
$ -> Status da interpretação:
    (E)xecutando
    (F)im da interpretação

# _
  
```

Figura 16 – Interface textual fornecida pelo firmware

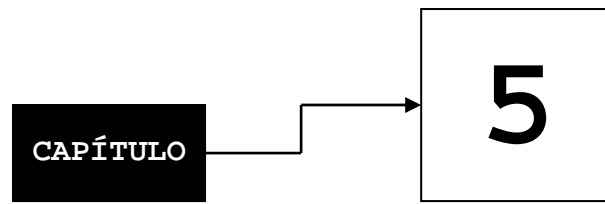
Sabendo que a interface com o usuário projetada traz mecanismos para a programação do *hardware* RoboFácil diretamente nos macrocódigos *assembly*, especificados para a linguagem de máquina virtual desse *hardware*, seria plausível pensar num exemplo de aplicação pedagógica desse Kit com público alvo diferente do especificado para essa pesquisa – alunos do ensino médio. Por exemplo, estudantes poderiam utilizar esse equipamento, fazendo uso apenas desse nível primitivo de interação com o usuário, em disciplinas de Graduação e Pós-Graduação em Ciência da Computação, tais como Organização de Computadores e Compiladores.

4.2.4 Direitos Autorais/Custo

O *firmware* do *hardware* RoboFácil é um conjunto de rotinas também desenvolvidas em um programa de Pós-Graduação em Informática de uma Universidade pública. Como o referido Curso tem financiamento parcial do governo para sua realização, é garantido à Universidade direitos autorais sobre as rotinas implementadas desse *firmware*. No entanto, sua distribuição será feita sem custo adicional para os usuários.

Após a apresentação das motivações e justificativas que alavancaram o desenvolvimento de um novo *firmware* para o *hardware* RoboFácil, foram apresentados, em detalhes, os diferentes recursos para criação de uma interface homem-computador simplificada entre o *software* ProgrameFácil e o *hardware* RoboFácil.

No próximo capítulo serão explorados os recursos eletrônicos do *hardware* RoboFácil que é controlado pelo *firmware* apresentado nesse capítulo.



ROBOFÁCIL

Nesse capítulo será feita a descrição detalhada dos diferentes elementos que compõem o artefato de *hardware* do Kit de Robótica Educacional RoboFácil.

5.1 Motivações e Justificativas

A existência de um equipamento parcialmente implementado e a possibilidade do emprego desse artefato tecnológico em projetos de aprendizado que façam uso de conceitos científicos e dispositivos da robótica permitiu a continuidade do desenvolvimento desse *hardware*.

Visando melhor aproveitamento e definição de sua aplicabilidade como ferramenta educacional em sala de aula, elementos adicionais – *plugins* eletrônicos¹⁸, ou simplesmente *plugins* – foram incluídos nesse artefato. Especificamente, foram implementados e incorporados ao *hardware* RoboFácil quatro *plugins* com diferentes objetivos de controle: sensor de luminosidade, sensor de temperatura, *leds* e motores de passo.

O *firmware* desenvolvido – descrito no capítulo anterior – e carregado a posteriori após a inicialização do *hardware* RoboFácil, proporciona a execução, nesse *hardware*, da programação realizada pelo “usuário não especialista” no *software* ProgrameFácil.

A arquitetura eletrônica adotada por esse *hardware* o torna diferenciado em relação à maioria dos kits de robótica para fins pedagógicos fabricados no Brasil, pois o mesmo possibilita ter seu comportamento alterado uma infinidade de vezes através da (re)programação via *software* – ProgrameFácil, não necessitando, portanto, da alteração eletrônica para mudança de comportamento. Essa característica proporciona flexibilidade de utilização em ambiente escolar.

Uma outra particularidade desse *hardware* é a possibilidade de manter esse equipamento desconectado do microcomputador quando da realização da programação¹⁹ do seu

¹⁸ Nesse projeto, um *plugin* eletrônico pode ser definido como um *hardware* externo ao circuito principal, que possui a flexibilidade de ser acoplado com uma certa facilidade ao sistema, sem que exista a necessidade de alteração no projeto eletrônico para utilizá-lo.

¹⁹ A concepção das soluções tecnológicas de *hardware* e *software* que compõem o Kit de Robótica Educacional RoboFácil permite que se faça uso do Método de Programação Off-line para realizar a programação do *hardware* RoboFácil. Segundo Bertotto e Güntzel (2000, p. 2), esse tipo de programação: “[...] permite programar um robô [hardware de um kit de robótica] sem que o mesmo esteja fisicamente conectado ao sistema [microcomputador] [...]”.

comportamento e execução de um programa, previamente concebido e transferido do microcomputador para o *hardware* em questão.

Serão pormenorizadas, a seguir, a arquitetura eletrônica, características e limitações tecnológicas desse artefato.

5.2 RoboFácil

5.2.1 Arquitetura Geral

O *hardware* do Kit de Robótica Educacional RoboFácil é baseado em uma CPU da família dos microcontroladores MCS-51 da Intel (INTEL, 2005b) e contém uma extensão de 16 portas digitais de entrada e 16 portas de saída com *buffer*.

O sistema provê a utilização de circuitos externos – denominados *plugins* – que permitem a interação do sistema com o mundo externo. Um plugin acoplado ao *hardware* pode assim controlar, por exemplo, motores de passo, sensor de luminosidade, sensor de temperatura, etc.

Os principais elementos eletrônicos disponíveis no *hardware* do kit são:

- ❑ CPU – Intel 8031AH (CPU CMOS de baixo consumo);
- ❑ 32 KB de memória ROM de programa e 32 KB de memória RAM de programa;
- ❑ 8 KB de memória RAM de dados;
- ❑ Interface de comunicação RS-232C utilizando conector externo DB-25;
- ❑ Mostrador (*display*) de 01 (uma) linha com 16 (dezesseis) colunas;
- ❑ Teclado (botoeira) com 05 (cinco) botões;
- ❑ Controle de 02 (dois) motores de passo;
- ❑ Controle de motores DC (apenas projetado/não implementado);
- ❑ 01 (um) sensor de temperatura;
- ❑ 01 (um) sensor de luminosidade;

- ❑ 01 (um) conversor digital-analógico; e
- ❑ 01 (um) conversor analógico-digital.

5.2.2 Unidade de Controle e Sistema de Memória

As características da unidade de controle derivam-se das propriedades operacionais das CPUs da linha Intel 8051, já projetada visando a implementação de sistemas de controle de equipamentos por microprocessador, fazendo uso de um *hardware* mínimo. Existem diversos modelos dessas CPUs, cada um podendo ser configurado de diferentes formas.

A CPU 8051 é composta por três conjuntos de portas externas bidirecionais de 8 bits – P0, P1 e P2 (Figura 17), endereçáveis bit a bit (vide tabela de endereçamentos no Anexo B). Alguns modelos dessa CPU podem ser utilizados com ROM e RAM externas, e nesse caso, duas portas (P0 e P2) são empregadas para controle dos barramentos externos de dados e endereços. Ao implementar o barramento, as portas P0 e P2 são concatenadas para formar um endereço de 16 bits (dando assim o máximo de $2^{16}=64$ KBytes de acesso externo). A porta P0 deve ser multiplexada através de um circuito 74HCT373, para fornecer o barramento bidirecional de dados.

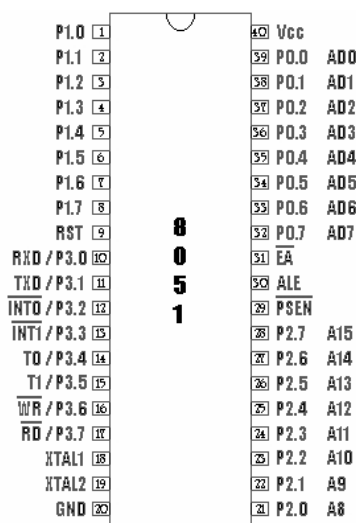


Figura 17 – Descrição dos pinos do microcontrolador Intel 8051

A CPU implementa internamente um serializador/desserializador, que serve para realizar a conexão serial do processador com o mundo externo. É possível configurá-lo para comunicação síncrona e assíncrona, embora nesse trabalho tenha sido utilizada apenas a forma assíncrona. A velocidade de comunicação é dada por um *timer* interno que, no caso da configuração de *hardware* desse projeto, permite conexão de até 2.400 bps (com a mudança do cristal utilizado, pode-se ir até 9.600 bps).

A CPU possui sinais diferentes para leitura de programa e leitura de dados, aumentando teoricamente a quantidade possível de memória. Esse fato foi aproveitado pelo projeto, servindo-se de uma área de ROM para rotinas básicas (32 KB), uma área de RAM para programa (32 KB) e uma área de RAM para dados (8 KB). Dessa forma, simplifica-se o processo de testagem de novas rotinas básicas, carregando-as, via *interface* serial, na área de RAM de programa – o *bootstrap* é feito a partir das rotinas já existentes em ROM. Numa próxima etapa do projeto, quando tais rotinas já estiverem suficientemente testadas, poderão então ser gravadas em ROM, eliminando-se assim, a necessidade do uso dessa RAM de programa.

Devido ao grande número de portas utilizadas nesse projeto, foi escolhido o modelo de endereçamento de portas no espaço em memória. Para os programas, as diversas *interfaces* externas são lidas e gravadas como posições de memória. Dois circuitos NAND (74HCT00) e dois registradores de 4 bits (74HCT139) controlam e selecionam esses endereços e suas incompatibilidades com o espaço real de memória.

5.2.3 Subsistemas Internos

5.2.3.1 Comunicações

Os sinais de comunicação serial da CPU são TXD e RXD (*Transmitter e Receiver Data*), que serão ligados ao exterior através de uma conversão de nível e corrente, realizada por um circuito MC145407. Esse circuito permite, na verdade, a interconexão de seis sinais, e assim,

aproveitamos para gerar quatro outros sinais que podem ser utilizados, por exemplo, para controle de um modem externo – permitindo mudanças na tecnologia de conectividade do *hardware* RoboFácil, proporcionando, até mesmo, sua conexão futura com à Internet. Esses sinais extras são lidos numa porta específica, a partir de um *latch* 74HCT244.

Os sinais gerados por essa placa são levados a um conector externo de 25 pinos (DB25), preparado para conexão a um cabo serial, que por sua vez é conectado a um microcomputador, permitindo a carga e descarga de dados e programas²⁰.

5.2.3.2 Teclado (Botoeira)

Foram implementados cinco botões, sendo quatro conectados a um registrador 74HCT244 e um ligado diretamente ao sinal INT0 da CPU (vide Apêndice A.3). Esse último foi pensado como uma forma dos programas serem interrompidos externamente, possivelmente numa aplicação de “Stop/Load/Run” de uma ROM de carga.

Na versão atual do *firmware* esse botão é utilizado quando do recebimento de um programa no *assembly* virtual do *hardware* RoboFácil, só iniciando sua execução após o seu pressionamento.



Figura 18 – Display e Teclado. (1) Botão que comanda o início da execução

²⁰ Nessa versão do RoboFácil ainda não foi implementada a interface infra-vermelha para conexão com o microcomputador, apesar de tal característica estar prevista na concepção do hardware.

5.2.3.3 Mostrador (Display)

Foi utilizado um *display*²¹ de cristal líquido (LCD) programável ALFACOM de uma linha com dezesseis caracteres, modelo LCM 1601 0630. Esse *display* foi escolhido por ter uma conexão e programação simples. Todo processo de controle da varredura e da memória interna é feito por comandos de alto nível, enviados ao *display*, segundo uma tabela fornecida pelo fabricante (ALFACOM, 2003).

A conexão desse *display* é feita diretamente ao *latch* de dados e o endereçamento pelo seletor, da forma mais simples possível, pois segue a mesma arquitetura empregada na implementação da conexão dos *plugins* nas portas de I/O do *hardware* RoboFácil.

5.2.3.4 Conversor Digital-Analógico (D/A)

O *hardware* provê um conversor digital-analógico, implementado através de um circuito convencional R-2R, construído sobre um conjunto de resistências. O valor obtido está na faixa de 0 à 4 volts, aproximadamente. A entrada do circuito é um registrador 74HC374, que provê razoável isolamento elétrico.

Apesar de eletronicamente implementado, o conversor D/A não se encontra em utilização nessa versão do *hardware*, todavia nada impede que seja criada uma aplicação para esse circuito, tal como a síntese de voz.

²¹ Por simplicidade, não foi implementada a leitura de dados do display, que se julgou não ser importante para esse projeto. Também foi mantida fixa (não programável) a luminosidade, ajustada apenas por um potenciômetro interno à placa.

5.2.3.5 Conversor Analógico-Digital (A/D)

Para realizar a conversão de sinal analógico para digital, optou-se por não fazer uso de nenhum circuito de conversão rápida, e sim realizar todo controle por comparação de nível. A razão para tal é de fácil entendimento: para medir um nível de entrada analógica, gera-se um valor de tensão no conversor digital-analógico, e compara-se o valor obtido com o valor de entrada (através de um comparador analógico construído com amplificadores operacionais LM324 e resistências). Por um processo de busca binária, no máximo em nove comparações, chega-se ao valor da voltagem (valor digital).

Devido à existência de um circuito conversor A/D no *hardware* RoboFácil, torna-se possível que o Kit de Robótica Educacional possa ser controlado, em futuras versões, por comandos de voz.

5.2.4 Interfaces Genéricas de Plugins

5.2.4.1 Técnicas para Conexão de Interfaces Plugins

Um dos problemas mais complexos encontrados na construção de sistemas robóticos é definir exatamente quais serão as *interfaces* utilizadas. Colocando poucas *interfaces*, o sistema fica restrito; se muitas, o sistema fica oneroso e grande. Para solucionar esse impasse foi utilizado no RoboFácil um esquema baseado em *plugins*.

São dezesseis sinais de saída e dezesseis de entrada, nesse esquema, isolados por registradores (74HCT374 e 74HCT244) que podem tolerar correntes relativamente altas, protegendo, razoavelmente, o exterior da placa principal do *hardware* RoboFácil. Nessas entradas e saídas, ligadas a dois conjuntos de conectores, se acoplarão os diversos circuitos – *plugins*.

A dificuldade que essa solução introduz é a necessidade de uma programação cuidadosa e dependente das conexões realizadas.

Nesse projeto, quatro *plugins* foram implementados:

- ❑ controle dos motores de passo;
- ❑ controle de *leds*;
- ❑ controle do sensor de luminosidade; e
- ❑ controle do sensor de temperatura.

5.2.4.2 Plugin de Controle dos Motores de Passo

O circuito de controle dos motores de passo necessita de chaveadores para cada um dos quatro fios de controle do motor, conectando-os a zero ou a um valor entre 9 e 12 V, dependendo do tipo de motor escolhido. A maioria dos motores utiliza uma tabela de códigos que informa os valores de programação como mostrada a seguir:

- ❑ 0000: Repouso (motor solto);
- ❑ 1100: Motor na posição 1;
- ❑ 0110: Motor na posição 2;
- ❑ 0011: Motor na posição 3;
- ❑ 1001: Motor na posição 4; e
- ❑ Qualquer outro valor: Instabilidade.



Figura 19 – Plugin de controle dos motores de passo. (1) Placa de circuito do plugin; e (2) Motores de passo

Para movimentá-lo no sentido horário, a seqüência contínua será a posição 1, 2, 3, 4, 1, 2, 3, 4, etc., e em sentido inverso, basta aplicar a seqüência inversa. O tempo de aplicação, histerese²² e aceleração permitida devem ser obtidos experimentalmente para cada motor (o manual do fabricante apresenta valores esperados e/ou máximos para essas operações). A lógica para o acionamento é totalmente realizada por *software*. O ângulo rodado em cada seqüência também é especificado no manual do motor.

A conexão dos motores foi facilitada com o uso de conectores de cinco pinos, não existindo, portanto, a necessidade dos mesmos serem soldados diretamente na placa de circuito impresso do *plugin*. Essa característica facilitará a ligação desses motores, em futuras versões, no primeiro andar do RoboFácil, podendo movimentar, efetivamente, o Kit.

²² Histerese é uma diferença entre o ponto de acionamento e desligamento, sendo necessária para evitar que o sinal de saída fique “trepidando”, ou seja: se não houver histerese, o sinal ficará oscilando quando o sinal monitorado estiver exatamente no valor pré-ajustado.

5.2.4.3 Plugin de Controle de Motores DC

Nessa versão do RoboFácil, fez-se a opção por não implementar o controle de motores de corrente contínua (DC), visto que os motores de passo permitem, por exemplo, o controle preciso da velocidade, direção, distância, entre outros. Todavia, é importante a implementação futura de controle para motores DC, uma vez que são motores mais baratos e mais simples de utilizar.

5.2.4.4 Plugin de Controle de Lâmpadas, Relés ou Leds

O circuito implementado permite a conexão tipo liga/desliga, o que pode ser realizado através de uma chave analógica. Entretanto, nesta versão do kit, por razões pedagógicas, acoplou-se uma barra com oito *leds* (2 vermelhos, 3 amarelos e 3 verdes), o que permite simular, por exemplo, aplicações educacionais que façam uso de um semáforo.

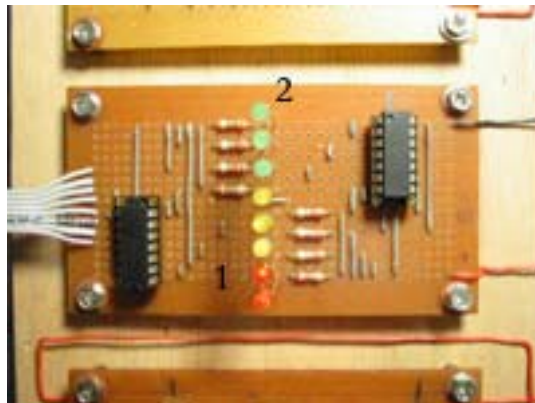


Figura 20 – Plugin de controle dos leds. (1) Placa de circuito do plugin; e (2) Oito leds

5.2.4.5 Plugin de Controle do Sensor de Luminosidade

O *plugin* de controle do sensor de luminosidade foi implementado fazendo uso de um resistor dependente de luz (LDR). O sensor de luminosidade é binário e através de um resistor variável – trimpot – é indicado o valor de referência.

O circuito pode oscilar, para intensidades de luz exatamente iguais ao valor de referência, pois o mesmo não implementa histerese. Provavelmente, a ausência de circuitos adicionais para controlar a histerese não compromete a sua aplicabilidade, pois o impacto será desprezível para a maior parte das aplicações educacionais que serão pensadas para esse Kit.

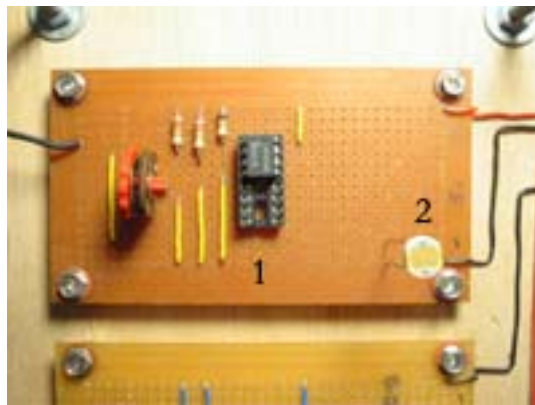


Figura 21 – Plugin de controle do sensor de luminosidade. (1) Placa de circuito do plugin; e (2) LDR

5.2.4.6 Plugin de Controle do Sensor de Temperatura

O *plugin* de controle do sensor de temperatura foi desenvolvido utilizando um termistor de coeficiente negativo (NTC). O sensor de temperatura também é binário – ativa/desativa a partir de um determinado valor.

Através de um resistor variável – trimpot – é indicado o valor de referência, sem implementação de histerese. Assim, para temperaturas exatamente iguais ao valor de referência, o circuito pode oscilar, sendo importante a realização de procedimentos de *software* para diminuir esse impacto.

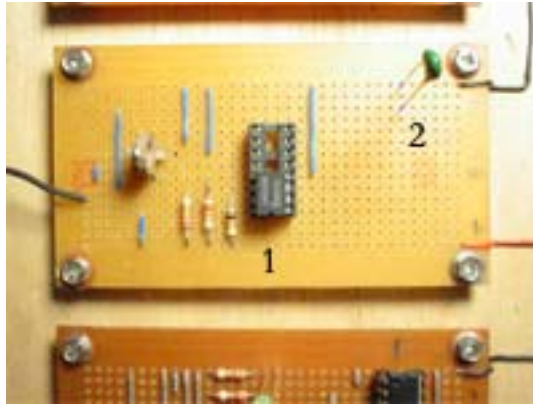


Figura 22 – Plugin de controle do sensor de temperatura. (1) Placa de circuito do plugin; e (2) NTC

5.2.5 Endereçamentos

É relevante, para os futuros implementadores do Kit de Robótica Educacional RoboFácil, conhecer as equivalências de endereços de memória empregados, pois somente através deles é possível utilizar os diversos recursos eletrônicos disponibilizados, tais como o *display* e os motores de passo. Contudo, no estágio atual do projeto, para os usuários-alunos, esses endereços não são imprescindíveis, pois os endereçamentos que controlam os diversos recursos são transparentes através da utilização do *software* ProgrameFácil.

As Tabelas 18 e 19 apresentam esses endereços:

Tabela 18 – Mapa de endereços de memória

Descrição	Endereço Inicial	Endereço Final
PROM Principal	0000h	7FFFh
RAM Principal	0000h	1FFFh
RAM / ROM	8000h	FFFFh

Tabela 19 – Mapa de endereços de I/O

Descrição	Endereço
Display (instruções)	6000h (somente escrita)
Display (dados)	6001h (somente escrita)
I/O 1	6002h (8 bits externos de mais baixa ordem)
I/O 2	6004h (8 bits externos de mais alta ordem)
I/O 3	6006h b0..b3 (botões) b4 (DSR) b5 (CTS) b6 (valor analógico maior do que o digital gerado) b7 (não usado) RTS (bit 0 de P1) DTR (bit 1 de P1) LDR (bit 2 de P1) NTC (bit 3 de P1)

5.2.6 Alimentação Elétrica

Todo o circuito do *hardware* de controle, inclusive o circuito RS-232²³, foi preparado para ser alimentado por uma fonte de 5 volts simples. Opcionalmente é possível alimentá-lo com uma fonte um pouco maior (6 volts = 4 pilhas de lanterna), mas não maior que isso, sob risco de causar danos irreversíveis.

Havendo a necessidade de fazer uso de uma fonte de alimentação maior, ou uma fonte não regulada, será imprescindível introduzir no projeto um circuito regulador de voltagem convencional. Nesse caso, haverá desperdício de energia no regulador, e sendo consumida mais potência, haverá diminuição da vida útil das pilhas²⁴.

²³ RS-232 é um padrão da indústria para conexões de comunicação serial. Adotado pela Electrical Industries Association (EIA), este padrão recomendado – Recommended Standard (RS) – define as linhas características específicas e sinais utilizados por controladores de comunicação serial para padronizar a transmissão de dados seriais entre equipamentos.

²⁴ Na versão atual do hardware RoboFácil, os motores de passo estão sendo alimentados por uma fonte externa ao Kit. Essa fonte deve fornecer tensões da ordem de 9 a 12 volts para produzir o acionamento desses motores. Numa versão industrializada do RoboFácil, essa fonte deverá ser suprimida.

5.2.7 Conector Externo

Com o intuito de facilitar o interfaceamento entre a placa principal e os *plugins*, adotou-se um cabo *flat* de 34 vias para realizar a ligação do conector da placa principal – que contém algumas entradas e saídas lógicas – com os *plugins* de controle dos motores de passo e *leds*²⁵. Para realizar a ligação dos *plugins* de controle do sensor de temperatura e luminosidade, optou-se por utilizar apenas um fio para cada *plugin*, ligando-se o mesmo através de um conector fêmea em uma das entradas do conector macho, disponível na placa principal do *hardware* RoboFácil.

O mapeamento com a especificação de algumas saídas lógicas da placa principal do *hardware* RoboFácil pode ser consultado na referência (MIRANDA, L.C.; BORGES, J.A.S.; SAMPAIO, F.F.; TAKANO, D.F., 2004).

5.2.8 Interface

A interface homem-computador pensada para o *hardware* RoboFácil foi concebida visando facilidades operacionais, obtidas através da proposta de utilização dos *plugins*. A adoção dos *plugins* permite que o usuário personalize os recursos eletrônicos disponíveis no *hardware* RoboFácil de modo a atender as peculiaridades de cada projeto. Essa característica além de proporcionar modularidade, também garante manejo facilitado para sua execução.

²⁵ Pelo fato de existir a possibilidade de na próxima versão do Kit se utilizar os outros sinais elétricos disponíveis no RoboFácil, não foram cortadas as vias atualmente não utilizadas do cabo *flat*.

5.2.9 Custo

Para a implementação do *hardware* RoboFácil foram utilizados alguns materiais alternativos e sucatas, e outras peças foram adquiridas. Apresenta-se, na Tabela 20, o custo total de aquisição de todos os componentes eletrônicos/eletromecânicos necessários para confecção desse artefato.

A mensuração do custo de montagem de um *hardware* RoboFácil levou em conta, exclusivamente, os componentes eletrônicos descrito no seu projeto eletrônico (Apêndices A), não levando em consideração o custo de acabamento desse equipamento, pois esse item não foi explorado durante a realização desse trabalho.

No decorrer da pesquisa de preço²⁶ foram constatadas algumas diferenças que podem influenciar de forma significativa o valor final desse produto dependendo do fabricante e/ou fornecedor dos componentes. Por exemplo, a diferença de preço entre um potenciômetro e um trimpot – mesma funcionalidade eletrônica – foi de até 66,66%.

Tabela 20 – Custo detalhado de um hardware RoboFácil

Produto	Qnt.	Vl./UM	Loja	Vl./P	Apêndice	Referência
27256	1	5,00	A	5,00	A.1	U4
62256	1	7,00	A	7,00	A.1	U5
6264	1	6,00	A	6,00	A.1	U3
7406	4	2,00	D	8,00	A.5, A.6	U2A, U2B, U2C, U2D, U1A, U1B, U1C, U1D
74HCT00	1	2,00	D	2,00	A.1, A.3	U6A, U6B, U6C, U6D
74HCT02	2	2,00	D	4,00	A.2, A.3, A.4	U13A, U13C, U13D, U18C, U18D
74HCT139	2	2,00	D	4,00	A.1, A.2	U7A, U7B, U8A, U8B
74HCT244	3	2,00	D	6,00	A.2, A.3	U11, U12, U17
74HCT373	1	2,00	D	2,00	A.1	U2
74HCT374	3	2,00	D	6,00	A.2, A.4	U9, U10, U14
Cabo paralelo com,	3	1,50	D	4,50	A.7	–

²⁶ O levantamento de preço dos componentes eletrônicos e/ou eletromecânicos foi realizado em cinco lojas – nessa dissertação identificadas com Lojas A, B, C, D e E – especializadas na venda de produtos dessa natureza numa das mais tradicionais ruas de comercialização de produtos desse gênero na cidade do Rio de Janeiro. Os valores coletados, e aqui apresentados, foram obtidos na primeira semana de fevereiro de 2006.

no mínimo, 3 vias						
Capacitor 1 μ F	1	0,20	D	0,20	A.1	C1
Capacitor 10 μ F	4	0,30	D	1,20	A.3	C2, C3, C4, C5
Conector DB-25 Fêmea	1	0,90	A	0,90	A.7	–
Conector DB-25 Macho	1	0,80	A	0,80	–	–
Conector DB-9 Fêmea	1	0,70	A	0,70	A.7	–
Cristal	1	3,50	D	3,50	A.1	X1
Diodo 1N4007	8	0,15	A	1,20	A.5	–
Display	1	48,00	D	48,00	–	–
LDR	1	3,60	A	3,60	A.5	–
Led – Amarelo	3	0,20	B	0,60	A.6	–
Led – Verde	3	0,20	B	0,60	A.6	–
Led – Vermelho	2	0,20	B	0,40	A.6	–
LM324	1	1,00	B	1,00	A.4	U15A, U15B, U15C
LM393	1	1,30	A	1,30	A.5	–
MC145407	1	16,00	D	16,00	A.3	U16
Microcontrolador Intel 8031AH	1	9,00	A	9,00	A.1	U1
Mosfet IRF630	8	2,60	A	20,80	A.5	–
Motor de Passo	2	66,00	D	132,00	–	–
NTC	1	3,30	C	3,30	A.5	–
Push bottom	5				A.3	–
Resistor 10K Ω	11	0,10	A	1,10	A.4, A.5	R10, R12, R40, R41, R42, R43, R44, R45, R46
Resistor 12K Ω	8	0,10	A	0,80	A.5	–
Resistor 20K Ω	10	0,10	D	1,00	A.4	R32, R33, R34, R35, R36, R37, R38, R39, R47, R48
Resistor 220 Ω	8	0,10	A	0,80	A.6	–
Resistor 22K Ω	9	0,10	A	0,90	A.3, A.5	–
Resistor 3,3K Ω	2	0,10	A	0,20	A.4	R13, R15
Resistor 330K Ω	1	0,10	A	0,10	A.4	R14
Trimpot 100K Ω	1	2,20	B	2,20	A.4	R11
Trimpot 10K Ω	1	2,20	B	2,20	A.5	–
Trimpot 1K Ω	1	2,20	B	2,20	A.5	–
Trimpot 47K Ω	1	2,20	B	2,20	A.2	–
CUSTO TOTAL				313,30		

NOTA: Qnt. representa a quantidade numérica; Vl./UM denota valor (R\$) unitário ou por metro; Vl./P significa valor (R\$) por produto (quantidade multiplicada pelo preço unitário ou metro).

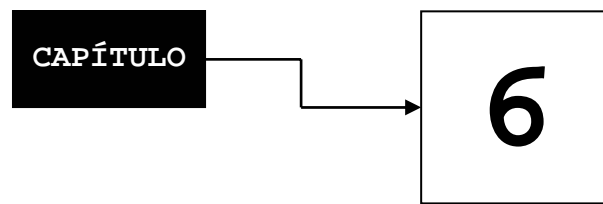
O exposto no Capítulo 5 foi vanguarda às soluções associadas – detalhadas nos capítulos anteriores – que culminaram no kit de robótica resultado dessa dissertação.

As singularidades atribuídas ao Kit RoboFácil transformam-no em um produto tangível ao contexto nacional, podendo esse ser almejado por educadores e educandos como ferramenta que auxilia a construção do conhecimento de forma criativa e prazerosa, acabando por envolver a todos nesse processo em colaboração. O fato desse artefato ser escalonável no aspecto físico e modular no aspecto lógico pode torná-lo chave-mestra das diferentes necessidades do público alvo, em diferentes contextos sócio-econômicos que o Kit RoboFácil estará inserido.

Assim, metaforicamente, no último capítulo dessa dissertação, o kit abandona os laços criados com os seus implementadores, com todas as qualidades e limitações já expostas, registrando os resultados obtidos e as contribuições auferidas para as pesquisas na área de robótica educacional do Brasil.

SEÇÃO
III

A Seção III, sendo a última dessa dissertação, tem como objetivo central expor os resultados, trabalhos futuros e as conclusões obtidas ao atingir o ponto fim do escopo definido dessa pesquisa.



CONCLUSÃO

Finda-se, no Capítulo 6, o relato da pesquisa realizada, explorando as contribuições auferidas no processo de materialização do Kit de Robótica Educacional RoboFácil e suas perspectivas futuras.

6.1 Considerações Finais

Essa pesquisa realizou estudos sobre o uso da robótica com fins pedagógicos e de kits de robótica educacional nacionais e estrangeiros, com o objetivo de conhecer em toda sua amplitude os requisitos de projeto para a utilização desse artefato tecnológico em aplicações de cunho pedagógico-educativo. Através dessas explorações, examinaram-se os motivos, os contextos e onde a robótica educacional foi/poderá ser empregada.

Foi constatado que há o interesse pelos envolvidos – alunos e professores – em realizar projetos/aulas com a utilização de kits de robótica educacional. Todavia não o fazem, ou não na frequência almejada, devido aos altos custos operacionais para os padrões nacionais, pelas restrições intrínsecas aos equipamentos disponíveis, ou pela necessidade de conhecimentos avançados de eletrônica analógica/digital e linguagens de programação de baixo e/ou alto nível e, ainda, pela associação desses problemas relatados.

Para conhecer esses anseios e vivenciar a prática do ensino mediado pela robótica foram realizados estudos bibliográficos e de campo – *in loco* em escolas e em oficinas/cursos que exploravam essa temática, buscando levantar os requisitos junto ao público alvo e seus mentores.

Os requisitos levantados foram as molas-mestras para uma proposta ímpar, materializada neste trabalho em um kit de robótica nomeado de Kit de Robótica Educacional RoboFácil – Kit RoboFácil, permitindo demonstrar a viabilidade no que tange ao desenvolvimento de um kit de robótica com fins educacionais nacional de baixo custo. Sendo assim, corroborou-se a primeira hipótese dessa pesquisa – é possível desenvolver um kit de robótica educacional a um baixo custo, e que não seja limitado nos seus recursos de *hardware* e *software* (funcionamento autônomo e (re)programado por *software*, respectivamente).

A diferenciação desse projeto se dá nos seguintes aspectos: (1) Kit de Robótica Educacional de baixo custo; (2) programação e reprogramação do *hardware* do Kit por *software* (linguagem icônica) – essa característica proporciona maior flexibilidade no emprego do Kit em

projetos, sem, contudo, aumentar significativamente seu preço; (3) simulação dos programas desenvolvidos (ProgrameFácil) na tela do monitor – esse recurso permite a redução do número de *hardwares* de robótica quando da utilização desse Kit em ambientes educacionais, não precisando de um Kit para cada grupo de alunos; (4) possibilidade de desconectar o *hardware* RoboFácil do microcomputador durante sua execução – essa qualidade amplia as possibilidades de utilização do Kit em díspares projetos, sem com isso, necessariamente, elevar demasiadamente seu custo; e (5) *hardware* RoboFácil escalonável – plugins.

O Kit RoboFácil oferece um ambiente integrado – composto do *hardware* RoboFácil e por uma ferramenta icônica de apoio à programação desse *hardware* (*software* ProgrameFácil) – para auxiliar a realização de projetos didáticos em sala de aula.

A criação e o desenvolvimento do *firmware* e do *hardware* RoboFácil, como descrito nos Capítulos 4 e 5, respectivamente, foi justificado pela carência imposta da maioria dos kits de robótica educacional desenvolvidos no Brasil, que não permitem que os *hardwares* dos kits permaneçam desacoplados do microcomputador durante o processo de execução e de serem (re)programados por *software* – alterar seu comportamento – uma gama variada de vezes.

A linguagem computacional ProgrameFácil – Capítulo 3 – apresenta características como a possibilidade de realização de simulação na tela do monitor do modelo de comportamento desenvolvido pelo usuário antes de, efetivamente, transferi-lo para ser executado sobre o *hardware* do kit de robótica. Essa característica viabiliza a redução na quantidade de *hardwares* necessários quando do emprego desse Kit em ambientes reais de ensino. Por sua vez a meta-linguagem presente nesse ambiente gráfico tem como objetivo tornar intuitiva e de fácil manejo sua interface, tentando viabilizar a programação do referido *hardware* por usuários leigos.

A disponibilização do Kit RoboFácil, embora propicie sua utilização, não provoca a demanda pelo seu emprego como instrumento de apoio às atividades escolares contemporâneas. A procura por novas alternativas aos materiais empregados na educação traz em seu seio a resistência ao ainda não explorado, barreiras que poderão frear a sua utilização, mas sem impedi-

la, confirmando a necessidade de investigações minuciosas para estudo do aumento e facilidades de seu emprego – robótica com fins educacionais – tanto pelo baixo custo de fabricação do Kit RoboFácil quanto pela simulação em tela permitida pelo *software* ProgrameFácil.

Portanto, a segunda hipótese – um *software* para robótica educacional que simule na tela do monitor o funcionamento de um *hardware* de robótica facilitará o emprego dessa ferramenta pedagógica nas escolas brasileiras – não foi confirmada no âmbito dessa pesquisa.

Espera-se, com a integração e miniaturização dos circuitos eletrônicos, a redução do custo de fabricação do Kit RoboFácil. Assim, parcelas cada vez maiores de estudantes poderão ter acesso a experiências práticas no processo educacional diário, sem que instituições necessitem despender vultuosos recursos com a aquisição e manutenção de kits de robótica.

6.2 Peculiaridades dos Produtos Resultantes

A partir do recorte de dois problemas identificados nessa área de pesquisa e da proposta elaborada para saná-los, foram concebidos os artefatos de *hardware* e *software*, e a junção desses foi denominado de Kit de Robótica Educacional RoboFácil.

No *hardware* RoboFácil estão disponíveis os seguintes recursos: 08 (oito) *leds*, 01 (um) *display* alfanumérico, 01 (um) sensor de luminosidade, 01 (um) sensor de temperatura e 02 (dois) motores de passos que poderão ser (re)programados pelas vias tradicionais, fazendo uso das linguagens *assembly* e C, e utilizando o *software* de comunicação serial – Hyperterminal – para realizar a transferência desses programas, ou através da linguagem computacional icônica ProgrameFácil.

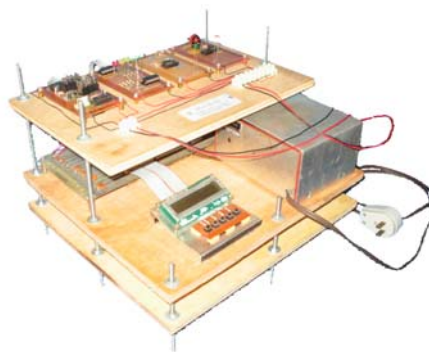


Figura 23 – Hardware RoboFácil atual

O ambiente visual proporcionado pelo *software* ProgameFácil, através de uma interface icônica, tem como objetivo ser de fácil entendimento pelo aluno-usuário. Essa aplicação possui ferramentas integradas de simulação, compilação e interpretação de modelos vislumbrados pelos alunos, ou sugeridos pelos educadores, com a finalidade de permitir a construção do conhecimento dos discentes através da integração da tríade: imaginação, exploração e prática.

Grife-se a possibilidade de simulação nesse ambiente, tornando a plataforma viável para realização de experimentos flexíveis e seguros, e treinamento a baixo custo, atribuindo ainda ao usuário “poderes” de criação, reorganização e destruição, se assim lhe for conveniente, usufruindo totalmente do modelo concebido, e tornando um portal compartilhado entre o virtual e o real.

O *software* ProgameFácil estará disponível para *download* no *site* do Projeto RoboFácil (ROBOFÁCIL, 2005), possibilitando sua utilização para simulação dos modelos de comportamento de *hardwares* de robótica sem, pelo menos num primeiro momento, ser necessário o *hardware* RoboFácil.

6.3 Trabalhos Futuros

Essa pesquisa, sob orientação dos pesquisadores do GINAPE, explorou um tema embrionário para o referido Grupo: robótica educacional. Por esse motivo, e devido aos prazos

estipulados, algumas perguntas ainda não foram respondidas com êxito, e outras surgiram no decorrer da realização do presente estudo. Dessa maneira, sinalizam-se alguns dos possíveis trabalhos posteriores a essa pesquisa, nas seções que se seguem, distribuídas por contexto.

6.3.1 Projeto RoboFácil

Discrimina-se ações de investigações futuras referentes ao Projeto RoboFácil.

- ❑ Realizar um estudo aprofundando *in loco* em escolas para constatação das potencialidades educacionais das soluções tecnológicas apresentadas nesse trabalho; e
- ❑ Desenvolver material didático-pedagógico, visando associar os conhecimentos teóricos, explorados nos currículos dos diferentes níveis do ensino, com atividades experimentais que façam uso da robótica como ferramenta de apoio ao trabalho do professor em sala de aula, permitindo assim a utilização desse Kit em ambientes reais de ensino.

6.3.2 Software ProgrameFácil

Seguem-se uma série de sugestões de refinamento/aprimoramento do *software* ProgrameFácil.

- ❑ Concepção e implementação de um Objeto de Programação “inteligente”, com a finalidade de viabilizar o reaproveitamento de estruturas – parte de programas – previamente desenvolvidos pelo usuário nessa ferramenta. Pode-se pensar nesse objeto como um bloco de comando dentro de um procedimento, que pode ser reutilizado – através de “chamadas” – por qualquer modelo desenvolvido, em qualquer parte do seu programa;

- Outro trabalho importante seria o estudo sobre interpretadores léxicos, de modo a viabilizar a incorporação de um analisador léxico à ferramenta. Essa funcionalidade garantiria a verificação sintática – não a semântica – dos programas em desenvolvimento. Dessa forma, em tempo de design a ferramenta faria a sinalização de problemas de sintaxe de um programa, tal como usar o Objeto Fim de Looping antes de utilizar o Objeto Início de Looping;
- Modificar o algoritmo de formulação do objeto “Linha de Programação” de forma a permitir, inclusive, flexibilidade por parte do usuário na disposição dessas linhas na janela “Meu Programa”;
- Uma próxima etapa do projeto – não explorado nessa dissertação – pode ser um estudo de caso para verificar, junto ao público alvo dessa pesquisa, sua eficiência e eficácia quanto aos processos de construção de modelos de comportamento, compilação e simulação nessa ferramenta educacional;
- Estudo detalhado sobre a interface definida para esse *software*, validando sua facilidade de manejo, e também, de entendimento dos conceitos e metáforas utilizadas nesse ambiente;
- e
- A concretização de estudos no sentido de possibilitar a utilização da linguagem de programação visual ProgrameFácil com outros kits de robótica encontrados no Brasil, tais como o *hardware* do Lego MindStorms (LEGO, 2005c), GoGo Board (GOGO BOARD, 2004), Super Robby (SUPER ROBBY, 2004) e Robótica Fácil (ROBÓTICA FÁCIL, 2003).

6.3.3 Firmware

Apresenta-se uma proposta de desenvolvimento futuro para o *firmware* do Kit.

- ❑ Modificar a interface entre o usuário e o *hardware* RoboFácil, possibilitando o armazenamento temporário de mais de um programa na memória desse *hardware*. Para a execução de um desses programas, bastará selecioná-lo – pelo teclado do *hardware* – e comandar sua inicialização.

6.3.4 Hardware RoboFácil

O *hardware* RoboFácil poderá sofrer, para agregar qualidades, alterações já identificadas abaixo.

- ❑ Realizar alterações no projeto eletrônico, a fim de que a velocidade de transferência de programas do microcomputador para o *hardware* RoboFácil seja de 9.600 bps – atualmente a velocidade de transferência é de 300 bps. Espera-se reduzir o tempo de transferência do *firmware* com maiores funcionalidades em 32 vezes;
- ❑ Realizar um projeto eletrônico visando contemplar a alimentação elétrica de todos os circuitos e componentes eletrônicos e/ou eletromecânicos, que compõem o *hardware* RoboFácil, através de bateria(s) ou pilha(s); e
- ❑ Implementar o controle de histerese nos *plugins* do sensor de temperatura e luminosidade através de circuitos eletrônicos, no momento, executada pelo *software*.

Os resultados preliminares dessa pesquisa são promissores, demonstrando as potencialidades do emprego da robótica educativa em ambiente escolar e a diversidade de problemas que ainda precisam ser explorados. Particularmente, no Brasil, o Kit RoboFácil apresenta-se como alternativa viável para os potenciais usuários da robótica aplicada à educação, logrando êxito nos contextos tecnológicos e sócio-econômicos.

Acredita-se que a miniaturização – com redução dos circuitos eletrônicos – e a posterior produção industrial do Kit RoboFácil, permitirão alavancar o processo de maior emprego da robótica com fins pedagógico-educacional no ambiente escolar nacional – principalmente junto aos atores sociais intrínsecos à educação (professores e alunos). Estimativas iniciais apontam, como produto dessas ações, menor custo se comparado a outros kits de robótica, com características desejáveis e semelhantes, atualmente comercializados no Brasil.

Almeja-se a expansão da área de pesquisa de robótica educacional pelos pesquisadores do GINAPE e alunos da Pós-Graduação em Informática do UFRJ/IM-NCE, garantindo um maior fomento e visibilidade desse projeto de ação pedagógica perante o meio acadêmico e a comunidade escolar, num futuro não tão distante. Conjectura-se, assim, o preenchimento de lacunas deixadas ou vindouras dessa iniciativa de vanguarda no Grupo, por meio de novas pesquisas realizadas sobre os alicerces da Universidade.

REFERÊNCIAS BIBLIOGRÁFICAS

ALFACOM. ALFATRONIC S.A. **Módulos Multi-Matrix ALFACOM – Manual de Utilização**. 2003.

ALVES, A.C.; BLIKSTEIN, P.; LOPES, R.D. **Robótica na periferia? Uso de tecnologias digitais na rede pública de São Paulo como ferramentas de expressão e inclusão**. In: XI Workshop sobre Informática na Escola (WIE) / XXV Congresso da Sociedade Brasileira de Computação (CSBC), 2005, São Leopoldo – RS. Anais do XXV Congresso da Sociedade Brasileira de Computação. São Leopoldo – RS, 2005. p. 2.594-2.602.

AZEVEDO JÚNIOR, J.B. **TTL/CMOS: Teoria e Aplicação em Circuitos Digitais**. São Paulo: Érica, 1984. 2 v.

BACON, F. **Novum Organum ou Verdadeiras Indicações Acerca da Interpretação da Natureza**. São Paulo: Nova Cultura, 2005. 255 p. ISBN: 85-13-01240-8. (Os Pensadores).

BARR, M. **Programming Embedded Systems in C and C++**. Sebastopol: O'Reilly & Associates, 1999. 194 p. ISBN: 1-56592-354-5.

BERTOTTO, C.A.; GÜNTZEL, J.L. **Ambiente de Programação Off-line para o Kit de Robótica ROBIX RCS-6**. In: XIX Concurso de Trabalhos de Iniciação Científica (CTIC) / XX Congresso da Sociedade Brasileira de Computação (CSBC), 2000, Curitiba – PR. Anais do XX Congresso da Sociedade Brasileira de Computação. Curitiba – PR, 2000. Disponível em: <<http://www.niee.ufrgs.br/SBC2000/eventos/ctic/ctic013.pdf>>. Acesso em: 14 jan. 2006. 11h03.

BITTENCOURT, J.V. **Brincando de Conversar na Cooperativa do Conhecimento: O Processo de Apropriação da Interface de Chat por Crianças de Séries Iniciais**. Porto Alegre – RS, 2004. 246 f. Dissertação (Mestrado em Educação) – Faculdade de Educação, Universidade Federal do Rio Grande do Sul, Porto Alegre – RS, 2004.

BORATTI, I.C. **Programação Orientada a Objetos Usando DELPHI**. Florianópolis: Visual Books, 2001. 272 p. ISBN: 85-7502-018-8.

CARDOSO DE SÁ, M. **Programação C para Microcontroladores 8051**. São Paulo: Érica, 2005. 334 p. ISBN: 85-365-0077-8.

CHELLA, M.T. **Ambiente de Robótica Educacional com Logo**. In: VIII Workshop sobre Informática na Escola (WIE) / XXII Congresso da Sociedade Brasileira de Computação (CSBC),

2002, Florianópolis – SC. Anais do XXII Congresso da Sociedade Brasileira de Computação. Florianópolis – SC, 2002. Disponível em: <http://www.nied.unicamp.br/~siros/doc/artigo_sbc2002_wie_final.PDF>. Acesso em: 04 fev. 2006. 13h05.

_____. **Ambiente de Robótica para Aplicações Educacionais com SuperLogo**. Campinas – SP, 2002. 186 f. Dissertação (Mestrado em Engenharia Elétrica) – Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas – SP, 2002.

CYBERBOX. **CyberBox BeSafe Ltda.** Disponível em: <<http://www.cyberbox.com.br/cb.htm>>. Acesso em: 30 abr. 2004. 18h56.

D'ABREU, J.V.V. **Desenvolvimento de Ambientes de Aprendizagem Baseados no Uso de Dispositivos Robóticos**. In: X Simpósio Brasileiro de Informática na Educação (SBIE), 1999, Curitiba – PR. Anais do X Simpósio Brasileiro de Informática na Educação. Curitiba – PR, 1999. Disponível em: <http://www.nied.unicamp.br/oea/mat/telerobotica_joao_nied4_sbie1999.pdf>. Acesso em: 25 fev. 2006. 15h32.

_____. **Desenvolvimento de Projetos em Parceria Professor – Aluno na Oficina de Robótica Pedagógica**. In: IV Congresso Iberoamericano de Informática Educativa, 1998, Brasília – Brasil. Actas do IV Congresso Iberoamericano de Informática Educativa. Brasília – Brasil, 1998. Disponível em: <http://www.nied.unicamp.br/oea/mat/telerobotica_joao_nied3_ribie1998.pdf>. Acesso em: 25 fev. 2006. 15h39.

D'ABREU, J.V.V.; CHELLA, M.T. **Ambiente Colaborativo de Aprendizagem a Distância Baseado no Controle de Dispositivos Robóticos**. In: XII Simpósio Brasileiro de Informática na Educação (SBIE), 2001, Vitória – ES. Anais do XII Simpósio Brasileiro de Informática na Educação. Vitória – ES, 2001. Disponível em: <<http://www.inf.ufes.br/~sbie2001/figuras/artigos/a185/a185.htm>>. Acesso em: 25 fev. 2006. 19h24.

_____. **Ambiente de Telerobótica em EaD**. In: IX Workshop sobre Informática na Escola (WIE) / XXIII Congresso da Sociedade Brasileira de Computação (CSBC), 2003, Campinas – SP. Anais do XXIII Congresso da Sociedade Brasileira de Computação. Campinas – SP, 2003. Disponível em: <http://www.nied.unicamp.br/~siros/doc/artigo_wie2003_final.PDF>. Acesso em: 16 fev. 2006. 7h37.

D'ABREU, J.V.V.; GONÇALVES, L.M.G.; GARCIA, M.F.; GARCIA, L.T.S. **Uma Abordagem Prático-Pedagógica para o Ensino de Robótica em Ciência e Engenharia de Computação**. In: XIII Simpósio Brasileiro de Informática na Educação (SBIE), 2002, Porto Alegre – RS. Anais do XIII Simpósio Brasileiro de Informática na Educação. Porto Alegre – RS, 2002. p. 428-439.

DELPHI. **Borland Delphi**. Disponível em: <<http://www.borland.com.br/delphi>>. Acesso em: 07 set. 2005. 11h37.

EDACOM. **EDacom Tecnologia em Sistemas de Informática Ltda**. Disponível em: <<http://www.edacom.com.br>>. Acesso em: 06 set. 2005. 16h06.

GIRAFFA, L.; MARCZAK, S.; PRIKLADNICKI, R. **PDS-E: Em direção a um processo para desenvolvimento de Software Educacional**. In: XI Workshop sobre Informática na Escola (WIE) / XXV Congresso da Sociedade Brasileira de Computação (CSBC), 2005, São Leopoldo – RS. Anais do XXV Congresso da Sociedade Brasileira de Computação. São Leopoldo – RS, 2005. p. 2.833-2.841.

GOGO BOARD. **GoGo Board**. Disponível em: <<http://www.gogoboard.org>>. Acesso em: 30 abr. 2004. 21h32.

HILGRAEVE. **Hilgraeve, Inc**. Disponível em: <<http://www.hilgraeve.com>>. Acesso em: 12 set. 2005. 7h06.

HOUAISS, A. **Dicionário Houaiss da Língua Portuguesa**. 1. ed. Rio de Janeiro: Objetiva, 2001. 2.925 p. ISBN: 85-7302-383-X.

IMAGINE. **Imagine**. CNOTINFOR Brasil Educação e Tecnologia. Disponível em: <<http://www.cnotinfor.com.br/cnotinfor/ imagine.htm>>. Acesso em: 03 fev. 2005. 19h52.

INEP. INSTITUTO NACIONAL DE ESTUDOS E PESQUISAS EDUCACIONAIS ANÍSIO TEIXEIRA. **Sinopse Estatística da Educação Básica: Censo Escolar 2003**. Brasília, 2004. p. 155-156.

INTEL. **Intel Corporation**. Disponível em: <<http://www.intel.com>>. Acesso em: 02 nov. 2005. 1h22.

_____. Intel Corporation. **Hexadecimal Object File Format Specification**. rev. A, 1988. 11 p.

_____. **Intel MCS 51/251 Microcontrollers**. Disponível em: <<http://www.intel.com/design/mcs51>>. Acesso em: 02 nov. 2005. 1h24.

KEIL. Keil Software, Inc. **8051 Development Tools**. Disponível em: <<http://www.keil.com/c51>>. Acesso em: 27 dez. 2005. 22h14.

LEGO. **LEGO Educational Division**. Disponível em: <<http://www.lego.com/education>>. Acesso em: 06 set. 2005. 16h15.

_____. **LEGO Group**. Disponível em: <<http://www.lego.com>>. Acesso em: 06 set. 2005. 16h18.

_____. **LEGO MindStorms**. Disponível em: <<http://mindstorms.lego.com>>. Acesso em: 06 set. 2005. 16h19.

LOUREIRO, R.C. **Avaliação de Softwares Educativos: Procurando Romper as Barreiras da Ingenuidade**. Fortaleza – CE, 1998. 135 f. Dissertação (Mestrado em Educação Brasileira) – Faculdade de Educação, Universidade Federal do Ceará, Fortaleza – CE, 1998.

MACHADO, F.B.; MAIA, L.P. **Arquitetura de Sistemas Operacionais**. 2. ed. Rio de Janeiro: LTC, 1999. 232 p. ISBN: 85-216-1097-1.

MEGALOGO. **MegaLogo**. CNOTINFOR Brasil Educação e Tecnologia. Disponível em: <<http://www.cnotinfor.com.br/cnotinfor/megalogo.htm>>. Acesso em: 03 fev. 2005. 20h36.

MICROWORLDS. **Microworlds**. Logo Computer Systems Inc. (LCSI). Disponível em: <<http://www.microworlds.com>>. Acesso em: 03 fev. 2005. 23h02.

MIRANDA, L.C.; BORGES, J.A.S.; SAMPAIO, F.F. **RoboFácil – Kit de Robótica Educacional Reprogramável por Software**. In: XXXI Seminário Integrado de Software e Hardware (SEMISH) / XXIV Congresso da Sociedade Brasileira de Computação (CSBC), 2004, Salvador – BA. Anais do XXIV Congresso da Sociedade Brasileira de Computação. Salvador – BA, 2004. p. 67-82.

MIRANDA, L.C.; BORGES, J.A.S.; SAMPAIO, F.F.; TAKANO, D.F. **RoboFácil – Kit de Robótica Educacional Reprogramável: Design e Implementação**. 2004. 33 f. Relatório Técnico (NCE – 06/04), Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro, Rio de Janeiro – RJ, 2004.

MIRANDA, L.C.; SAMPAIO, F.F.; BORGES, J.A.S. **ProgrameFácil: Linguagem Computacional Icônica com Simulação para Aprendizado de Robótica**. In: Workshop de Interfaces e Interação em Ambientes Educacionais / XVI Simpósio Brasileiro de Informática na Educação (SBIE), 2005, Juiz de Fora – MG. Anais do XVI Simpósio Brasileiro de Informática na Educação. Juiz de Fora – MG, 2005.

MIT. **Massachusetts Institute of Technology** (MIT). Disponível em: <<http://www.mit.edu>>. Acesso em: 18 fev. 2006. 16h24.

NI. **National Instruments Corporation** (NI). Disponível em: <<http://www.ni.com>>. Acesso em: 18 fev. 2006. 17h03.

OLIVEIRA NETTO, A.A. **IHC – Interação Humano Computador: Modelagem e Gerência de Interfaces com o Usuário**. Florianópolis: VisualBooks, 2004. 120 p. ISBN: 85-7502-138-9.

PECH, S.M. **Sistemas Embarcados: Características e Procedimentos para Inicialização do Sistema Operacional**. Rio de Janeiro – RJ. 2002. 136 f. Dissertação (Mestrado em Informática) – Instituto de Matemática, Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro, Rio de Janeiro – RJ, 2002.

PFAFFENBERGER, B. **Webster's New World: Dicionário de Informática**. Tradução da sexta edição. Rio de Janeiro: Campos, 1998. 797 p. ISBN: 85-352-0372-9.

POSSAMAI, C.L.; PASSERINO, L.M. **O Processo de Cooperação e Resolução de Problemas Apoiado pela Robótica Educativa com Adolescentes**. In: VII Congresso Iberoamericano de Informática Educativa, 2004, Monterrey – México. Actas do VII Congresso Iberoamericano de Informática Educativa. Monterrey – México, 2004. Disponível em: <<http://www.niee.ufrgs.br/ribie2004/Trabalhos/Posters/poster1324-1333.pdf>>. Acesso em: 19 fev. 2006. 11h21.

QUEIROZ, L.R. et al. Educação a Distância em Robótica e Visão Computacional. **Revista Brasileira de Informática na Educação** (RBIE). Sociedade Brasileira de Computação (SBC), n. 3, p. 17-26, set. 1998. ISSN 1414-5685. Disponível em: <<http://gmc.ucpel.tche.br/rbie-artigos/nr3-1998/Queiroz03.htm>>. Acesso em: 04 out. 2005. 15h21.

ROBOFÁCIL. **Projeto RoboFácil**. Grupo de Informática Aplicada à Educação do Instituto de Matemática/Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro (UFRJ/IM-NCE/GINAPE). Disponível em: <<http://www.nce.ufrj.br/ginape/robofacil>>. Acesso em: 04 nov. 2005. 4h48.

ROBOLAB. **ROBOLAB Software**. Disponível em: <<http://www.lego.com/eng/education/mindstorms/home.asp?pagename=robolab>>. Acesso em: 30 abr. 2004. 18h51.

ROBÓTICA FÁCIL. Robótica Fácil. **Manual do Kit Robótica Fácil**. 2003. 16 p.

ROBOTICANDO. **RobotIcanDo**. Laboratório de Estudos Cognitivos da Universidade Federal do Rio Grande do Sul (UFRGS/LEC). Disponível em: <<http://oea.psico.ufrgs.br/roboticando>>. Acesso em: 28 jun. 2004. 11h18.

ROCHA, H.V.; BARANAUSKAS, M.C.C. **Design e Avaliação de Interfaces Humano-Computador**. São Paulo: NIED/UNICAMP, 2003. 244 p. ISBN: 85-88833-04-2.

SANTOS, C.F.; MAGALHÃES NETTO, J.F.; MENEZES, C.S. **Uma Proposta para Robótica Educacional usando Lego Mindstorms**. In: X Workshop sobre Informática na Escola (WIE) / XXIV Congresso da Sociedade Brasileira de Computação (CSBC), 2004, Salvador – BA. Anais do XXIV Congresso da Sociedade Brasileira de Computação. Salvador – BA, 2004. p. 584-594.

SANTOS, C.F.; MENEZES, C.S. **A Aprendizagem da Física no Ensino Fundamental em um Ambiente de Robótica Educacional**. In: XI Workshop sobre Informática na Escola (WIE) / XXV Congresso da Sociedade Brasileira de Computação (CSBC), 2005, São Leopoldo – RS. Anais do XXV Congresso da Sociedade Brasileira de Computação. São Leopoldo – RS, 2005. p. 2.746-2.753.

SCHONS, C.; PRIMAZ, E.; WIRTH, G.A.P. **Introdução a Robótica Educativa na Instituição Escolar para alunos do Ensino Fundamental da disciplina de Língua Espanhola através das Novas Tecnologias de Aprendizagem**. In: I Workshop de Computação da Região Sul (WORKCOMP-SUL), 2004, Florianópolis – SC. Anais do I Workshop de Computação da Região Sul. Florianópolis – SC, 2004. Disponível em: <<http://inf.unisul.br/~ines/workcomp/cd/pdfs/2217.pdf>>. Acesso em: 02 fev. 2006. 20h53.

SIROS. **Sistemas Robóticos com SuperLogo**. Equipe de Robótica Pedagógica do Núcleo de Informática Aplicada a Educação da Universidade de Campinas (UNICAMP/NIED). Disponível em: <<http://www.nied.unicamp.br/~siros>>. Acesso em: 28 jun. 2004. 12h47.

STEFFEN, H.H. **Robótica Pedagógica na Educação: Um Recurso de Comunicação, Regulagem e Cognição**. São Paulo – SP, 2002. 111 f. Dissertação (Mestrado em Comunicação) – Escola de Comunicação e Artes, Universidade de São Paulo, São Paulo – SP, 2002.

SUPERLOGO. **SuperLogo**. Disponível em: <<http://www.nied.unicamp.br/~siros/slogo.htm>>. Acesso em: 03 fev. 2005. 22h16.

SUPER ROBBY. **Super Robby**. ARS Consult. Disponível em: <<http://www.arsconsult.com.br/produtos/srobby>>. Acesso em: 30 abr. 2004. 18h41.

TAURION, C. **Software Embarcado: A nova onda da Informática**. Rio de Janeiro: Brasport, 2005. 178 p. ISBN: 85-7452-228-7.

TAVARES, D.M.; ANTUNES, V.A.; GONÇALVES, L.M.G. Em Evidência o Potencial e Limitações dos Compiladores NQC e BrickOS e seus Respetivos Sistemas Operacionais. **Revista de Informática Teórica e Aplicada (RITA)**. Universidade Federal do Rio Grande do Sul (UFRGS), São Leopoldo – RS, v. 10, n. 2, p. 79-98, jan. 2004. 112 p. Disponível em:

<http://www.inf.ufrgs.br/~revista/docs/rita10/rita_v10_n2_p79a98.pdf>. Acesso em: 08 fev. 2006. 08h31.

TUFTS. **Tufts University**. Disponível em: <<http://www.tufts.edu>>. Acesso em: 18 fev. 2006. 19h11.

VISUAL BASIC. **Microsoft Visual Basic Developer Center**. Disponível em: <<http://msdn.microsoft.com/vbasic>>. Acesso em: 07 jan. 2006. 12h57.

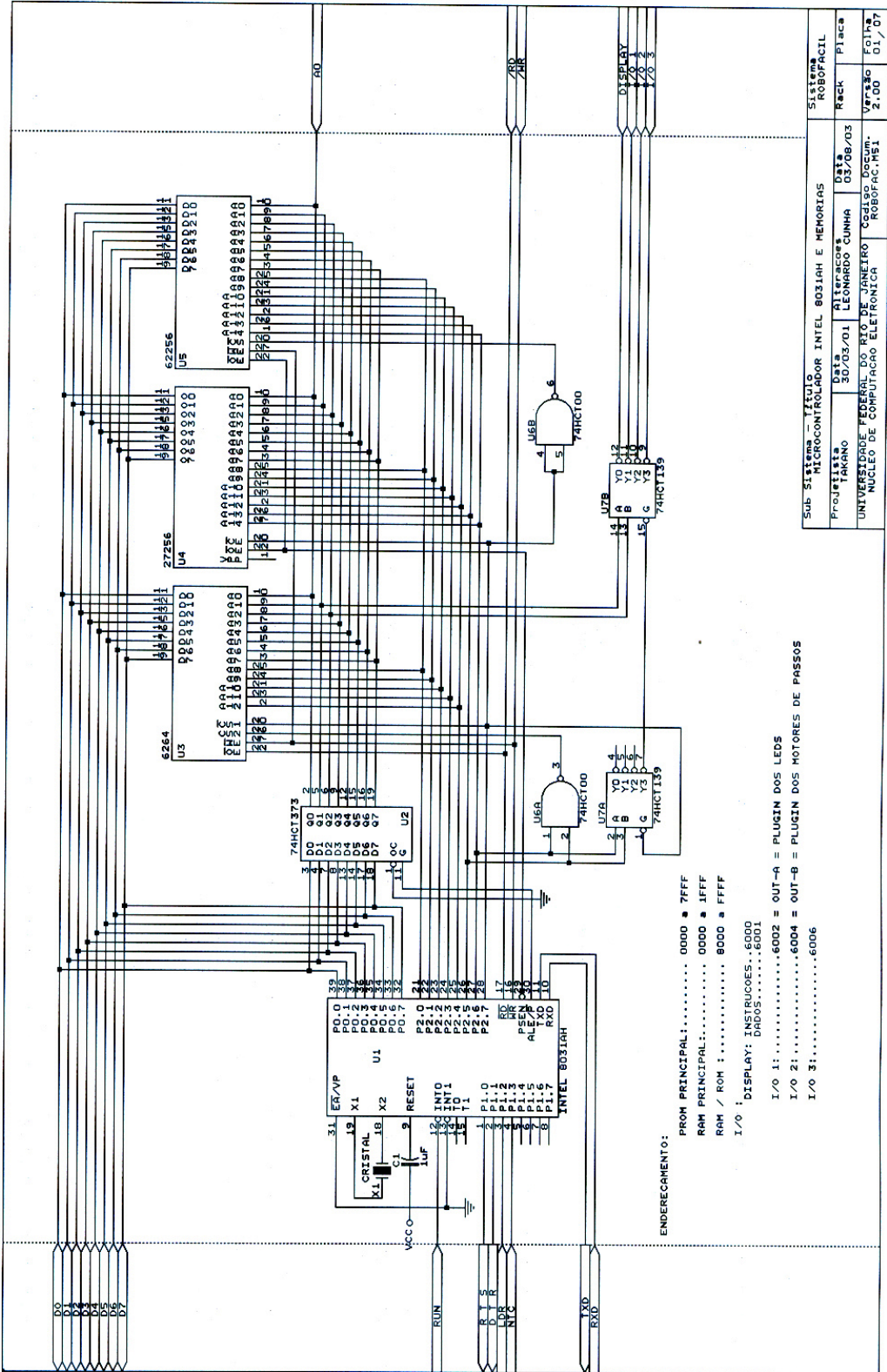
VISUAL C++. **Microsoft Visual C++ Developer Center**. Disponível em: <<http://msdn.microsoft.com/visualc>>. Acesso em: 07 jan. 2006. 13h22.

ZILLI, S.R. **A Robótica Educacional no Ensino Fundamental: Perspectivas e Prática**. Florianópolis – SC, 2004. 89 f. Dissertação (Mestrado em Engenharia de Produção) – Universidade Federal de Santa Catarina, Florianópolis – SC, 2004.

APÊNDICES

Apêndice A – Esquemas dos Circuitos Eletrônicos

Apêndice A.1 – Circuito Eletrônico Principal

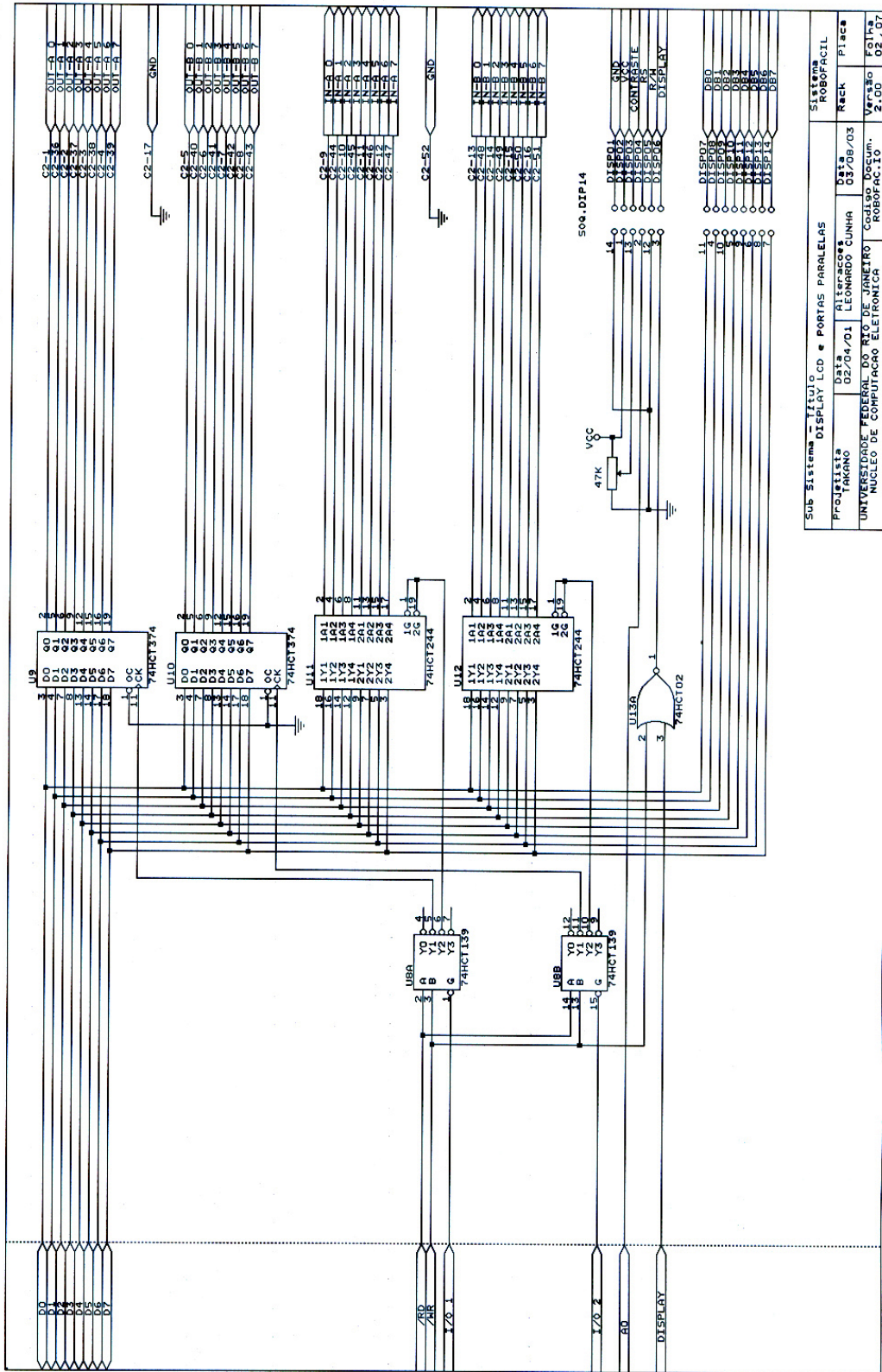


ENDERECHAMENTO:

- PRGM PRINCIPAL:..... 0000 ▶ 7FFF
- RAM PRINCIPAL:..... 0000 ▶ 1FFF
- RAM / ROM :..... 8000 ▶ FFFF
- I/O 1 :..... 6000 ▶ 6001
- I/O 2 :..... 6002 ▶ 6004 = OUT-A = PLUGIN DOS LEDES
- I/O 3 :..... 6004 = OUT-B = PLUGIN DOS MOTORES DE PASSOS
- DISPLAY: INSTRUCCOES...6000
- DADOS.....6001

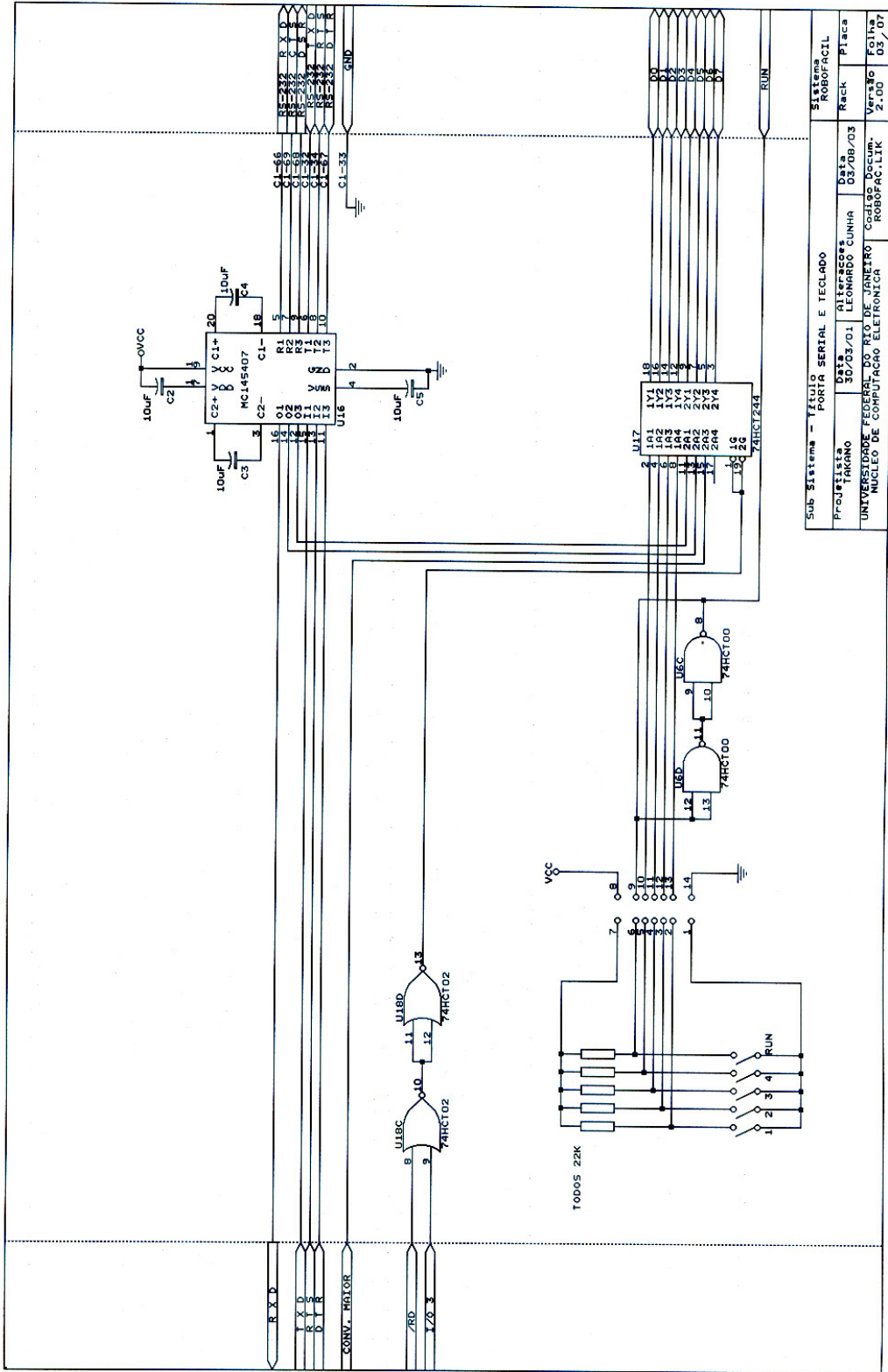
SUB SISTEMA – TIPO			SISTEMA	
MICROCONTROLADOR INTEL 8031BH E MEMORIAS			ROBOFACIL	
Proj.ista	Data	Alteracoes	Data	Placa
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO	30/03/01	LEONARDO CUNHA	03/08/03	
NUCLEO DE COMPUTACAO ELETRONICA		ROBOFACIL.MEI	Ver.3.0	Folha
			2.00	01/07

Apêndice A.2 – Display e Portas Paralelas



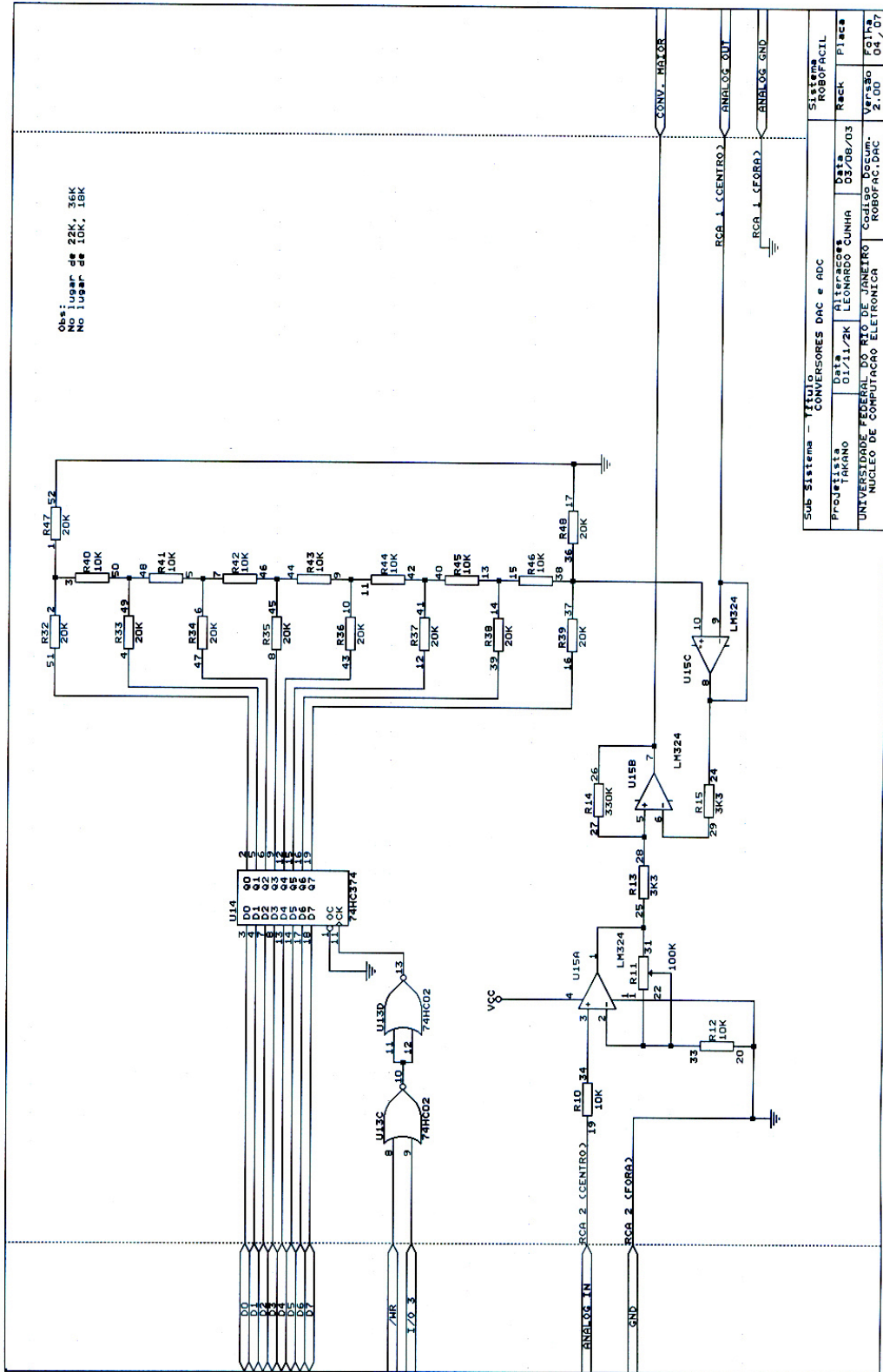
Projeto		Data		Data	
Sistema DISPLAY LCD e PORTAS PARALELAS		Alterações		03/08/03	
ROBOFACIL		LEONARDO CUNHA		Rack	
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO		Codigao Docum.		Versao	
NÚCLEO DE COMPUTAÇÃO ELETRÔNICA		ROBOFACIL-10		2.00	
				Folha	
				02 / 07	

Apêndice A.3 – Porta Serial e Teclado

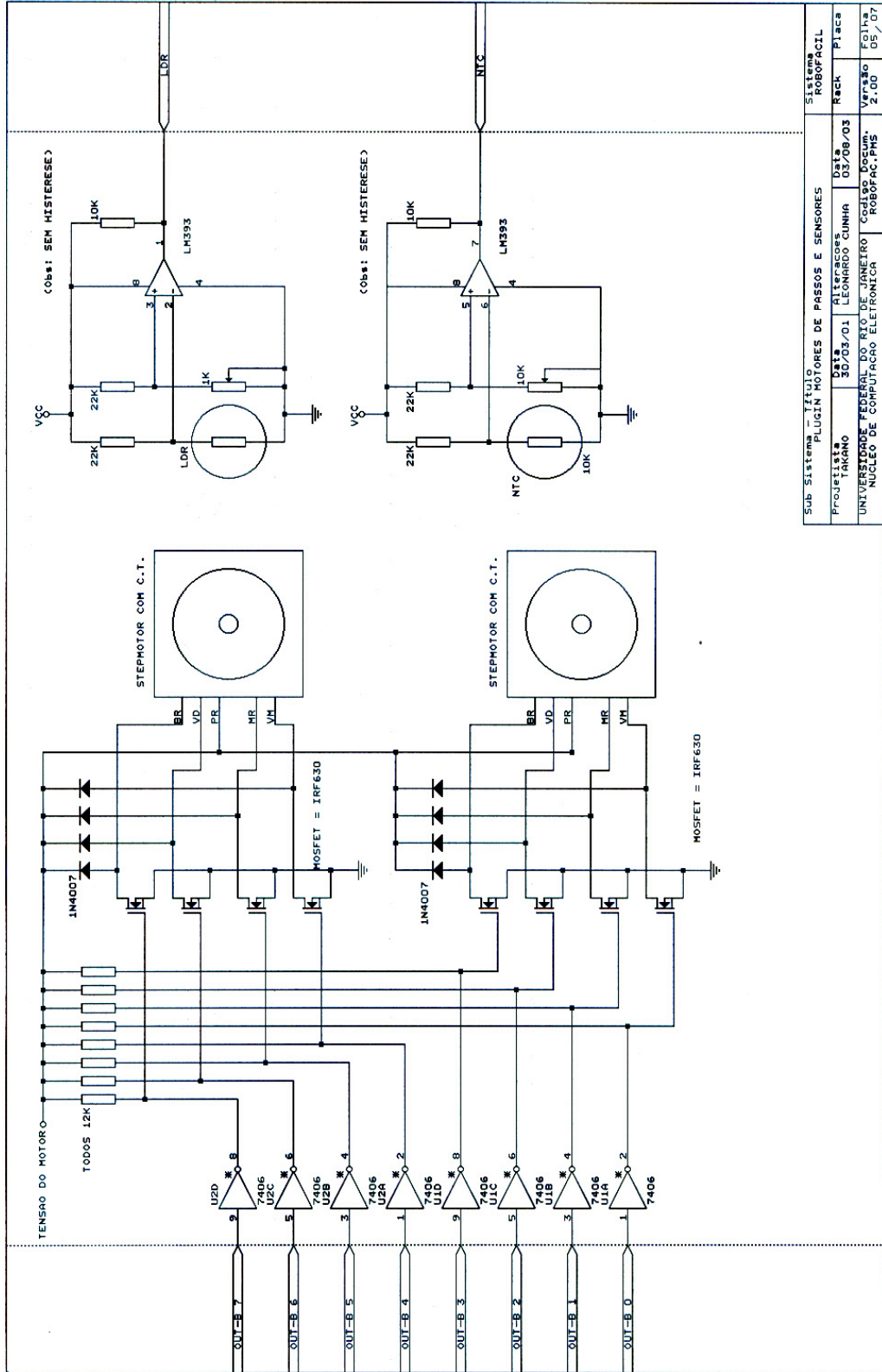


Sub Sistema - Título SERIAL E TECLADO		Sistema ROBOFACIL	
Proj.ista	Data	Alterações	Rack
TAHANO	30/03/01	LEONARDO CUNHA	03/08/03
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO	NÚCLEO DE COMPUTAÇÃO ELETRÔNICA		Código Discum. ROBOFAC.LIK
Versão 2.00		Folha 05	07

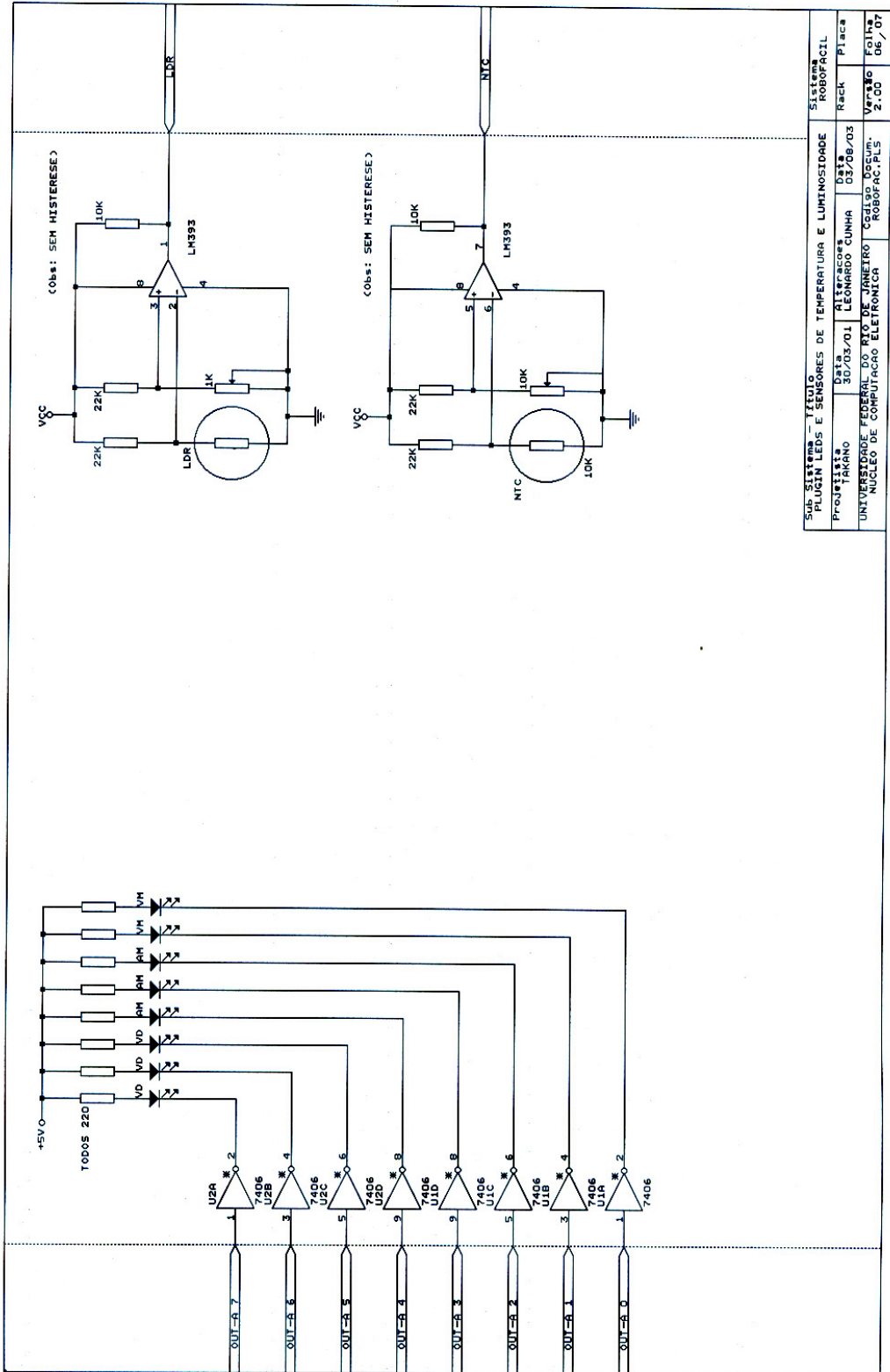
Apêndice A.4 – Conversores DAC e ADC



Apêndice A.5 – Plugin de Controle dos Motores de Passos e Sensores

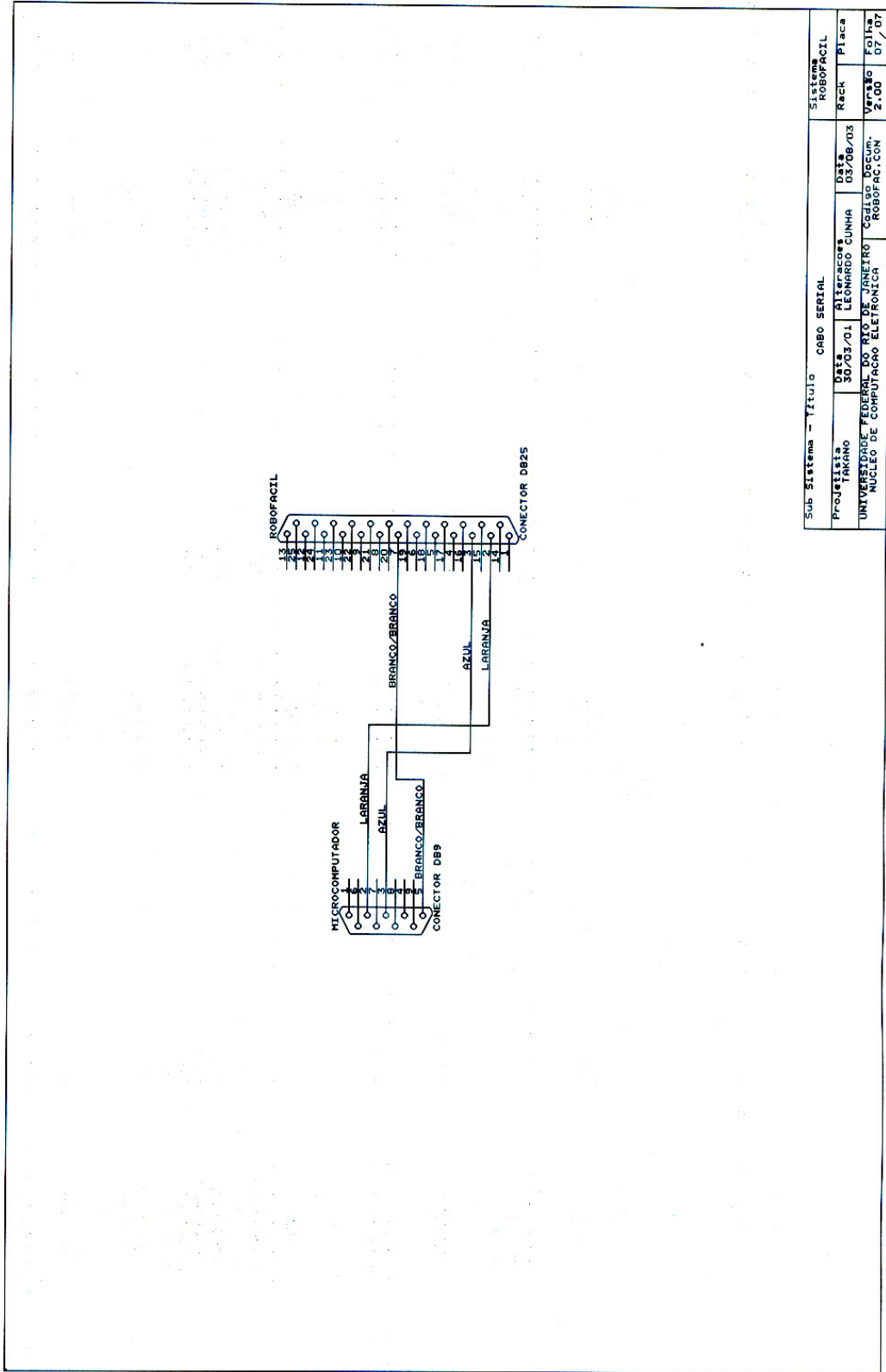


Apêndice A.6 – Plugin de Controle dos Leds e Sensores



SUB SISTEMA - TÍTULO		SISTEMA	
PLUGIN Leds e Sensores de Temperatura e Luminosidade		ROBOFACIL	
PROJETA	DATA	ALTERAÇÕES	DATA
TAKANO	30/03/01	LEONARDO CUNHA	03/08/03
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO		Código Docum.	
NÚCLEO DE COMPUTAÇÃO ELETRÔNICA		ROBOFAC-PLS	
VAPOR		FOLIO	
2,00		06/07	

Apêndice A.7 – Cabo Serial



Sub Sistema - Título		CABO SERIAL		Sistema	
Proj. Para		Alterações		ROBOFACIL	
30/03/01		LEONARDO CUNHA		Rack	
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO		CÓDIGO DOCUM.		Placa	
NÚCLEO DE COMPUTAÇÃO ELETRÔNICA		ROBOFACIL.CON		Versão	
				2.00	
				Folha	
				07 / 07	

ANEXOS

Anexo A – Tabela ASCII Standard

Dec	Hex	Caractere	Dec	Hex	Caractere	Dec	Hex	Caractere	Dec	Hex	Caractere
0	0x00	NULL	32	0x20	Espaço	64	0x40	@	96	0x60	
1	0x01	SOH	33	0x21	!	65	0x41	A	97	0x61	a
2	0x02	STX	34	0x22	"	66	0x42	B	98	0x62	b
3	0x03	ETX	35	0x23	#	67	0x43	C	99	0x63	c
4	0x04	EOT	36	0x24	\$	68	0x44	D	100	0x64	d
5	0x05	ENQ	37	0x25	%	69	0x45	E	101	0x65	e
6	0x06	ACK	38	0x26	&	70	0x46	F	102	0x66	f
7	0x07	BEL	39	0x27	'	71	0x47	G	103	0x67	g
8	0x08	BS	40	0x28	(72	0x48	H	104	0x68	h
9	0x09	HT	41	0x29)	73	0x49	I	105	0x69	i
10	0x0A	LF	42	0x2A	*	74	0x4A	J	106	0x6A	j
11	0x0B	VT	43	0x2B	+	75	0x4B	K	107	0x6B	k
12	0x0C	FF	44	0x2C	,	76	0x4C	L	108	0x6C	l
13	0x0D	CR	45	0x2D	-	77	0x4D	M	109	0x6D	m
14	0x0E	SOH	46	0x2E	.	78	0x4E	N	110	0x6E	n
15	0x0F	SI	47	0x2F	/	79	0x4F	O	111	0x6F	o
16	0x10	DLE	48	0x30	0	80	0x50	P	112	0x70	p
17	0x11	DC1	49	0x31	1	81	0x51	Q	113	0x71	q
18	0x12	DC2	50	0x32	2	82	0x52	R	114	0x72	r
19	0x13	DC3	51	0x33	3	83	0x53	S	115	0x73	s
20	0x14	DC4	52	0x34	4	84	0x54	T	116	0x74	t
21	0x15	NAK	53	0x35	5	85	0x55	U	117	0x75	u
22	0x16	SYN	54	0x36	6	86	0x56	V	118	0x76	v
23	0x17	ETB	55	0x37	7	87	0x57	W	119	0x77	w
24	0x18	CAN	56	0x38	8	88	0x58	X	120	0x78	x
25	0x19	EM	57	0x39	9	89	0x59	Y	121	0x79	y
26	0x1A	SUB	58	0x3A	:	90	0x5A	Z	122	0x7A	z
27	0x1B	ESC	59	0x3B	;	91	0x5B	[123	0x7B	{
28	0x1C	FS	60	0x3C	<	92	0x5C	\	124	0x7C	
29	0x1D	GS	61	0x3D	=	93	0x5D]	125	0x7D	}
30	0x1E	RS	62	0x3E	>	94	0x5E	^	126	0x7E	~
31	0x1F	US	63	0x3F	?	95	0x5F	_	127	0x7F	DEL

Anexo B – Mapa dos registradores da RAM interna do microcontrolador 8051

BYTE ADDRESS	BYTE ADDRESS															
7F	GENERAL PURPOSE RAM															
"NOT BIT ADDRESSABLE"																
30	BIT ADDRESSABLE LOCATION															
2F									7F	7E	7D	7C	7B	7A	79	78
2E									77	76	75	74	73	72	71	70
2D									6F	6E	6D	6C	6B	6A	69	68
2C									67	66	65	64	63	62	61	60
2B									5F	5E	5D	5C	5B	5A	59	58
2A									57	56	55	54	53	52	51	50
29									4F	4E	4D	4C	4B	4A	49	48
28									47	46	45	44	43	42	41	40
27									3F	3E	3D	3C	3B	3A	39	38
26									37	36	35	34	33	32	31	30
25									2F	2E	2D	2C	2B	2A	29	28
24									27	26	25	24	23	22	21	20
23									1F	1E	1D	1C	1B	1A	19	18
22									17	16	15	14	13	12	11	10
21	0F	0E	0D	0C	0B	0A	09	08								
20	07	06	05	04	03	02	01	00								
1F	BANK 3															
18	BANK 3															
17	BANK 2															
10	BANK 2															
0F	BANK 1															
08	BANK 1															
07	DEFAULT REGISTER BANK FOR R0-R7															
00	DEFAULT REGISTER BANK FOR R0-R7															

RAM

BYTE ADDRESS	BYTE ADDRESS								
FF									
F0	F7	F6	F5	F4	F3	F2	F1	F0	B
E0	E7	E6	E5	E4	E3	E2	E1	E0	ACC
D0	D7	D6	D5	D4	D3	D2	D1	D0	PSW
B8	.	.	.	BC	BB	BA	B9	B8	IP
B0	B7	B6	B5	B4	B3	B2	B1	B0	P3
A8	AF	.	.	AC	AH	AA	A9	A8	IE
A0	A7	A6	A5	A4	A3	A2	A1	A0	P2
99	NOT BIT ADDRESSABLE								SBUF
98	9F	9E	9D	9C	9B	9A	99	98	SCON
90	97	96	95	94	93	92	91	90	P1
8D	NOT BIT ADDRESSABLE								TH1
8C	NOT BIT ADDRESSABLE								TH0
8B	NOT BIT ADDRESSABLE								TL1
8A	NOT BIT ADDRESSABLE								TL0
89	NOT BIT ADDRESSABLE								TMOD
88	8F	8E	8D	8C	8B	8A	89	88	TCON
87	NOT BIT ADDRESSABLE								PCON
83	NOT BIT ADDRESSABLE								DPH
82	NOT BIT ADDRESSABLE								DPL
81	NOT BIT ADDRESSABLE								SP
80	87	86	85	E4	83	82	81	80	P0

SPECIAL FUNCTION REGISTER