

PAULO CESAR GONÇALVES

**PROTÓTIPO DE UM ROBÔ MÓVEL DE BAIXO CUSTO
PARA USO EDUCACIONAL**

MARINGÁ

2007

PAULO CESAR GONÇALVES

**PROTÓTIPO DE UM ROBÔ MÓVEL DE BAIXO CUSTO
PARA USO EDUCACIONAL**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Estadual de Maringá, como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Dante Alves Medeiros
Filho

MARINGÁ

2007

Dados Internacionais de Catalogação-na-Publicação (CIP)
(Biblioteca Central - UEM, Maringá – PR., Brasil)

G635p Gonçalves, Paulo Cesar
 Protótipo de um robô móvel de baixo custo para
 uso educacional / Paulo Cesar Gonçalves. -- Maringá
 : [s.n.], 2007.
 86 p. : il., figs., tabs.

 Orientador : Prof. Dr. Dante Alves Medeiros
 Filho.

 Dissertação (mestrado) - Universidade Estadual de
 Maringá. Programa de Pós-graduação em Ciência da
 Computação, 2007.

 1. Robótica educacional. 2. Ambientes de
 aprendizagem. 3. Alfabetização digital. 4. Fluência
 digital. 5. Tecnologia digital - Baixo custo. 6.
 Ferramentas tecnológicas. I. Universidade Estadual
 de Maringá. Programa de Pós-graduação em Ciência da
 Computação.

CDD 21.ed. 004.165071

PAULO CESAR GONÇALVES

**PROTÓTIPO DE UM ROBÔ MÓVEL DE BAIXO CUSTO
PARA USO EDUCACIONAL**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Estadual de Maringá, como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Aprovado em 30/11/2007.

BANCA EXAMINADORA

Prof. Dr. Dante Alves Medeiros Filho
Universidade Estadual de Maringá – DIN/UEM

Prof. Dr. João Angelo Martini
Universidade Estadual de Maringá – DIN/UEM

Prof. Dr. Jorge Bidarra
Universidade Estadual do Oeste do Paraná – CCET/UNIOESTE

*Dedico esta dissertação à
pessoa mais importante da minha vida, Mônica,
minha esposa, que com sua paciência e amor me ajudou a chegar até aqui.*

Agradecimentos

Aos meus pais, Antônio e Lúcia, pelo carinho e dedicação.

À minha esposa Mônica, meu filho Vítor, pelo amor, compreensão e apoio, principalmente nos momentos difíceis.

Ao meu orientador, Professor Dante Alves Medeiros Filho, pela oportunidade, confiança, amizade, dedicação e conhecimentos compartilhados.

Aos amigos Márcio Germano Perissato, Bruno de Pádua Melo e Danilo Rodrigues César, pela amizade, incentivo e contribuições.

A todos os funcionários do Departamento de Informática da UEM.

“Viver de acordo com a mente pressupõe que você tem ação sobre as coisas, que você tem domínio, que você tem controle.

Viver de acordo com a consciência, você não tem outra alternativa a não ser aceitar as coisas como elas são.”

Satyaprem

Resumo

A Robótica Educacional é utilizada como ferramenta computacional didática, e embora esteja relacionada com robôs móveis, ela é mais abrangente, podendo ser definida como o ambiente constituído pelo computador, componentes eletrônicos, eletromecânicos e programas, que integrados têm por objetivo explorar conceitos das diversas áreas do conhecimento. A robótica educacional teve a sua expansão atual, após o desenvolvimento de um bloco programável, que é um computador muito pequeno inserido dentro de um bloco de montagem LEGO, sendo posteriormente comercializado como parte de um conjunto de robótica chamado LEGO Mindstorms. No entanto, este conjunto de robótica custa em torno de US\$700,00 no Brasil, sendo um custo proibitivo para a realidade da maioria das escolas brasileiras. Este alto custo torna a sua utilização restrita a poucas escolas, tendo surgido como alternativa a placa GoGo, um dispositivo eletrônico programável que tem projeto e código abertos e baixo custo dos componentes, o que possibilita a construção pelo próprio usuário. A partir da construção e utilização da placa GoGo, este trabalho constatou ser possível desenvolver projetos na área de robótica educacional de baixo custo e de fácil implementação, portanto acessíveis à realidade daquelas escolas brasileiras.

Palavras-Chaves: Robótica Educacional, Tecnologia Digital, Ambientes de Aprendizagem.

Abstract

Educational-Robotics is used as a teaching tool, although most times it is related to mobile robots, it is in fact much broader, can be defined as an environment built from computers, electronic components, electro mechanicals and programs, which together have the goal to explore several areas of knowledge. The area of educational-robotics had its main growth after the development of the programmable brick, which is a small computer inserted within an assembling LEGO brick, afterwards sold as a robotic kit called LEGO Mindstorms. However, the cost of this robotic-kit lies around US\$ 700.00 in Brazil, and because of its high cost, cannot be used in many brazilian schools. In order to achieve the mainstream of the brazilian schools, a replacement kit was designed, it is called the GoGo board, which is also a programmable brick, whose source code and project are free and available, and it is built using low cost components, allowing the user or student himself to build the board. From the usage of the GoGo board, this study revealed that it is possible to design and develop projects in the educational-robotics at low costs and easy to build and implement, therefore accessible to the reality of the brazilian schools.

Key-words: Robotics on Education, Digital Tecnology, Learning Environment.

Sumário

1	Introdução	15
2	As Origens da Robótica Educacional	19
2.1	As Primeiras Tartarugas Robóticas	19
2.2	Iniciativas Educacionais	21
2.3	A Tartaruga Logo	21
2.4	O LEGO/Logo	23
3	Tijolos Programáveis	27
3.1	Introdução	27
3.2	O Tijolo Logo “ <i>Logo Brick</i> ”	27
3.3	Tijolo Programável “ <i>Red Brick</i> ”	29
3.4	O LEGO Mindstorms	30
3.4.1	Introdução	30
3.4.2	Especificação do Hardware do RCX	31
3.4.3	A Estrutura Física do RCX	32
3.4.4	A Estrutura Lógica do RCX	34
3.5	Cricket	35
3.5.1	Introdução	35
3.5.2	O Software do Cricket	36
4	A Placa GoGo	39
4.1	Introdução	39
4.2	Especificação do Hardware da Placa GoGo	39
4.3	Como a Placa GoGo Trabalha	40
4.3.1	A Estrutura Física da Placa GoGo	42
4.3.2	A Estrutura Lógica da Placa GoGo	45
4.4	Módulos Adicionais	47
4.4.1	Módulo <i>Display</i>	47
4.4.2	Módulo Amplificador	48
4.5	Montagem da placa GoGo	48
4.5.1	Compra dos componentes	49
4.5.2	Fabricação da Placa de Circuito Impresso	50
4.5.3	Gravação do <i>BootLoader</i> no Microcontrolador	52
4.5.4	Montagem dos Componentes na Placa de Circuito Impresso	53
4.5.5	Transferência do Firmware para a Placa GoGo	54
5	Protótipo de um Robô Móvel de Baixo Custo	57

5.1 Introdução	57
5.2 Sensores	57
5.2.1 Sensor de Toque.....	58
5.2.2 Sensor de Luminosidade Refletivo	58
5.3 Motores	59
5.4 Estrutura Mecânica do Protótipo	60
5.4.1 Rodas de Tração e Apoio	61
5.4.2 O Chassi do Protótipo	62
5.4.3 Engrenagens e Eixos	63
5.5 A Montagem do Protótipo	64
5.6 Programação do Protótipo utilizando o Logo Cricket	66
5.6.1 Um Programa para Movimentar o Robô.....	66
5.6.2 Mudar a Direção do Robô.....	67
5.6.3 Repetindo Comandos	68
5.6.4 Movendo em Espiral	69
5.6.5 A Instrução <i>se</i>	70
5.6.6 Agindo através de um Sensor de Toque.....	71
5.6.7 Sensor de Luminosidade	71
6 Experimentos e Resultados	75
6.1 Introdução	75
6.2 Programando o Robô Coletor	77
6.2.1 Procedimento “Enlatado”	78
7 Conclusão	81
Referências	83

Índice de Figuras

Figura 1 - Machina Speculatrix.....	19
Figura 2 - Veículo Braitenberg 1.....	20
Figura 3 - Veículos Braitenberg 2a e 2b	20
Figura 4 - Veículos Braitenberg 2a e 2b encontrando a luz.....	21
Figura 5 - Tartaruga de solo (LOGO FOUNDATION, 2007).....	22
Figura 6 - Tartaruga gráfica	22
Figura 7 - Interface externa LEGO (CHELLA, 2002).....	24
Figura 8 - Interface interna LEGO (CHELLA, 2002).....	25
Figura 9 - Tijolo Logo (MARTIN et al., 2000).....	28
Figura 10 - Tijolo programável Red Brick (MARTIN et al., 2000).....	29
Figura 11 - RCX (Robotics Command Explorer).....	31
Figura 12 - Torre de comunicação infravermelho.....	32
Figura 13 - Placa do RCX.....	33
Figura 14 - Estrutura lógica do RCX	34
Figura 15 - Cricket (MARTIN et al., 2000).....	35
Figura 16 - Placa GoGo.....	40
Figura 17 - Modo conectado da placa GoGo.....	40
Figura 18 - Interface GoGo Console.....	41
Figura 19 - Modo autônomo da placa GoGo.....	41
Figura 20 - Interface Procedimentos Logo.....	42
Figura 21 - Placa GoGo (GOGO, 2007a).....	43
Figura 22 - Portas dos sensores	44
Figura 23 - Estrutura lógica da placa GoGo.....	45
Figura 24 - Procedimento Logo Cricket.....	46
Figura 25 - Mapa da memória de programa.....	47
Figura 26 - Módulo display.....	47
Figura 27 - Módulo amplificador.....	48
Figura 28 - Processo de montagem da placa GoGo.....	49
Figura 29 - Impressão do circuito da placa GoGo.....	50
Figura 30 - Processo térmico de transferência.....	51
Figura 31 - Placa após remoção do papel glossy.....	51
Figura 32 - Placa após o processo de corrosão.....	52
Figura 33 - Placa após limpeza e perfuração	52
Figura 34 - Mapa da memória de programa.....	53

Figura 35 - Circuito de fornecimento de energia.....	54
Figura 36 - Circuito da porta serial	54
Figura 37 - Circuito completo	54
Figura 38 – Cone Conector para sensor passivo.....	57
Figura 39 - Conector para sensor ativo	58
Figura 40 - Sensores de toque utilizando micro chaves	58
Figura 41 - Circuito do sensor de luminosidade ativo.....	59
Figura 42 - Invólucro do sensor de luminosidade ativo	59
Figura 43 - Motor 13V	60
Figura 44 - Motores 5,9 a 13V	60
Figura 45 - Veículo triciclo com tração traseira	61
Figura 46 - Mini CD 8 cm	61
Figura 47 - Roda giratória	62
Figura 48 - Rodas e eixo do robô.....	62
Figura 49 - Chassi do robô.....	63
Figura 50 - Conjunto de engrenagens e eixo.....	63
Figura 51 - Engrenagens fixadas nos eixos.....	64
Figura 52 - Montagem das rodas	64
Figura 53 - Montagem dos motores	65
Figura 54 - Montagem do suporte de pilhas	65
Figura 55 - Montagem dos sensores.....	65
Figura 56 - Montagem da placa GoGo no chassi.....	66
Figura 57 - Protótipo montado com o display.....	66
Figura 58 - Robô com sensor de luminosidade	72
Figura 59 - O desafio Enlatado.....	75
Figura 60 - Robô móvel LEGO Mindstorms	76
Figura 61 - Robô móvel com a placa GoGo	76
Figura 62 - Robô coletor de latas	77
Figura 63 - Estrutura do programa utilizando o NQC - RCX.....	77
Figura 64 - Estrutura do programa utilizando o Logo Cricket - GoGo.....	78

Índice de Tabelas

Tabela 1 - Projetos selecionados na FEBRACE	17
Tabela 2 - Descrição da placa GoGo	44
Tabela 3 - Lista de componentes placa GoGo 3.0 em março/2007.....	50

Índice de Listagens

Listagem 1 - Procedimento para acionamento dos motores do robô.....	67
Listagem 2 - Procedimento para mudança de direção do robô.....	68
Listagem 3 - Procedimento para o robô mover-se ‘em quadrado’	68
Listagem 4 - Procedimento para o robô mover-se ‘em quadrado’ duas vezes	69
Listagem 5 - Procedimento para o robô mover-se em espiral.....	70
Listagem 6 - Procedimento para o robô mover-se para esquerda e direita	71
Listagem 7 - Procedimento para utilização dos sensores de toque	71
Listagem 8 - Procedimento para o robô retornar quando encontrar uma linha preta ...	73
Listagem 9 - Procedimento principal para o desafio Enlatado	79
Listagem 10 - Procedimento que monitora linha através do sensor7	80
Listagem 11 - Procedimento que define a direção que o robô irá virar	80
Listagem 12 - Procedimento que controla a quantidade de latas já encontradas	80
Listagem 13 - Procedimento que monitora o robô quando encontra uma lata	80

1 Introdução

O computador pessoal está cada vez mais presente no cotidiano dos lares, escolas e ambientes de trabalho, mas existem outros computadores dedicados a tarefas específicas e contidos em outros equipamentos, como nos telefones celulares, nos eletrodomésticos, nos semáforos, no controle dos motores dos automóveis, e que passam despercebidos. Em tais equipamentos, diferente dos computadores de propósito geral, como o computador pessoal, um sistema embarcado realiza um conjunto de tarefas pré-definidas, geralmente com requisitos específicos. Já que os referidos sistemas são dedicados a tarefas específicas, através de engenharia, pode-se otimizar os projetos reduzindo tamanho, recursos computacionais, consumo de energia, e custo do produto.

Os computadores pessoais estão nas escolas, mas em algumas atividades, em que são necessários recursos computacionais com características como baixo consumo de energia, mobilidade, tamanho reduzido, e dispositivos de entrada e saída não convencionais, tais como sensores e motores, não podem ser utilizados. Estas características normalmente são necessárias quando os alunos utilizam recursos computacionais inseridos, por exemplo, em seus protótipos para uma feira de Ciências.

Visando atender tais necessidades foi desenvolvido, no Massachusetts Institute of Technology (MIT) (MIT, 2007), um tijolo¹ programável (*Programmable Bricks*) que é um computador muito pequeno inserido dentro de um bloco de montagem LEGO do tamanho de uma caixa pequena de suco. Este pequeno computador pode ser programado para interagir com o ambiente externo através de sensores de temperatura, de toque e de intensidade de luz; e atuadores como motores, buzinas e luzes. A programação do tijolo é realizada em um computador pessoal utilizando uma versão da linguagem Logo, conhecida como Logo de tijolo (*Logo Bricks*), sendo em seguida efetuada a transferência do programa para o tijolo programável através de um cabo serial ou através de comunicação infravermelha. A partir deste ponto podem ocorrer duas situações: o tijolo continuar conectado com o computador pessoal e continuar trocando informações ou tornar-se autônomo e independente (RESNICK et al., 1996).

A partir do protótipo do primeiro bloco programável foi desenvolvido o conjunto de robótica educacional² LEGO Mindstorms, que pretendia suprir a lacuna descrita no que

¹ Tijolo é o nome dado ao bloco de montar LEGO pelo fabricante e aceito pelos usuários.

² Robótica Educacional é o ambiente constituído pelo computador, componentes eletrônicos, eletromecânicos e programa, onde o aprendiz, por meio da integração destes elementos, constrói dispositivos automatizados com o objetivo de explorar conceitos das diversas áreas do conhecimento (CHELLA, 2002, p. 23, apud D'ABREU, 1993).

concerne à possibilidade de criação de dispositivos automatizados, com funcionamento autônomo e independente do computador pessoal. Este conjunto de robótica é comercializado pela empresa LEGO, sendo o RIS (*Robotic Invention System*) o nome do conjunto, que contém o tijolo programável RCX (*Robotic Command Explorer*), a torre de comunicação infravermelha (*Infrared Tower*), dois sensores de toque, um sensor de luminosidade, dois motores, e mais de 700 peças LEGO (LEGO, 2007).

A utilização de dispositivos de computação embarcada, como o RCX que faz parte do conjunto LEGO Mindstorms, em ambientes educacionais de robótica pode oferecer aos estudantes, do ensino fundamental até a Universidade, uma ferramenta para o desenvolvimento de projetos que necessitem de um computador de tamanho reduzido, com facilidade de mobilidade e de aprendizado da programação (FORTES, 2007; RIBEIRO, 2006; TEIXEIRA, 2006; KLASSNER; ANDERSON, 2003; STEFFEN, 2002; VALLIM, 2000).

Para Lopes e Fagundes (2006),

Esta efetividade da aprendizagem na construção de robôs foi comprovada em vários estudos, como os do Media Lab (Resnick & Ocko, 1991; Hancock, 2001), do Laboratório de Estudos Cognitivos/IP/UFRGS (Lopes & Fagundes, 1995), na UNICAMP/NIED (D'Abreu, 1993), entre outros. Estes estudos comprovam que a atividade de programação, de design e de depuração de protótipos permitem ao sujeito enriquecer seus esquemas de significação com novos esquemas de representação lógico-matemáticos, lingüísticos e estéticos, elementos essenciais da aprendizagem. (LOPES; FAGUNDES, 2006, p. 2).

Porém, o conjunto LEGO Mindstorms, que inclui o RCX, além de outros materiais de construção, por exemplo, custa em torno de US\$ 200,00 nos Estados Unidos e em torno de US\$700,00 no Brasil, devido a impostos de importação e taxas, sendo um custo proibitivo para a realidade da maioria das escolas brasileiras. Este alto custo torna a sua utilização restrita a poucas escolas (SIPITAKIAT et al., 2004; ALVES et al., 2005; MIRANDA, 2006).

Por exemplo, nos anais dos últimos cinco anos da Feira Brasileira de Ciências e Engenharia (FEBRACE) (LOPES et al., 2003, 2004, 2005, 2006, 2007) podemos constatar, conforme Tabela 1, que os projetos desenvolvidos por algumas equipes de escolas relacionados à automação de processos, monitorações ambientais e protótipos robóticos, utilizaram o conjunto comercial LEGO Mindstorms, mas que também algumas equipes de escolas utilizaram outras alternativas, com a construção e utilização de protótipos feitos com vários modelos de microcontroladores³ disponíveis no mercado, como da família PIC, 8051, AVR, BasicStep, entre outros.

³ Microcontrolador é um pequeno computador programável contido em um único *chip* e utilizado para controlar dispositivos eletrônicos.

A diversidade de microcontroladores utilizados pode indicar a inexistência de um produto alternativo ao conjunto comercial LEGO Mindstorms, e que tenha custo mais acessível e que seja tão eficiente quanto um tijolo programável.

Feira Brasileira de Ciência e Tecnologia	2003	2004	2005	2006	2007
Projetos que utilizaram o LEGO Mindstorms	2	2	3	2	2
Projetos que utilizaram microcontroladores	1	4	5	3	6
Projeto que utilizaram a placa GoGo	0	0	2	2	4
Projetos onde não foram possíveis identificar o recurso utilizado	21	34	28	31	30
Total dos projetos relacionados a automação de processos, monitorações ambientais e protótipos robóticos	24	40	38	38	42
Total Geral dos Projetos Selecionados	93	194	201	207	228

Tabela 1 - Projetos selecionados na FEBRACE

Como alternativa ao tijolo programável RCX do conjunto de robótica LEGO Mindstorms surgiu a placa GoGo, uma placa programável, também desenvolvido no MIT, mas com alguns compromissos diferenciados, como simplicidade de projeto; baixo custo dos componentes; e projeto e código abertos, o que possibilita a construção pelo próprio usuário (GOGO, 2007a; GOGO, 2007b).

A placa GoGo, assim como o RCX do conjunto de robótica LEGO Mindstorms, é apenas a parte computacional da implementação de um robô móvel, necessitando ainda, de sensores, de motores e de uma estrutura mecânica. Assim, para a utilização da placa GoGo na montagem de um robô móvel é necessária a compra de componentes de baixo custo e a montagem dos sensores, dos motores e da estrutura mecânica do robô. Uma possibilidade interessante para a estrutura mecânica é a utilização de sucata eletrônica, devido à redução de custos, e ainda, por reutilizar um material que seria descartado e que possivelmente iria para o lixo (CAVALLO; BLIKSTEIN, 2004; CÉSAR; BONILLA, 2007).

Conforme observou César e Bonilla (2007), a utilização de componentes de sucata eletrônica pode ser uma fonte de materiais alternativos para a Robótica Educacional,

Atualmente existe uma situação de descarte de equipamentos obsoletos ou inutilizados que não justificam sua manutenção. Tais equipamentos costumam conter dispositivos eletromecânicos, como motores e sensores, além de materiais que podem ajudar o educando na montagem de seus projetos de controle do DEC [Dispositivos Eletrônicos a serem Comandados], como eixos, roldanas, engrenagens, fiações, bornes de ligação, resistores, transistores, reguladores de tensão, etc. Estes dispositivos e materiais podem ser reaproveitados. Esta possibilidade não se limita a equipamentos de informática. Aparelhos eletrônicos em geral, máquinas fotográficas e brinquedos fora de uso, podem ser também aproveitados

integralmente ou em parte no projeto de controle do DEC. (CÉSAR; BONILLA, 2007, p. 4).

Neste contexto, o presente trabalho propõe a construção de um robô móvel com a placa GoGo e componentes de baixo custo ou reaproveitados de sucata eletrônica, tendo como objetivo principal contribuir no estudo e aplicação de ferramentas tecnológicas acessíveis à realidade das escolas brasileiras, podendo tais ferramentas serem utilizadas em situações de ensino-aprendizagem através da Robótica Educacional.

A dissertação está organizada da seguinte forma: o capítulo 2 apresenta um breve histórico da robótica educacional até a primeira geração de tijolos programáveis; o capítulo 3 discute o surgimento e a evolução de dispositivos autônomos para a robótica educacional; o capítulo 4 descreve o hardware e o software da placa de baixo custo GoGo; o capítulo 5 discorre sobre a montagem de um protótipo de robô com a placa GoGo e materiais alternativos de sucata eletrônica, bem como apresenta a programação do robô utilizando a linguagem Logo Cricket; o capítulo 6 apresenta um experimento utilizando o protótipo do robô chamado desafio robótico “Enlatado”; o capítulo 7 apresenta a conclusão sobre este trabalho e as perspectivas para futuras pesquisas.

2 As Origens da Robótica Educacional

2.1 As Primeiras Tartarugas Robóticas

No início da década de 50 o neurologista William Grey Walter construiu robôs móveis⁴ que denominou de tartarugas, conforme Figura 1, com os nomes de Elmer e Elsie, tendo também, com sentido de humor, criado um nome para essa espécie de tartaruga robótica: "*Machina Speculatrix*" (SABBATINI, 2007). Cada tartaruga tinha um conjunto de dois comportamentos elementares: no primeiro a tartaruga após colidir com um obstáculo afastava-se dele, e no segundo, ela se aproximava de fontes de luz, exceto se a fonte de luz fosse muito forte, caso em que se afastava. Embora limitados a estes comportamentos os pequenos robôs eram capazes, ainda assim, de exibir uma interação complexa com o ambiente e entre si, por exemplo, quando um deles possuía uma fonte de luz.

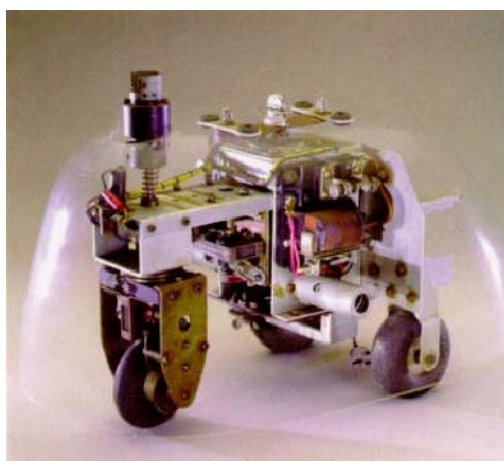


Figura 1 - Machina Speculatrix

O objetivo inicial de Walter foi o de criar robôs que exibissem comportamentos, ainda que simples, plausíveis de serem encontrados em animais.

No livro, publicado em 1984, "Vehicles: Experiments in Synthetic Psychology" (BRAITENBERG, 1984), Valentino Braitenberg descreveu uma série de veículos que eram máquinas hipotéticas autônomas construídas através de conexões de elementos eletrônicos simples. Eles exibiam uma gama de comportamentos, evoluindo desde da atração simples e repulsão a "emoções" complexas e "inteligência" (CORREA, 2005).

O veículo de Braitenberg número 1 é mostrado na Figura 2. Este veículo consiste em

⁴ Um robô móvel é um agente autônomo capaz de extrair informações do ambiente e utilizar esse conhecimento do mundo para deslocar-se de modo significativo e intencional, atuando e executando tarefas.

um sensor de luminosidade conectado (por um fio) em um motor. Quanto mais luminosidade recebida pelo sensor, mais rápido o motor gira.

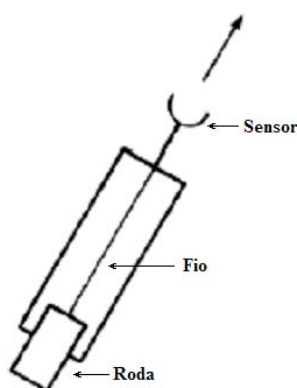


Figura 2 - Veículo Braitenberg 1

Os veículos números 2a e 2b fazem uso de dois destes circuitos com sensor de luminosidade e motor, para formar um veículo que tem a habilidade para virar, conforme Figura 3. Além disso, como indicado na Figura 4, estes veículos, ou exibem um tropismo positivo com atração para a luz, ou um tropismo negativo com repulsa para a luz.

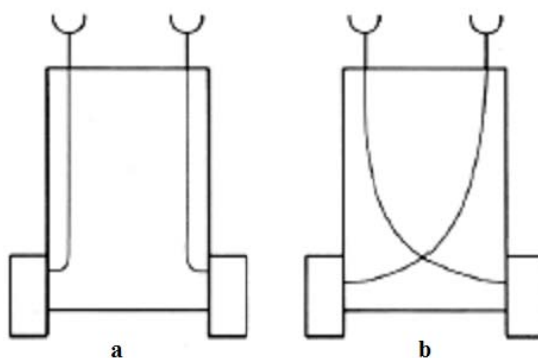


Figura 3 - Veículos Braitenberg 2a e 2b

Se uma luz é posicionada à sua esquerda, como no veículo 2b, mostrado na Figura 4, então seu sensor de luminosidade esquerdo será estimulado em um maior grau que seu sensor direito, e sua roda direita ficará mais rápida que sua roda esquerda, isto fará o veículo virar para a luz, sendo que a interação de sensor e o motor continuará dirigindo para a luz. Em um modo complementar, o veículo 2a, mostrado na Figura 4, dirige-se para longe da luz (MARTIN, 1988).

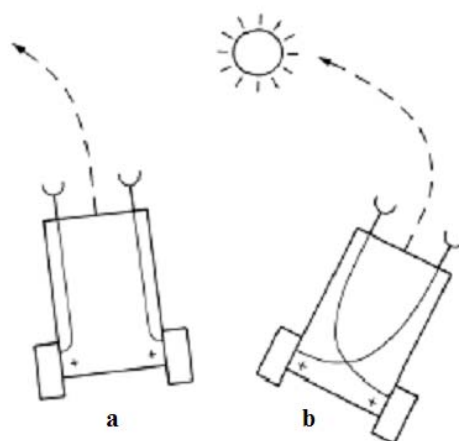


Figura 4 - Veículos Braitenberg 2a e 2b encontrando a luz

2.2 Iniciativas Educacionais

Seymour Papert foi pioneiro em utilizar a robótica para fins educacionais. Ele desenvolveu a linguagem Logo no MIT, tendo, a princípio, criado uma tartaruga de solo para utilização da versão inicial da linguagem Logo, inspirado nas tartarugas desenvolvidas por Grey Walter. Era um dispositivo móvel pequeno que poderia ser controlado através de comandos do computador. Ele pretendia que a tartaruga fosse um objeto no qual as crianças poderiam usar o conhecimento de seu próprio corpo para compreender o movimento da tartaruga, podendo, deste modo, a tartaruga se tornar um “objeto para pensar com” (*thing to think with*). Com o surgimento de computadores pessoais, a tartaruga de solo deu lugar a uma tartaruga virtual, que se movimentava na tela do computador (MARTIN, 1988).

No sistema chamado LEGO/Logo este "ambiente animal artificial" foi levado novamente para fora da tela do computador, para o mundo físico. Projetado pelo esforço comum entre a empresa dinamarquesa LEGO, e Seymour Papert, Stephen Ocko e Mitchel Resnick, estes do MIT, o LEGO/Logo é um conjunto de montagem que consiste em material plástico de montagem LEGO - vigas, blocos, engrenagens, polias, rodas e motores - e uma interface de computador que envolve sensor e programação Logo.

Os materiais fazem parte de um ambiente de aprendizagem para crianças, no qual elas constroem máquinas motorizadas e então as conecta ao computador, permitindo que programas possam ser desenvolvidos para animar as suas construções.

2.3 A Tartaruga Logo

Quando a linguagem de programação para crianças Logo foi desenvolvida por Seymour

Papert no MIT, nos anos 60, os computadores utilizados na época eram de médio porte e os recursos gráficos de vídeo eram uma novidade pouco acessível. Devido a esta limitação de tecnologia da época o Logo não era como hoje, em que a principal característica é a interação do usuário através de uma interface gráfica para controlar uma tartaruga na tela visando desenvolver o raciocínio em geometria e nas regras matemáticas do movimento (CHELLA, 2002).

Para contornar a limitação de interação dos computadores da época e também como uma representação mais concreta para que as crianças, segundo Papert, pudessem imaginar-se no lugar da tartaruga para facilitar a programação, ele criou, inspirado na abordagem de W. Grey Walter, uma tartaruga mecânica que tinha uma caneta em seu corpo e ficava no solo sobre um papel. Ela era ligada ao computador por um longo cabo serial, por onde recebia os comandos de programação Logo e desenhava no papel. Esta primeira tartaruga Logo é conhecida na literatura por “tartaruga de solo” em contraste com a tartaruga atual “tartaruga gráfica”, conforme Figuras 5 e 6. Esta iniciativa foi pioneira na utilização da robótica na educação e inspirou os trabalhos posteriores.



Figura 5 - Tartaruga de solo (LOGO FOUNDATION, 2007)

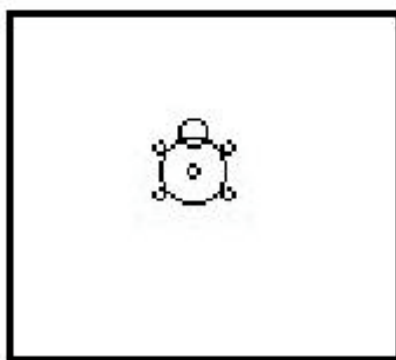


Figura 6 - Tartaruga gráfica

A tartaruga de solo foi uma tentativa de dar uma expressão “corporal” para a programação Logo e possibilitar um nível de comunicação através do movimento com o computador. O computador passou a ter “pernas” e “braços” e até alguns “sentidos” como o tato através de sensores de toque e uma “pequena visão” através dos sensores de luminosidade.

Um exemplo bastante esclarecedor sobre a abordagem de programação da tartaruga de solo é o desenvolvimento de um programa para a tartaruga movimentar-se ao redor de um objeto utilizando os sensores de toque para “sentir” o objeto. Ela vai se movimentando para frente e “tateando” o objeto através do sensor de toque, e o mais interessante, observa Papert, é que esta estratégia funciona para qualquer forma de objeto, seja uma caixa ou um cilindro. No livro “A Máquina das Crianças”, publicado em 1994 (PAPERT, 1994), Papert volta a tratar desta forma de utilizar a programação e a define como abordagem cibernética, mais inspirada na biologia do que na abordagem lógica, como contar os passos necessários para a tartaruga andar para frente e então virar 90° para a esquerda e assim por diante. Ele afirma que são abordagens epistemológicas bem diferentes, enquanto a cibernética trabalha com um conhecimento incerto controlado, a abordagem lógica trabalha com o conhecimento certo e preciso.

2.4 O LEGO/Logo

O sistema LEGO/Logo é um ambiente computacional que possibilitou a construção de dispositivos com comportamentos programáveis. Ele é o resultado do trabalho conjunto de um grupo de pesquisadores do Epistemology and Learning Group do MIT e da indústria dinamarquesa LEGO (OCKO et al., 1987).

O LEGO/Logo é formado pelos seguintes elementos:

- a) Peças tradicionais da LEGO, tais como, tijolos, eixos, polias, engrenagens, correntes etc..
A essas peças foram acrescentados motores e sensores de toque, de luz e contáctos;
- b) A linguagem Logo foi acrescida de comandos específicos relativos aos motores e sensores, tais como, comandos para ligá-los (ligamotor) e desligá-los (desligamotor), comandos para ativar os sensores (ligasensor), e para contar pulsos emitidos pelos sensores (contapulso); e
- c) Uma interface que estabelece um canal de comunicação entre o computador e o dispositivo, sendo que esse canal de comunicação permite a troca de informações nos dois sentidos: informações do computador para o dispositivo e vice-versa.

A construção de dispositivos é obtida através da combinação das peças da LEGO,

motores e sensores, e uma vez feita a ligação entre o computador e a interface, pode-se usar a linguagem Logo ampliada para escrever procedimentos que, explorando a troca de informações entre o computador e o dispositivo, determinam o comportamento do dispositivo.

O sistema LEGO/Logo resgata a idéia da tartaruga mecânica controlada pelo Logo, porém há uma diferença entre o antigo ambiente de tartarugas mecânicas e o ambiente proporcionado pelo LEGO/Logo. No antigo ambiente a tartaruga era um objeto mecânico já pronto, sendo que a única tarefa possível era controlar o seu movimento. Já no ambiente LEGO/Logo o usuário tem a oportunidade de construir o objeto mecânico que desejar, sendo que tal objeto pode ser uma tartaruga, um outro animal qualquer, um carro, um braço mecânico, um semáforo, etc..

Portanto, o usuário do LEGO/Logo participa de duas atividades: o projeto do objeto mecânico, que não se restringe às tartarugas, e o projeto dos programas computacionais que controlam o comportamento do objeto.

Com o LEGO/Logo as crianças podem construir vários dispositivos mecânicos, como um elevador, uma criatura robótica, e conectá-los à uma caixa de interface externa, mostrada na Figura 7, sendo que esta é ligada a uma interface interna no computador, conforme Figura 8. Somente após a ligação entre os dispositivos mecânicos montados e o computador, através destas duas interfaces, é possível escrever e executar um programa Logo para controlar o movimento deles.

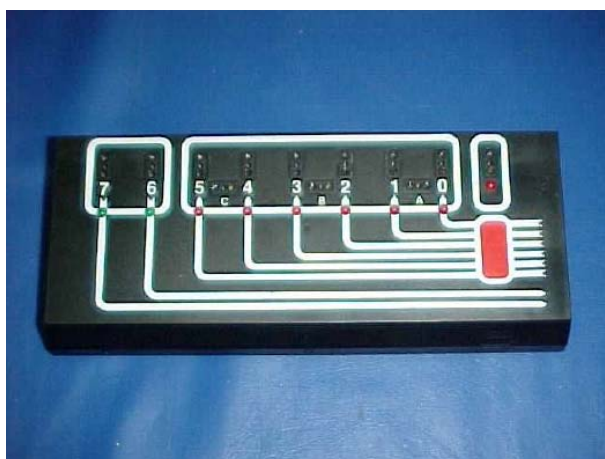


Figura 7 - Interface externa LEGO (CHELLA, 2002)



Figura 8 - Interface interna LEGO (CHELLA, 2002)

O sistema de LEGO/Logo ficou comercialmente disponível no final dos anos 80, vendido para as escolas pela LEGO com o nome LEGO tc logo. O LEGO/Logo foi uma inovação porque foi o primeiro conjunto de construção robótica disponibilizado amplamente, embora tivesse algumas limitações. As máquinas construídas pelas crianças tinham que ser conectadas a um computador através de cabos. Se a máquina tivesse apenas um motor não havia inconveniente, mas uma vez que a máquina tivesse um par de motores e um sensor, por exemplo, cada um com seu próprio cabo, haveria um conjunto de cabos conectados entre a interface e a máquina, limitando a sua mobilidade (MARTIN et al.,2000).

3 Tijolos Programáveis

3.1 Introdução

Um tijolo programável (*Programming Bricks*) é um computador muito pequeno inserido dentro de um tijolo de montagem LEGO do tamanho de uma caixa pequena de suco. Este pequeno computador pode ser programado para interagir com o ambiente externo através de sensores de temperatura, de toque e de intensidade de luz, e atuadores como motores, buzinas e luzes (RESNICK et al., 1996).

Os tijolos programáveis tiveram sua origem nas pesquisas efetuadas nos laboratórios do MIT com o LEGO/Logo, tendo sido uma evolução de uma pesquisa anterior do MIT, “*Braitenberg Brick*”, onde um tijolo eletrônico (*Electronic Bricks*), que embora contivesse algumas funções lógicas predefinidas como portas lógicas, *flip-flops* e cronômetros, ainda não era programável através de software (HOGG et al., 1991).

A programação do tijolo é realizada em um computador pessoal utilizando uma versão Logo, conhecida como Logo de tijolo, sendo em seguida efetuada a transferência do programa para o tijolo programável através de um cabo serial ou através de comunicação infravermelha. A partir deste ponto podem ocorrer duas situações: o tijolo continuar conectado com o computador pessoal e continuar trocando informações ou tornar-se autônomo e independente.

Para alguns projetos desenvolvidos com o tijolo programável, com a monitoração de variáveis ambientais como temperatura e intensidade de luminosidade, o modo conectado com o computador era viável. Já nas aplicações com robôs móveis a autonomia era fundamental para o desempenho da aplicação e a ligação via cabo tornou-se um aborrecimento prático para o movimento do robô.

O objetivo do desenvolvimento dos tijolos programáveis foi proporcionar ambientes educacionais de computação que pudessem ser utilizados por estudantes para desenvolvimento de projetos que necessitassem de um computador de tamanho reduzido, com facilidade de mobilidade e de aprendizado da programação.

3.2 O Tijolo Logo “*Logo Brick*”

Na versão original do LEGO/Logo, uma caixa de interface, a qual era ligada através de cabos à máquina, devia ser conectada através de um cabo longo em um computador pessoal. As limitações deste sistema conduziram à idéia de um "computador em um tijolo LEGO" isso

poderia ser construído na própria máquina de LEGO.

Conseqüentemente o desenvolvimento do Tijolo Logo, ou seja, um computador dentro de um tijolo LEGO, possibilitou ao usuário programá-lo na linguagem Logo, conectado em um computador “grande”, que tem um teclado e uma tela, e depois utilizá-lo, desconectado e autonomamente, para controlar até quatro motores e receber informações de vários sensores.

O Tijolo Logo mede 9 cm de comprimento, 5 cm largura e 3 cm altura, o tamanho aproximadamente de uma caixa de suco pequena, conforme Figura 9, e é alimentado por um conjunto de baterias recarregáveis das mesmas dimensões.

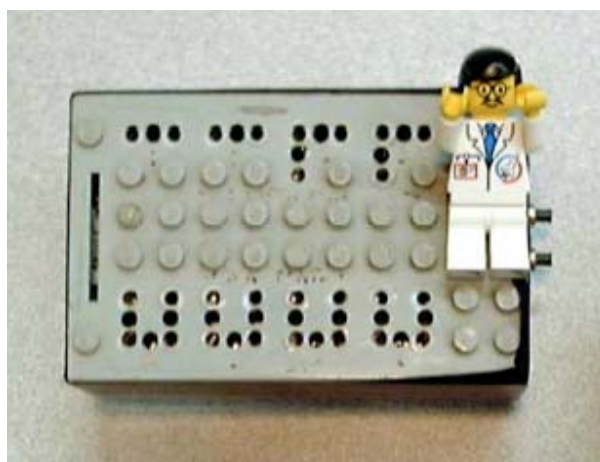


Figura 9 - Tijolo Logo (MARTIN et al., 2000)

Há várias implicações para projetos que são construídos utilizando o Tijolo Logo, em lugar de usar o LEGO/Logo original. A principal diz respeito à possibilidade das máquinas serem projetadas para ter mobilidade sem preocupação com fios enroscados. Isto encoraja mais invenções, que reproduzem comportamentos mais parecidos com o de "animais", em vez de máquinas estacionárias, isto porque cada máquina pode levar seu próprio programa a bordo do Tijolo Logo, sendo que projetos que utilizam mais de um Tijolo Logo são possíveis, sem técnicas de programação complexas, e tais projetos que envolvem mais que uma máquina revelam, freqüentemente, fenômenos emergentes imprevisíveis de interação entre as máquinas que são interessantes ao estudo.

Ao mesmo tempo em que o LEGO/Logo estava chegando às escolas, os pesquisadores do MIT começaram a pensar em seu sucessor, no qual pretendia-se introduzir os componentes eletrônicos necessários nos tijolos, sendo que isso ocorreu no ano de 1987, quando foram criados os primeiros protótipos de um “Tijolo Programável” prontos para serem usados com crianças em atividades de pesquisas em aprendizagem. Esses protótipos ficaram conhecidos como Tijolos Logo.

Um conjunto de experiências foi realizado usando o Tijolo Logo com estudantes. Essas experiências são descritas em “Children, Cybernetics, and Programmable Turtles” (MARTIN, 1988). Trabalhou-se de perto com um número pequeno de estudantes que construíram projetos que usavam o Tijolo Logo junto com uma tartaruga de LEGO (um robô pequeno), e com auxílio, as crianças escreveram programas para dar comportamentos diferentes à tartaruga, como a busca pela luz e o desvio de obstáculos. Foi observado com atenção a relação das crianças com a tecnologia. Algumas crianças gostaram de tratar a tartaruga robô como um animal de estimação, e gostavam quando este exibia comportamento imprevisível. Outras estavam mais interessadas em ter uma tartaruga para executar um conjunto específico de ações, e foram desafiadas pela dificuldade de fazê-lo.

3.3 Tijolo Programável “Red Brick”

A primeira geração do Tijolo Programável, conhecido como Tijolo Logo, foi um sucesso, na medida em que permitiu monitorar de perto o trabalho com as crianças. No entanto, foi necessário um desenvolvimento aplicado na construção dos Tijolos Programáveis, para que pudessem, efetivamente, serem utilizados em ambiente de sala de aula.

Durante o período de 94 a 96, foi criada a segunda geração de Tijolos Programáveis, que ficou conhecida como o “Red Brick” porque foi alojado em um invólucro plástico vermelho, conforme a Figura 10. Este tijolo foi projetado especificamente para ser robusto e de fácil fabricação. Durante a sua vida útil mais de 100 cópias do mesmo foram construídas para uso prolongado em escolas e centros comunitários.



Figura 10 - Tijolo programável Red Brick (MARTIN et al., 2000)

3.4 O LEGO Mindstorms

3.4.1 Introdução

A versão comercial do LEGO Mindstorms foi liberada no final de 1998, mas suas origens datam de muitos anos antes. Ele realmente é o resultado de dois processos separados: inovação e pesquisa.

O primeiro processo é representado pelo desenvolvimento contínuo de produtos novos da companhia LEGO, desde a primeira aparição dos blocos de plástico de montagem em 1949, Automatic Binding Brick, os quais conduziram à criação da série Technic em 1977. A série Technic abriu novas possibilidades para que crianças e adultos pudessem criar modelos articulados com crescente complexidade.

O segundo processo é o da pesquisa conduzida no Epistemology and Learning Group no Media Lab do MIT, conduzido por Fred Martin, Brian Silverman e Randy Sargent, sob a orientação dos professores Seymour Papert e Mitchel Resnick e financiado pela companhia LEGO. Eles iniciaram o trabalho em 1986, culminando no desenvolvimento do assim chamado “*Programmable Brick*”, uma unidade pequena capaz de se conectar ao mundo externo com uma variedade de sensores e atuadores, projetada para a criação de robôs e de outras aplicações em que um computador pode interagir com objetos do dia-a-dia.

A soma destes dois esforços trouxe à vida o RCX (*Robotics Command Explorer*), um microcomputador da companhia LEGO, baseado em parte na tecnologia desenvolvida no Media Lab do MIT, conforme Figura 11. O RCX programável foi completado com sensores, outras peças especiais e partes existentes da série LEGO Technic. O software utilizado pelo RCX foi especificamente projetado para possibilitar a conexão com um computador PC. Este conjunto de ferramentas transformou-se no LEGO Mindstorms Robotics Invention System 1.0 (FERRARI et al., 2002).



Figura 11 - RCX (*Robotics Command Explorer*)

3.4.2 Especificação do Hardware do RCX

O RCX é o núcleo do conjunto inteiro, podendo ler eventos externos através de sensores e controlar movimentos através de motores, e isto pode ser programado facilmente com o RCX Code que é uma linguagem de programação específica contida no software CD-ROM, e também pode ser programado com a ajuda de ferramentas desenvolvidas por terceiros como o ambiente de desenvolvimento RCXCC (*RCX Command Center*) e o compilador NQC⁵ (TAVARES et. al, 2004).

Conforme a Figura 11, o equipamento contém ambos os dispositivos de entrada: sensor ativo de luz (*light sensor*) e sensor passivo de toque (*touch sensor*), e dispositivos de saída: dois motores. Estes componentes são fundamentais para a robótica, eles possibilitam um modo de interagir com o mundo. Os sensores de toque e luz podem fornecer informações para o robô sobre seus movimentos, posicionamento e comportamento geral. Os motores são o ponto de partida para todo movimento e atividade, podendo controlar engrenagens, rodas, eixos e outras partes móveis.

A torre preta e cinza, conforme Figura 12, que faz parte do conjunto, é o componente

⁵ O “Not Quite C” (NQC) é um compilador para uma linguagem simples com sintaxe semelhante à linguagem ‘C’. O NQC gera códigos para a máquina virtual presente no firmware do RCX.

através do qual o robô LEGO se comunica com o PC. É na realidade um transmissor de infravermelho (IR) que envia código e informação do computador para o RCX e vice-versa.



Figura 12 - Torre de comunicação infravermelho

3.4.3 A Estrutura Física do RCX

Desmontando o RCX, dentro, encontra-se a placa principal na qual são montados e soldados muitos componentes, conforme Figura 13. Na visão frontal da placa, existe no centro um grande *display* de LCD cercado por quatro contatos para os botões de borracha e doze conectores para as seis portas de entrada e de saída. No lado esquerdo da placa há dois Leds de infravermelho (IR) e um receptor de infravermelho (IR), estes permitem comunicação do RCX com um PC, um controle remoto, ou outro RCX, através da torre. Ainda na visão frontal da placa existem um alto-falante, dois grandes capacitores e muitos outros pequenos elementos que a completam.

No verso da placa, existem um *chip* quadrado grande (microprocessador) e uma memória RAM. Finalmente, existem três controladores à direita para os motores, e dois *chips* para processamento de lógica digital no meio da placa.

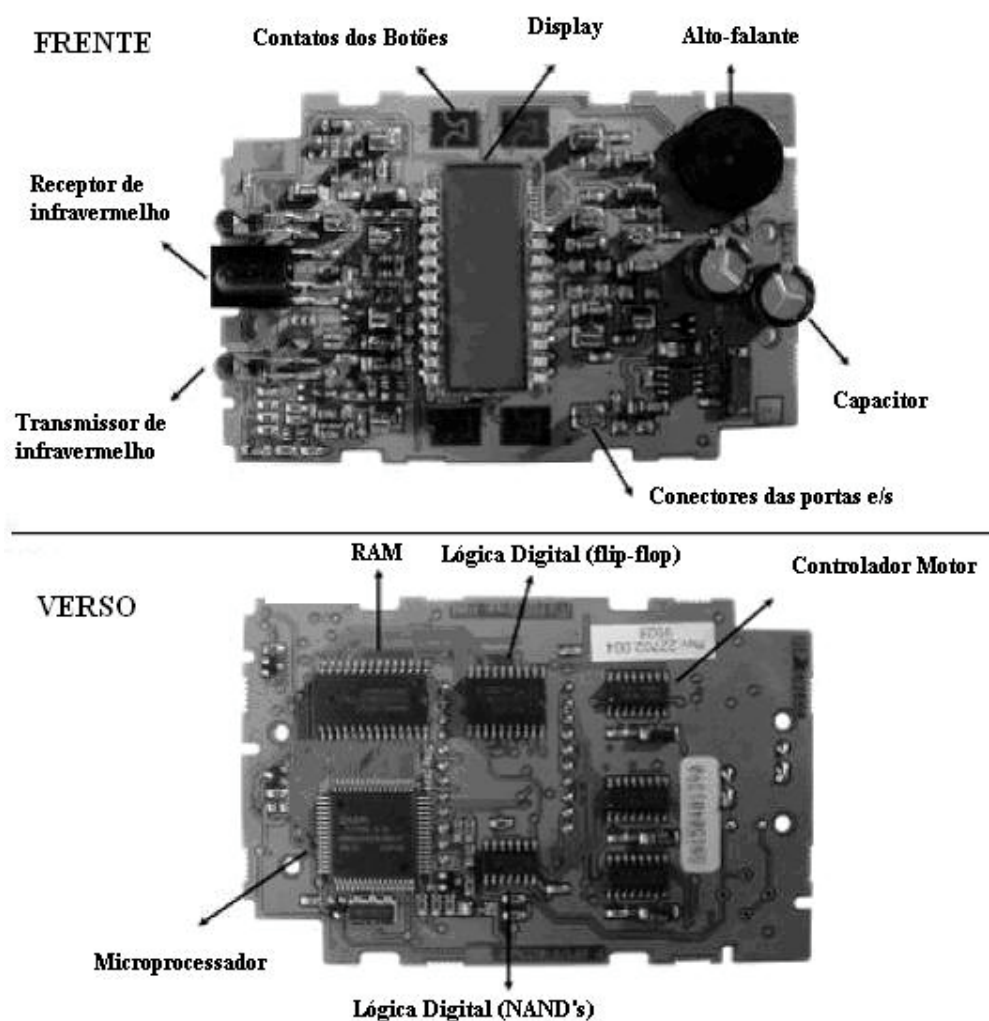


Figura 13 - Placa do RCX

Nos computadores, memória é o dispositivo que possibilita armazenar dados e programas do usuário, e como na maioria dos computadores, a memória do RCX é constituída de dois tipos:

- Memória Somente de Leitura (ROM): Este tipo de memória é permanentemente gravada no *chip* e não pode ser alterada;
- Memória de Acesso Aleatório (RAM): Este tipo de memória pode ser facilmente modificada ou acessada em qualquer ponto. Este dispositivo tem a necessidade de ser constantemente provido com energia para evitar que o conteúdo seja apagado.

O RCX possui 16 Kb de memória ROM embutida no microprocessador e 32 Kb de memória RAM em um *chip* externo.

3.4.4 A Estrutura Lógica do RCX

Um modelo lógico pode ser esboçado para melhor entender como o hardware do RCX trabalha. Através de uma estrutura de camadas múltiplas, este modelo é construído, conforme Figura 14.

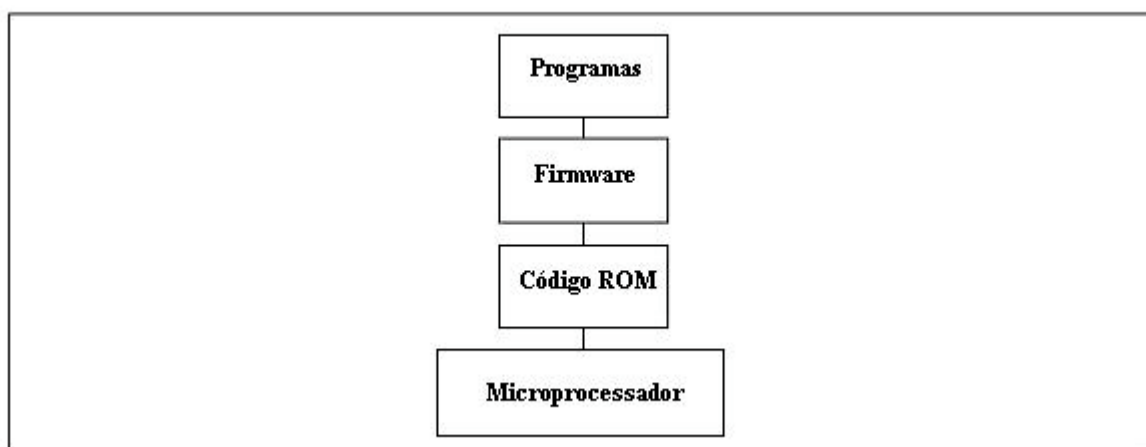


Figura 14 - Estrutura lógica do RCX

Na base, está o microprocessador Hitachi da série H8300. Ele executa as instruções de código de máquina. Componentes adicionais têm a tarefa de converter sinais de entrada das três portas de sensores em dados digitais.

A próxima camada no sistema é o código em ROM. O RCX vem com um conjunto de instruções que possibilitam todas as funcionalidades básicas à unidade, como controle de portas, display e comunicações de infravermelho. Estando familiarizado com esta arquitetura, podemos comparar o código em ROM ao código na BIOS de um computador pessoal que o inicializa e permite que se comunique com seus periféricos. Sem este comportamento de baixo nível não poderia ser feita qualquer coisa com o RCX, porque ele não poderia ser conectado com o mundo externo.

Acima do código em ROM está o *firmware*. Ele é um tipo de sistema operacional para o RCX, provendo funcionalidade para o sistema inteiro. A palavra *firmware* identifica um tipo particular de software que normalmente não é alterado pelo usuário final. Entretanto, neste caso, ocorre uma exceção importante: no RCX, o *firmware* pode ser alterado.

O *firmware* interpreta o *bytecode* (código intermediário, compacto e legível ao RCX) e o converte em instruções de código de máquina, chamando as rotinas de código em ROM para executar as operações padrões do sistema. Ele é carregado na memória RAM quando da primeira instalação do sistema, e sendo necessária recarga pela torre de IR toda vez que forem

retiradas as baterias do RCX por mais de alguns segundos.

Na camada seguinte, está o código de programa e dados do usuário. O software armazenado no PC converte o programa em um formato que é mais compacto e legível pelo processador do RCX (*bytecode*). A memória RAM é logicamente dividida em seções diferentes: 16 Kb para o firmware, 6 Kb para armazenar o programa do usuário, e o resto é usado para interpretar o *bytecode* e controlar dados durante a execução do programa (FERRARI et al.,2002).

3.5 Cricket

3.5.1 Introdução

Em algumas aplicações, o Tijolo Programável *Red Brick*, o qual deu origem ao LEGO Mindstorms, mostrou-se, ainda, inadequado em razão de seu tamanho, tendo havido necessidade do desenvolvimento de tijolo programável ainda menor. Assim surgiu o Cricket.

O Cricket é um pequeno computador programável do tamanho de uma bateria de 9 Volts que pode autonomamente controlar motores e receber informações de sensores, conforme Figura 15, sendo uma evolução do projeto do *Programmable Brick*, desenvolvido no MIT. Ele utiliza um microcontrolador PIC da empresa Microchip (MICROCHIP, 2007) e pode controlar atuadores básicos como motores e lâmpadas conectadas em um das duas saídas do Cricket, e monitorar simples sensores resistivos como interruptores, fotocélulas e termistores que são conectados em uma das entradas analógicas do Cricket (MARTIN et al., 2000).

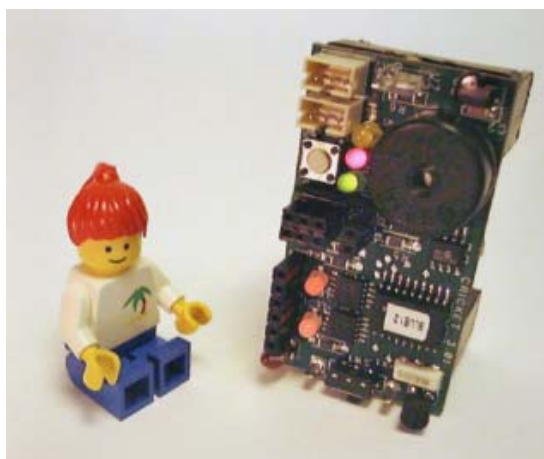


Figura 15 - Cricket (MARTIN et al., 2000)

Além disso, o Cricket possui a capacidade de comunicação infravermelha bidirecional que pode ser utilizada na comunicação dele com o Computador (para carregar programas no Cricket ou visualizar dados de sensores) e na comunicação entre Crickets, sendo que ele também inclui uma porta de expansão periférica, que amplia a capacidade do Cricket em se comunicar com outros dispositivos.

O Cricket possui um ambiente de software desenvolvido para ele, conhecido como Logo Cricket. O Logo Cricket é uma linguagem *procedural* que inclui variáveis globais e locais; procedimentos; argumentos e valores de retorno; estruturas de controle como “repeat”, “loop”, “if” e “ifelse”; e funções primitivas especializadas para interagir com motores e com o hardware de sensores.

3.5.2 O Software do Cricket

O sistema de software do Cricket envolve o uso de um computador pessoal, onde o Logo Cricket é desenvolvido e traduzido para o código objeto no formato de *bytecode*, os quais são transferidos para o Cricket propriamente dito, onde posteriormente é executado, através de uma máquina virtual desenvolvida na linguagem de montagem do microcontrolador PIC.

A máquina virtual do Cricket está gravada na memória ROM (memória somente de leitura) interna do microcontrolador PIC, sendo que a memória RAM (memória de acesso aleatório volátil) interna do microcontrolador PIC é usada durante sua execução.

O código do usuário reside em uma memória EEPROM (memória não volátil) e está conectado permanentemente ao microcontrolador. As rotinas de comunicação infravermelha embutidas incluem um protocolo para ler e escrever em uma memória EEPROM externa, e para pedir que a máquina virtual comece a execução do *bytecode* carregado na memória EEPROM.

Nas versões iniciais, o Cricket era implementado utilizando-se um microcontrolador PIC16C84, depois um PIC16F84 e finalmente o PIC16C715, bem como microcontroladores PIC compatíveis. O PIC16C715 executa instruções a 1 megahertz (MHz) com só 128 bytes de memória RAM interna e 2048 palavras de ROM.

Procedimentos definidos pelo usuário completam as funções primitivas providas pela linguagem, sendo que tais procedimentos podem aceitar um número arbitrário de parâmetros e opcionalmente um retorno. A mesma pilha usada para cálculo de resultados numéricos é usada para armazenar endereços de retorno de chamadas de procedimento. Por causa da quantidade pequena de memória RAM disponível no microcontrolador PIC utilizado, o

aninhamento e as chamadas recursivas de procedimento estão limitados a aproximadamente 16 níveis de profundidade.

Um sistema numérico de 16 bits disponível possibilita a aritmética de números inteiros com sinal, sendo também providos de operadores aritméticos padrão, operadores de comparação e operadores lógicos. Também há uma função de geração de número pseudo-aleatório, bem como variáveis globais para armazenar o estado de programa.

Os parâmetros dos procedimentos se comportam como variáveis locais, e adicionalmente 2048 bytes de 4096 bytes da memória física é posta de lado para armazenamento estendido de dados, sendo que um número arbitrário de vetores de dados pode ser declarado nesta área.

4 A Placa GoGo

4.1 Introdução

Os projetos desenvolvidos pelos pesquisadores do Future of Learning Group no Media Laboratory do MIT, utilizando o LEGO Mindstorms com alunos em países em desenvolvimento, encontraram dificuldade na sustentabilidade de suas atividades porque, ou eram extremamente caros ou não estavam disponíveis em tais países. Diante deste cenário, em 2001, os estudantes do MIT, Arnan (Roger) Sipitakiat e Paulo Blikstein, desenvolveram uma placa alternativa ao RCX do LEGO Mindstorms chamada GoGo (GOGO, 2007a).

Eles se inspiraram no Cricket e no LEGO Mindstorms que, a princípio, eram os únicos utilizados em seus projetos de pesquisa com estudantes.

A meta principal do projeto da placa GoGo foi criar uma plataforma de projeto aberto que fosse barata e permitisse a montagem local por todos aqueles que estivessem interessados. Foram escolhidos os componentes e o formato da placa de circuito impresso de forma que pudessem ser montados sem a necessidade de equipamentos especiais de soldagem. Os componentes foram selecionados cuidadosamente de forma que eles estivessem disponíveis em muitos países, podendo a placa ser construída nos Estados Unidos, Tailândia, Brasil, México, Malásia, China e em muitos outros países em desenvolvimento. O custo da placa foi mantido a um mínimo possível (BLIKSTEIN; CAVALLO, 2003; SIPITAKIAT et al., 2004).

4.2 Especificação do Hardware da Placa GoGo

A GoGo é constituída por uma placa de circuito impresso de face simples com dimensões de 11cm por 8cm, conforme Figura 16. Ela utiliza um microcontrolador PIC da empresa Microchip e controla atuadores básicos como motores e lâmpadas conectados em uma das quatro saídas da placa GoGo, e monitorar simples sensores resistivos como interruptores, fotocélulas e termistores que são conectados em uma das suas oito entradas analógicas.

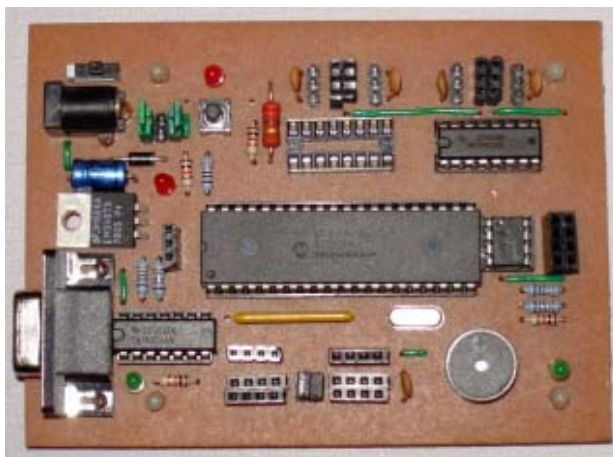


Figura 16 - Placa GoGo

Além disso, a placa GoGo possui a capacidade de comunicação bidirecional que pode ser utilizada na comunicação dela com o computador para carregar programas Logo Cricket ou visualizar dados de sensores, bem como também inclui uma porta de expansão periférica que amplia a capacidade da placa GoGo em se comunicar com outros dispositivos.

4.3 Como a Placa GoGo Trabalha

No *modo conectado*, conforme Figura 17, a placa funciona sempre ligada ao computador por um cabo serial, onde o aplicativo GoGo Board Monitor, através da interface “GoGo Console”, conforme Figura 18, possibilita a leitura dos sensores e o controle direto das portas da placa onde estão os motores. Este modo é indicado para algumas aplicações, como alguns tipos de jogos, trabalhos de artes interativos e outras aplicações onde a falta de mobilidade e a necessidade de ter um computador sempre junto com a placa não sejam limitantes.

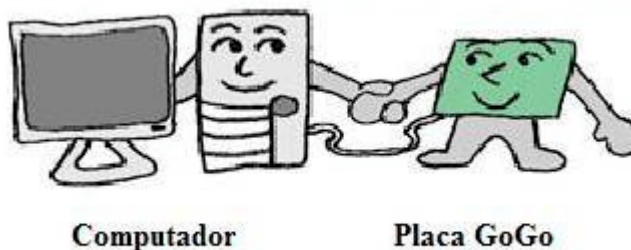


Figura 17 - Modo conectado da placa GoGo

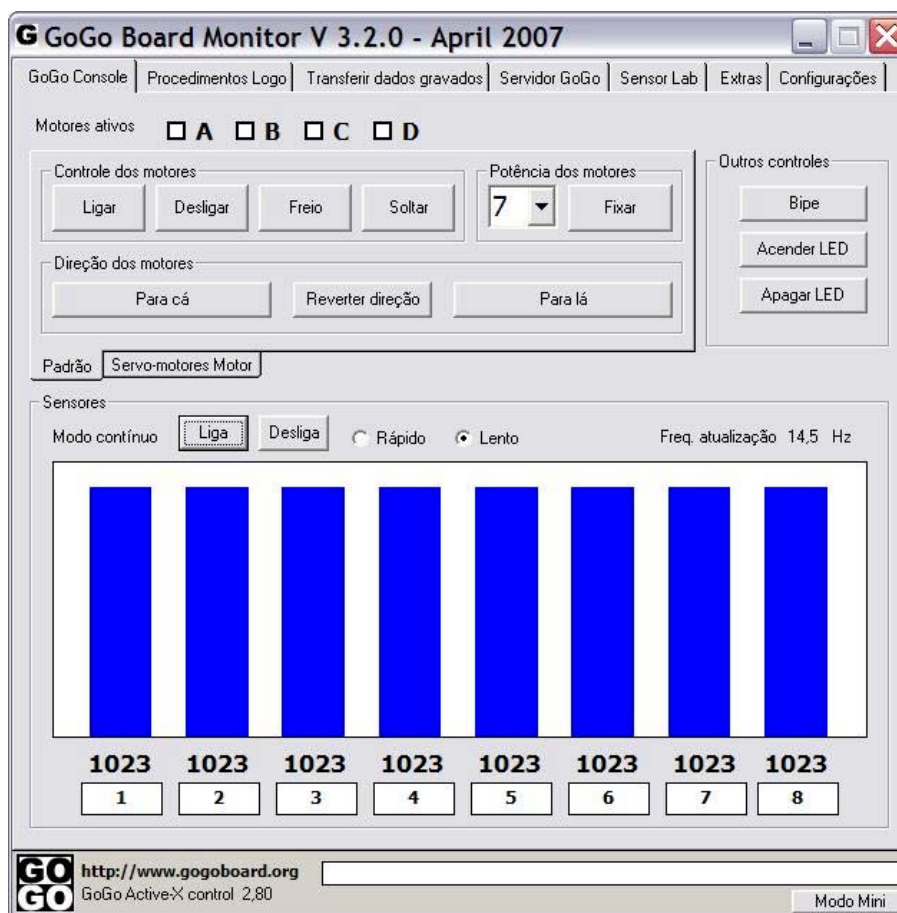


Figura 18 - Interface GoGo Console

Já o modo autônomo, conforme Figura 19, permite aos usuários, através do Monitor GoGo Board, criar procedimentos utilizando uma linguagem de programação chamada Logo Cricket, conforme Figura 20, e transferi-los para a placa GoGo para serem executados enquanto a mesma estiver desconectada do computador. Isto permite aos usuários criar robôs autônomos, dispositivos de sensoriamento ambiental e outros objetos inteligentes sem a presença do computador.

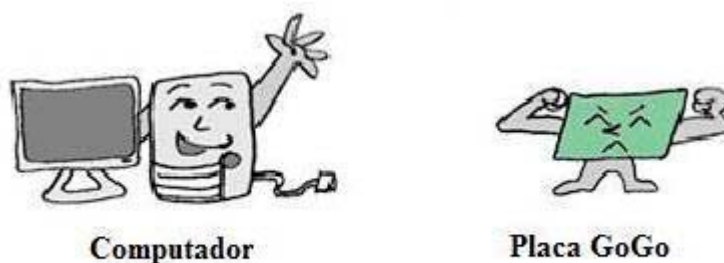


Figura 19 - Modo autônomo da placa GoGo

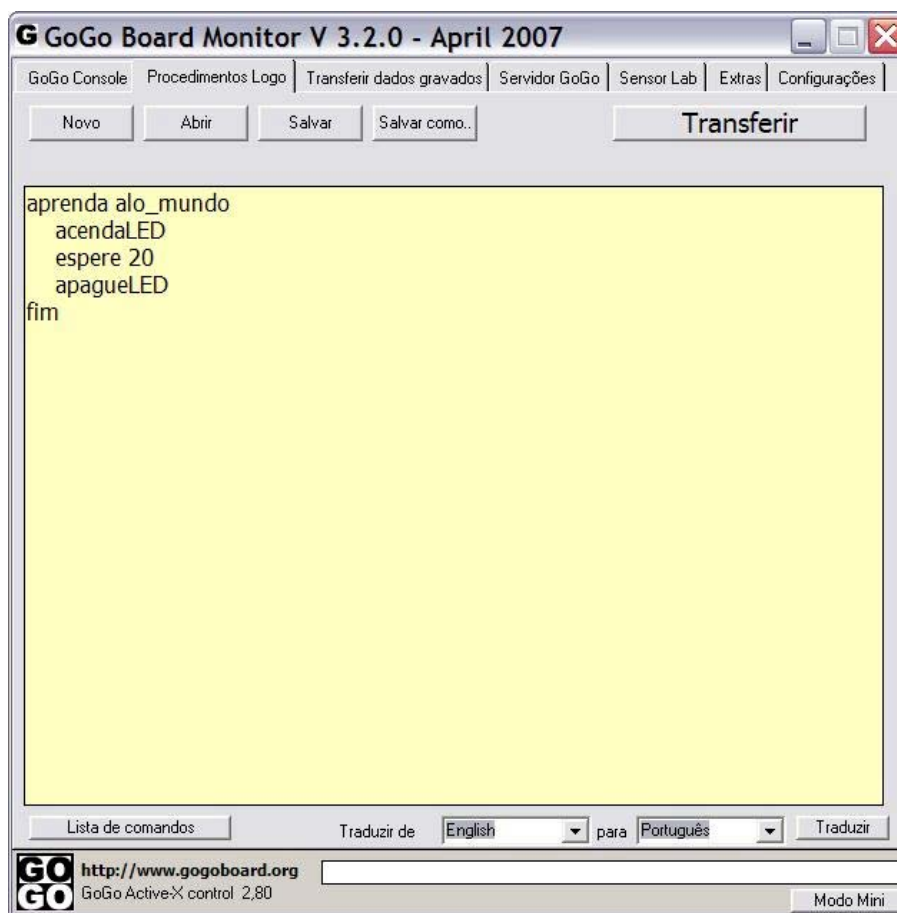


Figura 20 - Interface Procedimentos Logo

4.3.1 A Estrutura Física da Placa GoGo

Na placa, conforme Figura 21, o *chip* maior é o microcontrolador PIC16F877 de 40 pinos, conectado em um soquete, que possibilita a sua inserção e remoção com facilidade. Algumas das características técnicas deste microcontrolador são:

- Memória de programa EEPROM *Flash* com 8K palavras de 14 bits, que permite a gravação rápida do programa diversas vezes no mesmo *chip*, inclusive pelo próprio código interno;
- Memória EEPROM com 256 bytes;
- Comunicações seriais: SPI, I²C e USART;
- Conversores analógicos de 10 bits (8x) e comparadores analógicos (2x);
- Dois módulos CCP: Capture, Compare e PWM;

Logo acima do microcontrolador se encontram os dois *chips* SN754410NE que, atuando entre o microcontrolador e as saídas analógicas, fornecem a corrente elétrica, de até 1 Ampère, para o funcionamento das saídas.

O *chip* ao lado esquerdo do microcontrolador é uma memória serial EEPROM 24LC256 com capacidade de 256 K bits (32 K bytes), utilizada para armazenamento de dados. Já o *chip* do lado direito e abaixo é um SN74HC04N, que possibilita a compatibilização dos sinais das portas de comunicação do microcontrolador (0 e 5 volts) com os sinais utilizados na interface serial do computador (-12 e +12 volts). Ainda na placa, existem um regulador de voltagem 7805, localizado no lado esquerdo e acima do conector da porta serial, que garante que uma fonte externa de voltagem seja compatibilizada com os 5 volts necessários para a placa.

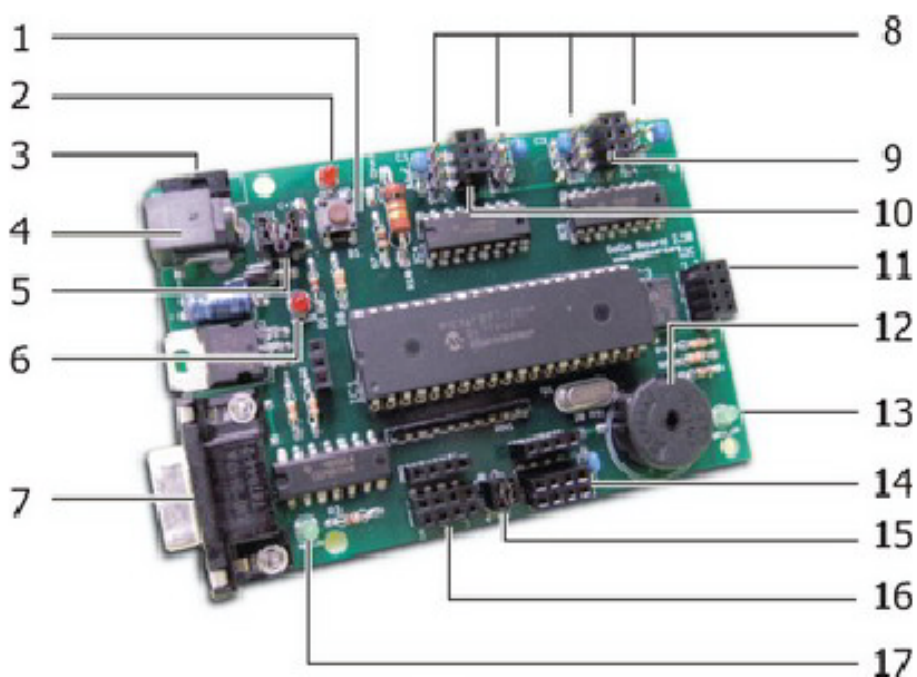



Figura 21 - Placa GoGo (GOGO, 2007a)

A Tabela 1 descreve resumidamente os componentes da placa GoGo.

	Nome	Descrição
1	Botão Iniciar/Parar	Inicia ou pára os procedimento armazenados na placa.
2	Led Execução	Quando aceso indica que o procedimento armazenado está sendo executado.
3	Chave Liga/Desliga	Chave principal que liga e desliga a placa.
4	Entrada de Energia	Esta é a entrada principal de energia da placa. Normalmente esta entrada é fornecida entre 5 a 9 Volts.
5	<i>Jumpers</i> Configuração de Energia	Estes <i>jumpers</i> permitem configurar como as portas de saídas dos motores serão energizadas.  <i>Jumper</i> na posição 1 e 3: Saída com 5 volts


		 <i>Jumper</i> na posição 1 e 2: Saída com a mesma voltagem fornecida na entrada de energia principal.
6	Led Energia	Este led sempre estará aceso quando a placa estiver ligada.
7	Porta Serial	A placa GoGo utiliza esta porta para se conectar ao computador.
8	Porta Servo Motores	Permite conectar até quatro servo motores.
9,10	Saídas A/B e C/D	Nas portas de saída podem ser conectados motores, lâmpadas e relês. Estas portas compartilham recursos com as portas do servo. Assim, portas que estão sendo usadas para servo não podem ser utilizadas para controlar um motor ao mesmo tempo.
11	Barramento de Extensão	Esta porta permite a placa GoGo se comunicar com outros dispositivos (como um <i>display</i>)
12	Buzina Piezo-elétrica	A buzina piezo-elétrica que pode ser programada para emitir um bipe.
13	Led Usuário	Um led que pode ser programado pelo usuário da placa GoGo para acender ou apagar.
14,16	Entradas de Sensores	Permite conexão de oito sensores (por exemplo, de luz, de toque, temperatura, distância, etc.).
15	<i>Jumper</i> Configuração de Sensor	Estes dois <i>jumper</i> s permitem configurar as portas de entrada 4 e 5. Se o <i>jumper</i> estiver ausente, a porta evitará um divisor de voltagem na placa. Isto permite à placa trabalhar com sensores que geram voltagem (como o sensor de proximidade Sharp).
17	Led Atividade Serial	Indica comunicação entre a placa e o computador.

Tabela 2 - Descrição da placa GoGo

A placa GoGo tem oito portas para sensores. Cada uma delas tem três pinos, como mostrado na Figura 22. A fileira superior e a inferior (Entrada do Sensor e Terra, respectivamente) são as que os sensores mais simples utilizam. A fileira do meio é um fornecimento extra de energia para sensores ativos que precisam de uma fonte de energia extra para funcionar corretamente.

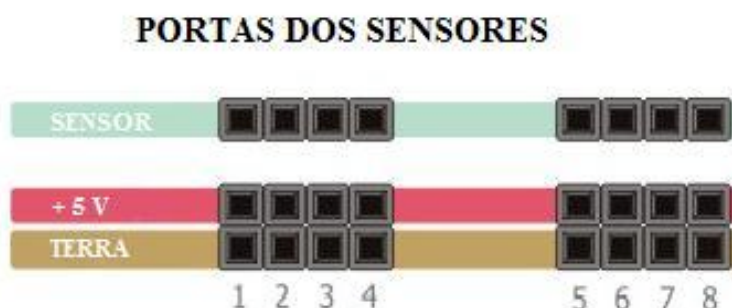


Figura 22 - Portas dos sensores

4.3.2 A Estrutura Lógica da Placa GoGo

Um modelo lógico pode ser esboçado para melhor entender como o hardware da placa GoGo trabalha. O modelo é construído através de uma estrutura de camadas, conforme a Figura 23.

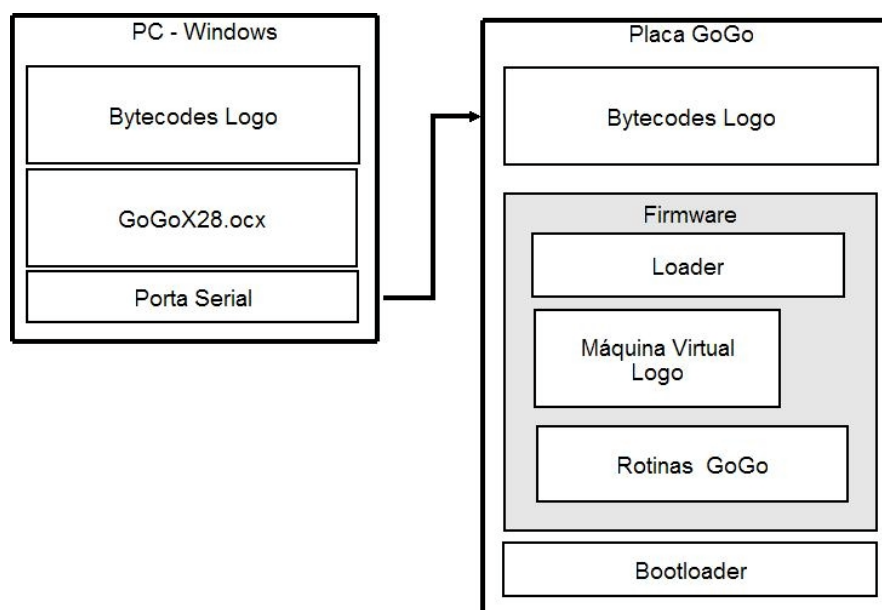


Figura 23 - Estrutura lógica da placa GoGo

O software da placa GoGo está dividido em duas partes: o GoGo Board Monitor, uma interface com o usuário e um compilador Cricket Logo, executados no computador pessoal; e o *Firmware* executado na placa GoGo. Esta técnica é chamada de *cross-compiling*, o que significa que o programa é escrito em um computador, porém este programa será executado em outro (GOGO, 2007a; GOGO, 2007b).

Através da interface “Procedimento Logo” no GoGo Board Monitor é escrito um programa na linguagem Logo Cricket, conforme a Figura 24. Ao se clicar no botão “Transferir”, o compilador traduz o procedimento Logo Cricket “alo_mundo” em uma seqüência de *bytecodes*, que em seguida são transferidos através da porta serial para a placa GoGo de forma transparente ao usuário, onde posteriormente serão executados, através de uma máquina virtual Logo que faz parte do *firmware*.

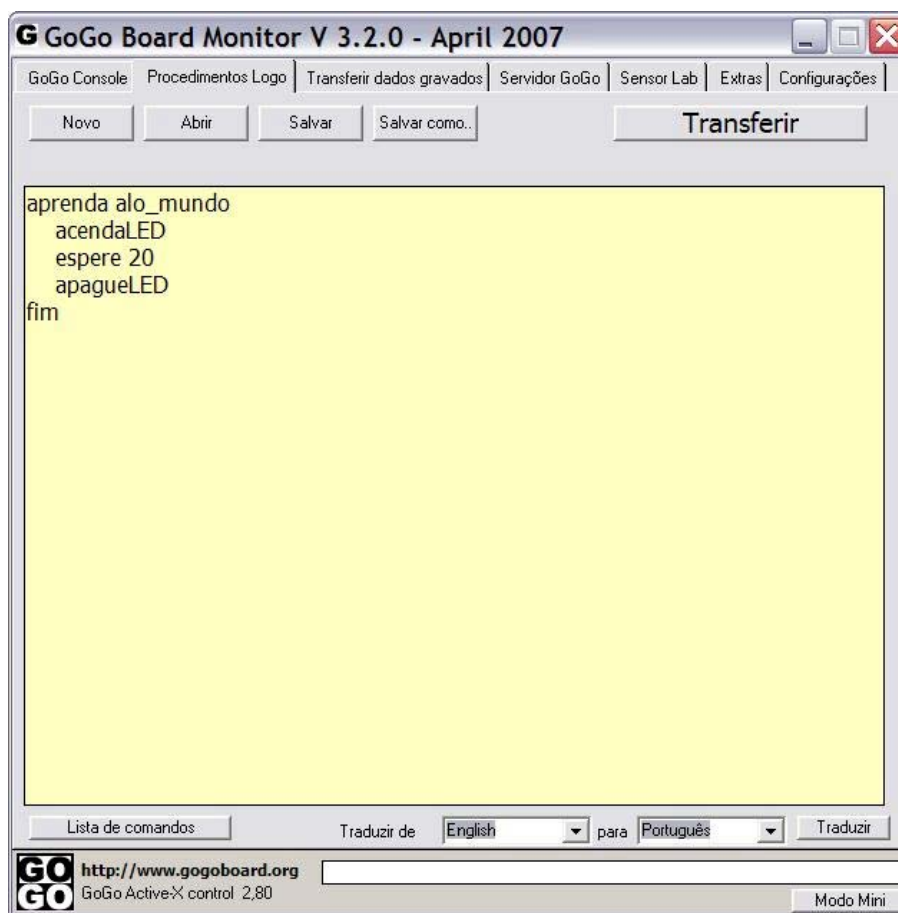


Figura 24 - Procedimento Logo Cricket

As funções primitivas no nível de usuário, desenvolvidas no Logo Cricket, são traduzidas para um, dois, ou três bytes de código de objeto para a máquina virtual do Cricket. Durante a execução, a máquina virtual, através de uma pilha, retém a computação de resultados de expressões e argumentos para procedimentos. Na linguagem de *bytecode*, a expressão “1 + 1” seria traduzida para número 1, número 1, mais. Durante a execução o primeiro número 1 causaria o valor 1 a ser empilhado na pilha. Depois com o segundo número 1, outro 1 seria empilhado na pilha. E então o operador '+' desempilharia os dois números 1 para fora da pilha e empilharia 2, como o resultado da operação.

Esta abordagem, assim como na implementação do Cricket, possibilita vários benefícios importantes, entre os quais se destacam:

- O código objeto, *bytecode*, que é o resultado da compilação do programa do usuário, é bastante pequeno e facilmente alocado em uma fatia de memória consecutiva de baixo custo do microcontrolador;
- A máquina virtual tem uma arquitetura simples, baseada em pilha, o que permite que, tanto a sua implementação na linguagem C do PIC quanto a implementação do compilador sejam escritas com uma quantidade mínima de códigos.

A memória de programa EEPROM Flash de 8K palavras é dividida em três regiões de endereçamento, o *firmware* ocupa 7k palavras, o *bootloader* ocupa 768 palavras e a região de *bytecodes* Logo ocupa apenas 256 palavras, conforme Figura 25.

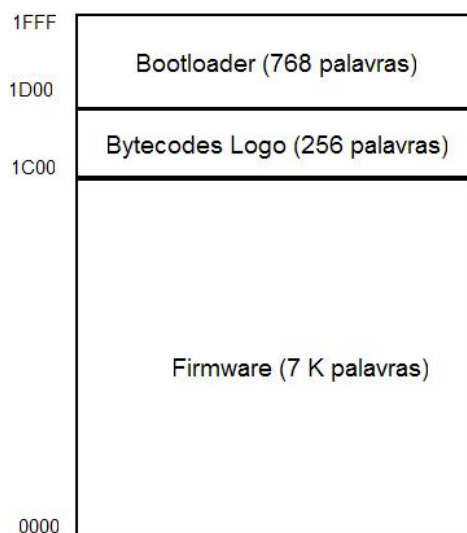


Figura 25 - Mapa da memória de programa

4.4 Módulos Adicionais

4.4.1 Módulo *Display*

O *display* possibilita a exibição de quatro dígitos, podendo ser números e ou as seguintes letras: A, b, c, d, E, F, G, h, i, J, L, n, o, P, q, r, S, t, u, y. Ele é particularmente útil para mostrar valores, por exemplo, o valor lido por um sensor quando a placa não está conectada a um computador, conforme Figura 26.

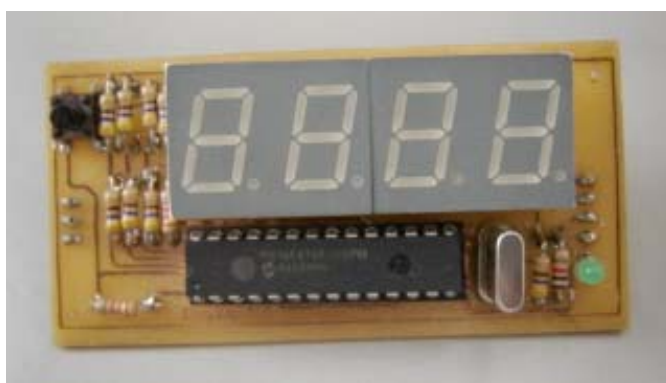


Figura 26 - Módulo display

O módulo do *display* é conectado na placa GoGo através de pinos que se encaixam na porta do barramento de expansão da placa GoGo. Uma vez que o *display* esteja conectado à placa GoGo é possível enviar uma mensagem de até quatro dígitos ou caracteres para ser exibidos no *display*, através do GoGo Monitor utilizando a interface "Extras/Display". A outra maneira de exibição de valores é através da utilização da primitiva "mostre" em um procedimento Logo Cricket, como nos exemplos abaixo:

- *mostre sensor1 ;exibe no display o valor do sensor1*
- *mostre "GoGo ;exibe a palavra "GoGo" no display*

4.4.2 Módulo Amplificador

O módulo amplificador permite a placa GoGo utilizar sensores ativos que geram voltagem alternada (sinal CA). Tais sensores podem ser um microfone para detecção de som ou um sensor de vibração utilizando um dispositivo piezoelétrico, conforme Figura 27.

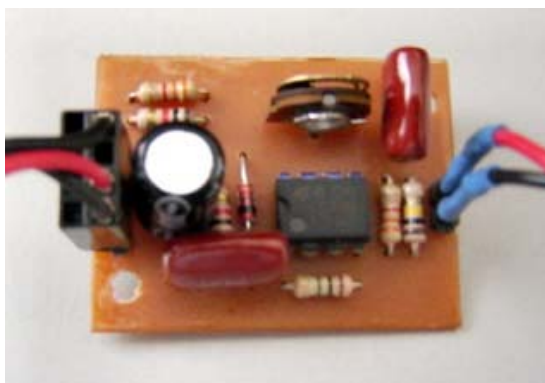


Figura 27 - Módulo amplificador

4.5 Montagem da placa GoGo

A placa GoGo foi projetada para ser construída pelo próprio usuário, e em função disso todas as informações estão disponíveis no site oficial da placa (GOGO, 2007a). O processo consiste de cinco etapas, conforme a Figura 28.

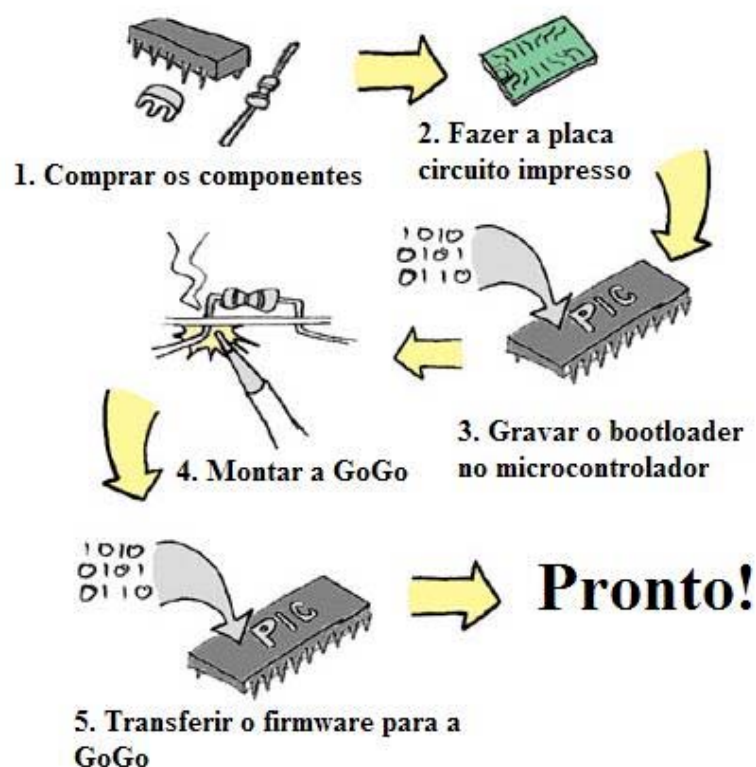


Figura 28 - Processo de montagem da placa GoGo

4.5.1 Compra dos componentes

Uma lista de componentes necessários está disponível no site da placa GoGo com a descrição e os códigos para aquisição em uma empresa nos Estados Unidos. No entanto, para facilitar aos usuários a aquisição dos componentes no Brasil, inclusive para aqueles que moram em regiões que não dispõem de lojas especializadas, foi elaborada uma lista de componentes de um fornecedor nacional, conforme Tabela 2, que efetua venda à distância e providencia o envio dos componentes pelo correio (EVATRON, 2007).

Qtd.	Descrição	Valor Unitário	Valor Parcial
1	Conector <i>Power Jack</i> Mini .08"	R\$ 0,36	R\$ 0,36
1	Rede de Resistores 33K	R\$ 1,20	R\$ 1,20
1	Cristal 20.000MHz 20pF HC-49/US	R\$ 0,80	R\$ 0,80
1	Capacitor Eletrolítico 100uF 16V radial	R\$ 0,27	R\$ 0,27
1	Regulador de Tensão 7805	R\$ 1,65	R\$ 1,65
1	Soquete para CI ST 40 POS TIN	R\$ 1,70	R\$ 1,70
1	CI 24LC256-I/P Memória EEPROM Serial 32KX8	R\$ 5,10	R\$ 5,10
1	<i>Buzzer</i> 12MM para circuito impresso 6V	R\$ 1,60	R\$ 1,60
1	Conector DB9 Fêmea para circuito impresso	R\$ 1,20	R\$ 1,20
2	LED 3mm verde	R\$ 0,10	R\$ 0,20

1	Chave Tactil 6mm SQ H=5.0mm 100GF	R\$ 0,16	R\$ 0,16
2	Soque de Barra 1X40 180°	R\$ 2,20	R\$ 4,40
5	Resistor 3.3K 1/4W 5% <i>Carbon Film</i>	R\$ 0,02	R\$ 0,10
2	CI SN754410NE HALF-H DRVR QUAD 16-DIP	R\$ 9,00	R\$ 18,00
1	Chave deslizante DPDT .2A L=4mm	R\$ 1,70	R\$ 1,70
4	Mini <i>Jumper</i> com aba	R\$ 0,09	R\$ 0,36
1	Resistor 2.0 Metal Film 2W 5%	R\$ 0,20	R\$ 0,20
1	Microcontrolador PIC16F877A-I/P	R\$ 15,00	R\$ 15,00
1	CI SN74HC04 HEX <i>Inverter</i> CMOS HS 14-DIP	R\$ 0,58	R\$ 0,58
5	Capacitor Cerâmico .1UF 50V 20% radial	R\$ 0,06	R\$ 0,30
1	Diodo retificador 4004	R\$ 0,10	R\$ 0,10
2	LED 3mm vermelho	R\$ 0,10	R\$ 0,20
3	Resistor 1.0K 1/4W 5% <i>Carbon Film</i>	R\$ 0,02	R\$ 0,06
2	Barra de pinos BSPI 2 lados 1x40	R\$ 0,45	R\$ 0,90
Valor Total		R\$	56,14

Tabela 3 - Lista de componentes placa GoGo 3.0 em março/2007

4.5.2 Fabricação da Placa de Circuito Impresso

A confecção da placa pode ser realizada pelo próprio usuário, sendo indicada para pequenas quantidades, e utiliza um conjunto de ferramentas de baixo custo disponível em lojas de componentes eletrônicos. O processo envolve a transferência do desenho do circuito para uma fina superfície de cobre em uma placa de material dielétrico (isolante), normalmente fenolite ou fibra de vidro. As placas GoGo, utilizadas neste trabalho, foram fabricadas utilizando a técnica de transferência térmica, sendo que este processo consiste em imprimir o desenho do circuito em uma folha de papel *glossy*, utilizando uma impressora laser, conforme a Figura 29.

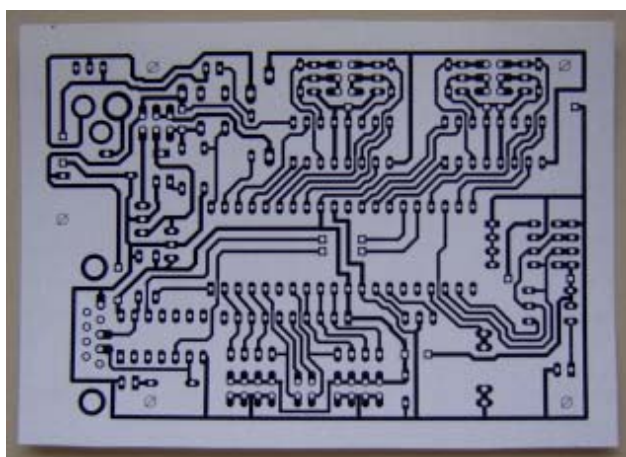


Figura 29 - Impressão do circuito da placa GoGo

Este papel é utilizado para impressão de fotos em impressoras jato de tinta e possui, na face de impressão, uma fina película plástica, que acaba aderindo na superfície de cobre da placa no processo de transferência térmica com um ferro de passar roupas, conforme a Figura 30. Após a placa esfriar, ela é colocada em uma vasilha com água e detergente para retirada do papel *glossy*, deixando apenas o *toner* da impressão na placa, conforme a Figura 31.



Figura 30 - Processo térmico de transferência

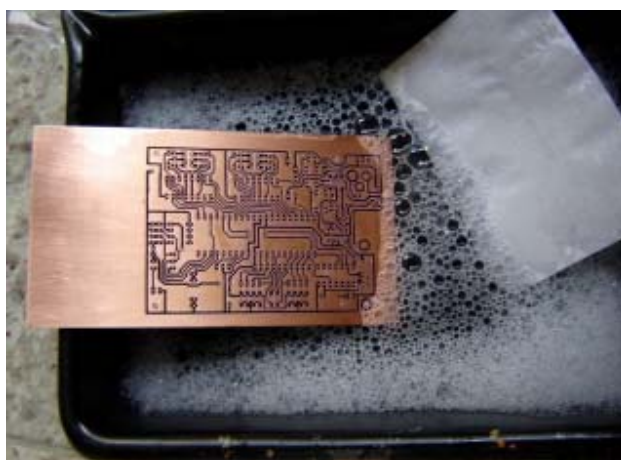


Figura 31 - Placa após remoção do papel *glossy*

A placa contendo, em sua superfície de cobre, o desenho do circuito é então imersa em uma solução corrosiva que retira todo o cobre exposto, deixando somente o cobre protegido pelo desenho do circuito, conforme Figura 32. A fase final consiste na limpeza da placa e perfuração dos pontos onde serão inseridos os componentes para soldagem. A placa, após estes passos, é mostrada na Figura 33.

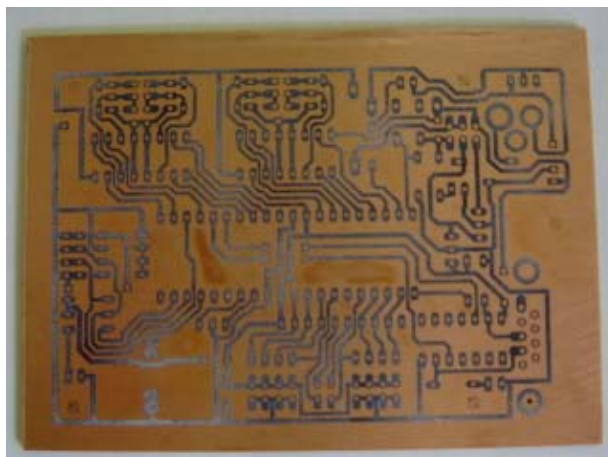


Figura 32 - Placa após o processo de corrosão

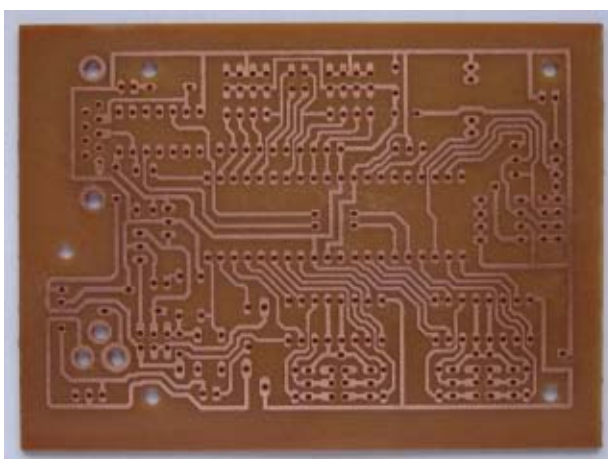


Figura 33 - Placa após limpeza e perfuração

Para grandes quantidades de placas são indicados a confecção em uma empresa especializada, sendo necessário o envio dos arquivos técnicos do projeto da placa, chamados de arquivos *Gerber*⁶.

4.5.3 Gravação do *BootLoader* no Microcontrolador

Para a placa GoGo receber o *firmware*, através da comunicação serial com o computador, primeiro é necessário que seja gravado no microcontrolador um programa chamado *BootLoader*. O propósito do *BootLoader* é, sendo gravado apenas uma vez no PIC, permitir a reprogramação do PIC tão freqüentemente quanto necessário utilizando somente a porta serial.

⁶ É um formato padrão universal de arquivo eletrônico composto de uma combinação de comandos gráficos utilizados pelo equipamento fotoploter para a formação das imagens da placa de circuito impresso.

Após gravado no microcontrolador PIC, ele usa uma pequena parte dos 8K palavras da memória do PIC, conforme a Figura 34.

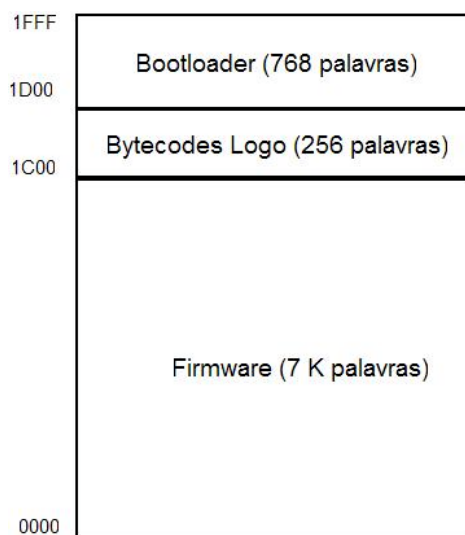


Figura 34 - Mapa da memória de programa

Para a gravação do *BootLoader* nas placas GoGo usadas neste trabalho foi utilizado o gravador *McFlash PIC*, da empresa Labtools (LABTOOLS, 2007). No entanto, para reduzir os custos da placa, pode ser utilizado um gravador de microcontrolador PIC, construído pelo próprio usuário. Uma opção, é o gravador PLMS OziPIC'er (OZIPICER, 2007) que, por utilizar a porta paralela para se comunicar com o computador, tem um projeto simples e de baixo custo. Este gravador utiliza o software de gravação WinPicProg (WINPICPROG, 2007).

4.5.4 Montagem dos Componentes na Placa de Circuito Impresso

A montagem significa soldar os componentes na placa de circuito impresso. Para esta tarefa são necessárias ferramentas básicas para solda, como um ferro solda, solda e alicates. No site do projeto da placa GoGo (GOGO, 2007a) um documento, chamado Guia de Montagem, recomenda uma seqüência de três passos a serem seguidos, conforme Figuras 35, 36 e 37. Seguindo esta recomendação, é possível testar cada parte da placa montada em cada passo. Isto torna mais fácil a depuração quando algo der errado.

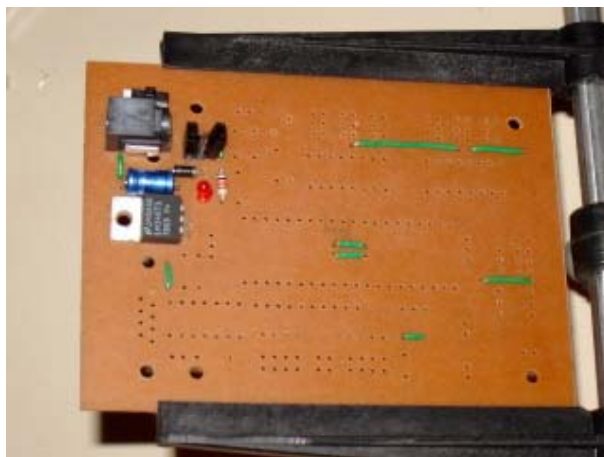


Figura 35 - Circuito de fornecimento de energia

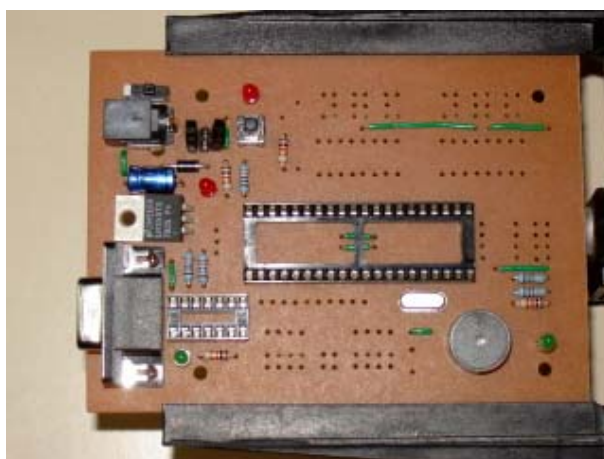


Figura 36 - Circuito da porta serial

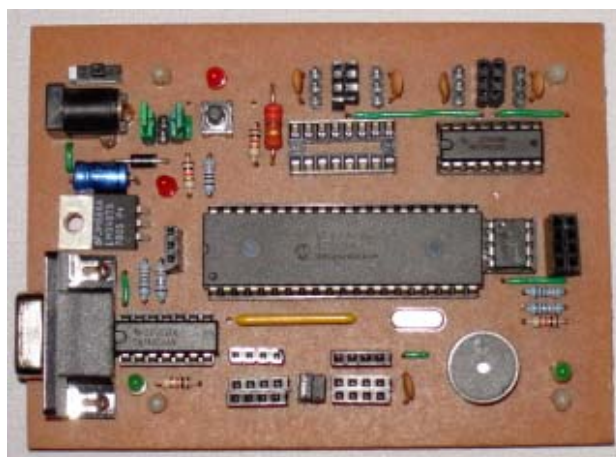


Figura 37 - Circuito completo

4.5.5 Transferência do Firmware para a Placa GoGo

O *firmware* contém as funcionalidades principais da placa GoGo. Ele é carregado

para a placa nesta fase de montagem e quando for disponibilizada uma nova versão, para corrigir algum problema da versão corrente, ou para agregar uma nova funcionalidade a ela.

Para o carregamento do *firmware* na placa GoGo, ela deverá estar conectada a um computador com o aplicativo *FirmwareLoader* instalado nele. Este aplicativo, arquivo do *firmware* propriamente dito e as informações sobre este procedimento estão disponíveis no site da GoGo.

Uma vez que o *firmware* foi carregado corretamente, a placa deverá estar totalmente funcional, podendo então ser testada utilizando o GoGo Board Monitor. Para o modo conectado, a placa poderá ser testada através das funcionalidades da interface GoGo Console, já para o modo autônomo, utilizando um procedimento Logo Cricket, através da interface Procedimentos Logo.

5 Protótipo de um Robô Móvel de Baixo Custo

5.1 Introdução

A placa GoGo, assim como o RCX do conjunto de robótica LEGO Mindstorms, é apenas a parte computacional da implementação de um robô móvel, necessitando, ainda, de sensores, de motores e de uma estrutura mecânica. O conjunto de robótica LEGO Mindstorms possui dois sensores de toque, um sensor de luminosidade, dois motores, e mais de 700 peças LEGO, como vigas, eixos, engrenagens, polias e rodas, utilizadas na montagem das estruturas mecânicas dos robôs.

Já para a utilização da placa GoGo na montagem de um robô móvel é necessária a compra de componentes de baixo custo e a montagem dos sensores, dos motores e da estrutura mecânica do robô. Uma possibilidade interessante para a estrutura mecânica é a utilização de sucata eletrônica, devido à redução de custos, e ainda, por reutilizar um material que seria descartado e que possivelmente iria para o lixo.

5.2 Sensores

No site da placa GoGo, existem informações para a montagem de vários tipos de sensores e atuadores, utilizando componentes eletrônicos de baixo custo encontrados em lojas especializadas. Para a construção do protótipo do robô móvel, aqui apresentado, foram montados quatro sensores de toque e um sensor de luminosidade.

Os sensores de toque são considerados passivos porque eles não necessitam de tensão elétrica para funcionar e são ligados na placa GoGo através somente de dois pinos, conforme a Figura 38. Já o sensor de luminosidade refletiva necessita de uma tensão de 5 volts para o seu funcionamento, e a sua conexão com a placa se dá através de três pinos, conforme a Figura 39.

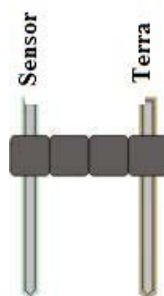


Figura 38 – Cone Conector para sensor passivo

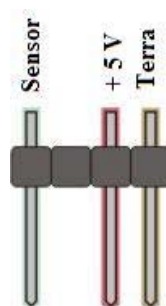


Figura 39 - Conector para sensor ativo

5.2.1 Sensor de Toque

O sensor de toque é um sensor passivo muito útil e simples, e pode ser construído com uma micro chave (*lever microswitch*) ou uma chave táctil (*push button*), e seu funcionamento se dá através, no seu interior, de dois objetos condutivos que se tocam quando o sensor é externamente pressionado, conforme Figura 40.



Figura 40 - Sensores de toque utilizando micro chaves

5.2.2 Sensor de Luminosidade Refletivo

O sensor de luminosidade refletivo é um sensor ativo e necessita de dois componentes para o seu funcionamento: um LED (diodo emissor de luz) que irradia uma quantidade de luz, que por sua vez, é refletida no objeto e então percebida pelo fotoresistor (LDR), sendo que este tipo de sensor é útil quando se deseja descobrir a presença de um objeto sem tocá-lo, ou para "enxergar" sua cor, através da quantidade de luz refletida por ele, conforme Figura 41. O circuito do sensor de luminosidade refletivo foi acondicionado dentro de um invólucro construído através da dobradura de uma parte de uma embalagem cartonada de longa vida,

conforme Figura 42.

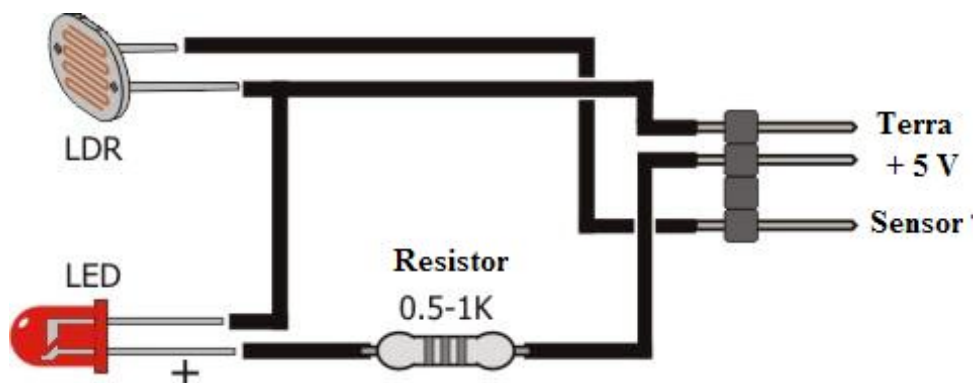


Figura 41 - Circuito do sensor de luminosidade ativo



Figura 42 - Invólucro do sensor de luminosidade ativo

5.3 Motores

Os dois motores usados na construção do protótipo utilizam uma tensão de trabalho de 13V, conforme Figura 43, e foram retirados do mecanismo que movimenta a bandeja de duas unidades leitoras de CD inutilizadas (sucata eletrônica). No entanto, estes motores utilizados no mecanismo da bandeja da unidade de CD podem ser encontrados com uma faixa de tensão de trabalho entre 5,9V e 13V, dependendo do modelo e marca da unidade, conforme Figura 44.



Figura 43 - Motor 13V



Figura 44 - Motores 5,9 a 13V

5.4 Estrutura Mecânica do Protótipo

A estrutura de sustentação e movimento do robô autônomo desenvolvido neste trabalho utiliza os conceitos de um veículo triciclo, duas rodas traseiras e uma dianteira, sendo que a tração e a direção são dadas pelas rodas traseiras e a roda dianteira é utilizada apenas para apoio, conforme a Figura 45.

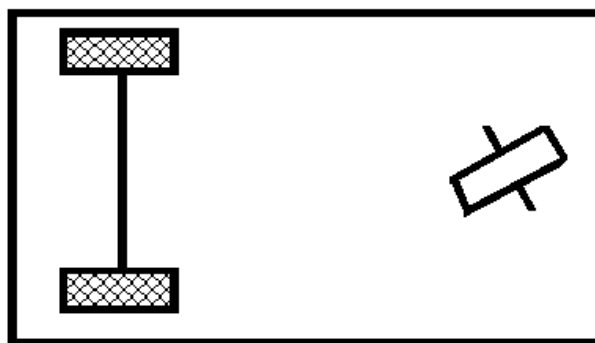


Figura 45 - Veículo triciclo com tração traseira

5.4.1 Rodas de Tração e Apoio

Para as rodas traseiras do robô, foram utilizados dois mini CDs de 8 cm, conforme Figura 46, e a para a roda dianteira foi usada uma roda giratória utilizada na fabricação de móveis, mostrada na Figura 47. A escolha do mini CD, não tão comum quanto o CD convencional de 12 cm, foi motivada pelas dimensões da placa GoGo de 11cm por 8cm, pois o CD normal com raio de 6 cm deixaria o robô com uma altura elevada do solo, o que poderia atrapalhar a sua estabilidade e estética. Ainda para aumentar a aderência das rodas do robô, os mini CDs foram envoltos com uma tira de borracha de câmara de motocicleta, conforme Figura 48.



Figura 46 - Mini CD 8 cm

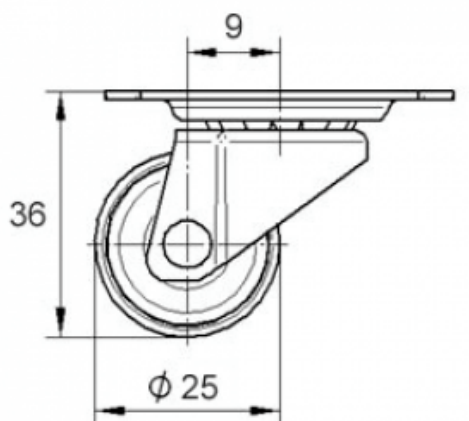


Figura 47 - Roda giratória



Figura 48 - Rodas e eixo do robô

5.4.2 O Chassi do Protótipo

O chassi foi construído com uma placa de fenolite com as mesmas dimensões da placa GoGo e com ranhuras e furos para fixação dos demais componentes que compõem o robô, conforme Figura 49. A fixação dos elementos ao chassi foi realizada através de parafusos com porcas e cintas plásticas.

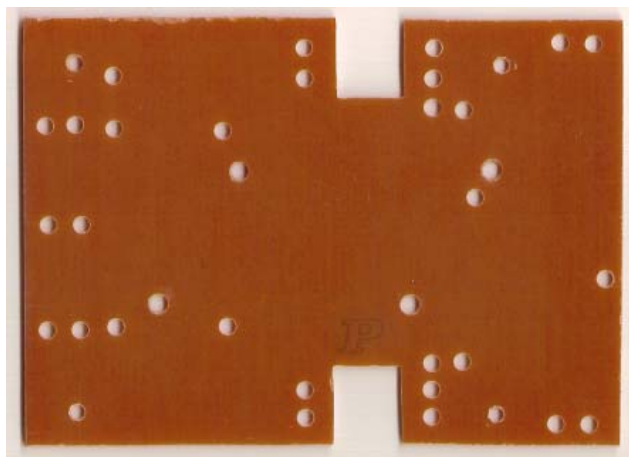


Figura 49 - Chassi do robô

5.4.3 Engrenagens e Eixos

As engrenagens utilizadas na redução da rotação e no aumento do torque, para a construção do robô, conforme Figuras 50 e 51, foram retiradas do mecanismo de movimentação da bandeja de duas unidades leitoras de CD inutilizadas (sucata eletrônica). Estas engrenagens podem variar de número e tamanho, dependendo do mecanismo utilizado nos diversos modelos de unidades leitoras de CD disponíveis, sendo que o mecanismo mais comum encontrado, utiliza uma combinação de engrenagens, polias e correias.



Figura 50 - Conjunto de engrenagens e eixo



Figura 51 - Engrenagens fixadas nos eixos

5.5 A Montagem do Protótipo

A montagem do protótipo é descrita através de seis passos, partindo da fixação dos eixos e rodas ao chassi e terminando com a união do protótipo do robô com a placa GoGo e o módulo display de expansão, conforme mostrado nas Figuras 52 a 57.



Figura 52 - Montagem das rodas

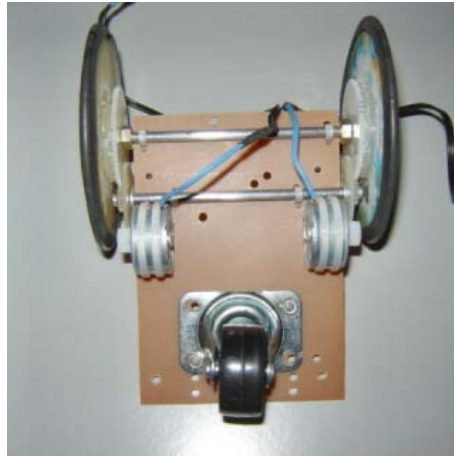


Figura 53 - Montagem dos motores

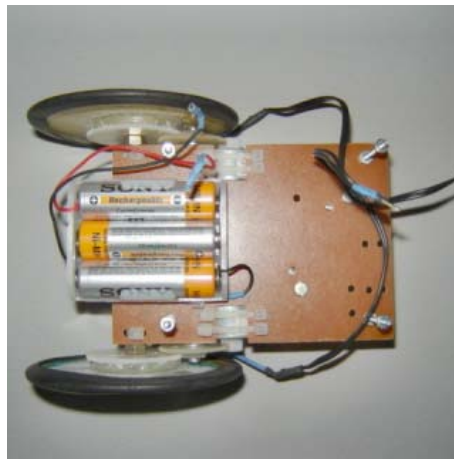


Figura 54 - Montagem do suporte de pilhas

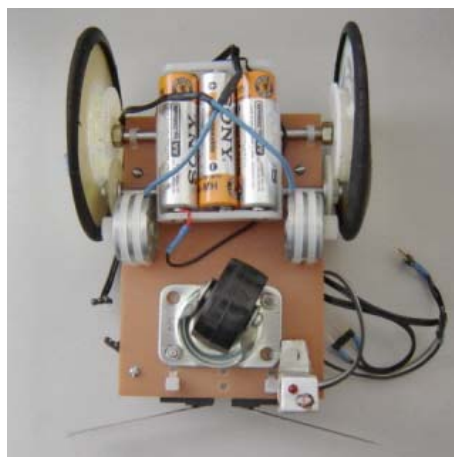


Figura 55 - Montagem dos sensores

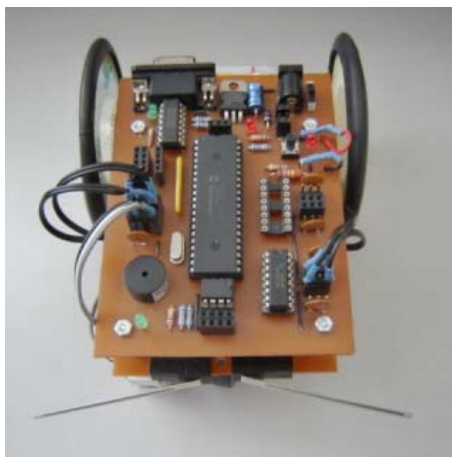


Figura 56 - Montagem da placa GoGo no chassi



Figura 57 - Protótipo montado com o *display*

5.6 Programação do Protótipo utilizando o Logo Cricket

Utilizando a linguagem Logo e partindo de um programa simples são incorporadas novas funcionalidades para tornar o comportamento do robô mais interessante. Com isso, pretende-se ter apresentado um programa com todas as funcionalidades necessárias para ser utilizado na programação do robô coletor de latas implementado com a placa GoGo, conforme descrito no Desafio Enlatado (LEGO, 1999a), que será apresentado no próximo capítulo.

5.6.1 Um Programa para Movimentar o Robô

Programas escritos na linguagem Logo Cricket consistem em procedimentos. Este programa tem somente um procedimento, chamado `exemplo1`. O procedimento `exemplo1`,

conforme Listagem 1, configura a direção do movimento dos motores conectados nas portas de saída *c* e *d* da placa GoGo, sendo que em seguida os motores são ligados e após quatro segundos a direção dos mesmos é mudada e permanece revertida por mais quatro segundos, quando finalmente os dois motores são desligados.

```
aprenda exemplo1
```

```
cd,paralá
```

```
c,ligue
```

```
d,ligue
```

```
espere 40
```

```
cd,md
```

```
espere 40
```

```
cd,desligue
```

```
fim
```

Listagem 1 - Procedimento para acionamento dos motores do robô

O programa apresentado move o robô para frente durante quatro segundos e depois muda a direção do movimento dos motores, fazendo o robô retornar ao ponto de origem. Aos próximos exemplos, são acrescentados novos comandos da linguagem Logo Cricket, permitindo ao robô apresentar um comportamento mais interessante.

5.6.2 Mudar a Direção do Robô

O procedimento exemplo2 permite que o robô possa fazer voltas invertendo a direção de um dos dois motores. O procedimento exemplo2, conforme Listagem 2, configura, através de variáveis, o tempo do movimento e o tempo do giro do robô, em seguida a direção do movimento dos motores conectados nas portas de saída *c* e *d* da placa GoGo, sendo estes, na seqüência, ligados e após, de acordo com o tempo definido na variável *tempo_movimento*, o robô movimenta-se para frente, depois o motor *d* muda de direção, fazendo com que o robô vire para a direita no tempo definido na variável *tempo_giro*, alterando a direção do robô em aproximadamente 90 graus, quando finalmente os dois motores são desligados.

```
aprenda exemplo2
```

```
faça "tempo_movimento 20
```

```
faça "tempo_giro 30
```

```
cd,paralá
```

```

cd,ligue
espere :tempo_movimento
d,md
espere :tempo_giro
cd,desligue
fim

```

Listagem 2 - Procedimento para mudança de direção do robô

5.6.3 Repetindo Comandos

O procedimento exemplo3, conforme Listagem 3, faz o robô mover-se em um quadrado. Mover-se em quadrados significa: mover-se adiante por um determinado tempo, virando 90 graus, mover-se adiante novamente, virando 90 graus, etc. Pode-se simplesmente repetir o procedimento exemplo2 por quatro vezes, mas isto pode ser feito de maneira mais fácil com o comando repita.

O número na instrução repita 4, indica com que frequência algo deve ser repetido. São postas as instruções que devem ser repetidas entre colchetes.

```

aprenda exemplo3
faça "tempo_movimento 20
faça "tempo_giro 25
repita 4
[
cd,paralá
cd,ligue
espere :tempo_movimento
d,md
espere :tempo_giro
]
cd,desligue
fim

```

Listagem 3 - Procedimento para o robô mover-se 'em quadrado'

No procedimento exemplo4, conforme Listagem 4, existe uma instrução repita dentro de outra. Esta instrução repita está "aninhada". A primeira instrução repita inicia no primeiro colchete e termina no último, já a segunda instrução repita aninhada, inicia no segundo colchete e termina no terceiro.

```

aprenda exemplo4
  faça "tempo_movimento 20
  faça "tempo_giro 25
  repita 2
  [
    repita 4
    [
      cd,paralá
      cd,ligue
      espere :tempo_movimento
      d,md
      espere :tempo_giro
    ]
  ]
  cd,desligue
fim

```

Listagem 4 - Procedimento para o robô mover-se 'em quadrado' duas vezes

5.6.4 Movendo em Espiral

No procedimento exemplo5, conforme Listagem 5, o procedimento exemplo3 é adaptado de tal modo que o robô mova-se em uma espiral. Isto pode ser alcançado fazendo o tempo de espera seja maior para cada próximo movimento reto. Quer dizer, aumentar o valor da variável tempo_movimento a cada repetição.

```

aprenda exemplo5
  faça "tempo_movimento 20
  faça "tempo_giro 25
  repita 40
  [
    cd,paralá
    cd,ligue
    espere :tempo_movimento
    d,md
    espere :tempo_giro
    faça "tempo_movimento :tempo_movimento + 10

```

```

]
  cd,desligue
fim

```

Listagem 5 - Procedimento para o robô mover-se em espiral

5.6.5 A Instrução *se*

O procedimento exemplo6, conforme Listagem 6, permite que uma parte particular do programa só seja executada em certas situações. Neste caso a instrução *se* é usada. Por exemplo, é desejado que o robô dirija ao longo de uma linha reta e então faça uma volta para a esquerda ou para a direita, alternadamente. Se a variável *sentido* for igual a zero, será feita uma volta para a esquerda, se for igual a um a volta será para a direita.

```

aprenda exemplo6
  faça "tempo_move 20
  faça "tempo_gira 25
  faça "sentido 0
  repita 10
  [
    cd, fixepotência 5
    cd, parará
    cd, ligue
    espere :tempo_move
    mostre :tempo_gira
    espere 4
    se :sentido = 0
    [
      c, paracá
      mostre "esqu
      faça "sentido 1
    ]
    se :sentido = 1
    [
      d, paracá
      mostre "dire
      faça "sentido 0
    ]
  ]

```

```

        espere :tempo_gira
    ]
    cd, desligue
fim

```

Listagem 6 - Procedimento para o robô mover-se para esquerda e direita

5.6.6 Agindo através de um Sensor de Toque

No procedimento exemplo7, conforme Listagem 7, é apresentado um programa muito simples, no qual o robô move-se adiante até que bata em algo e sejam desligados os motores.

```

aprenda exemplo7
    cd, fixepotência 4
    cd, parala
    cd, ligue
    parasempre
    [
        se ( (sensor1 < 500) ou (sensor2 < 500) )
            [
                cd, desligue
            ]
        ]
    ]
fim

```

Listagem 7 - Procedimento para utilização dos sensores de toque

5.6.7 Sensor de Luminosidade

O sensor de luminosidade mede a quantidade de luz emitida em uma direção particular. O sensor de luminosidade também emite luz, deste modo é possível apontar o sensor de luminosidade em uma direção particular e fazer uma distinção entre a intensidade de reflexão do objeto naquela direção. Isto é em particular útil quando se está tentando fazer um robô seguir uma linha no chão, conforme Figura 58.

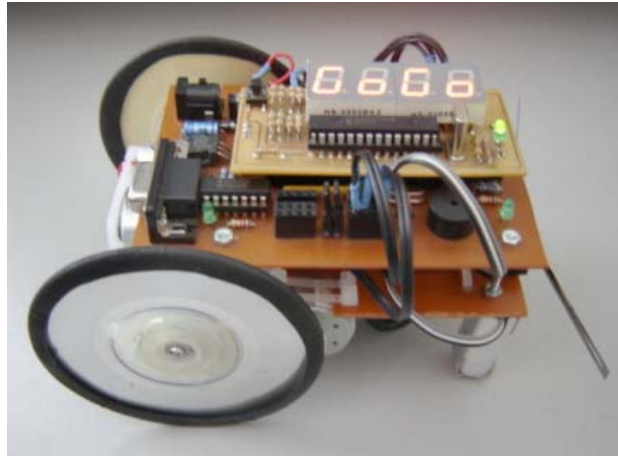


Figura 58 - Robô com sensor de luminosidade

Também é necessário um campo de provas, construído em uma superfície clara com um círculo marcado com uma fita isolante preta.

A idéia agora é que o robô mova-se, aleatoriamente, dentro do círculo e quando a intensidade da luz refletida diminuir, o sensor de luminosidade terá encontrado a linha preta e deverá mudar de direção para o robô manter-se dentro do círculo.

O procedimento exemplo8, conforme Listagem 8, permite que sempre que o valor de luminosidade seja maior ou igual a 400, indicando que o robô encontrou a linha preta, este reverte os motores por um determinado tempo, indo para trás, gira e move-se novamente para frente até encontrar novamente a linha preta.

aprenda exemplo8

```

faça "limiar 400
cd, fixepotência 6
espere 1
cd, parala
cd, ligue
parasempre
[
  mostre sensor7
  se sensor7 > :limiar
  [
    cd, freie
    cd, paraca
    cd, liguepor 8
    cd, desligue
    c, parala
  ]
]

```

d, paracá
cd, liguepor 4
cd, paralá
cd, ligue

]]

]]

fim

Listagem 8 - Procedimento para o robô retornar quando encontrar uma linha preta

6 Experimentos e Resultados

6.1 Introdução

Este exemplo mostra que o conjunto proposto neste trabalho é capaz de desenvolver as funcionalidades apresentadas no conjunto de robótica LEGO Mindstorms. O Desafio Enlatado envolve a construção de um robô com a capacidade de movimento autônomo, que tem como objetivo empurrar latas, percebidas pelos sensores de toque, para fora de um círculo, e manter-se dentro dos limites deste círculo com o auxílio do sensor de luminosidade reflexivo.

O Desafio Enlatado é um treinamento para os desafios robóticos, descrito nas Notas do Professor do Conjunto LEGO Mindstorms para Escolas 9790 (LEGO, 1999a). Ele consiste na montagem e na programação de um robô que pode retirar a maioria das latas de um campo específico, conforme Figura 59, e segue as seguintes regras:

- a) Os robôs iniciam o desafio dentro do círculo limitante;
- b) Os robôs podem empurrar, levantar e rolar as latas para fora do círculo limitante;
- c) Uma lata é contada como “fora” no momento em que está completamente fora do círculo limitante;
- d) Os robôs somente poderão usar a área próxima ao lado de fora do círculo limitante para girar;
- e) Cada rodada dura exatamente 2 minutos. Após o final da rodada, o robô deve parar e todas as latas do lado de fora do círculo limitante serão contadas. Este será o placar da rodada;
- f) Através de acordo, a competição do desafio poderá consistir de 1 a 3 rodadas, sendo feita a média dos placares.

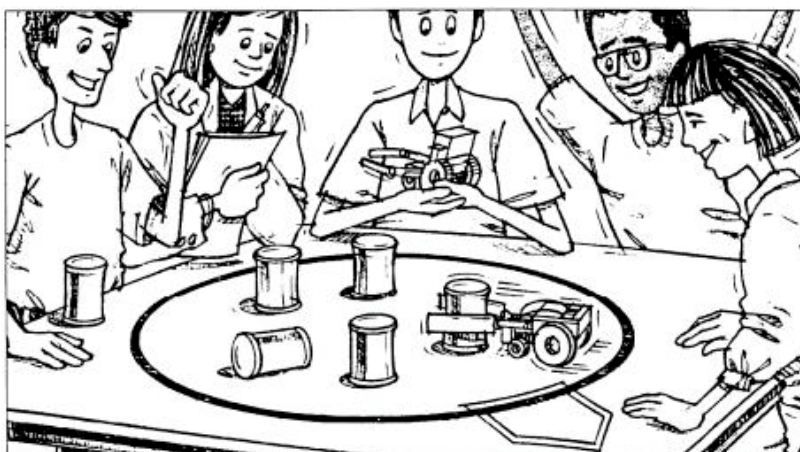


Figura 59 - O desafio Enlatado

A princípio, o robô utilizado para empurrar as latas do campo é uma versão simples do robô descrito nas páginas 4-6, do guia de montagens que acompanha o conjunto LEGO Mindstorms para Escolas (LEGO, 1999b). Este robô possui um chassi de duas rodas, movimentadas por dois motores, conforme Figura 60.

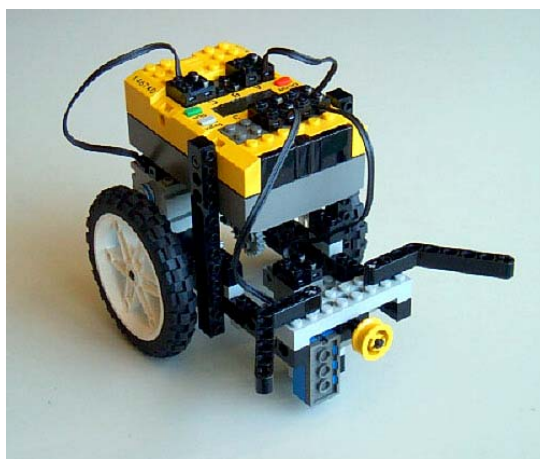


Figura 60 - Robô móvel LEGO Mindstorms

Neste trabalho, utilizamos o protótipo do robô, que foi construído com a utilização da placa GoGo, com as mesmas características e funcionalidades do robô construído com o conjunto LEGO Mindstorms, conforme Figuras 61 e 62.



Figura 61 - Robô móvel com a placa GoGo

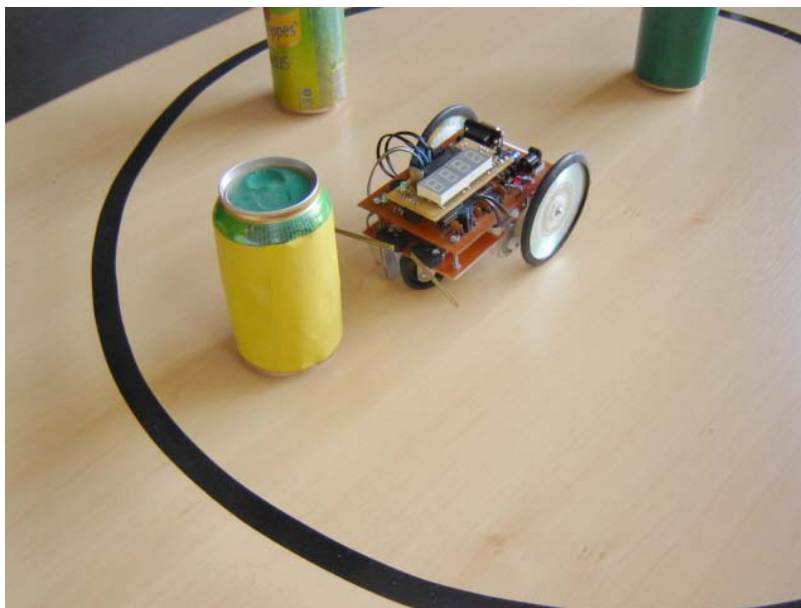


Figura 62 - Robô coletor de latas

6.2 Programando o Robô Coletor

O “Desafio Enlatado” descrito nas Notas do Professor do Conjunto LEGO Mindstorms para Escolas 9790 (LEGO, 1999a) pode ser implementado utilizando o recurso de programação através de múltiplas tarefas que podem ser executadas concorrentemente no RCX (MUDEN, 2000), tornando o controle do robô mais simples, conforme a Figura 63. Porém, a linguagem Logo Cricket não possui este recurso, sendo que todas as instruções devem estar em um procedimento que é executado de maneira seqüencial, ou seja, uma instrução após outra, conforme Figura 64.

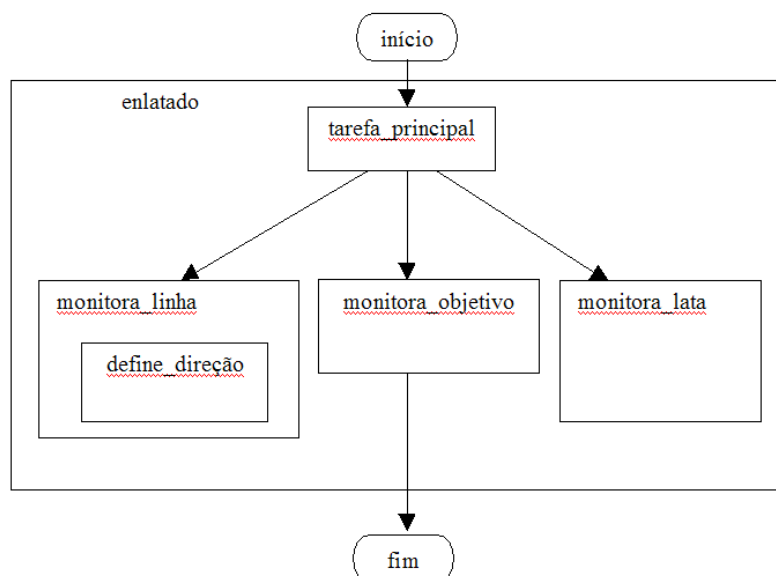


Figura 63 - Estrutura do programa utilizando o NQC - RCX

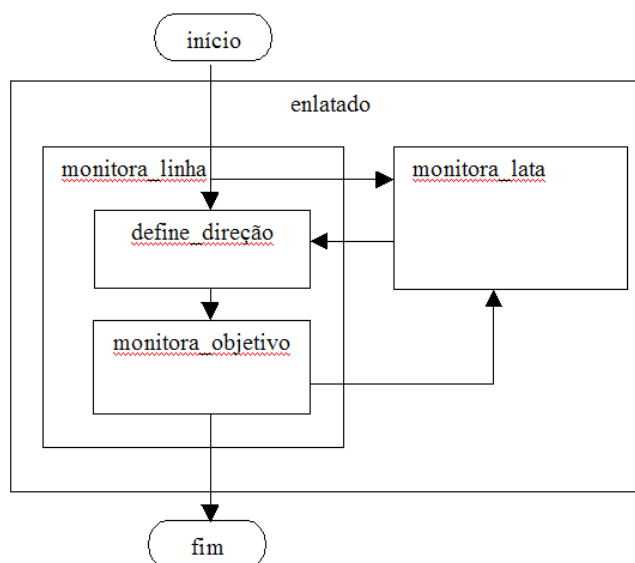


Figura 64 - Estrutura do programa utilizando o Logo Cricket - GoGo

Utilizando as idéias da programação do robô apresentadas no capítulo 5, foi criado um conjunto de procedimentos para definir o comportamento para o Robô Coletor.

6.2.1 Procedimento “Enlatado”

Conforme o programa adiante apresentado, o procedimento *enlatado*, descrito pela Listagem 9, contém os procedimentos:

- a) *monitora_linha* - Neste procedimento, conforme Listagem 10, através do *sensor7*, o robô quando encontra a linha preta, pára e retorna para o campo. Dentro deste procedimento, há ainda, outros dois procedimentos:
 - a.1) *define_direção* – Este procedimento, conforme Listagem 11, define a direção que o robô irá virar após ter encontrado a linha preta.
 - a.2) *monitora_objetivo* – Este procedimento, conforme Listagem 12, controla a quantidade de latas já encontradas pelo robô, e quando a segunda lata é encontrada o procedimento *enlatado* é encerrado
- b) *monitora_lata* - Neste procedimento, conforme Listagem 13, através do *sensor1* e *sensor2*, o robô quando encontra uma lata, incrementa a variável *contador* e empurra a lata para fora do campo.

```

aprenda enlatado
  faça "limiar 400
  faça "direção 0
  faça "lata 0
  faça "linha 0
  faça "contador 0
  ab, fixepotência 6
  espere 1
  cd, parará
  cd, ligue
  parasempre
  [
    monitora_linha
    espere 1
    monitora_lata
  ]
fim

```

Listagem 9 - Procedimento principal para o desafio Enlatado

```

aprenda monitora_linha
  mostre sensor7
  se sensor7 > :limiar
  [
    cd, freie
    faça "linha 1
    cd, fixepotência 6
    espere 1
    cd, parará
    cd, liguepor 8
    cd, desligue
    defina_sentido
    cd, liguepor 4
    cd, parará
    faça "linha 0
    monitora_objetivo
    faça "lata 0
    cd, ligue
  ]

```



```

]
fim

```

Listagem 10 - Procedimento que monitora linha através do sensor7

```

aprenda define_direção
  se :senão :direção = 0
  [
    c, parala
    d, paraca
    faça "direção 1
  ]
  [
    c, paraca
    d, parala
    faça "direção 0
  ]
fim

```

Listagem 11 - Procedimento que define a direção que o robô irá virar

```

aprenda monitora_objetivo
  se :contador = 2
  [
    cd, desligue
    pare
  ]
fim

```

Listagem 12 - Procedimento que controla a quantidade de latas já encontradas

```

aprenda monitora_lata
  se ( ( (sensor1 < 500) ou (sensor2 < 500) ) e (:lata = 0) )
  [
    bipe
    faça "lata 1
    cd, fixepotência 7
    faça "contador :contador + 1
  ]
fim

```

Listagem 13 - Procedimento que monitora o robô quando encontra uma lata

7 Conclusão

A construção de um robô móvel utilizando a placa GoGo e componentes de baixo custo ou reaproveitados de sucata eletrônica, pode contribuir para o estudo e aplicação de ferramentas pedagógicas acessíveis à realidade das escolas brasileiras, podendo tais ferramentas serem utilizadas em situações de ensino-aprendizagem através da Robótica Educacional.

Embora neste trabalho apenas tenha sido desenvolvida uma única aplicação prática, o Desafio Enlatado, a placa GoGo pode também ser utilizada em outras atividades pedagógicas como investigações científicas e automação de tarefas.

Além disso, devido ao compromisso do projeto da placa GoGo com o “mais simples e mais compreensível”, como a utilização de componentes não miniaturizados e uma placa de circuito impresso de fácil construção, a compreensão de conceitos computacionais se torna mais acessível aos alunos do que com a utilização do bloco programável RCX do Conjunto LEGO Mindstorms, que se apresenta como uma “caixa preta”, escondendo os detalhes constitutivos e computacionais.

Durante a montagem da placa GoGo e do robô móvel constatou-se a necessidade de conhecimentos técnicos específicos e a ausência de informações na língua portuguesa para auxílio de usuários iniciantes. Nesse aspecto, a sugestão para o complemento deste trabalho é o desenvolvimento de materiais didáticos, treinamento e suporte, em língua portuguesa, para os usuários da placa GoGo.

Referências

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 WIE - Workshop sobre Informática na Escola, 2005, São Leopoldo. Anais do XI WIE - Workshop sobre Informática na Escola (SBC), 2005 CD-ROM.

BLIKSTEIN, P.; CAVALLO, D. **"God hides in the details":** design and implementation of technology-enabled learning environments in public education. being reviewed for Eurologo 2003, Porto, 2003.

BRAITENBERG, V. **Vehicles: Experiments in Synthetic Psychology.** MIT Press, 1984.

CAVALLO, D.; BLIKSTEIN, P. et al. **The City that We Want:** Generative Themes, Constructionist Technologies and School/Social Change. In Proceedings from the IEEE International Conference on Advanced Learning Technologies, Finland, September 2004.

CÉSAR, D. R; BONILLA, M. H. S. **Robótica Livre:** Implementação de um Ambiente Dinâmico de Robótica Pedagógica com Soluções Tecnológicas Livres no Cet CEFET em Itabirito - Minas Gerais – Brasil. In: XIII WIE – Workshop sobre Informática na Escola, 2007, São Paulo. Anais do XII WIE – Workshop sobre Informática na Escola (SBC), 2007 CD-ROM.

CHELLA, M. T. **Ambiente de Robótica para Aplicações Educacionais com SuperLogo.** 2002. 186 f. Dissertação (Mestrado em Engenharia Elétrica)-Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, 2002.

CORREA, L. **Vida Artificial.** In: 25 Congresso da Sociedade Brasileira de Computação, 2005, São Leopoldo. Anais do Encontro Nacional de Inteligência Artificial. São Leopoldo, 2005.

EVATRON, **Evatron.** Disponível em: <<http://www.evatron.com.br/>>. Acesso em 05/11/2007.

FERRARI, G.; STUBER, J.; LAVERDE, D. **Programming Lego Mindstorms with Java,** United States of America, Syngress, 2002.

FORTES, R. M. **Interpretação de Gráficos de Velocidade em um Ambiente Robótico.** 2007. 133 f. Dissertação (Mestrado em Educação Matemática), Pontifícia Universidade Católica de São Paulo, São Paulo, 2007.

GOGO, **GoGo Board.** 2007a. Disponível em: <<http://www.gogoboard.org/>>. Acesso em 05/11/2007.

GOGO, **GoGo Board Source Code.** 2007b. Disponível em: <<http://code.google.com/p/gogoboard/>>. Acesso em 05/11/2007.

HOGG, D. W. MARTIN, F.; RESNICK, M. **"Braitenberg Creatures" Epistemology and Learning,** Memo #13, MIT Media Laboratory, Cambridge, 1991.

KLASSNER, F.; ANDERSON, S. D. (2003). **LEGO Mindstorms:** Not Just for K-12

Anymore. IEEE Robotics & Automation Magazine, 10(2):12-18, June 2003.

LABTOOLS, **Labtools**. Disponível em: <<http://www.labtools.com.br/>>. Acesso em 05/11/2007.

LEGO, **Mindstorms RIS 2.0**. Disponível em : < <http://mindstorms.lego.com>>. Acesso em 05/11/2007.

LEGO, **Sistema Robótico**: Notas de Professor e Copymasters para LEGO DACTA™ conjunto #9790. 1999a. Billund, Denmark: The LEGO Group, 1999.

LEGO, **Subassembly Constructopedia**. In LEGO Group, LEGO MINDSTORMS™ set for Schools # 9790. 1999b. Billund, Denmark: The LEGO Group, 1999.

LOGO FOUNDATION, **Logo Foundation**. Disponível em :<<http://el.media.mit.edu/logo-foundation/>>. Acesso em 05/11/2007.

LOPES, D. Q. ; FAGUNDES, L. C. **As Construções Microgenéticas e o Design em Robótica Educacional**. RENOTE. Revista Novas Tecnologias na Educação, v. 4, p. 1-10, 2006.

LOPES, R. D; FICHEMAN, I. K; ALVES, A. C; SAGGIO, E. (Org). **Feira Brasileira de Ciências e Engenharia**. Anais FEBRACE 2007. São Paulo. Disponível em: < <http://www.lsi.usp.br/febrace/sobrefebrace/historia/anaisfebrace2007.pdf>> Acesso em 05/11/2007.

LOPES, R. D; FICHEMAN, I. K; ALVES, A. C; SAGGIO, E. (Org.). **Feira Brasileira de Ciências e Engenharia**. Anais FEBRACE 2006. São Paulo. Disponível em: < <http://www.lsi.usp.br/febrace/sobrefebrace/historia/anaisfebrace2006.pdf>> Acesso em 05/11/2007.

LOPES, R. D; FICHEMAN, I. K; ALVES, A. C; FILHO, I. L. (Org.). **Feira Brasileira de Ciências e Engenharia**. Anais FEBRACE 2005. São Paulo. Disponível em: < <http://www.lsi.usp.br/febrace/sobrefebrace/historia/anaisfebrace2005.pdf>> Acesso em 05/11/2007.

LOPES, R. D; FICHEMAN, I. K; ALVES, A. C; FILHO, I. L. (Org.). **Feira Brasileira de Ciências e Engenharia**. Anais FEBRACE 2004. São Paulo. Disponível em: < <http://www.lsi.usp.br/febrace/sobrefebrace/historia/anaisfebrace2004.pdf>> Acesso em 05/11/2007.

LOPES, R. D; FICHEMAN, I. K; ALVES, A. C. (Org.). **Feira Brasileira de Ciências e Engenharia**. Anais FEBRACE 2003. São Paulo. Disponível em: < <http://www.lsi.usp.br/febrace/sobrefebrace/historia/anaisfebrace2003.pdf>> Acesso em 05/11/2007.

MARTIN, F. **Cybernetics, and Programmable Turtles**. 1988. 87 p. Thesis (Master of Science in Mechanical Engineering) - Departament of Mechanical Engineering, Massachusetts Institute of Technology (MIT), Cambridge, 1988.

MARTIN, F.; MIKHAK, B.; RESNICK, M.; SILVERMAN, B.; BERG, R. **To Mindstorms and Beyond**: Evolution of a construction kit for magical machines. In A. Druin (Ed.), *Robots*

for Kids: Exploring New Technologies for Learning Experiences. San Francisco: Morgan Kaufman / Academic Press, 2000.

MICROCHIP. **Microchip Technology Inc.** Disponível em: <<http://www.microchip.com>>. Acesso em 05/11/2007.

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. 2006. 124 f. Dissertação (Mestrado em Informática)-Instituto de Matemática, Núcleo de Computação, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2006.

MIT. **Massachusetts Institute of Technology (MIT).** Disponível em: <<http://www.mit.edu>>. Acesso em 05/11/2007.

MUDEN, G. E. **Concurrency Issues in Programmable Brick Languages.** 2000. 117 p. Thesis (Master of Science in Computer Science)-The Graduate School, The University of Maine. Orono, 2000.

OCKO, S; PAPERT, S.; RESNICK, M. (1987) **LEGO, Logo, and Science.** *Technology and Learning*, v. 2, n.1.

OZIPICER, **PLMS OziPIC'er.** Disponível em: <<http://www.plmsdevelopments.com>>. Acesso em 05/11/2007.

PAPERT, S. M. **Mindstorms: Children, Computers, and Powerful Ideas.** New York: Basic Books, 1980.

PAPERT, S. **A máquina das crianças: repensando a escola na era da informática.** 1 ed., Porto Alegre. Artes Médicas, 1994.

RESNICK, M. MARTIN, F.; SARGENT, R.; SILVERMAN B. **Programmable Bricks: Toys to Think With.** IBM Systems Journal, vol. 35, no. 3-4, pp. 443-452, 1996.

RIBEIRO, C. **RobôCarochinha: Um Estudo Qualitativo sobre a Robótica Educativa no 1º ciclo do Ensino Básico.** 2006. 207 f. Dissertação (Mestrado em Educação)-Instituto de Educação e Psicologia, Universidade do Minho, Braga, 2006.

SABBATINI, R. M. E. **Imitação da Vida: A História dos Primeiros Robôs.** Revista Cérebro & Mente, Número 9. Disponível em <<http://www.cerebromente.org.br/n09/historia/robots.htm>>. Acesso em 05/11/2007.

SIPITAKIAT, A. BLIKSTEIN, P.; CAVALLO, D.. **GoGo Board: Augmenting Programmable Bricks for Economically Challenged Audiences,** In Proceedings of the International Conference of the Learning Sciences (ICLS 2004), Los Angeles, USA, 2004.

STEFFEN, H. H. **Robótica Pedagógica na Educação: Um Recurso de Comunicação, Regulagem e Cognição.** 2002. 114 f. Dissertação (Mestrado em Ciências da Comunicação)-Escola de Comunicação e Artes, Universidade de São Paulo, São Paulo, 2002.

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 05/11/2007.

TEIXEIRA, J. **Aplicações da Robótica no Ensino Secundário: o Sistema LEGO Mindstorms e Física.** 2006. 187 f. Dissertação (Mestrado em Ensino da Física)-Faculdade de Ciências e Tecnologia, Universidade de Coimbra, Coimbra, 2006.

VALENTE, J. A. ; CANHETE, C. C. **LEGO-Logo: Explorando o Conceito de Design.** in Valente, J. A. (Org). **Computadores e Conhecimento: Repensando a Educação,** Campinas, Unicamp, 1993.

VALLIM, M. B. R. **Em Direção à Melhoria do Ensino na Área Tecnológica: A Experiência de uma Disciplina de Introdução à Engenharia de Controle e Automação.** 2000. 101 f. Dissertação (Mestrado em Engenharia Elétrica)-Universidade Federal de Santa Catarina, Florianópolis, 2000.

WINPICPROG, **WinPicProg.** Disponível em: < <http://www.winpicprog.co.uk/>>. Acesso em 05/11/2007.