LEGO Dacta - Trabalhando com materiais alternativos

www.mecatronicafacil.com.br

MECATRÔNICA FÁCIL

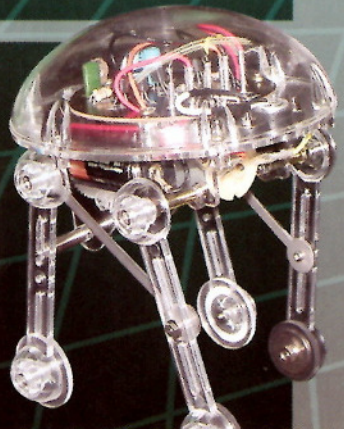
Ano 1 - nº6 - Setembro-Outubro/2002 - Brasil: R$ 8,50 - Europa: € 4,30



Microcontrolador PIC



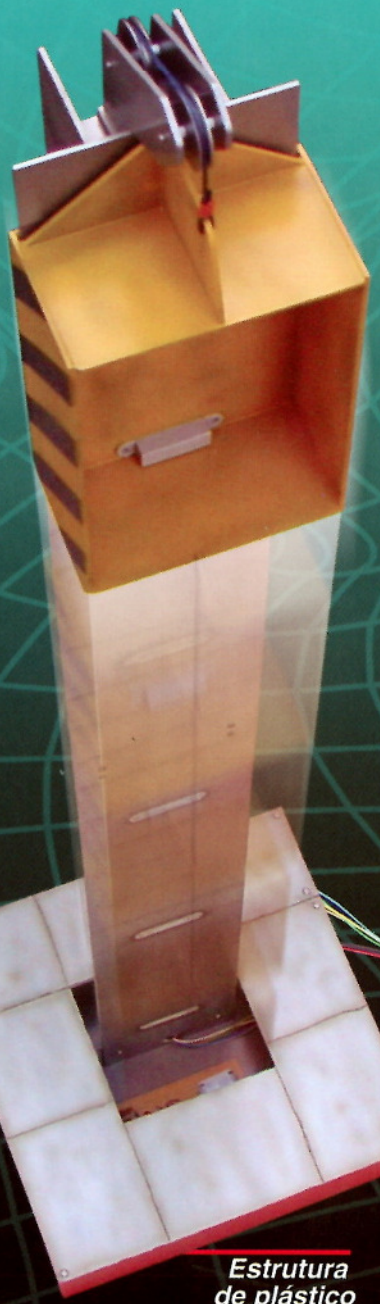
Os robôs que jogam futebol



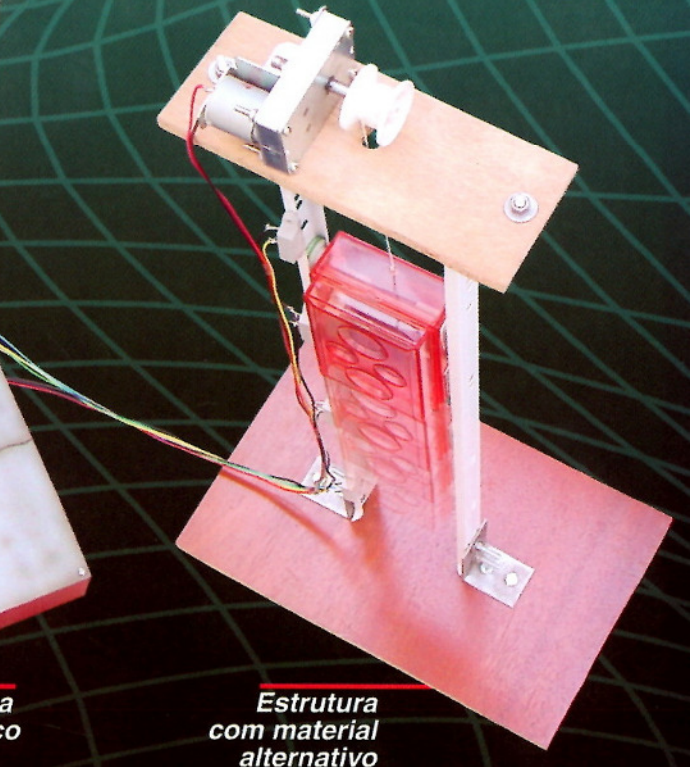
MECHA-MEDUSA
Aprenda a montar o kit

CONSTRUA UM ELEVADOR CONTROLADO PELO PC

Usando a linguagem LOGO



Estrutura de plástico



Estrutura com material alternativo

ISSN 1676-0980

LOGO - Basic Step - Introdução à mecânica

NAS BANCAS



www.revistapcecia.com.br

Editora Saber Ltda.
Diretores
Hélio Fittipaldi
Thereza M. Ciampi Fittipaldi

MECATRÔNICA FÁCIL
www.mecatronicafacil.com.br

Mecatrônica Fácil
Editor e Diretor Responsável
Hélio Fittipaldi

Editor Assistente
Paulo Gomes dos Santos

Publicidade
Eduardo Anion - Gerente
Ricardo Nunes Souza
Carla de Castro Assis
Melissa Rigo Peixoto

Conselho Editorial
Alexandre Capelli
Juliano Matias
Newton C. Braga

Colaboradores
Ewaldo Luiz de Mattos Mehl
Fabiano M. Paludetti
José Francci Júnior
Luís Henrique C. Bernardes
Márcio José Soares
Newton C. Braga
Octávio Nogueira
Wilson Roberto dos Santos

ASSINATURAS
www.mecatronicafacil.com.br
Fone/Fax: (11) 6192-4700
Atendimento das 8:30 às 17:30 h

Impressão
W.Roth

Distribuição
Brasil: DINAP
Portugal: Midesa

MECATRÔNICA FÁCIL
(ISSN - 1676-0980) é uma publicação da Editora Saber Ltda.
Redação, administração, números atrasados, publicidade e correspondência:
R. Jacinto José de Araújo, 315
CEP.: 03087-020 - São Paulo
SP - Brasil
Tel.: (11) 6192-4700

Empresa proprietária dos direitos de reprodução:
EDITORA SABER LTDA.

Associado à:
ANER - Associação Nacional dos Editores de Revistas

ANER

ANATEC - Associação Nacional das Editoras de Publicações Técnicas, Dirigidas e Especializadas.

ANATEC
PUBLICAÇÕES ESPECIALIZADAS
www.anatec.org.br

EDITORIAL



Atendendo a inúmeros pedidos, é com muita satisfação que trazemos a partir deste número, uma série de artigos sobre os "Microcontroladores PIC". Nesta série, você encontrará diversas informações sobre como utilizar o microcontrolador PIC16F84 nos seus projetos de Mecatrônica e aprenderá a construir uma placa para gravação de PIC de modo que possa fazer suas próprias gravações de programas.

Estréia também neste número a série "Introdução à Mecânica", onde o autor irá abordar diversos conceitos fundamentais dessa área da Mecatrônica que, muitas vezes, traz problemas para aqueles que estão mais familiarizados com a Eletrônica. Neste primeiro artigo, você irá conhecer melhor alguns sistemas de medidas, uma vez que isso é fundamental para se projetar qualquer peça mecânica.

Após o sucesso dos artigos da série "Trabalhando com plásticos", passaremos a trazer artigos que dão dicas de como trabalhar com materiais alternativos (um nome mais bonito para "sucata"). Neste mês, você irá conferir uma matéria sobre como trabalhar com o famoso "papelão". Como cortá-lo de modo a aproveitar melhor a sua resistência, como dobrá-lo, colá-lo e até mesmo como colocar parafusos nele!

O projeto prático deste mês é o "Elevador controlado pelo LOGO" que, assim como foi feito no último número, trazemos em duas versões: uma em plástico e outra em material alternativo (isso mesmo, a sucata). Neste projeto, você irá desenvolver suas habilidades para controlar um motor de corrente contínua através do PC e os professores do Ensino Fundamental e Médio poderão utilizá-lo com seus alunos devido a sua facilidade e riqueza de informações, principalmente em relação à Física.

Falando em professores, na seção "Escola" deste número, você vai conferir a "construção de uma estação de reciclagem", que é um projeto muito simples utilizando os *kits* da *LEGO dacta* e que visa trabalhar a interdisciplinaridade no ensino. Aproveito para fazer um convite aos professores que tenham idéias de projetos, para que as compartilhem com os demais leitores, entrando em contato conosco.

Boa leitura!

Paulo Gomes dos Santos

Atendimento ao leitor: a.leitor.mecatronicafacil@editorasaber.com.br

Os artigos assinados são de exclusiva responsabilidade de seus autores. É vedada a reprodução total ou parcial dos textos e ilustrações desta Revista, bem como a industrialização e/ou comercialização dos aparelhos ou idéias oriundas dos textos mencionados, sob pena de sanções legais. São tomados todos os cuidados razoáveis na preparação do conteúdo desta Revista, mas não assumimos a responsabilidade legal por eventuais erros. Caso haja enganos em texto ou desenho, será publicada errata na primeira oportunidade. Preços e dados publicados em anúncios são por nós aceitos de boa fé, como corretos na data do fechamento da edição. Não assumimos a responsabilidade por alterações nos preços e na disponibilidade dos produtos ocorridas após o fechamento.

Nº 6 - SETEMBRO/2002

ÍNDICE

Leitor 3

Robonews 5

Elevador controlado pelo LOGO 06

Mais um projeto onde apresentamos duas versões: uma em plástico e outra utilizando materiais de baixo custo. A idéia é controlar a subida, descida e parada de um elevador pelo PC através da Linguagem LOGO.

Construção de uma estação de reciclagem 16

Mais um projeto educacional utilizando o LEGO dacta. Esta estação é, na verdade, uma esteira que separa objetos metálicos de não metálicos. O principal objetivo deste projeto é mostrar como é possível trabalhar a Educação Ambiental na escola através da Mecatrônica.

Basic Step - 4ªparte 23

Aprenda a utilizar o Basic Step para enviar informações ao operador através de um visor de cristal líquido e conheça também as vantagens de empregar uma interface de comunicação serial para ligar o microcontrolador ao *display*.

Robô Mecha-Medusa MV-949 26

Conheça este robô andarilho que é acionado pelo som e veja, passo-a-passo, como montar o *kit* e também várias dicas úteis.

Os robôs que jogam futebol 32

Conheça os minúsculos e velozes robôs que jogam futebol que, apesar de parecerem simples brinquedos, são equipamentos muito sofisticados. Saiba quais as características desses robôs e as especificações das competições internacionais.

Trabalhando com materiais alternativos 38

Conheça algumas dicas para executar soluções mecânicas simples utilizando o papelão "poliondas", que é largamente empregado em embalagens industrializadas e, portanto, de facílima obtenção.

Introdução à Mecânica 41

Finalmente vamos desbravar esse universo que, muitas vezes, constitui um verdadeiro "nó" para quem deseja realizar projetos de Mecatrônica. Neste artigo, são discutidos alguns sistemas de medidas de comprimento e ângulo, bem como alguns dos instrumentos de medidas mais utilizados

Microcontroladores PIC 45

Os microcontroladores estão presentes em grande parte dos aparelhos que utilizamos no nosso dia-a-dia. Neste artigo, você vai conhecer a estrutura básica do microcontrolador e suas características básicas.

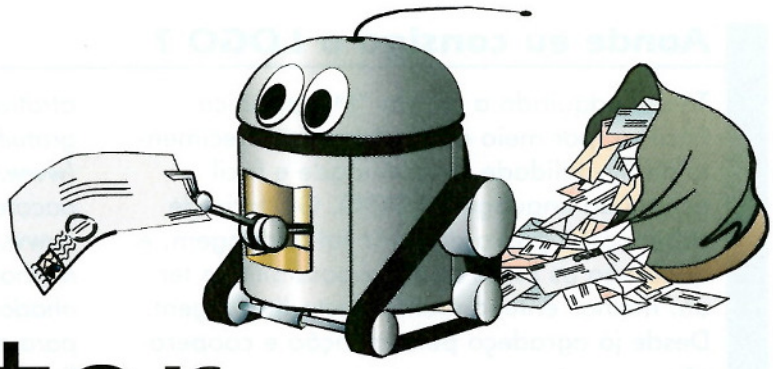
O LOGO controlando aplicações do Basic Step 52

Neste artigo, o autor discute como controlar um servomotor através do PC usando a linguagem de programação LOGO. E para interfacear esse controle, é utilizado o Basic Step, visto que não é possível controlar o servo diretamente pela porta paralela.

Controles PWM de potência 57

Uma forma de obter um controle mais preciso sobre o torque e velocidade de motores de corrente contínua ou até de outras cargas, é através de circuitos que empregam a tecnologia PWM.

Seção do Leitor



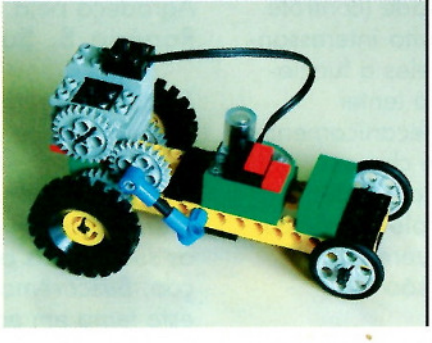
ENVIEM SUAS CARTAS PARA:

Editora Saber
Rua Jacinto José de Araújo, 315
CEP 03087-020 – São Paulo – SP
Ou *e-mails* para:
a.leitor.mecatronicafacil@editorasaber.com.br

Lego na escola

Gostaria de adquirir os exemplares anteriores, e parabenizá-los por utilizarem recursos como LEGO dacta, de fácil manuseio e compreensão. Vamos iniciar em nosso colégio (Ensino Médio) a Mecatrônica a partir dos projetos da revista que empregam o LEGO dacta. Portanto, gostaríamos, se possível, que sempre a revista abordasse projetos mais próximos ao currículo escolar, pois assim poderíamos até propor aos nossos alunos a assinatura da revista. Atenciosamente,
Profª Leny Pimenta - Colégio M. Lobato - COC Franca

Prezada Leny, que bom que as matérias publicadas estão auxiliando a escola onde você trabalha a implementar a Mecatrônica no Ensino Médio! Realmente, os kits da LEGO dacta são muito indicados para esse fim. Desejamos que vocês tenham um grande sucesso nessa nova empreitada que será mais um passo no crescimento da divulgação da Mecatrônica no Brasil. Continuaremos, sempre que possível, a trazer artigos que auxiliem os professores a implantar projetos de Mecatrônica com seus alunos, não só utilizando o LEGO Dacta, mas também outros materiais. Quanto aos números anteriores, estes podem ser adquiridos no site **www.mecatronicafacil.com.br** *ou pelo telefone* **11 61924700.**



Informações sobre o PIC



Adorei as revistas e gostaria de saber se é possível disponibilizar algum projeto de placa de programação dos PICs. Onde eu moro, é meio complicado conseguir cursos nas áreas de microcontroladores e robótica. As novas revistas são 10! Valeu!
Marcelo Gaudard

Caro Marcelo, atendendo ao seu pedido (e aos de inúmeros outros leitores também), começamos a publicar, neste número, uma série de artigos sobre os microcontroladores PIC. Em um dos próximos artigos, o autor ensinará como construir uma placa de gravação de PIC. Esperamos que você aprecie e obrigado pelos elogios.

Aonde eu consigo o LOGO ?

Tenho adquirido a revista "Mecatrônica Fácil" e, por meio desta, tomei conhecimento da flexibilidade, versatilidade e fácil acesso da linguagem LOGO. Gostaria de saber como posso adquirir tal linguagem, e quais são os livros que me possibilitem ter um melhor entendimento desta linguagem. Desde já agradeço pela atenção e cooperação.
Amarildo Rios - Americana – SP

Caro Amarildo, de fato a linguagem LOGO é bastante flexível e versátil, mas o seu principal atrativo é o fácil acesso. Ele pode ser baixado gratuitamente no site do Nied (www.nied.unicamp.br). Existe também um pacote educacional disponibilizado pela Divertire (www.divertire.com.br) em conjunto com a Melhoramentos, onde o software vem acompanhado com vários projetos e exemplos voltados para a educação. Este pode ser adquirido nas livrarias. Não conhecemos ainda nenhum livro que trate da Linguagem LOGO da forma que abordamos. Recomendamos que continue nos acompanhando, pois sempre procuraremos trazer informações sobre esta linguagem de programação.

Assinatura e números atrasados

Coleciono as revistas "Mecatrônica Atual" e "Mecatrônica Fácil" desde seu lançamento. Entretanto, perdi a revista "Mecatrônica Fácil" nº 3. Queria adquiri-la e também fazer a assinatura de ambas as revistas, para não precisar ir à banca comprar. Por favor, mandem-me a proposta de assinatura ou boleto bancário para que eu possa ter a facilidade de recebê-las em minha casa. Obrigado.
Eduardo Silva Santos

Caro Eduardo, é muito gratificante saber que você nos acompanha desde o início. Sobre a revista que você perdeu, não se preocupe, pois ainda dispomos dela em nosso estoque e você pode adquiri-la através do nosso site (www.mecatronicafacil.com.br) ou pelo telefone (11 61924700). Aproveite e faça a assinatura neste mesmo site ou telefone.



Porta serial

Gostaria de saber onde encontro exemplos de comunicação entre o PC e um equipamento externo através da porta serial. Preciso entender como são recebidos e enviados os dados pela porta serial. Grato,
Nelson

Caro Nelson, na edição anterior publicamos um artigo sobre comunicação com a porta serial do PC utilizando a Linguagem LOGO. Nele, o autor aborda alguns conceitos básicos sobre comunicação serial. Pretendemos, no futuro, trazer mais artigos explorando esta via de comunicação entre o PC e equipamentos externos.

Controle eletrônico da caixa de redução

Acompanho a revista "Mecatrônica Fácil" desde o primeiro número, e estou interessado em algumas aplicações que vêm sendo mostradas nas edições.
Na verdade, achei a funcionalidade (controle eletrônico) dos servomotores muito interessante. Contudo, preciso associar a eles a funcionalidade da caixa de redução. Já tentei implementar alguns controles mecanicamente do tipo chaves de fim-de-curso e chaves próximas ao eixo de rotação da caixa de redução. Porém eu acho o controle eletrônico mais 'limpo' e preciso, não requerendo ajustes mecânicos que às vezes são bem trabalhosos.
Daí, pergunto: como faço para agregar à minha caixa de redução (que tem a forçaque preciso) as funcionalidades de controle (eletrônico) dos servos?
Agradeço pela atenção.
Rogério S. Suguiura

Prezado Rogério, neste número, trouxemos um projeto prático onde controlamos uma caixa de redução com o PC através de um circuito de interface. Certamente, os conceitos trabalhados neste artigo ajudarão você a desenvolver melhor as suas idéias para a utilização da caixa de redução. Buscaremos trazer mais artigos que trabalhem este tema em edições futuras. Aguarde.

ROBONEWS

Newton C. Braga

Termina com sucesso a Robocup 2002

A cerimônia de encerramento da "Robocup 2002" aconteceu em 23 de junho, contando com 1004 jogadores em 288 equipes de 29 países. Este foi o maior evento no gênero, com 117 300 fãs indo ao Fukuoka Dome Stadium durante os 5 dias da competição.

Na final, uma partida entre os robôs da Four-legged Robot League (Liga dos Robôs de Quatro Pernas), que foi a mais popular do evento, acabou com a vitória da equipe CMPack'02, da Carnegie Mellon, por pênaltis.

Na liga dos Humanóides, que é uma categoria nova nesta competição, a equipe Nagara, da Gifu Prefecture Industry Association, ganhou o troféu na classe de 80 cm, também na decisão por pênaltis.

O principal ponto observado na competição foi uma evolução das tecnologias empregadas nos robôs, que se tornaram mais hábeis e sofisticados.

A próxima competição será na Itália, onde se esperam robôs ainda mais avançados.

Robô interativo tem personalidade

Combinando os mais avançados recursos da interação entre humanos e computadores, com a forma mais antiga de divertimento que é a marionete, foi criado o Horatio Beardsley.

"Doc" Beardsley é um robô animatrônico, uma criatura descendente de bonecos, como os vistos nas antigas catedrais medievais da Europa, que batiam os sinos e realizavam outros movimentos complexos. A ciência moderna, entretanto, agregou a Doc Beardsley a capacidade de ver, entender palavras articuladas e levar avante uma conversação.

A criação é dos pesquisadores da Carnegie Mellon University que, simplesmente, colocaram um "corpo" numa interface com um computador. Entre os atributos de Doc está o de que, ao ser perguntado sobre sua origem, ele responde que nasceu no alto de uma montanha da Áustria e é filho de criadores de cabras.

As pesquisas que levaram à criação de Doc têm, todavia, finalidades mais sérias: podem levar a assistentes digitais pessoais dotados de "personalidade", brinquedos eletrônicos interativos, figuras de museus animadas, além de garçons e atendentes robôs com capacidades avançadas de interação com os humanos, segundo afirma Ron Weaver, estudante graduado da Carnegie Mellon University.



Microfones e câmeras que captam sons e movimentos fornecem a percepção de Doc, que reage aos movimentos das pessoas virando a cabeça para o lado certo quando capta alguma coisa.

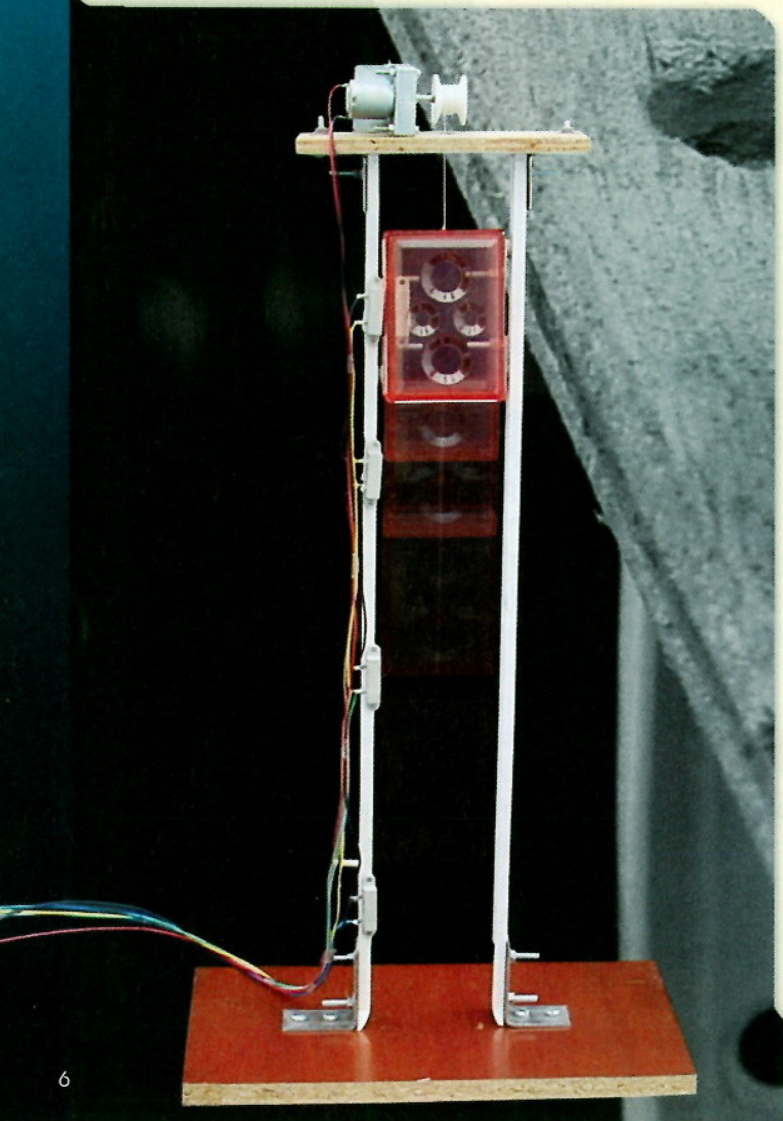
Além da conversação programada, o próximo passo a ser dado será dotar a criatura de senso de humor e até de temporização cômica e improvisação de modo a simular a dinâmica social, segundo afirmam os pesquisadores.

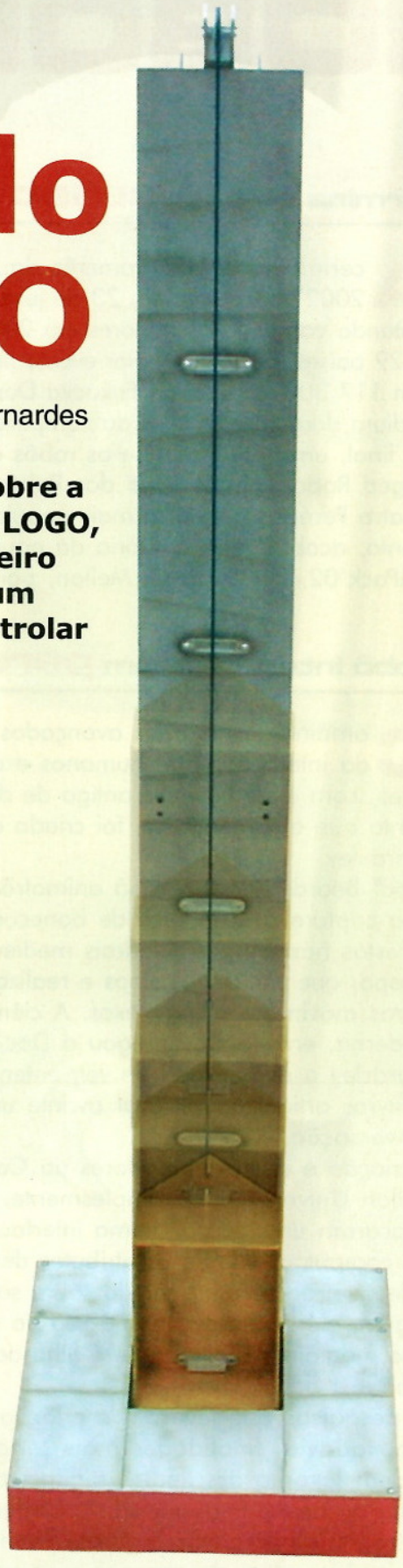
Mas, criaturas como Doc ainda não estão prontas para serem fabricadas. De acordo com os cientistas, somente daqui a uns 5 a 8 anos os primeiros robôs interativos com senso de humor estarão sendo fabricados e vendidos.

Elevador controlado pelo LOGO

Luiz Henrique Corrêa Bernardes

Após todos os artigos que escrevemos sobre a teoria e algumas montagens utilizando o LOGO, chegou a hora de fazermos o nosso primeiro projeto. Escolhemos para isso controlar um elevador, visto que já sabemos como controlar um motor de corrente contínua.





Mas, antes de falarmos de programação, vamos mostrar a montagem de uma maquete de um elevador que fizemos utilizando material alternativo. A **figura 1** mostra a ilustração do elevador montado. Antes de começarmos a explicar a montagem, gostaríamos de discorrer sobre segurança. O item mais importante em qualquer projeto que envolva montagem manual é, sem dúvida, a segurança no manuseio e operação de ferramentas e materiais. Na nossa montagem sugerimos o uso de equipamentos e

Figura 1 – Esquema da montagem.

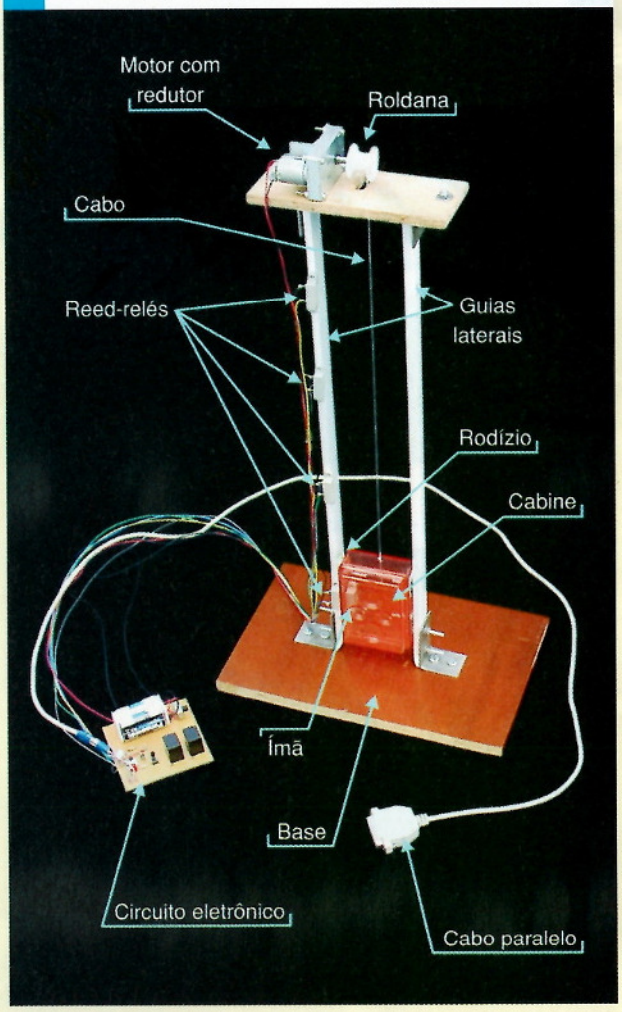


Figura 2 – Guias de gavetas utilizadas na montagem.

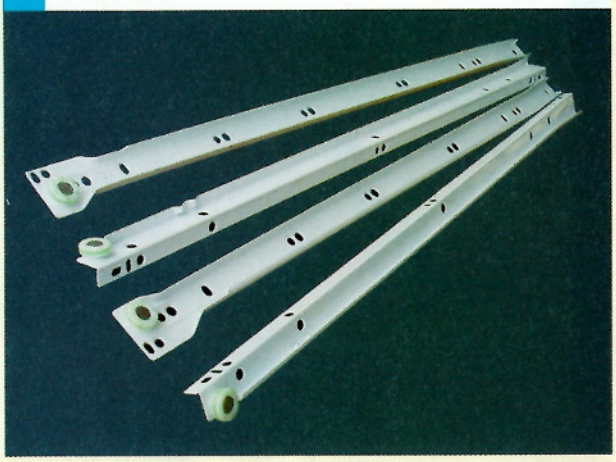


Figura 3 – Serrando as guias com um arco de serra.

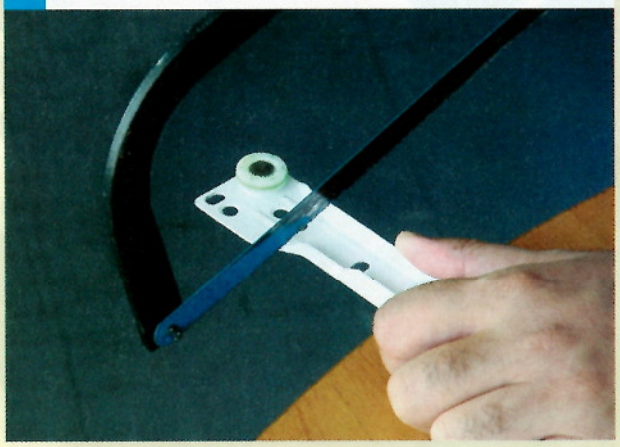


Figura 4 – Fazendo os furos na saboneteira.

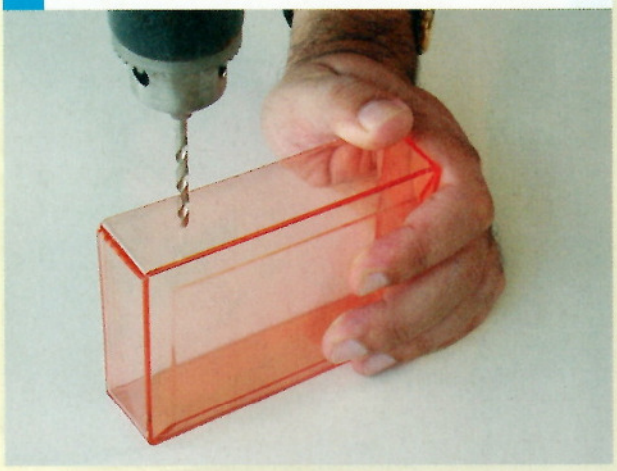
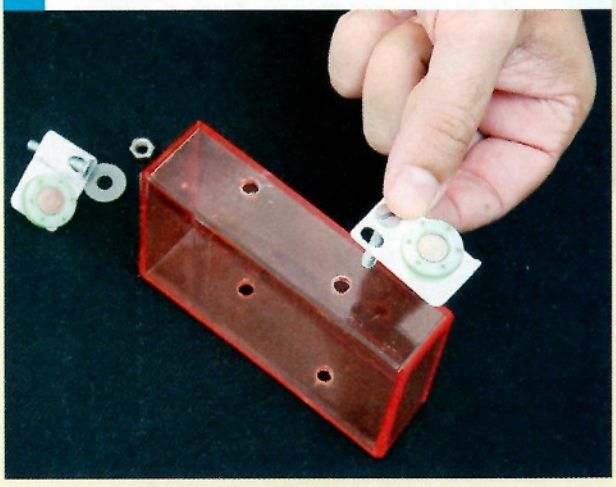


Figura 5 – Colocando os rodízios.



ferramentas que podem provocar ferimentos em caso de manuseio ou utilização incorreta. Portanto, caso tenha dificuldades ou nunca tenha utilizado os materiais e ferramentas propostas, consulte sempre uma pessoa mais experiente.

Figura 6 – Fixando os rodízios.



Figura 7 – Cabine montada.

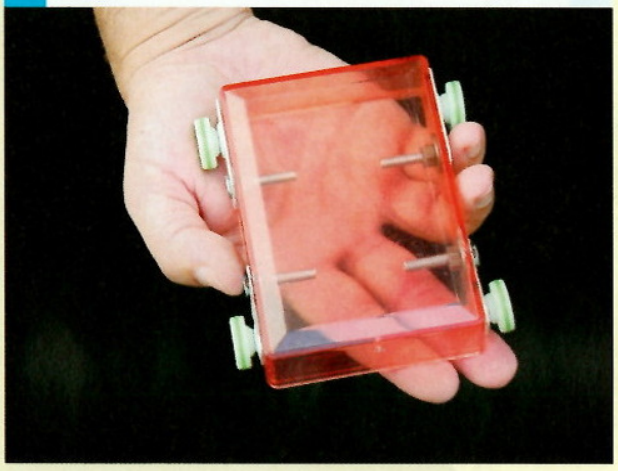


Figura 8 – Retirando a parte interna de uma roda de aeromodelo para fazer a roldana.



MONTAGEM

Para as guias laterais utilizamos guias de gavetas com rodízios (**figura 2**), e cortamos os rodízios com um arco de serra (**figura 3**). Para a montagem

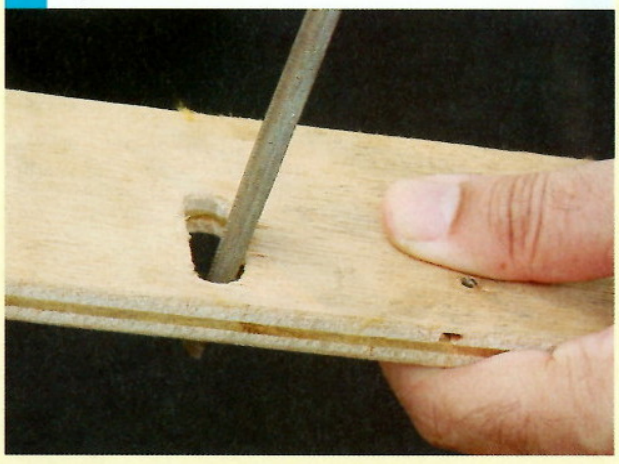
Figura 9 – Fazendo um furo na roldana.



Figura 10 – Fixando a roldana no motor.



Figura 11 - Fazendo um furo para passagem do cabo.



da cabine, utilize uma saboneteira e faça os furos para a fixação dos rodízios conforme a **figura 4**. Depois, com parafusos fixe os 4 rodízios que foram previamente cortados (**figuras 5 e 6**). Observe a cabine montada na **figura 7**.

Agora, vamos preparar uma roldana utilizando a parte interna de uma roda de aeromodelo (**figura 8**). Faça um furo na lateral para fixação do cabo que irá movimentar a cabine (**figura 9**). Pegue o motor com redutor e fixe a roldana (**figura 10**). Ob-

Figura 12 – Fazendo o furo para a cantoneira.



Figura 13 – Fixando a cantoneira.



Figura 14 – Fixando a guia.

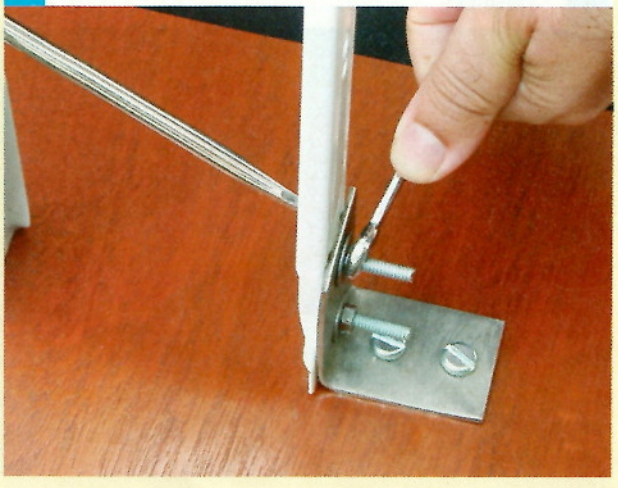


Figura 15 – Colocando a cabine entre as guias.

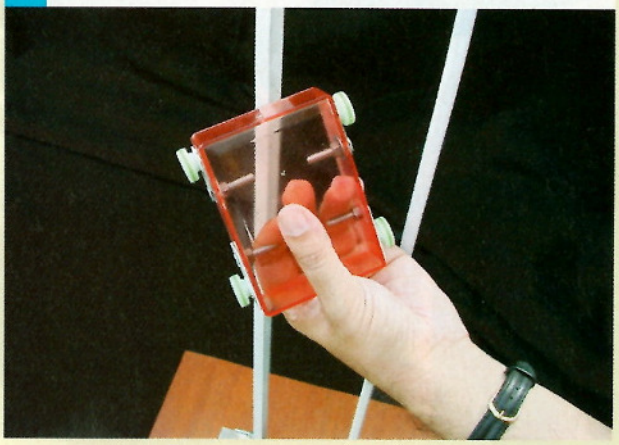


Figura 16 – Verificando se a cabine desliza livremente.

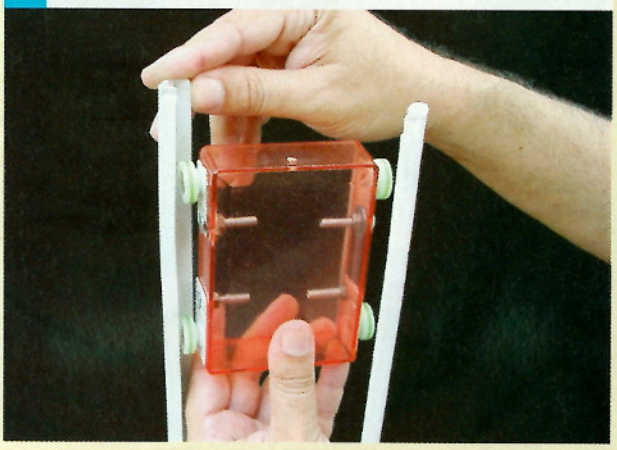
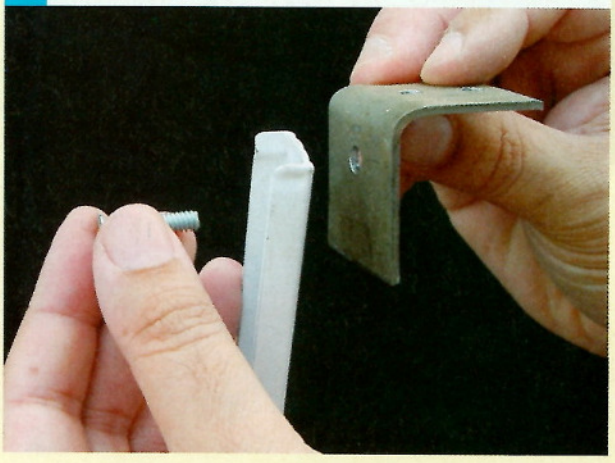


Figura 17 – Colocando a cantoneira para a fixação da base do motor.



servando a **figura 11**, prepare a base de fixação do motor com redutor fazendo um furo para passagem do cabo.

Na seqüência, vamos fixar as guias em uma base de madeira utilizando cantoneiras e parafusos conforme a série de **figuras 12, 13 e 14**. Lembre-se de deixar um espaço entre as guias de tal maneira que a cabine possa deslizar sem prender-se ou sair das guias (**figuras 15 e 16**). Feito isso, coloque a cabine em definitivo nas guias e prenda a base do motor

Figura 18 – Fixando a base do motor.



Figura 19 – Colocando o motor.

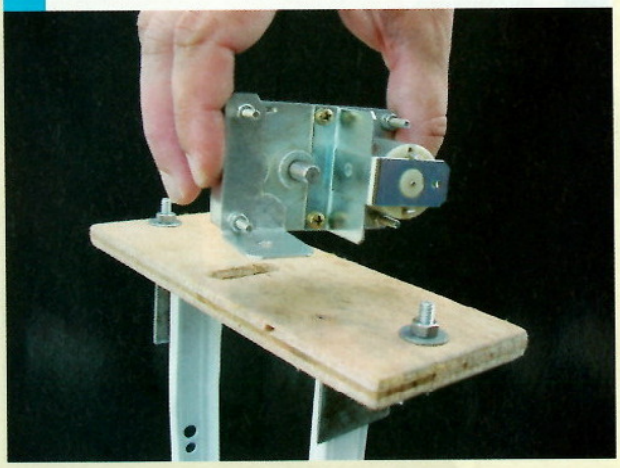


Figura 20 – Fixando o motor.

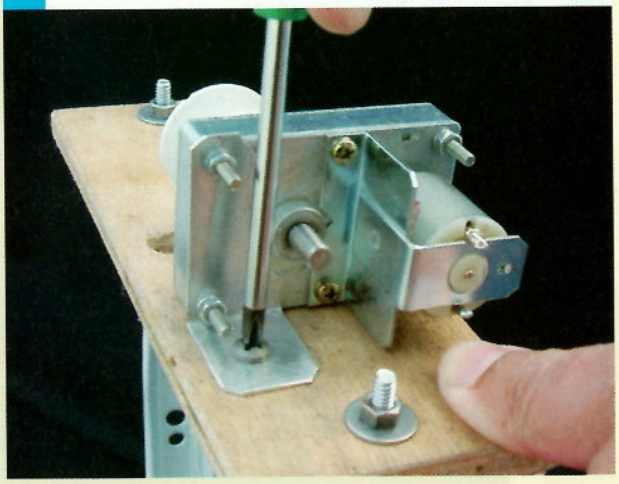


Figura 21 – Prendendo o cabo na roldana.

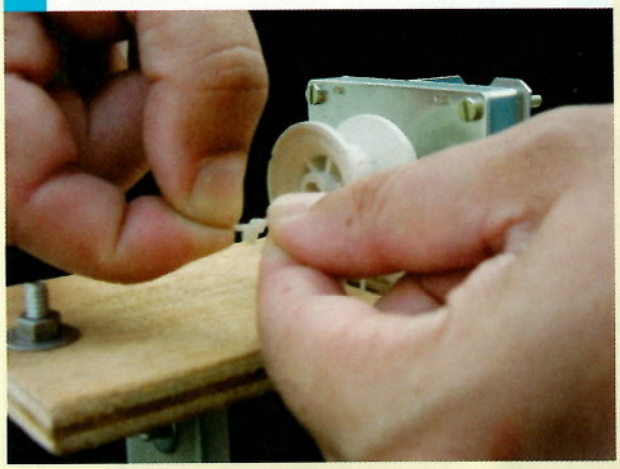
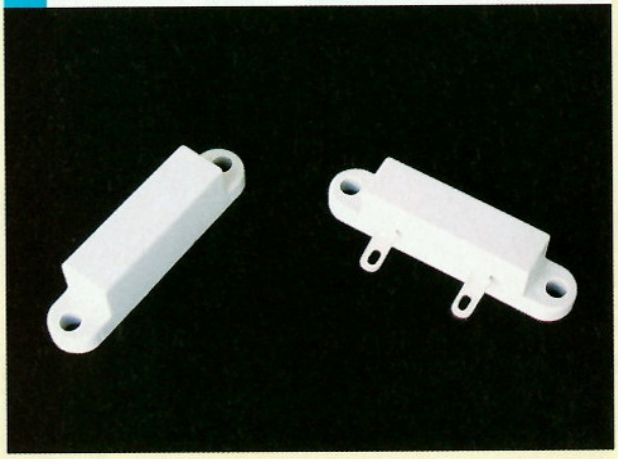


Figura 22 – Prendendo a cabo na cabine.



Figura 23 – O ímã (esquerda) e o reed-relé (direita).



com cantoneiras e parafusos (**figura 17 e 18**). Centralize e fixe o motor com a roldana em cima do furo da base (**figuras 19 e 20**). Prenda o cabo na roldana (**figura 21**) e na cabine (**figura 22**). Agora, só falta centralizar o ímã (**figura 23**) na cabine (**figura 24**) e os *reed-relés* nas hastes (**figura 25**) onde serão as paradas dos andares.

Ótimo! Estamos com a mecânica do elevador pronta. Vamos testá-la antes de prosseguir. Pegue um conjunto de 4 pilhas (6 volts) e conecte no motor de tal maneira que faça a cabine subir e descer (**figura 26**), e com um multímetro observe os "reed-relés", se estão funcionando perfeitamente quando o ímã da cabine passa próximos a eles, veja na **figura 27**.

Pronto, a montagem mecânica está pronta e testada! Agora faça a montagem do circuito de controle (**figura 28**) conforme o esquema elétrico descrito na **figura 29**.

Figura 24 – Centralizando o ímã na cabine.

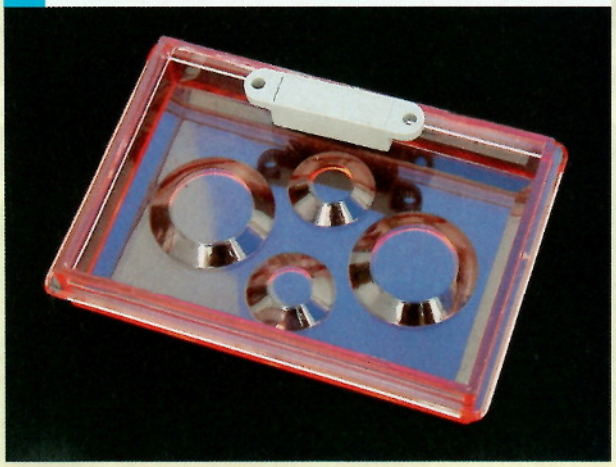


Figura 25 – Colocando os *reed-relés* nas hastes.

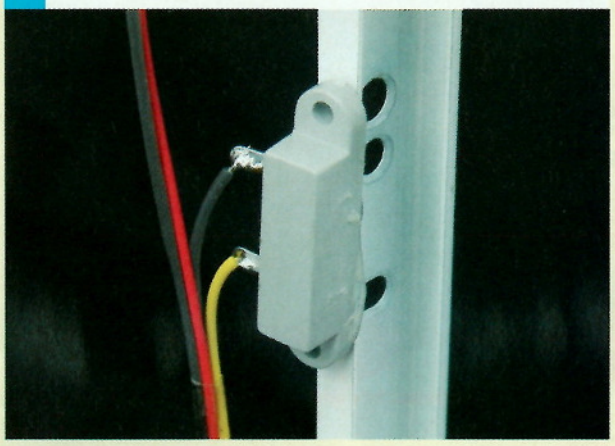


Figura 26 – Testando o motor.

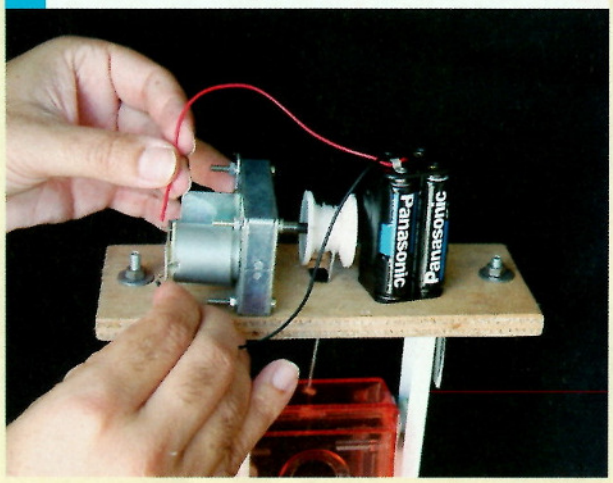


Figura 27 – Testando o sistema.

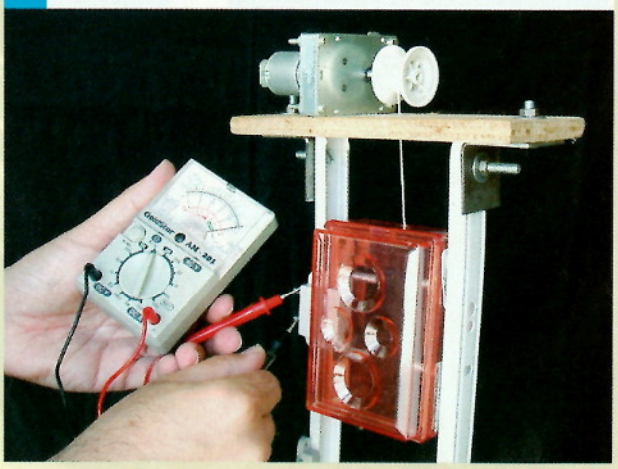
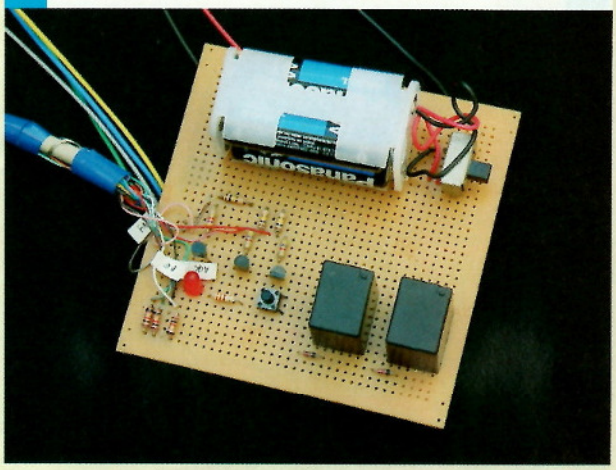


Figura 28 – Circuito elétrico.



PROGRAMAÇÃO

A proposta do programa de controle do elevador é a de um sistema semimanual (como os utilizados nos elevadores com ascensorista) onde os controles de subida, descida e parada são manuais.

A única parte automática são as paradas no terceiro andar e no térreo. Veja o fluxograma simplificado do funcionamento do programa (**figura 30**).

Figura 29 – Esquema elétrico.

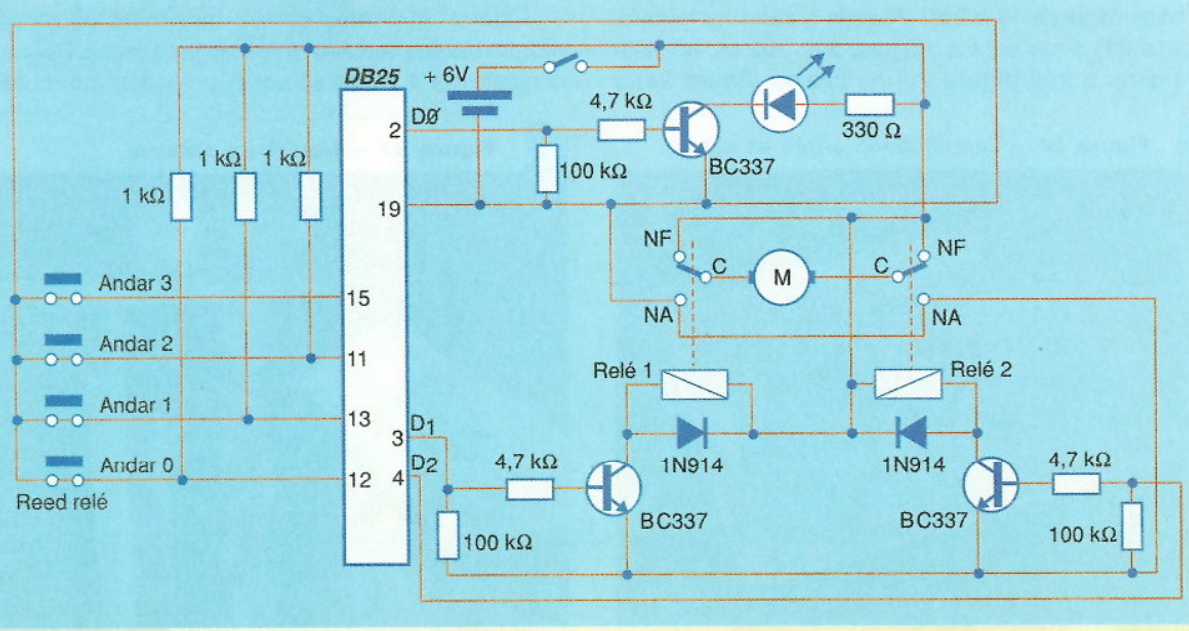
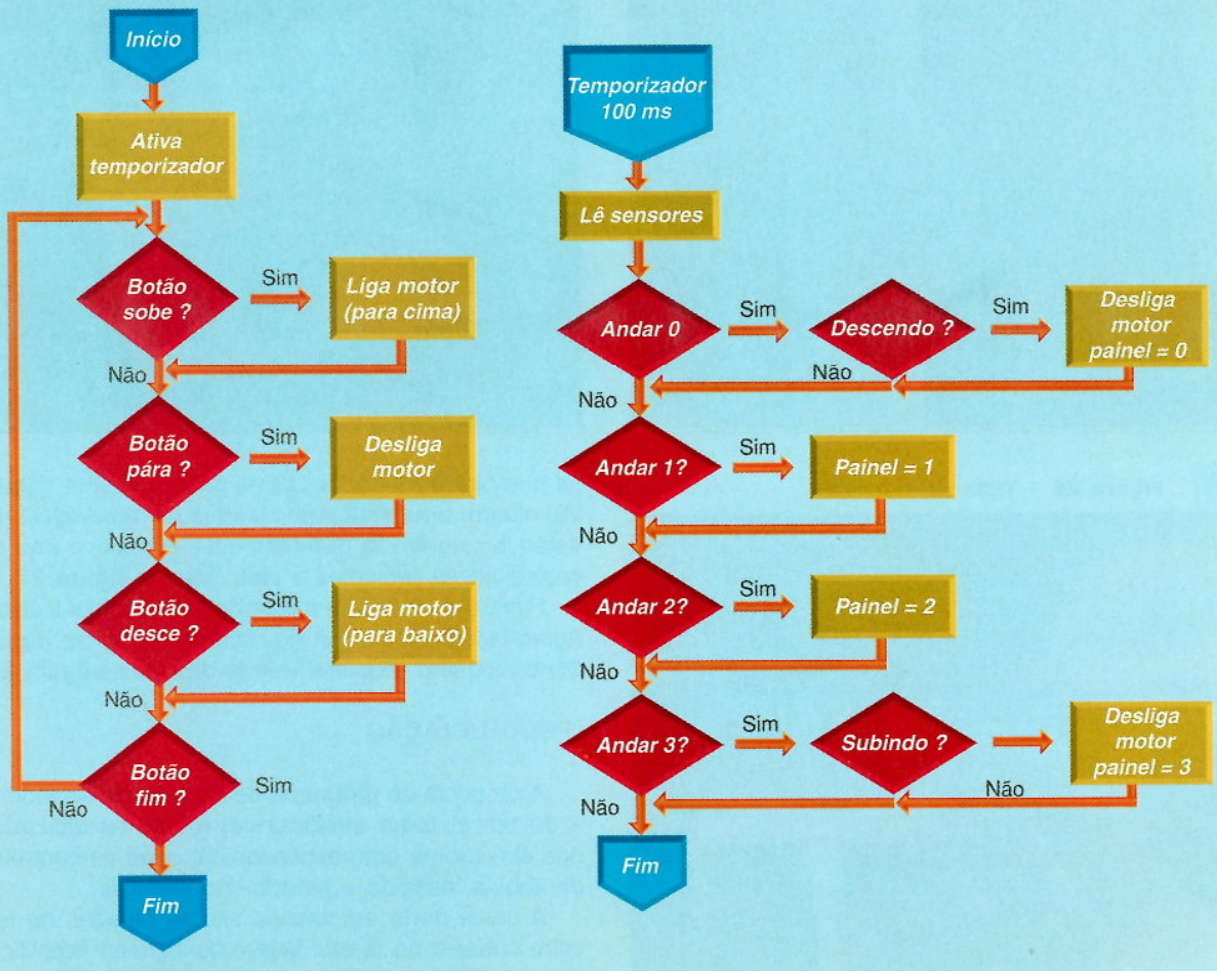


Figura 30 – Fluxograma do programa.



Código-fonte 1 - Programa do elevador.

```
aprenda desce
atribua "motor "Desce
atribua "sensor portaentradab 889
atribua "rasc bite :sensor 32
se não :rasc = 0 [portasaídab 888 2 ]
fim


aprenda elevador
dt
atribua "motor "xxxxx
atribua "andar "x
para
ativetemporizador 1 100 [ve_andar]
criejanela "main "j1 "elevador 20 20 100 150
criebotão "j1 "b1 "Sobe 30 40 40 20 [sobe ]
criebotão "j1 "b2 "Para 30 60 40 20 [para ]
criebotão "j1 "b3 "Desce 30 80 40 20 [desce]
criebotão "j1 "b4 "Fim 60 120 30 15 [termina]
crieestático "j1 "e1 :motor 40 10 25 15
crieestático "j1 "e2 "Andar 30 25 20 15
crieestático "j1 "e3 "X 60 25 20 15
fim

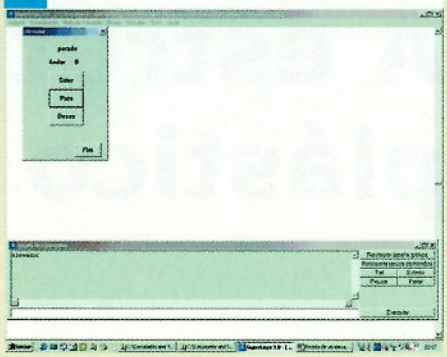

aprenda para
atribua "motor "parado
portasaídab 888 0
fim


aprenda sobe
atribua "motor "sobe
atribua "sensor portaentradab 889
atribua "rasc bite :sensor 8
se não :rasc = 0 [portasaídab 888 4 ]
fim


aprenda termina
apaguejanela "j1
desativetemporizador 1
at
fim


aprenda ve_andar
desativetemporizador 1
atribua "sensor portaentradab 889
atribua "rasc bite :sensor 8 ; Bit Fault
se (e sãoiguais :motor "sobe :rasc = 0 ) ~
                    [para atribua "andar 3]
atribua "rasc bite :sensor 16 ; Bit Sel In
se :rasc = 0 [ atribua "andar 1]
atribua "rasc bite :sensor 32 ; Bit PE
se (e sãoiguais :motor "desce :rasc = 0 ) ~
            [espere 23 para atribua "andar 0 ]
atribua "rasc bite :sensor 64 ; Bit ACK
se :rasc = 0 [atribua "andar 2 ]
mudeestático "e1 :motor
mudeestático "e3 :andar
ativetemporizador 1 100 [ve_andar]
fim
```

Figura 31 – Tela de controle.



Note que temos o programa principal onde o temporizador é ativado e os botões são controlados. Já na rotina de temporização, são monitorados os sensores para atualizar o painel do elevador que indica o andar. Dependendo do andar (0 ou 3) e do movimento do elevador, o motor será desligado. Por exemplo, se o elevador estiver subindo e chegar no andar 3, automaticamente o motor será desligado.

Agora, é só carregar o programa do elevador (**código-fonte 1**) e suas funções conforme a listagem, observando que todos os conceitos utilizados na programação já foram apresentados nos artigos da série sobre o LOGO.

Então, estamos prontos para ver se tudo funciona. Execute o programa "elevador" e aparecerá a tela de controle conforme a **figura 31**. Portanto, é só pressionar os botões desejados e ver se tudo funciona como desejamos.

CONCLUSÃO

Quando tudo está funcionando, é um grande prazer e esquecemos de todo o trabalho que deu para realizá-lo: encontrar os materiais; emprestar as ferramentas do vizinho ou do tio; entender porque esqueceu de digitar o comando "**ativetemporizador 1 100 [ve_andar]** " e gastar uma hora para descobrir porque o elevador arrebentava o cabo e o motor não parava! Enfim, o prazer de ver tudo funcionando é contagiante e não vemos a hora de modificar ou de fazer outro projeto. Então, fica a proposta: envie-nos suas modificações e sugestões de outras montagens do elevador.

Boas montagens e até o próximo projeto!

A estrutura de plástico do Elevador

José Francci Júnior

A estrutura é, de fato, muito simples, mas como as peças são grandes e os esforços também, foi dada atenção redobrada à colagem e muitos pontos críticos foram reforçados.

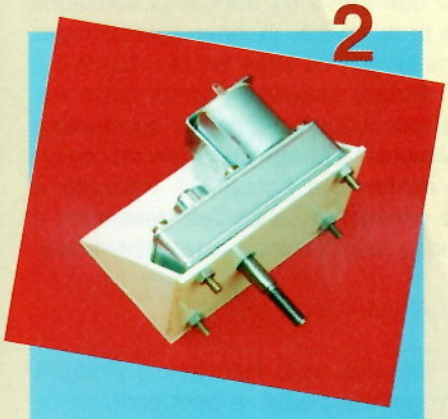
Toda a construção foi feita em chapa de plástico poliestireno de 2 mm à exceção das tampas e de alguns componentes das roldanas de passagem, onde utilizamos chapa de 1 mm.

Como o conjunto todo é muito grande ele foi construído em três módulos (a torre em duas metades e a base) que se unem por parafusos. A altura do conjunto (1.04 m) tornou necessária a instalação de pesos de chumbo nas extremidades da base para uma melhor estabilidade.

O *layout* do conjunto foi baseado nos elevadores de carga utilizados em obras e oficinas, com a parte motriz na base.



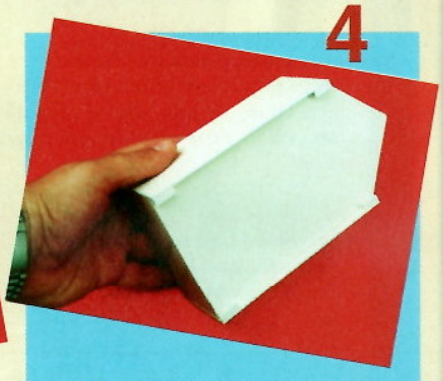
1 Detalhe da construção da base, note os barretes de chumbo nas extremidades para estabilizar o conjunto.



2 Vista do suporte do redutor.



3 Detalhe do carro do elevador.



4 Detalhe posterior do carro do elevador onde se vêem os engates guia da torre.



5 As metades da torre em contrução.



6 Detalhe da torre com o carro do elevador engatado.



7 Detalhe da contrução das roldanas de passagem, à esquerda as superiores e à direita a inferior.

8



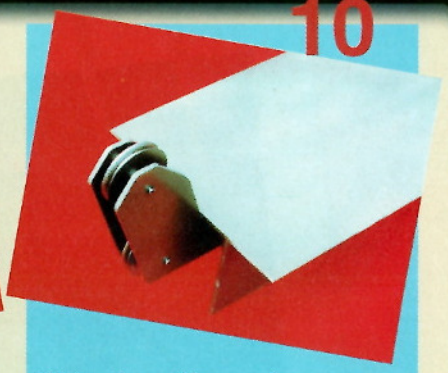
Vista do carretel do cabo do elevador instalado no redutor.

9



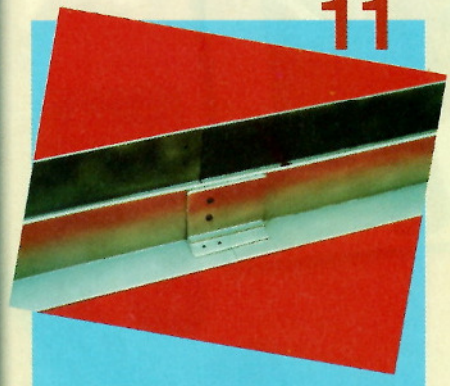
Vista da roldana de passagem inferior acabada.

10



Vista das roldanas de passagem superiores já instaladas na torre.

11



Detalhe da união das metades da torre.

12



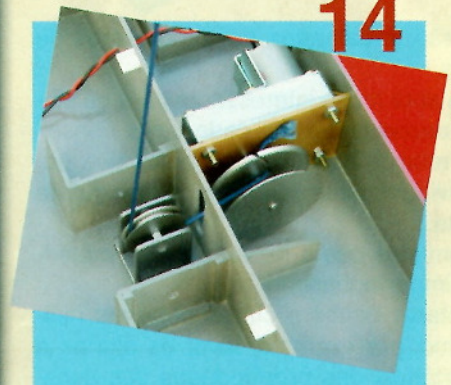
Detalhe do suporte da torre na base.

13



O conjunto motriz pronto antes da instalação na base.

14



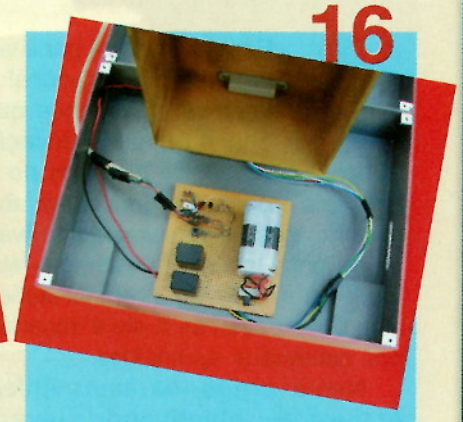
Detalhe do conjunto motriz na base.

15



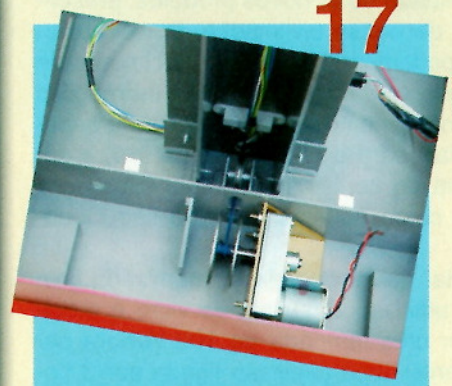
Colocação do ímã na cabine.

16



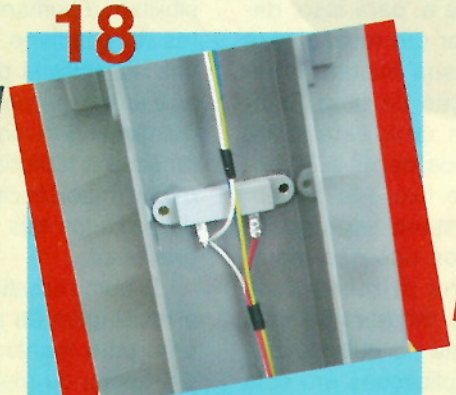
Conexões do circuito eletrônico.

17



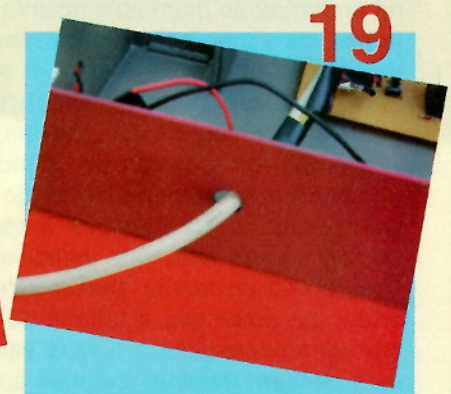
Conexões dos reed-relés e motor.

18



Detalhe da ligação do reed-relé.

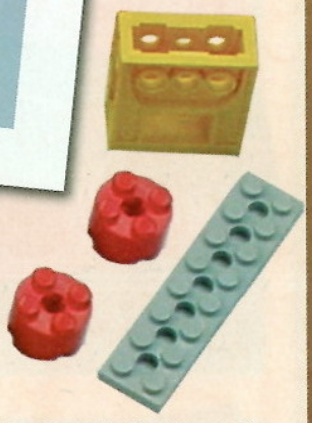
19



Furo para a passagem do cabo serial.



Construção de uma estação de reciclagem



Fabiano Miranda Paludetti

A **Escola Bosque** partilha novamente mais um projeto educacional com você leitor e com as instituições que vêm acompanhando nossa Educação Tecnológica e Robótica Pedagógica inseridas no currículo escolar, as quais trouxeram uma nova abordagem para a construção do conhecimento através da experiência. Vale a pena lembrar que sempre relacionamos o conteúdo de sala de aula com situações no cotidiano do aluno, e mostramos a ele a importância e a utilidade dos conteúdos aprendidos.

Neste artigo, daremos ênfase a Educação Ambiental objetivando levar ao aluno a conscientização da responsabilidade de cada um na preservação do meio ambiente e, para isso, daremos mais uma vez ao professor um roteiro para reproduzir esta experiência em sua escola, mostrando passo-a-passo a construção de uma Estação de Reciclagem que contém uma esteira de separação, programada por alunos de 4ª e 5ª série, que separa componentes do lixo que podem ser reciclados. Na verdade, a esteira separadora de lixo separa materiais metálicos dos não-metálicos como podemos observar na **primeira foto** onde, por exemplo, a latinha de refrigerante está em um armazenador e o vidro está no outro. Daremos também detalhes de suas aplicações em sala de aula, mencionando os conteúdos abordados e de que forma eles são trabalhados por professores de outras disciplinas.

Esta atividade será muito interessante para os alunos, pois estará despertando em cada um a consciência crítica em relação a pequenos gestos que podemos fazer individualmente para melhorar o futuro do Meio Ambiente.

*Comentar com eles que a separação dos materiais diferentes é importantíssimo.

Podemos destacar também outros objetivos muito relevantes como, por exemplo, levar os alunos a perceber a importância das propriedades das substâncias, sua composição e os processos químicos básicos e reconhecer que esses conhecimentos possibilitam à humanidade obter novos produtos, presentes no cotidiano de nossa civilização. Como objetivos específicos podemos citar o reconhecimento de que a matéria pode se transformar mediante reações químicas; o reconhecimento da importância da separação dos materiais para a reciclagem do lixo; e dar maior importância ao conhecimento das diferenças de produtos tais como aqueles que não são biodegradáveis.

É bom ressaltar que a energia economizada com a reciclagem de uma única garrafa de vidro é suficiente para manter uma lâmpada de 100 W/hora por 5,5 horas, ou manter uma televisão ligada por 3 horas; ou ainda, é igual a ½ lata de gasolina. O petróleo

Figura 1 - Peças para montar a Estação de reciclagem.

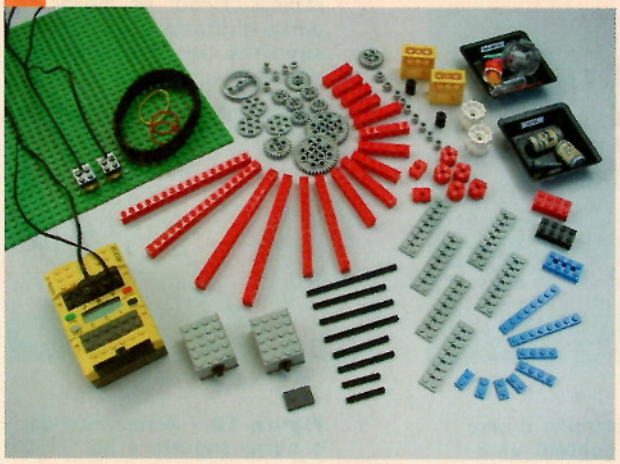
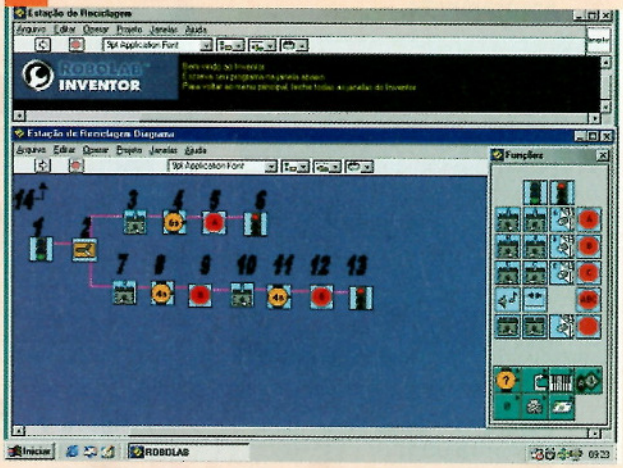


Figura 2 - Tela do programa elaborado no Sotfware Robolab.



é um combustível fóssil que leva milhões e milhões de anos para se formar e pode acabar em um futuro bem próximo se não revertermos esse processo já.

A principal diferença deste projeto em relação aos outros já publicados é que utilizaremos o RCX, um bloco amarelo que armazena a programação feita pelos alunos em um microcomputador e, para que as informações se transfiram do micro ao RCX, é preciso de uma Torre de Infravermelho. Tanto um quanto o outro e as demais peças, estão contidos no Kit 9790 da LEGO dacta, o qual utilizaremos neste projeto também.

Lembramos que antes de começar a montagem, o professor separa os alunos da classe em grupos de quatro componentes e explica a função de cada um. Um aluno será o construtor, outro o almoxarife que irá conferir e distribuir as peças necessárias para a montagem, a terceira função será a do relator que preencherá a ficha do projeto e a última função é a do apresentador, que descreverá todo o projeto e suas idéias. Outra diferença básica deste projeto para os anteriores é que sugerimos ao professor, atribuir as atividades de elaborar o programa no microcomputador ao relator e ao apresentador, feita no Software ROBOLAB da LEGO dacta, visto que o construtor e o almoxarife estarão já empenhados na construção do projeto. Após a distribuição das funções, vamos então ao começo da montagem do carrinho.

MONTAGEM

Enquanto o almoxarife fornece as peças da montagem (observadas na **figura 1**) ao construtor, o relator e o apresentador iniciam a elaboração do programa da Estação de Reciclagem, o qual detalharemos a seguir.

Este software é *iconizado*, facilitando a assimilação por parte dos programadores e tem uma gama de opções para enriquecer qualquer projeto de Robótica. Para começar a programação devemos *clicar* duas vezes no ícone do Robolab, escolher a opção Programador e logo após, Inventor 4. Aparecerá a tela demonstrada na **figura 2**. Vamos elaborar este programa dando um número a cada ícone utilizado. Devemos começar arrastando o ícone ***1*** ao canto esquerdo da tela, o qual dá inicio à programação. Adiciona-se na mesma direção o ícone ***2*** (bifurcação), possibilitando a execução de duas etapas simultaneamente. Acrescenta-se o ícone ***3*** (motor A com sentido para frente), motor que funcionará por tempo estipulado pelo ícone ***4*** (tempo de execução), no caso, 6 (seis) segundos. Depois, adicionamos o ícone ***5*** para cessar o funcionamento do motor A e terminamos a programação de um dos braços bifurcados, responsável pelo funcionamento da esteira da estação, arrastando o ícone ***6***. Começaremos agora a programação do segundo braço bifurcado com o ícone ***7*** (motor B com sentido para trás), abaixo e ligeiramente à esquerda do ícone ***3***, fazendo funcionar a parte giratória da estação que separará os materiais a serem reciclados. Este motor será executado por 4 (quatro) segundos como podemos observar no ícone ***8***. A adição do ícone ***9*** é para pausar o motor B, que passará a deslocar a parte giratória para o sentido inverso após arrastarmos o ícone ***10*** (motor B com sentido para frente). Acrescentaremos então o ícone ***11***, idêntico ao ícone ***8***, possibilitando ao motor B funcionar por mais 4 (quatro) segundos e depois parar, graças ao ícone ***12***. Finalizando a programação, coloca-se neste braço bifurcado o ícone ***13***, em processo semelhante à finalização da programação do braço da esteira.

Agora iniciaremos a etapa de transferência das informações efetuadas no microcomputador para o RCX, ligando-o pressionando o botão vermelho On-Off, posicionando-o em frente a torre de Infravermelho. Feito isso, *clica-se* no ícone ***14*** da **figura 2**. Após ouvirmos o efeito sonoro emitido pelo RCX, está concluída a transferência, fato observado na **figura 3**.

Figura 3 - Visualização da transferência da programação.



Figura 4 - Peça inicial para a montagem.

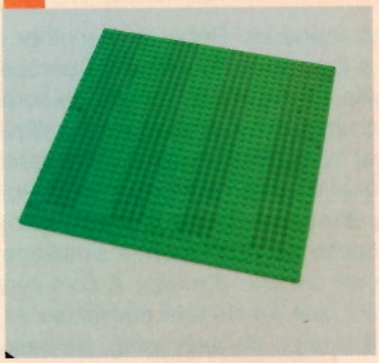


Figura 5 - Começando a base da esteira.

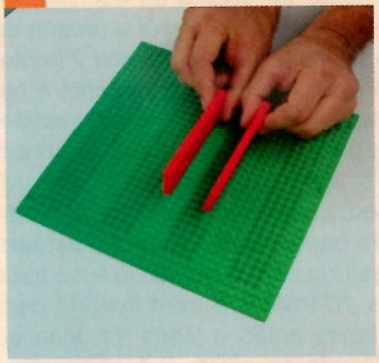


Figura 6 - Montando o eixo da esteira.



Figura 7 - Encaixando o eixo da esteira que contém uma roldana.

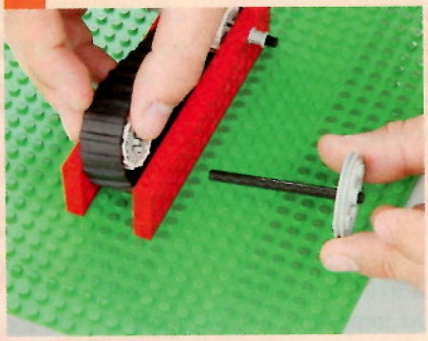


Figura 8 - Acoplando o redirecionador do motor da parte giratória.

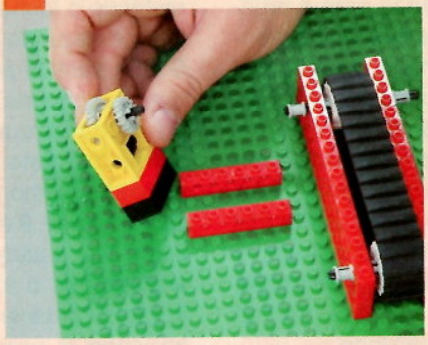


Figura 9 - Encaixando o eixo, a engrenagem do redirecionador do motor da parte giratória e um dos suportes das engrenagens da parte giratória.

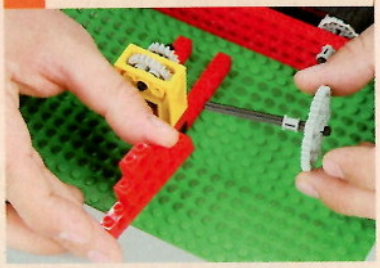


Figura 10 - Acrescentando o outro suporte e as engrenagens da parte giratória.

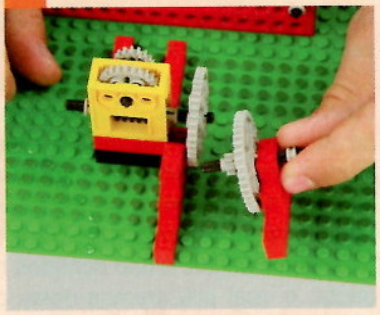
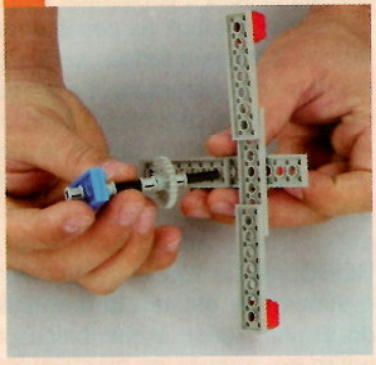


Figura 11 - Parte giratória vista de baixo (inacabada).



Mas, obviamente, efetuaremos a transferência somente após concretizar os passos de construção da estação que relataremos a seguir.

Na bancada onde se montará a estação coloca-se a peça verde fornecida pelo almoxarife e observada na **figura 4**. Para a montagem da base da esteira, adicionam-se duas peças vermelhas encaixadas paralelamente (**figura 5**). Na mobilização da esteira devemos acrescentar duas peças cinza à peça branca visualizadas na **figura 6** e depois envolvê-las na borracha que transportará os materiais. Encaixa-se um eixo em uma roldana e acopla-se a base do carrinho conforme a **figura 7**. Na **figura 8** se observa a inclusão, dentre outras peças já encaixadas, de uma amarela cuja função é redirecionar o movimento produzido pelo motor que irá fazer a esteira funcionar. A próxima etapa é adicionar a um eixo uma engrenagem com uma pecinha fixadora do eixo, encaixá-los na peça amarela, montar dois "triângulos" utilizando peças vermelhas (uma pequena, uma média e outra grande para cada "triângulo") e colocar entre eles outro eixo com outras duas engrenagens (uma pequenina e outra igual a última utilizada) – **figuras 9** e **10**. Agora se monta a parte giratória que

Figura 12 - Detalhe do ímã em um dos braços da parte giratória.

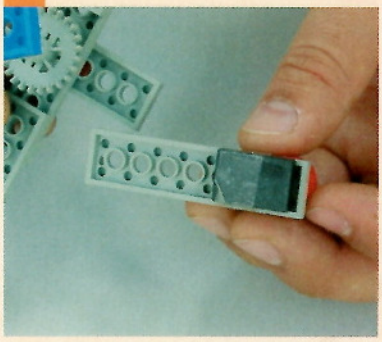


Figura 14 - Confecção da base do motor da parte giratória.

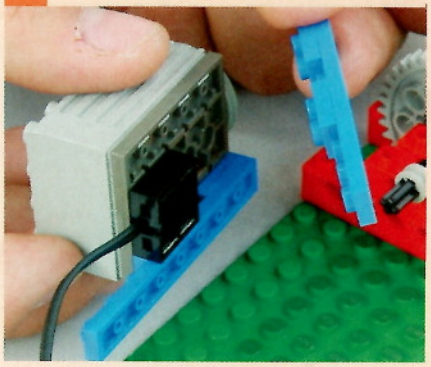


Figura 16- Acoplando o eixo menor e a roldana no redirecionador.

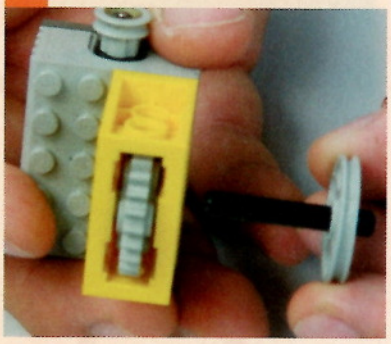


Figura 13 - Vista de cima do encaixe da parte giratória.

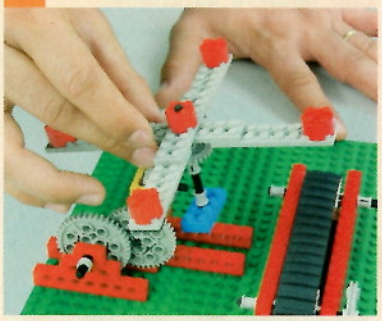


Figura 15 - Encaixando o redirecionador do motor da esteira.

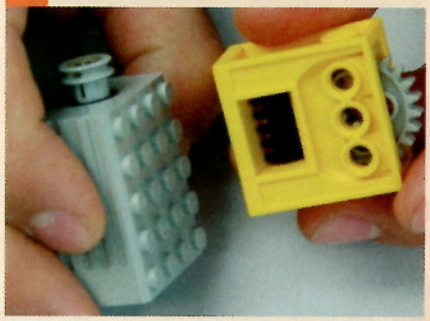
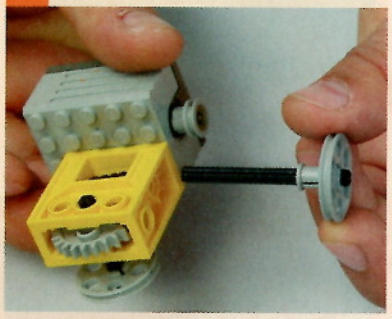


Figura 17 - Acrescentando o eixo maior e a roldana no redirecionador.



separará os componentes do lixo com um eixo que contém uma engrenagem dentada e uma peça azul que fixará o eixo com os quatro braços que girarão, sendo que parte de um deles foi retirado (**figura 11**) para destacar a acoplagem de um ímã em sua extremidade, como podemos ver na **figura 12**. Na base do redirecionador de movimento, onde está a peça amarela, encaixamos o eixo da parte giratória observada na **figura 13**. O almoxarife fornece então o motor que movimentará o último eixo fixado ao construtor, que nele adiciona duas bases feitas com peças azuis e um fio conector (**figura 14**), cuja outra extremidade é conectada na saída B do RCX.

Acrescentaremos o segundo motor ao projeto (aquele que irá movimentar a esteira) e para isso necessitaremos de um outro redirecionador que deve ser acoplado nesse motor da **figura 15**, contido de uma pequena roldana onde brevemente será colocado um elástico. Ao redirecionador é adicionado um eixo pequeno com uma roldana maior na ponta (**figura 16**). Visualizamos na **figura 17** que na peça amarela é encaixado um eixo um pouco maior com uma roldana igual a utilizada na figura anterior. Agora acrescentamos um elástico vermelho àquela pequena roldana há pouco mencionada, acoplando-os à roldana maior da **figura 18**. Em seguida montamos a base do motor e encaixamos outro fio conector (**figura 19**



Figura 18 - Vista do encaixe do elástico vermelho.



Figura 20 - Adicionando o fio conector ao motor da esteira.

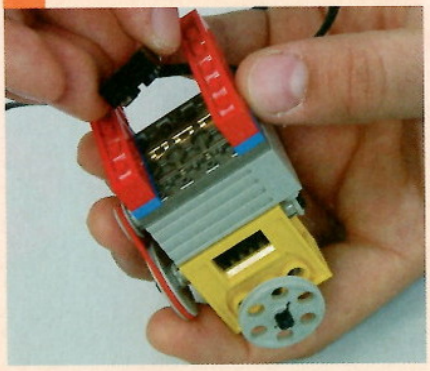


Figura 22 - Acoplando o motor e o redirecionador da esteira.

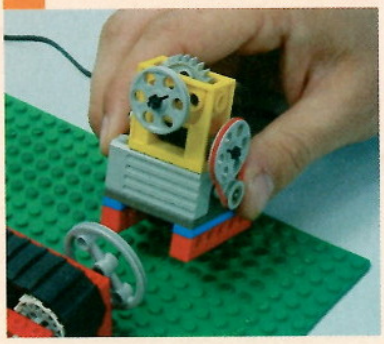


Figura 19 - Montando a base do motor da esteira.

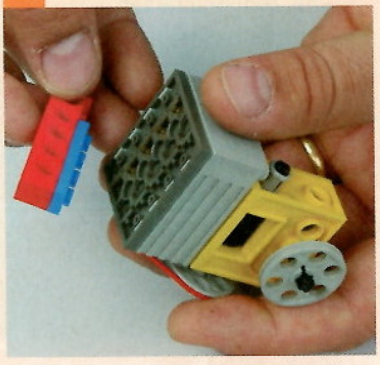
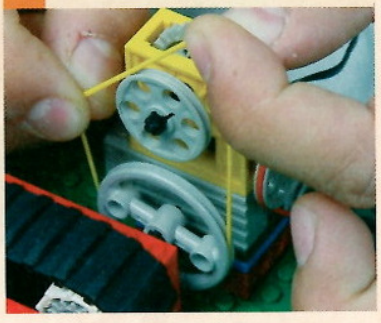


Figura 21 - Ligando o fio conector na saída A do RCX.



Figura 23 - Colocando o elástico amarelo que liga as roldanas do redirecionador e da esteira.



e **20**), cuja outra extremidade é conectada à saída A do RCX (**figura 21**). Concluída a base do motor da esteira, devemos encaixá-la na peça verde (conforme a **figura 22)** e colocar um elástico amarelo ligando a roldana do redirecionador à roldana conectada a esteira (**figura 23**).

Se quisermos enriquecer ainda mais este projeto poderemos acrescentar dois sensores de toque visualizados na **figura 24**, os quais podem interromper e/ou reiniciar o funcionamento da Estação.

Enfim, aqui termina a construção da nossa Estação de Reciclagem que, com mais alguns acessórios demonstrados na **figura 25** está pronta para propiciar aos alunos uma maneira inesquecível de se vivenciar e aprender os conteúdos envolvidos.

INTERDISCIPLINARIDADE

Utilizando uma forma dinâmica e abrangente do trabalho, não é tarefa difícil relacionar os projetos com os conteúdos das diversas disciplinas. Quando o aluno é um agente ativo no aprendizado, a fixação do conteúdo é maior e mais duradoura.O projeto de Robótica e Educação Tecnológica, mais do que um caráter interdisciplinar, tem um caráter transdisciplinar. Por exemplo neste caso do projeto da Estação de Reciclagem, sugerimos algumas aplicações e relações com outras disciplinas.

Em ***Geografia*** podemos fazer um estudo, por exemplo, sobre as condições de vida de uma população, principalmente a da periferia, que vive em condições precárias e até mesmo perto de lixões que estão cada vez mais volumosos e, para diminuir este impacto no meio ambiente, na vida desses indivíduos e alterações na natureza, pode-se abordar a aplicação do processo de reciclagem, minimizando o crescimento da quantidade de lixo próximo à casa dessas pessoas.

Em ***Matemática*** podemos trabalhar com cálculos de velocidade média de separação dos componentes, análise de tabelas e cálculos simples sobre o período de decomposição dos diferentes produtos e o porquê disso, função – distância percorrida e tempo gasto, inclusive com a utilização de gráficos mostrando a porcentagem dos produtos que são efetivamente reciclados e notificando o tempo que a Natureza demora para os degradar.

Em **História**, temos uma proposta que se baseia, em linhas gerais, na compreensão do passado para que possamos entender o mundo no qual vivemos hoje, e da necessidade que ele tem de buscar alternativas para um desenvolvimento sustentável e almejando ser capazes de levantar hi-

Figura 24 - Encaixando os sensores de toque.

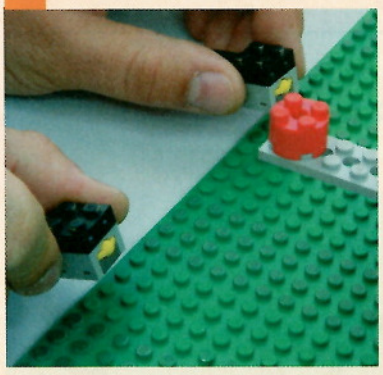
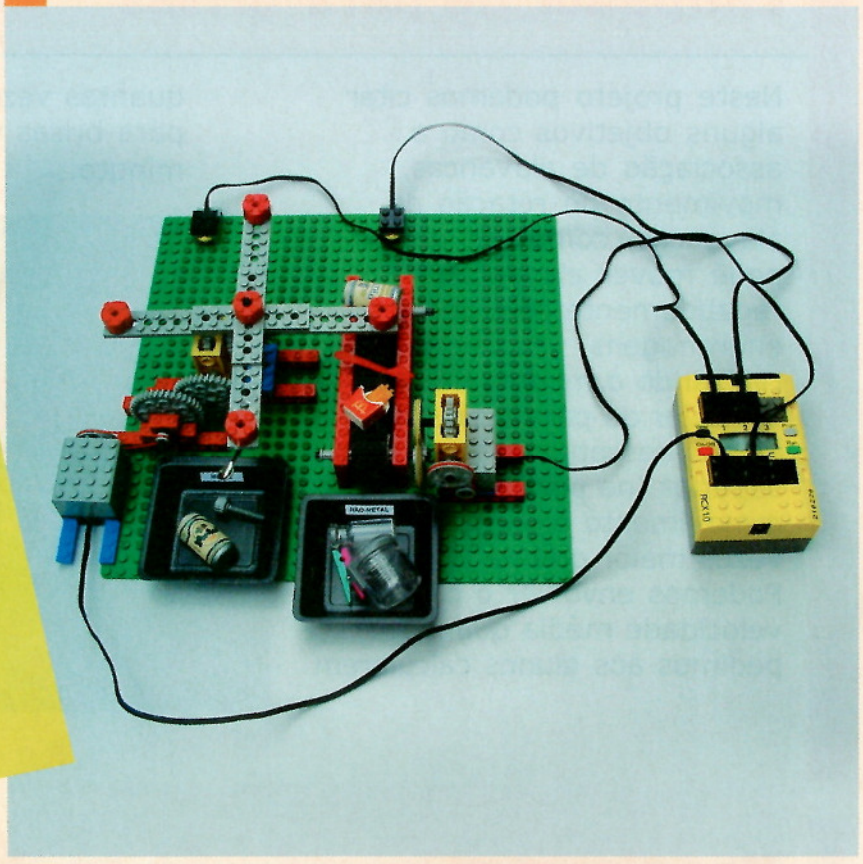


Figura 25 – Visão da Estação de Reciclagem finalizada.



Sites da Internet

Escola Bosque - www.escola-bosque.com.br

EDAcom Tecnologia - www.edacom.com.br

póteses futuras do desenvolvimento da humanidade. Num projeto como o da Estação de Reciclagem, podemos analisar o processo tecnológico empregado nos séculos 19, 20 e 21, concentrando nosso objetivo no aperfeiçoamento de novas técnicas em aproveitar melhor os recursos naturais retirados de nosso planeta e o surgimento da era da informática que pode em muito facilitar este processo de reciclagem.

Em ***Inglês*** podemos ampliar o vocabulário dos alunos utilizando algumas partes do software Robolab que, apesar de estar em português, tem palavras em inglês e apresenta um vídeo também em inglês. Compreensão e discussão do vídeo. Podemos também associar palavras relacionadas com esse tema com a língua inglesa.

Em ***Artes,*** a construção de maquetes, a expressão do mundo real e idéias para o futuro sem (e com) a reciclagem.

Em ***Ciências*** podemos abordar vários conteúdos como: Educação Ambiental (importância do conhecimento das diferenças de produtos como, por exemplo, os produtos que não são biodegradáveis), saúde e qualidade de vida (uma vida melhor, livre de poluentes e doenças relacionadas com o lixo e, conseqüentemente, com o solo), estudo da anatomia humana, dando ênfase aos malefícios das doenças adquiridas com a exposição ao lixo, prejudicando o bom funcionamento do corpo humano, estudo dos tipos de reciclagem existentes e sua utilização de forma indispensável ao homem, entre outros.

Os principais conceitos de ***Ciências e Tecnologia*** são: discriminação de atributos, seguir critérios, distinguir características e propriedades de diversos objetos e materiais, planejamento, construção e avaliação, aprendizado cooperativo e preservação do meio ambiente.

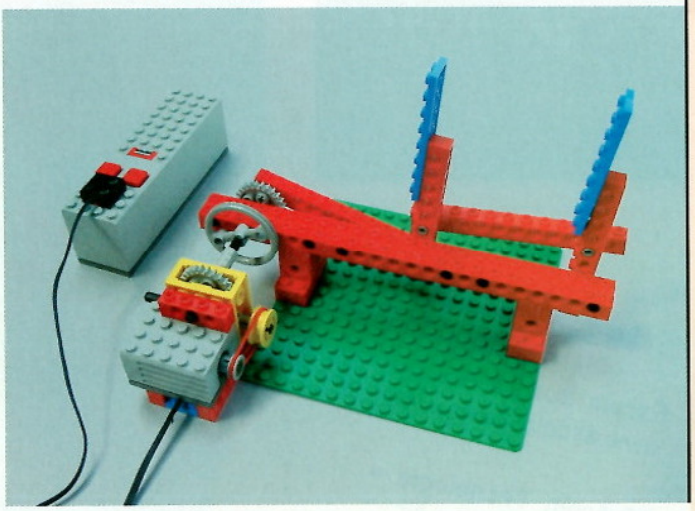
Na ***Física*** estudaremos princípios de magnetismo com relação aos ímãs utilizados na esteira, estudar a eletrostática e a eletrodinâmica, não esquecendo de se aprofundar no mecanismo de funcionamento das pilhas utilizadas no projeto, onde podemos abordar também o estudo da ***Química***.

Em ***Português,*** trabalhar a elaboração de redações, jornais e adoção de livros paradidáticos pertinentes aos assuntos e analisar textos referentes aos conteúdos aprendidos.

Esses são apenas alguns exemplos. As possibilidades de interdisciplinaridade com relação aos projetos são inumeráveis. É uma excelente oportunidade para que os professores se atualizem, pesquisando e se aprofundando em conteúdos que estimulam a criatividade do educando e do educador.

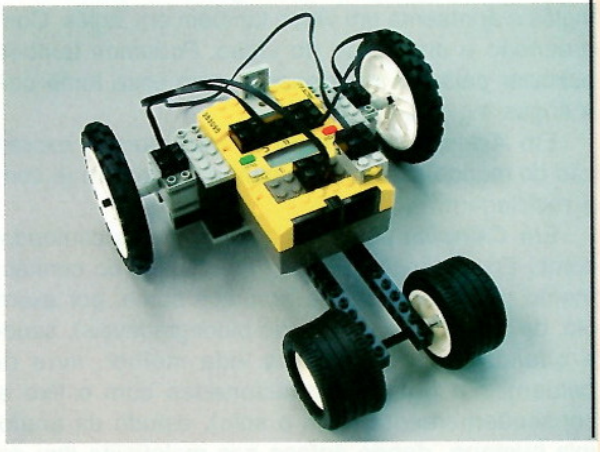
Limpador de pára-brisas

Neste projeto podemos citar alguns objetivos como a associação de alavancas, movimento de rotação de alavancas, como um motor pode mover alavancas repetidamente com polias e engrenagens.
Utilizando o modelo de um limpador de pára-brisas, os alunos descobrirão que na parte superior (na ponta do limpador) o movimento é entre 3 a 4 vezes maior que na base.
Podemos envolver o cálculo de velocidade média quando pedimos aos alunos calcularem quantas vezes os limpadores de pára-brisas vão e voltam em um minuto.



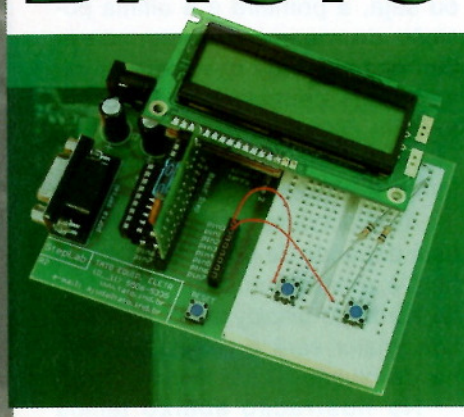
Modelo para introdução ao Robolab

Para que os alunos possam aprender a operar com mais facilidade o software Robolab, elaboramos um modelo bem simples. Com ele é que iniciamos a implantação da Robótica em nossa Escola.
A confecção deste projeto é feita com a utilização de dois motores que giram os eixos traseiros, que são independentes. É um modelo perfeito para a assimilação das primeiras programações no Robolab que começam nas opções Pilot 1, Pilot 2 e assim sucessivamente, onde o aluno aprende a fazer o carrinho, no caso, andar para frente e/ou para trás por tantos segundos, acender uma ou mais lâmpadas de freio, por exemplo, e parar após pressionar ou despressionar o sensor de toque.
É importante ressaltar que o carrinho é apenas um exemplo de projeto, pois os alunos terão muitos outros projetos em que utilizarão outras formas de robô e não somente a forma de um carro motorizado.



BASIC STEP

4ª Parte



Octávio Nogueira

Nos três primeiros artigos desta série, vimos os comandos básicos do microcontrolador BASIC Step 1 e como controlar alguns dispositivos. Nesta quarta parte, iremos ver algo mais sofisticado...

Como dissemos no primeiro artigo, os microcontroladores são usados basicamente para controlar máquinas e dispositivos. Nós já aprendemos como controlar diodos emissores de luz, servomotores e a ler chaves e botões. Apesar de serem empregados para controlar máquinas, estas geralmente têm um operador ou supervisor humano que precisa saber o que está acontecendo com a máquina ou em que fase de operação ela está.

O microcontrolador dispõe de diversos meios para informar ao operador o que está ocorrendo, e os indicadores mais básicos são luzes de advertência e *status*. Estas indicações nem sempre são muito informativas, por exemplo: quando vemos uma luz vermelha piscando, sabemos que algo não vai bem, mas não sabemos o que.

Um dos modos mais eficientes de informar o operador é através de um *display* de cristal líquido (LCD), pois este dispositivo permite ao microcontrolador informar exatamente o que está acontecendo. Na **figura 1** vemos uma foto de um *display* LCD de 16 caracteres e 2 linhas. Esse tipo de interface chama-se "interface homem-máquina" (IHM).

Os *displays LCD* são normalmente ligados aos microcontroladores através de uma interface paralela. Um dos problemas desse esquema de ligação é que necessitamos de um grande número de pinos de entrada e saída para controlar o *display*. Geralmente, nós desejamos usar estes pinos para controlar uma máquina, então, quanto menos o *display* precisar, melhor.

Uma outra forma de ligação é por meio de uma interface serial. A grande vantagem é que com este tipo de ligação, utilizamos apenas um pino do BASIC Step 1.

COMUNICAÇÃO PARALELA VERSUS SERIAL

Uma porta de comunicação serial envia e recebe 8 bits de dados de cada vez, utilizando 8 fios. Isso faz com que os dados sejam transmitidos muito rapidamente, entretanto, o cabo necessário para a transmissão é mais grosso e pesado pois ele precisa ter muitos fios. Nós encontramos as portas paralelas nos computadores PC e elas são utilizadas para controlar uma impressora, e são raramente empregadas para outra coisa.

Uma porta serial envia um bit de dado de cada vez, através de um único fio. Apesar de levar um tempo 8 vezes maior para transferir um *byte* de dados, ela necessita somente de uns poucos fios, na verdade apenas 2, se a comunicação for unidirecional (dados e terra).

Figura 1 - Display de cristal líquido.



Como temos apenas um fio transmitindo os dados, devemos usar algum tipo de protocolo para que o transmissor e o receptor tenham condições de saber onde o *byte* começa e onde termina. O tempo de transmissão de cada bit tem que ser muito preciso e deve ser o mesmo para o transmissor e o receptor.

A transmissão é iniciada com um bit de início, ele não faz parte do dado e serve apenas para informar ao receptor que iremos transmitir um dado. Após o bit de início, são transmitidos 8 bits de dados e mais um bit que é chamado bit de parada, que serve para indicar ao receptor que já transmitimos todos os bits desse *byte*.

O *display* que iremos utilizar possui duas linhas de 16 caracteres cada, esses caracteres podem ser letras maiúsculas ou minúsculas, números ou símbolos especiais criados por nós (ou não).

Por ser um *display* inteligente ele possui também um microcontrolador e, além dos caracteres para mostrar, aceita ainda diversos comandos para alterar o seu modo de operação.

O comando SEROUT

Nós já vimos que o display utiliza comunicação serial, mas como fazemos para que o BASIC Step 1 envie dados serialmente? Com o comando SEROUT, que é utilizado da seguinte forma:

SEROUT pino, velocidade, (dado)

Pino - é o pino do BASIC Step 1 que queremos usar para a comunicação serial; qualquer um dos 8 pinos pode ser utilizado e podemos ter ainda mais de uma saída serial.

Velocidade - indica ao BASIC Step 1 o tempo de transmissão de cada bit; o valor que o *display* aceita é de 2400 bits por segundo, então, devemos usar para este parâmetro o valor **N2400**.

Dado - é o que queremos transmitir.

Se desejássemos transmitir a palavra BASIC pelo pino 0 do BASIC Step 1 para o *display* serial, usaríamos o comando SEROUT 0,N2400,("BASIC"). É bem fácil, não?

Como dissemos antes, o *display* aceita diversos comandos de controle além dos dados, mas para que servem esses comandos? Os comandos mais utilizados são para apagar todo o display, ligar e desligar o cursor e posicionar o cursor na primeira ou segunda linha ou em algum posição da linha.

Nós temos que ter algum modo de dizer ao *display* que queremos enviar comandos e não dados, pois de outro modo ele não teria como distinguir entre ambos. O modo que fazemos isso é enviando um código especial que diz que após o mesmo, o próximo dado é um comando. Esse código é o número 254.

Por exemplo, o comando para apagar o *display* é "1", mas, se enviarmos apenas o dado "1", o *display* vai tentar mostrá-lo e não o interpretará como comando. Para apagar o *display* devemos enviar o comando do seguinte modo:

SEROUT 0,N2400,(254,1)

Deste modo, estamos enviando primeiro o código 254 que diz ao *display* que o próximo dado é um comando e, então, enviamos o código 1.

Os principais códigos de comando são:

```
- apaga o display  1
- posiciona o cursor na primeira linha     128
- posiciona o cursor na segunda linha 192.
```

PROJETO

Neste artigo vamos utilizar um *display LCD* para criar um jogo simples, similar ao tênis, para demonstrar os recursos do display e do BASIC Step 1.

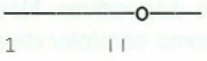
Para construir este projeto você irá precisar de:

1 - BASIC Step 1
1 – Placa StepLab
1 – Display LCD serial
2 – Resistores de 10 kohms
2 – Chaves de contato momentâneo
1 – Fonte de alimentação de 9 V
1 – Cabo de gravação
1 – Compilador BASIC Step 1.

O princípio de funcionamento do jogo é bastante simples: o *display* é dividido em dois campos, o da direita e o da esquerda, cada jogador fica com um campo, o objetivo é rebater a bola impedindo que ela atinja a borda do *display*, ou seja, a primeira e a última posição do mesmo. Na segunda linha ficam a rede que marca a divisão dos campos e o placar de pontos. O jogo termina assim que qualquer um dos participantes atinja 9 pontos.

A velocidade de devolução da bola depende da posição em que a bola é rebatida, quanto mais perto da borda do *display* a bola for rebatida, maior será a velocidade dela na devolução.

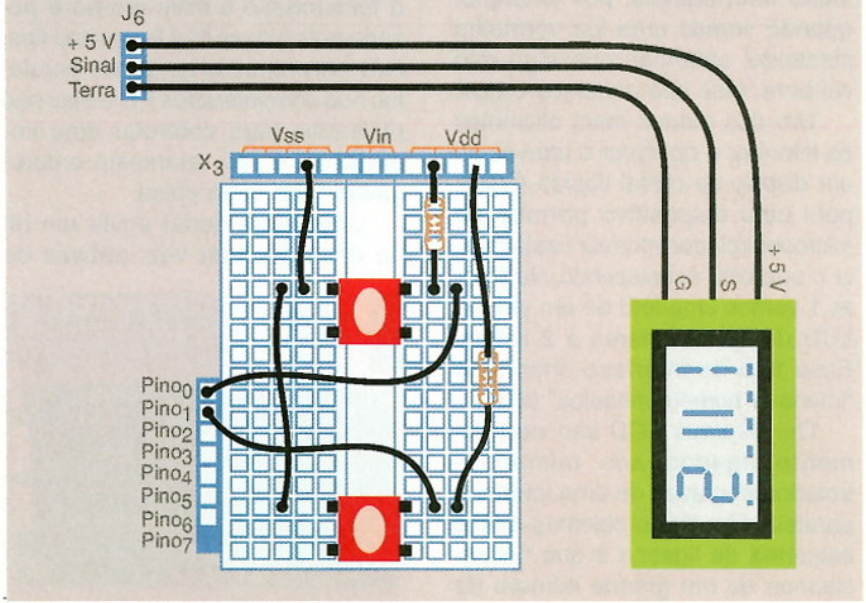
Na **ilustração abaixo** temos uma representação de como o *display* fica durante o jogo, aqui o jogador 1 tem um ponto e o jogador 2 tem quatro pontos.



Este programa, ao contrário dos outros, é bem complexo e utiliza praticamente toda a memória do BASIC Step 1 e nos dá uma idéia de suas possibilidades.

Monte o circuito conforme mostra a **figura 2**. Como podemos ver, o circuito é bem simples pois toda a sofisticação está no programa do BASIC Step 1.

Figura 2 – Montagem do circuito.



Listagem do programa

```
symbol cmd=254                ' comando
para o display
symbol clear=1                ' comando
limpa
symbol n1=192                 ' comando
Segunda linha
symbol pos=b0                 ' renomeação
da variáveis
symbol tempo=b3
symbol pontos1=b5
symbol pontos2=b6
symbol botao1=pin0
symbol botao2=pin1

goto config                   ' o programa
configura diversos parâmetros

vai0:                         '
movimentação de um jogador
   tempo=200
   lookup pos,(150,30,75,100),tempo    '
vê onde a bola foi rebatida
vai:
   b1=128+pos
   serout 6,N2400,(cmd,b1)          '
escreve
   serout 6,n2400,(0)               '     a
bola no display
   for b2=1 to tempo
      if botao2=0 and pos>8 then volta0 '
testa se o botão foi apertado
   next
   serout 6,N2400,(cmd,b1)
   serout 6,n2400,($B0)
   pos=pos+1
if pos<15 then vai
goto marca1

volta0:                       '
movimentação do outro jogador
b4=pos-12
tempo=200
lookup b4,(100,75,30),tempo
volta:
   b1=128+pos
   serout 6,N2400,(cmd,b1)
   serout 6,N2400,(0)
   for b2=1 to tempo
      if botao1=0 and pos<7 then vai0
   next
   serout 6,N2400,(cmd,b1)
   serout 6,N2400,($B0)
   pos=pos-1
if pos>0 then volta
goto marca2

marca1:                       ' marca ponto
do jogador 1
tempo=150
pontos1=pontos1+1
serout 6,n2400,(cmd,$C0)
serout 6,n2400,(#pontos1)
if pontos1<9 then volta
serout 6,n2400,(cmd,$C0)
goto ganhou

marca2:                       ' marca ponto
do jogador 2
tempo=150
pontos2=pontos2+1
serout 6,n2400,(cmd,$CF)
serout 6,n2400,(#pontos2)
if pontos2<9 then vai
serout 6,n2400,(cmd,$CA)

ganhou:
serout 6,n2400,("Ganhou")
espera:
if botao2=1 and botao1=1 then espera
goto inicio

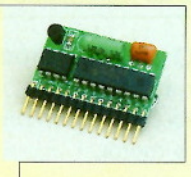
config:                       '    define    um
caracter novo e inicializa as variáveis
pause 500
serout 6,n2400,(cmd,$40)  'CG RAM
serout                    6,n2400,($00,
$0E,$1F,$11,$1F,$0E,$00,$00)
inicio:
serout 6, N2400, (cmd, clear)
for b0=1 to 16
   serout 6,n2400,($B0)
next
serout 6,N2400,(cmd,$C7)
serout 6,N2400,($B4,$B4)
pos=0
pontos1=0
pontos2=0
tempo=150
goto vai
```


TATO
Equipamentos Eletrônicos
Transformando idéias em realidade

Siga a série dos microcontroladores BASIC Step 1 e faça todas as experiências na prática com o material fornecido pela Tato Equipamentos Eletrônicos.

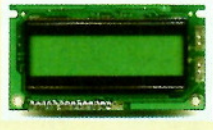
O kit inclui:

BASIC Step 1
Placa StepLab
Fonte de alimentação
Cabo de gravação
Disquete com o compilador



Tudo isto por apenas R$ 120,00.

Display serial



Apenas R$ 40,00.

Rua Ipurinas, 164 - São Paulo - SP - 04561-050
Tel (11) 5506-5335 - http://www.tato.ind.br

Robô Mecha-Medusa MV - 949

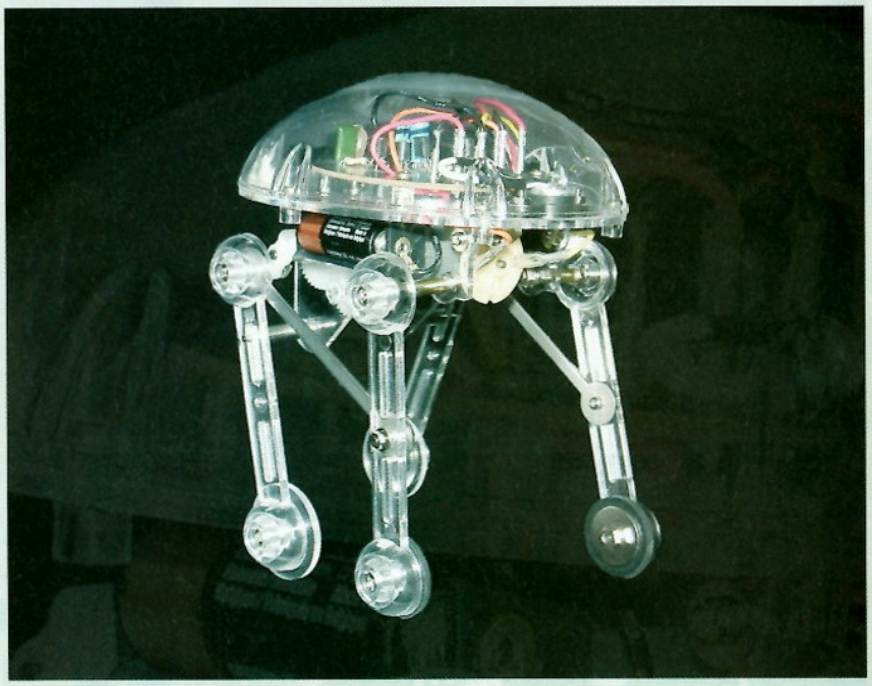
O que é Mecha-Medusa?
É um robô com quatro pernas e um sensor sonoro.

O modo de movimentar as quatro pernas se assemelha ao *Snow Walker* do filme "Star Wars".

Seu nome Mecha-Medusa está associado ao seu formato e transparência (**veja ao lado**).

Especificações:

- **Tensão:** 1,5 V (baterias "N" x2 – não incluídas)
- **Consumo:** 200 mA (com o motor operando)
- **Altura:** 132 mm (aproximadamente)
- **Diâmetro:** 102 mm (aproximadamente)
- **Peso:** 160 g (aproximadamente)
- **Tempo de caminhada:** 4 segundos (aproximadamente).



CIRCUITO ELETRÔNICO

O Mecha-Medusa possui um circuito eletrônico como cérebro e o controle do movimento é através do som.

O circuito eletrônico é formado por transistores, resistores, capacitores e microfone, que controlam o motor.

Amplificador de áudio e circuito de disparo monoestável

Os principais componentes neste bloco são TR_1 e TR_2. Ele amplifica os sinais sonoros captados pelo microfone de eletreto e fornece na saída um sinal por um certo tempo, voltando depois ao nível inicial. Em outras palavras, ele controla o tempo de movimento do motor (**figura 1**).

O tempo de controle ou movimento (tempo em que o Mecha-Medusa se move ao receber um comando sonoro) é da ordem de 3 a 4 segundos.

Circuito de acionamento do motor

O circuito de acionamento do motor amplifica a corrente o suficiente para que ela possa alimentar o motor. Isso é feito através de TR_3 e TR_4.

Quando a corrente é cortada no motor, a corrente em TR_3 e TR_4 também é desligada para economizar energia.

MECÂNICA

O robô Mecha-Medusa possui apenas um motor. O movimento é transmitido para as pernas através de um sistema de alavanca. Essas, por sua vez, são acionadas por duas "manivelas" (alocadas no eixo principal).

A redução de velocidade é feita por um pinhão e três engrenagens, vide **figura 2**.

MONTAGEM

O ponto crítico a ser observado na montagem das partes mecânicas é quanto ao "aperto" das porcas e parafusos. Nas "junções" que estão sujeitas a movimento, esses elementos não podem ser "apertados" totalmente.

Caso isso ocorra, a unidade perderá a capacidade de movimento.

Montando o robô:

1° passo: Instalando o motor e chave ao painel frontal (**figuras 3** a **8**).

2° passo: Instale os suportes das pilhas, o eixo da engrenagem e a alavanca no painel B (**figuras 9** a **13**).

* Cuidado! Há um erro no diagrama original. Consulte a errata anexa a ele.

3° passo: Interligue os painéis A e B (**figura 14**).

4° passo: Montagem das pernas (**figuras 15** a **18**).

5° passo: Monte as pernas traseiras, e una o painel A ao painel B (**figura 19**).

6° passo: Instale as pernas dianteiras no painel (**figuras 20** a **23**).

7° passo: Instale o painel de junção (**figura 24**).

8° passo: Instale as engrenagens da peça B (**figuras 25** e **26**).

9° passo: Colocando os painéis A e B no painel frontal (**figura 27**).

10° passo: Conectando os fios à placa de circuito impresso (**figura 28**).

11° passo: Teste.

Antes de colocar a cúpula, teste o robô inserindo as pilhas e ligando a chave. A unidade deve dar 4 passos cada vez que um "estalar" de dedos ocorrer próximo ao microfone. Caso isso não ocorra, ajuste a sensibilidade através de "trim-pot" (**figura 29**).

12° passo: Finalmente, instale a cúpula. (**figuras 30** a **31**).

A montagem desse *kit* é muito interessante, e é bastante útil para treinar as habilidades mecânicas do montador.

Até a próxima!

Figura 1 - Circuito elétrico do robô.

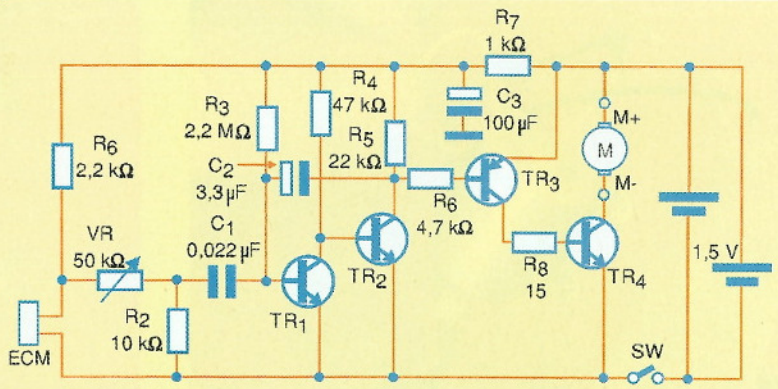


Figura 2 - Mecânica do Mecha-Medusa.

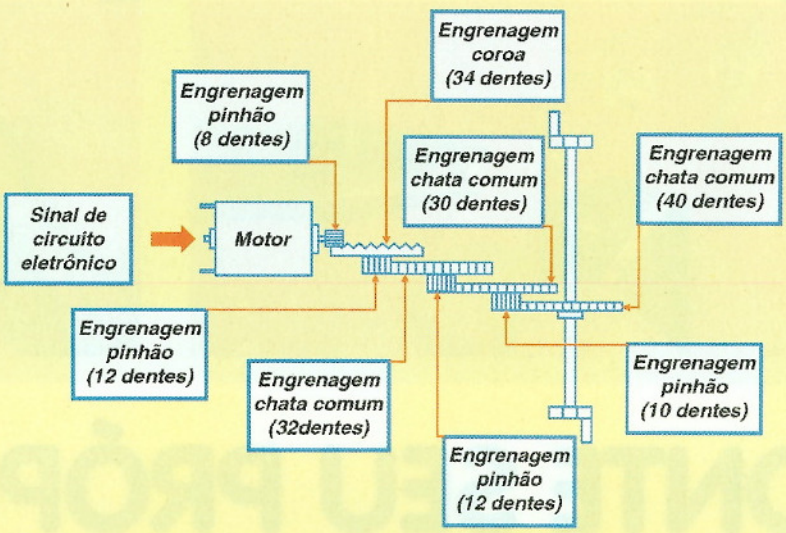


Figura 3 - Encaixe da engrenagem no motor.

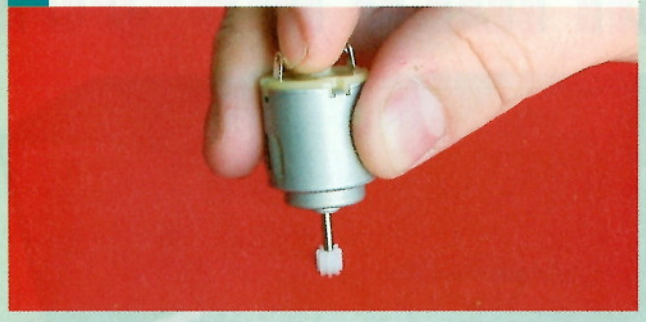


Figura 4 - Suporte do motor.



Figura 5 - Instalação do motor na base.

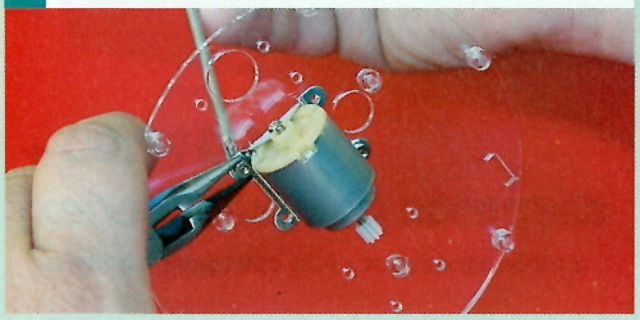


Figura 6 - Fixação da chave L/D na base.

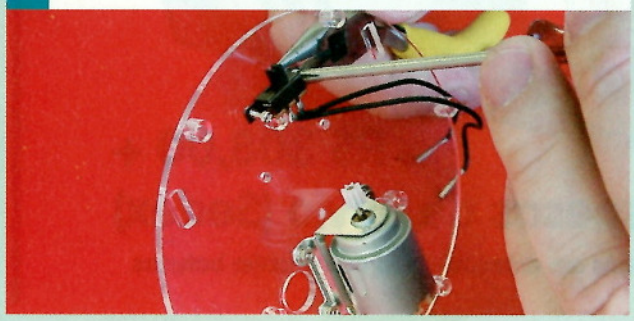


Figura 7 - Passagens dos fios pelos furos da base.

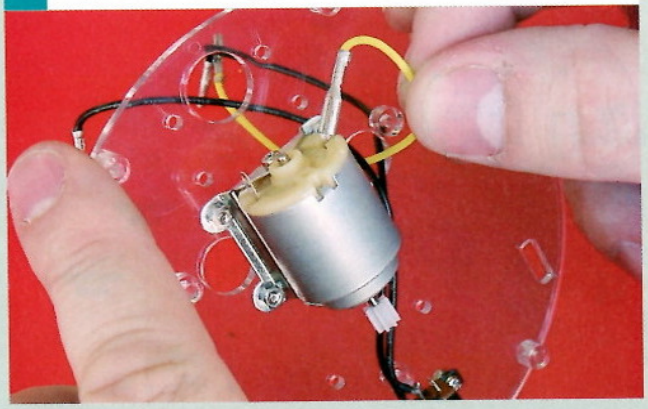


Figura 8 - Base principal pronta.

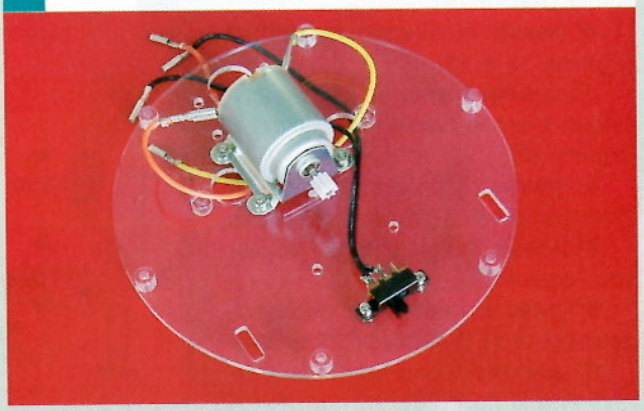


Figura 9 - Instalação do suporte de pilhas no painel A.

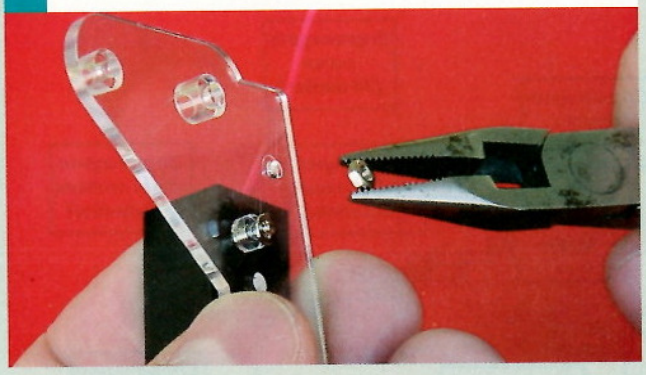


Figura 10 - Instalação do suporte de pilhas no painel B.



MONTE SEU PRÓPRIO ROBÔ

MECHA-MEDUSA MV-949



R$ 308,00 +

despesas de envio (Sedex)

27 robôs em estoque - Não acompanha baterias

SPIDER MV-962



R$ 397,00 +

despesas de envio (Sedex)

5 robôs em estoque - Não acompanha baterias

Saber Marketing Direto Ltda. Fone: 11 69428055 - E-mail:vendas@editorasaber.com.br

Figura 11 - Painel A pronto.

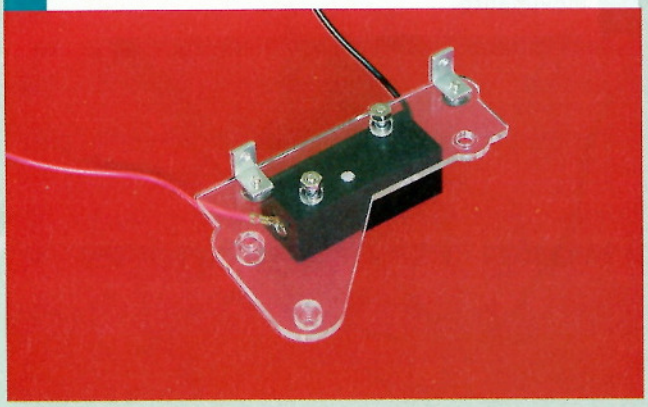


Figura 12 - Painel B pronto.

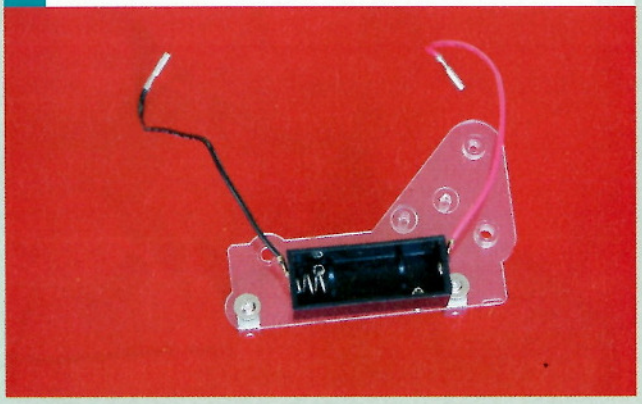


Figura 13 - Instalação do eixo principal.

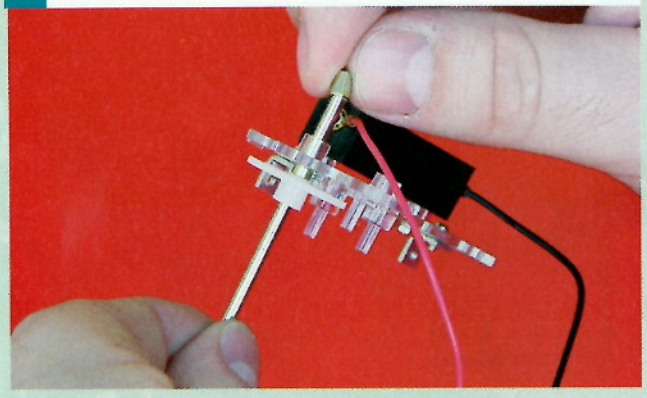


Figura 14 - Instalação da manivela do eixo.

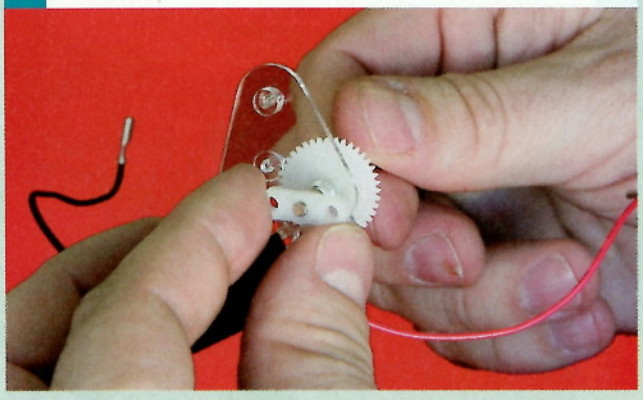


Figura 15 - Montagem das pernas dianteiras.

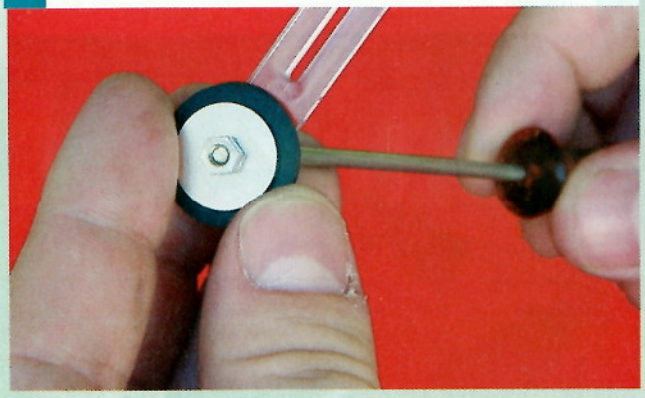


Figura 16 - Pernas dianteiras prontas.



Figura 17 - Montagem das pernas traseiras.

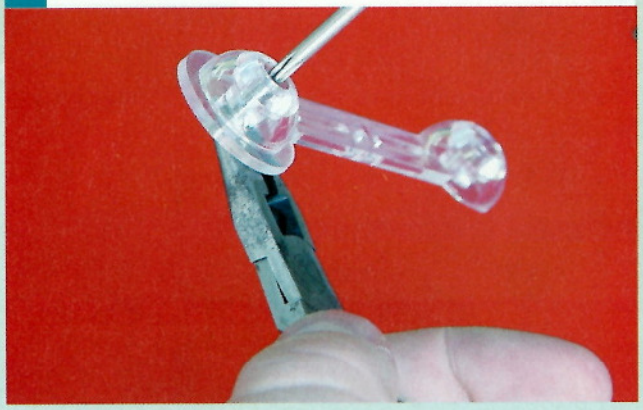


Figura 18 - Pernas traseiras prontas.

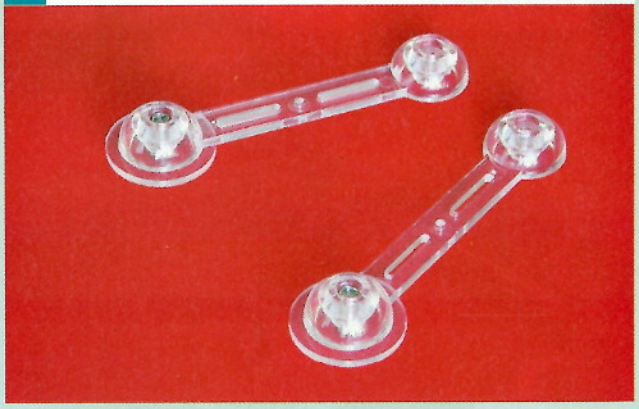


Figura 19 - União dos painéis A e B.

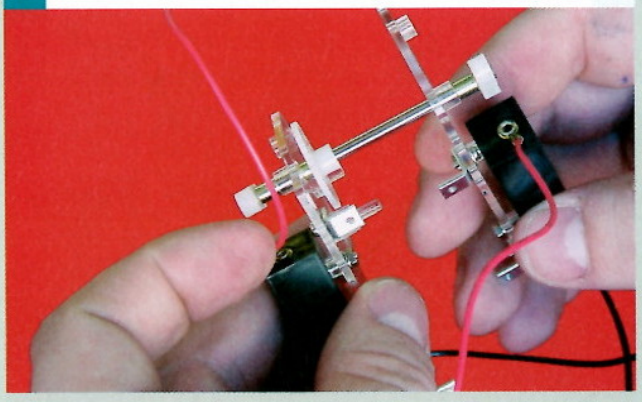


Figura 20 - Fixação da transmissão de movimento.

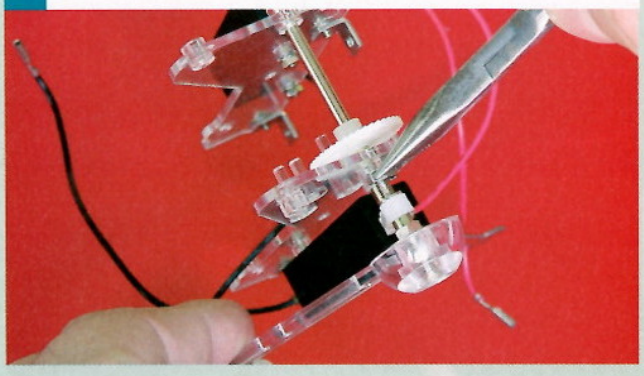


Figura 21 -Fixação da perna direita.

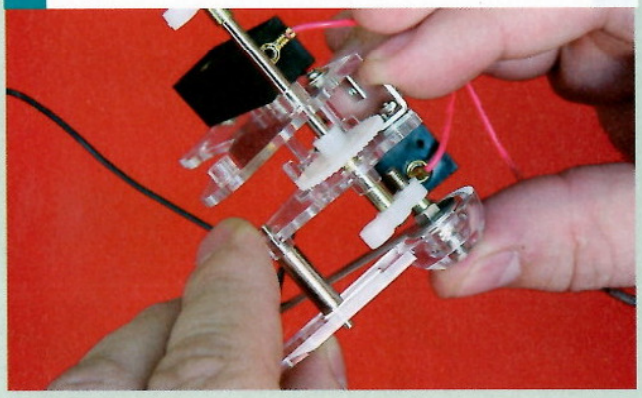


Figura 22 - Fixação da perna esquerda.

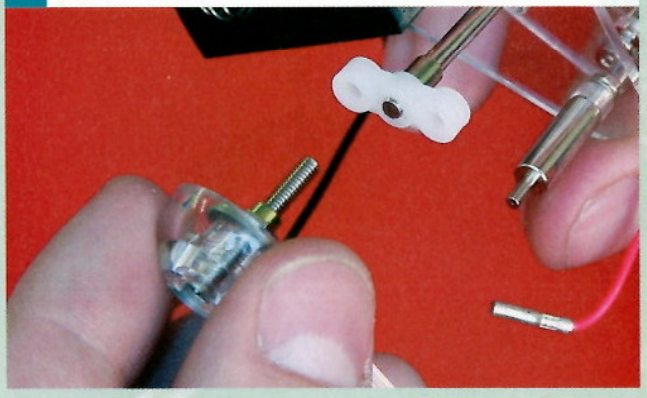


Figura 23 - Instalação dos suportes das pernas.

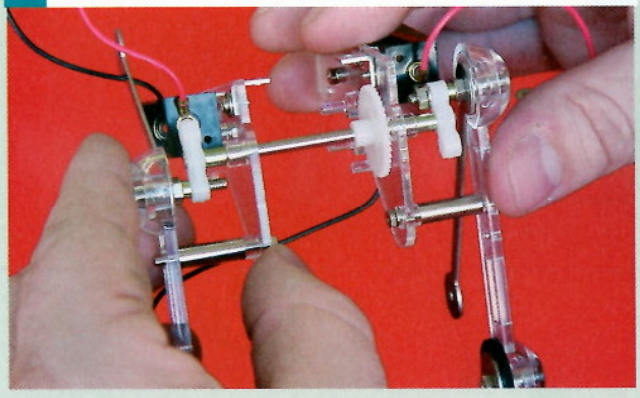


Figura 24 - Interligação das pernas do sistema "motor".

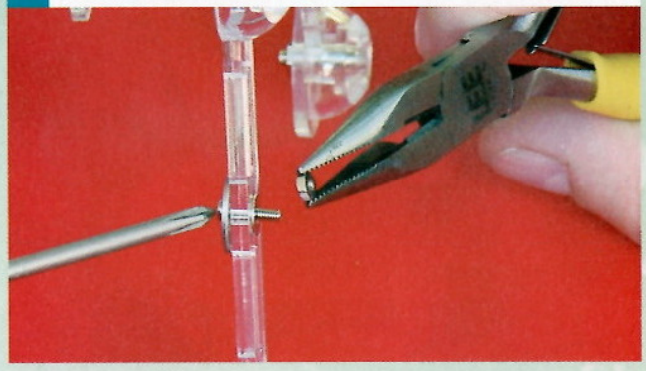


Figura 25 - Colocação de graxa nos eixos das engrenagens.

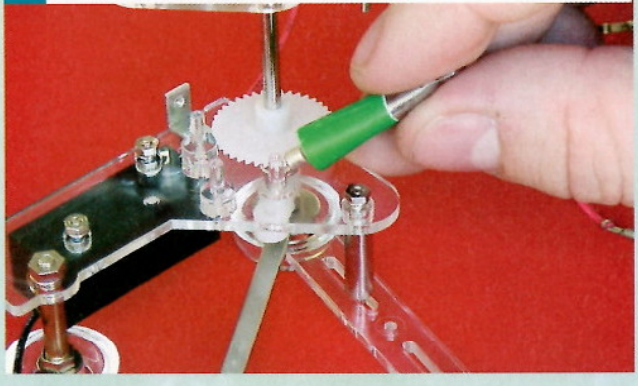


Figura 26 - Instalação das engrenagens.

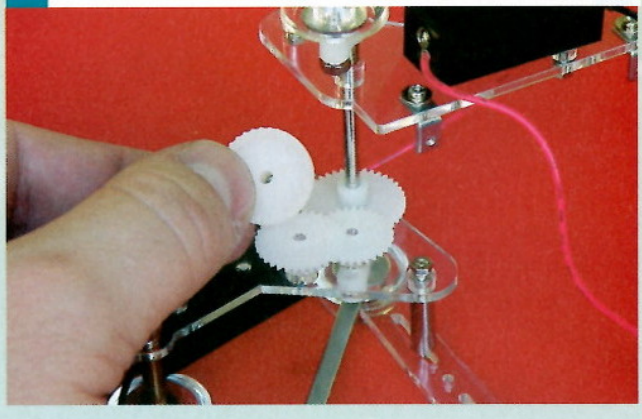


Figura 27 - Instalação dos painéis da base.

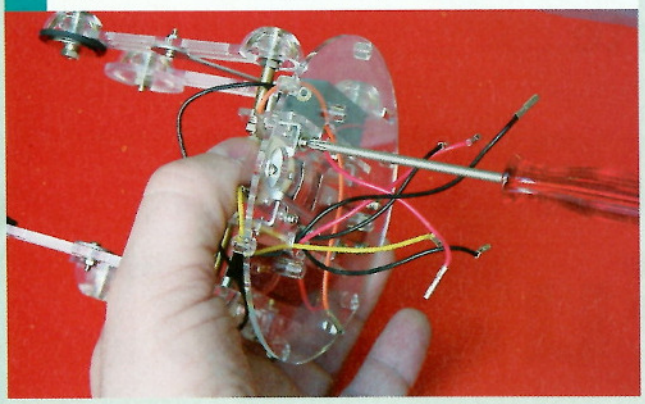


Figura 28 -Ligação dos fios à placa de circuito.

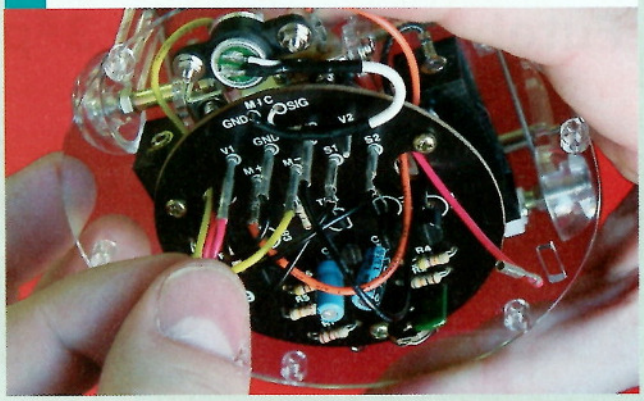


Figura 29 - Ajuste de sensibilidade (somente se necessário).

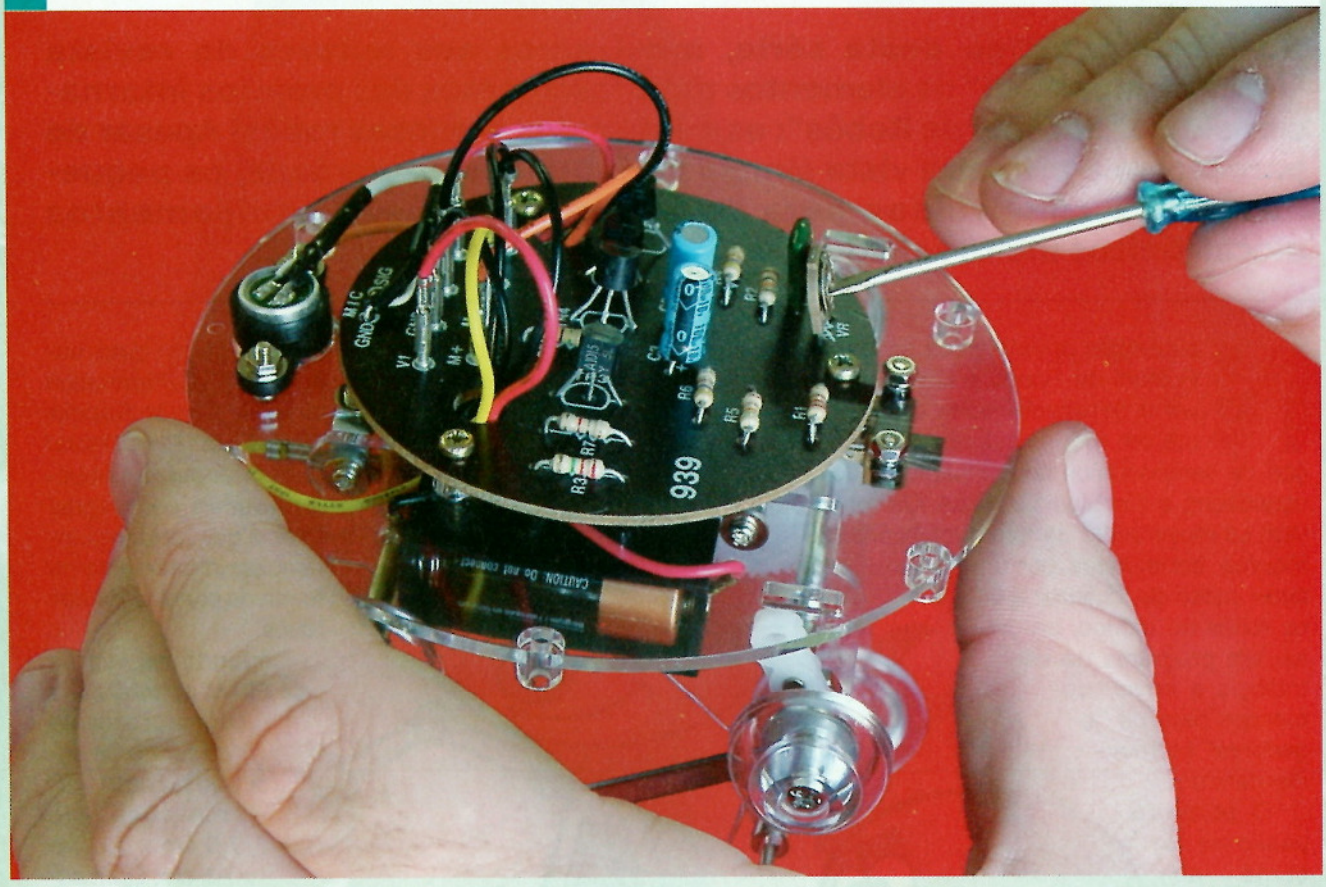


Figura 30 -Fixação da cúpula.

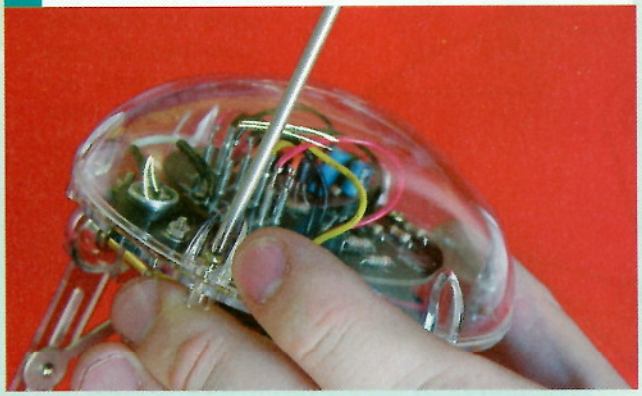


Figura 31 - Limpeza da unidade.



Futebol de Robôs - 3ªparte

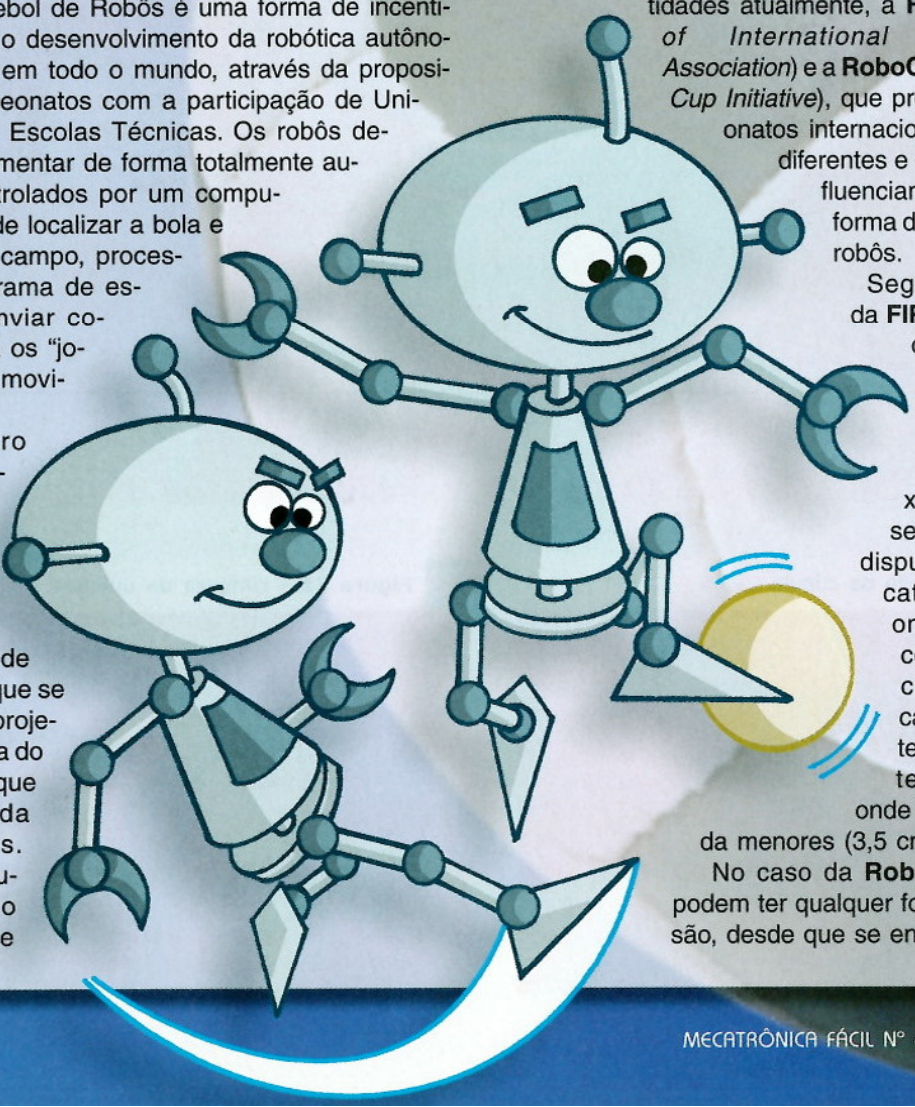
Os Robôs que Jogam Futebol

Ewaldo Luiz de Mattos Mehl

No terceiro artigo desta série, mostramos aos leitores da revista "Mecatrônica Fácil" os elementos mais fascinantes dos projetos internacionais de "Futebol de Robôs": os minúsculos e velozes robôs! Apesar de parecerem à primeira vista simples brinquedos, na verdade, os pequenos robôs construídos para os campeonatos são equipamentos mecatrônicos muito sofisticados.

Nos artigos anteriores da série, vimos que o Futebol de Robôs é uma forma de incentivar o desenvolvimento da robótica autônoma em todo o mundo, através da proposição de campeonatos com a participação de Universidades e Escolas Técnicas. Os robôs devem se movimentar de forma totalmente autônoma, controlados por um computador capaz de localizar a bola e os robôs no campo, processar um programa de estratégia e enviar comandos para os "jogadores" se movimentarem.

O primeiro passo no tocante ao projeto e construção dos robôs, após o estabelecimento de uma equipe que se dedique ao projeto, é a escolha do tipo de regra que será utilizada nas disputas. Devido ao sucesso que o Futebol de Robôs alcançou em todo o mundo, existem duas entidades atualmente, a **FIRA** (*Federation of International Robot-Soccer Association*) e a **RoboCup** (*Robot World Cup Initiative*), que promovem campeonatos internacionais com regras diferentes e que, portanto, influenciam diretamente na forma de construção dos robôs.



Segundo as regras da **FIRA**, os robôs são obrigatoriamente na forma de cubos, com dimensões de 7,5 cm x 7,5 cm x 7,5 cm. Com esses robôs pode-se disputar os jogos da categoria *Mirosot*, onde há disputas com três ou com cinco robôs em cada time. A **FIRA** tem também a categoria *Narosot*, onde os robôs são ainda menores (3,5 cm de aresta).

No caso da **RoboCup**, os robôs podem ter qualquer formato ou dimensão, desde que se encaixem num "ga-

barito" na forma de um cilindro com 18 cm de diâmetro e 25 cm de altura. Com esses robôs, disputam-se os jogos da categoria *small*, com cinco robôs em cada equipe. Há também na **RoboCup** outras categorias, com robôs maiores e até mesmo jogos com robôs dotados de pernas.

Aparentemente, à primeira vista, as regras da **RoboCup** podem parecer melhores para as equipes que estão iniciando seus projetos. Essas regras dão a liberdade de se projetar os robôs com formas diferentes e com dimensões maiores que os da **FIRA**, o que, em princípio, é uma vantagem. No entanto, o "campo" da **RoboCup** tem 152,5 cm x 274 cm, enquanto que a **FIRA** adota "campos" de 1,30 m x1,50 m (nos jogos "3 contra 3") e 2,20 m x 1,80 m (para os jogos "5 contra 5"). Apesar de parecer indiferente, os campos menores da **FIRA** facilitam muito a tarefa da aquisição das imagens: para o computador "ver" o campo da **RoboCup** com boa qualidade, é necessário dispor de uma câmera muito boa (e muito cara!). A decisão final terá que ponderar todos esses fatores, mas pelo menos no nosso caso pensamos que a adoção das regras da **FIRA** é mais conveniente para as equipes que estão surgindo no Brasil, mesmo com o desafio de se construir robôs relativamente menores que a **RoboCup**.

COMPONENTES DOS ROBÔS

No estágio atual da tecnologia, as equipes internacionais de Futebol de Robôs estão optando pelo chamado *enfoque descentralizado* para a montagem do sistema. Isso significa simplesmente que os robôs são entidades "cegas" ou "burras", ou seja, toda a "inteligência" do sistema reside no computador. O computador de cada equipe é encarregado de receber a imagem da câmera, processar essa informação e comandar os robôs a se movimentarem pelo "campo" em função dos resultados do programa de estratégia. Nada impede que se opte por instalar uma pequena câmera em cada robô, mas isto é algo bastante complicado sob o ponto de vista eletrônico. As regras da **FIRA** e da **RoboCup** sofrem periodicamente revisões, no sentido de introduzir elementos adicionais de dificuldade - prevê-se que futuramente será proibida a instalação da câmera externa ligada diretamente ao computador, devendo o sistema de visão ser deslocado para os robôs. Mas, pelo menos por enquanto, os robôs ainda podem ser completamente "cegos", o que na verdade facilita o projeto e a construção dos mesmos.

Com essas considerações iniciais, vamos dividir os robôs nos seguintes componentes:

- Motores e Sistema de Tração
- Receptor
- Microcontrolador e Circuito de acionamento dos Motores
- Baterias

MOTORES E SISTEMA DE TRAÇÃO

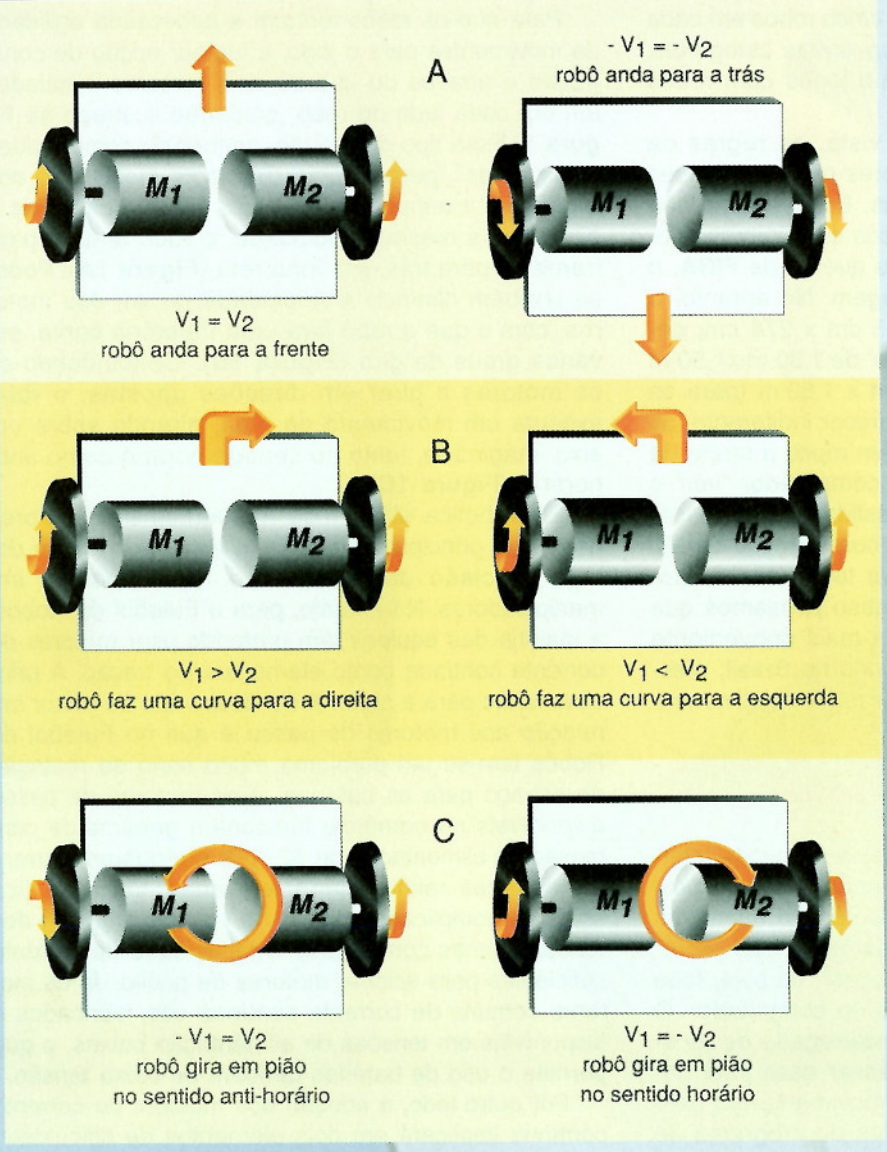
Para que os robôs tenham a necessária agilidade de movimentos para o jogo, a melhor opção de construção é através do uso de dois motores instalados um em cada lado do robô, conforme ilustrado na **Figura 1**. Esse tipo de arranjo, conhecido como "cadeira de rodas", permite uma excelente mobilidade dos robôs. Por exemplo, se as rodas forem colocadas a girar com a mesma velocidade, o robô andará para frente ou para trás, em linha reta (**Figura 1A**). Pode-se também diminuir a velocidade de um dos motores, com o que o robô fará uma trajetória curva, em vários graus de giro (**Figura 1B**). Comandando-se os motores a girar em direções opostas, o robô executa um movimento de pião, girando sobre um eixo imaginário, tanto no sentido horário como anti-horário (**Figura 1C**).

Em robótica utilizam-se freqüentemente motores de passo, principalmente nas aplicações onde se deseja precisão do movimento, como ocorre em manipuladores. No entanto, para o Futebol de Robôs, a maioria das equipes tem preferido usar motores de corrente contínua como elementos de tração. A principal razão para a preferência desse tipo de motor em relação aos motores de passo é que no Futebol de Robôs tem-se um problema muito sério de restrição de espaço para as baterias, e os motores de passo disponíveis no comércio funcionam geralmente com tensão de alimentação de 12 volts e requerem correntes elétricas relativamente elevadas. Ou seja, fica bastante complicado instalar no volume reduzido dos robôs, baterias com tensão e capacidade de corrente suficientes para acionar motores de passo. Já os motores comuns de corrente contínua são fabricados e disponíveis em tensões de alimentação baixas, o que permite o uso de baterias também de baixa tensão.

Por outro lado, a adoção dos motores de corrente contínua implicará em dois elementos de dificuldade para os projetistas dos robôs. O primeiro deles deve-se à necessidade de se ter velocidade variável em cada motor. Para se variar a velocidade em um motor de corrente contínua, pode-se optar por controlar a tensão de alimentação - é o que ocorre, por exemplo, nos carrinhos de *autorama*: o "acelerador" geralmente é simplesmente uma resistência variável, que permite controlar a tensão entregue na pista para o carrinho - quanto mais elevada for a tensão, mais rápido será o giro do motor. Esse tipo de controle, apesar de simples sob o ponto de vista de implementação eletrônica, é bem pouco eficiente porém sob o ponto de vista do torque obtido nos motores em baixa rotação.

Outra forma de controlar a velocidade de motores de corrente contínua é através da geração de pulsos de tensão fixa mas com larguras variáveis, técnica conhecida pela sigla PWM (*pulse width modulation*). Nesse caso, é necessário um circuito eletrônico um pouco mais sofisticado, que através de transistores

Figura 1 – Posicionamento dos motores no robô.



comandados por um microcontrolador, ligam e desligam rapidamente, estabelecendo um "trem de pulsos" que controla a velocidade do motor. Apesar de ser um circuito eletrônico mais complexo, o controle de velocidade por meio de PWM é uma alternativa muito interessante para sistemas onde existe um microcontrolador digital, como é o caso dos robôs para o projeto "Futebol de Robôs".

Outra necessidade, decorrente da adoção de motores de corrente contínua, é a implementação de um sistema de realimentação. Ao se comandar cada motor a girar, o sistema deve verificar o real deslocamento dos robôs por meio de algum tipo de sensor. A maneira geralmente adotada para isso é através da instalação de *encoders* nos motores ou nas rodas. Os *encoders* foram objeto de um artigo publicado na revista **Mecatrônica Atual** (ano 1, nº 3, abril/2002) e são transdutores que convertem o movimento angular do motor em uma série de pulsos, mostrando ao sistema de controle o resultado do comando efetuado. Cada equipe adota uma estratégia própria para a instalação do *encoder*, podendo-se usar, por exemplo, um dispositivo óptico acoplado à roda semelhante ao que se tem no interior dos *mouses* de computador. Outra alternativa é usar um motor de corrente contínua que já disponha de um *encoder* embutido - é um tipo de motor relativamente sofisticado, mas que facilita bastante a construção dos robôs.

A velocidade disponível no eixo dos motores de corrente contínua é normalmente elevada, chegando à faixa de milhares de rotações por minuto. Para se ter um termo de comparação, um pequeno motor elétrico de um carrinho de brinquedo, alimentado com a sua tensão nominal e sem nenhuma carga mecânica, gira a cerca de 5.000 rpm. É necessário, portanto, algum sistema mecânico de redução de velocidade de modo a acionar as rodas dos robôs em rotações bem mais baixas. Isso pode ser feito de várias formas, sendo o mais comum um sistema de engrenagens que transmite a rotação de cada motor a uma roda. A relação de redução irá depender de um cálculo no qual se leva em conta o diâmetro das rodas dos robôs e a velocidade máxima desejada. A maioria das equipes que atuam no projeto "Futebol de Robôs" usa uma velocidade máxima de deslocamento da ordem de 2 m/s. Acima disso o robô se tornaria tão rápido que fugiria totalmente do controle do sistema de visão instalado no computador. É importante também que as rodas tenham algum tipo de revestimento de borracha macia, para que não "patinem" ao girar, permitindo aos robôs se deslocarem na superfície de madeira do "campo".

RECEPTOR

É necessário transmitir, sem fios, as ordens do computador aos robôs. Há várias alternativas para se realizar isso, desde o uso de sistemas de rádio de

aeromodelismo até mesmo empregando emissores e receptores de luz infravermelha, semelhantes aos que se usam nos controles remotos das TVs. A título de exemplo, no artigo anterior desta série apresentamos os *transceivers* fabricados pela empresa inglesa *Radiometrix* que operam nas freqüências de 433,92 MHz e 418 MHz. Existem equipamentos similares de outros fabricantes, mas a grande vantagem dos módulos da *Radiometrix* é operarem tanto como receptor quanto como transmissor. Ou seja, para o sistema do Futebol de Robôs, usa-se um módulo ajustado como transmissor ligado ao computador, e instala-se um módulo da mesma freqüência ajustado como receptor em cada robô.

Na **Figura 2**, vê-se o esquema de ligação do módulo Radiometrix BIM-418 como receptor. A saída de dados recebidos é feita através do pino 12 em níveis TTL, ou seja, compatíveis com sinais digitais dos microcontroladores.

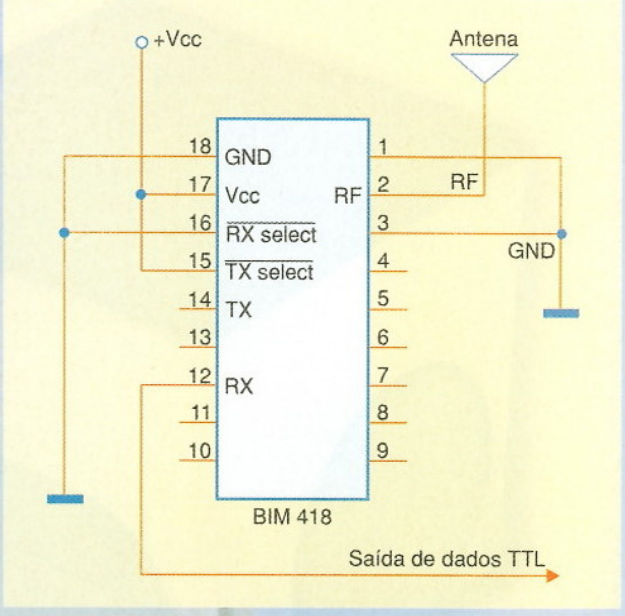
MICROCONTROLADOR E CIRCUITO DE ACIONAMENTO DOS MOTORES

A função do microcontrolador é dotar o robô de uma "inteligência" mínima, capaz de interpretar as ordens de movimentação recebidas do computador. Há uma gama enorme de microcontroladores disponíveis no mercado de componentes semicondutores e a escolha dependerá basicamente do grau de complexidade que se deseja para o projeto. Normalmente, não é necessário optar por microcontroladores com entradas ou saídas analógicas, uma vez que se pode utilizar um sistema totalmente digital. Consultando as diversas equipes internacionais que trabalham em projetos de Futebol de Robôs, encontramos desde o uso dos relativamente simples microcontroladores PIC (fabricante *Microchip Inc.*, EUA) até sofisticados sistemas com processadores digitais de sinais (DSP). A empresa sul-coreana *Yujin Robotics* comercializa robôs dentro das regras da FIRA na forma de ***kits*** completos, usando em cada robô um microprocessador modelo 80C296SA fabricado pela norte-americana *Intel*. Em nosso projeto na Universidade Federal do Paraná, estamos experimentando microcontroladores da *Atmel*, EUA, que têm se mostrado bastante poderosos e relativamente fáceis de serem programados.

O programa residente no microprocessador executa basicamente as seguintes funções:

• Interpretação das informações transmitidas do computador ao robô: Como todos os receptores funcionam na mesma freqüência, deve-se implementar um protocolo de comunicação que permita a cada robô identificar se o comando é destinado a ele ou ao "colega" do mesmo time. Isso pode ser feito instruindo o computador a incluir, no início de cada comando, alguns bits de controle, de modo que um robô específico seja movimentado.

Figura 2 – Ligação do módulo BIM-418 como receptor, através dos pinos 15 e 16. A ligação para os módulos BIM-433 (que funcionam em 433,92 MHz) é semelhante.



• Geração de pulsos de comando para os motores: conforme se viu, a maneira mais comum de se controlar a velocidade dos motores elétricos de corrente contínua é através de pulsos retangulares com largura variável (PWM). O programa embutido no microcontrolador deve, portanto, ser capaz de gerar esses pulsos, comandando a direção de giro e as velocidades de cada motor.

• Realimentação e Controle: Conforme vimos, o uso de motores de corrente contínua obriga a utilização de *encoders*, que geram pulsos à medida que as rodas do robô giram, permitindo dessa forma a realimentação do sistema. O sistema mais comum de controle da velocidade dos motores é feito através de um controlador tipo PID (proporcional-integral-derivativo), que deverá ter seus parâmetros ajustados a cada caso.

A saída dos microcontroladores é conectada a um circuito auxiliar para acionamento dos motores. Tipicamente, usa-se o circuito conhecido como *ponte H* (veja artigo sobre Caixas de Redução publicado na revista **Mecatrônica Atual** nº 1, Outubro-Novembro/2001). A *ponte H* permite a inversão da polaridade de alimentação do motor, ou seja, o motor pode ser comandado a girar no sentido horário ou anti-horário. Devido ao pequeno espaço disponível nos robôs, é conveniente usar *pontes H* na forma de circuitos integrados.

BATERIAS

Sendo entidades autônomas e sem fios, os robôs são alimentados por baterias. Há uma infinidade de

Figura 3 – Robô de futebol fabricado pela empresa brasileira Tecnomecânica.



alternativas disponíveis, desde o uso de pilhas comuns até baterias sofisticadas semelhantes às usadas em telefones celulares. É conveniente utilizar baterias recarregáveis, podendo-se adotar baterias do tipo Níquel-Cádmio (Ni-Cd), que já é bastante comum e facilmente encontrado no mercado de eletrônica. Existem também baterias recarregáveis chamadas Níquel-Metal-Hidreto (Ni-MH), que têm capacidade de carga maior que as de Níquel-Cádmio; apesar do preço superior, a maior capacidade de carga é interessante porque não se sofre o risco dos robôs ficarem sem carga durante a partida.

É importante fazer alguns testes referentes à capacidade das baterias manterem uma tensão constante nas mais variadas condições de operação dos motores. No nosso caso verificamos que, quando ocorre algum problema de obstrução no movimento dos robôs (por exemplo, o robô fica "entalado" num canto do "campo"), a corrente consumida pelos motores é muito elevada. Com isso, o nível de tensão fornecida pelas baterias caía sensivelmente, fazendo com que o microcontrolador operasse de forma errônea. A solução adotada por nós foi separar as alimentações, ou seja, cada robô passou a ser dotado de dois conjuntos independentes de baterias: um conjunto de baterias do tipo Ni-MH fornece energia para os motores, enquanto que uma pequena bateria de Ni-Cd alimenta apenas o circuito digital. Já verificamos que outras equipes partiram para uma solução diferente, dotando os robôs de um sistema de regulação da tensão da bateria. Ou seja, mais uma vez há diferentes alternativas, que deverão ser estudadas com atenção.

ROBÔS BRASILEIROS

As regras das competições da **FIRA** e da **RoboCup** permitem a cada equipe optar por construir seus robôs da maneira como achar conveniente, ou até mesmo usar robôs adquiridos prontos ou na forma de *kits*. Nas **Figuras 3, 4 e 5** mostra-se um robô construído no Brasil segundo as regras **FIRA**, categoria *Mirosot*, fabricado pela empresa brasileira Tecnomecânica Ltda. A idéia dessa empresa é fornecer às equipes interessadas robôs com toda a parte mecânica pronta e montada, deixando-se a cargo dos participantes o projeto e a montagem do circuito eletrônico. Sem dúvida essa é uma alternativa interessante, principalmente porque todas as equipes enfrentam mui-

Figura 4 – Espaço interno do robô brasileiro.

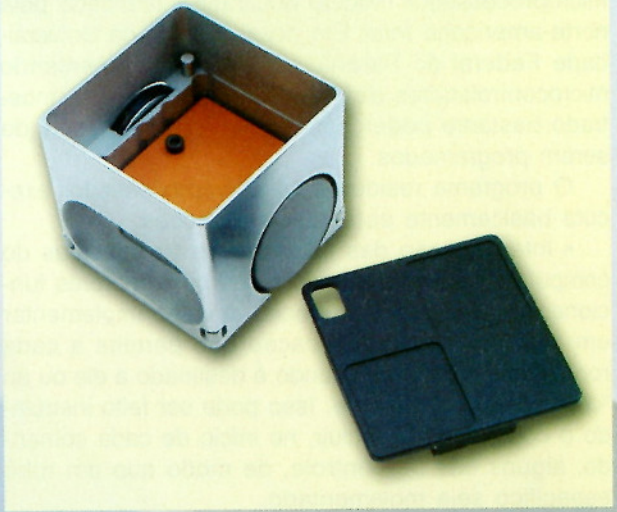


Figura 5 – Detalhe da roldana colocada no canto do robô para permitir que ele se livre de obstáculos mais rapidamente.



tas dificuldades ao se partir para a construção artesanal dos robôs. A estrutura mecânica dos robôs deve ser muito bem construída, pois é necessária uma alta precisão dos movimentos. Por exemplo, se as rodas tiverem uma diferença mínima de diâmetro, ao se comandar o robô para andar para frente, ele executará na verdade uma curva suave, deslocando-se portanto para um ponto completamente diverso daquele estabelecido pelo programa de estratégia.

Os robôs produzidos pela Tecnomecânica são construídos em um chassi metálico monobloco. As rodas são torneadas em aço inoxidável, com precisão de 0,05 mm, sendo aplicada em cada roda uma banda de rodagem em EPDM (um tipo de borracha macia). Para apoio do eixo das rodas foram usados rolamentos grau P6 em aço inoxidável e eixos balanceados. Tudo isso é importante para garantir a precisão dos movimentos dos robôs. A idéia da empresa é fornecer os robôs já com motores elétricos instalados, sendo a transmissão às rodas feita por um sistema de pinhão e coroa em aço e bronze fosforoso e dotado de sistema de apoio com microrrolamentos. Conforme as fotos demonstram, o robô dispõe de uma tampa superior encaixada, permitindo fácil acesso à parte eletrônica. O volume interno disponível para o circuito eletrônico é de 72 mm x 72 mm x 25 mm, com espaço para cinco baterias de tamanho equivalente às pilhas AA. Os motores fornecidos são da empresa suíça Minimotor Faulhaber, modelo 2233U4.5S-74/160 com pinhões de 12 dentes. Esses motores possuem alto torque e contatos em ouro, com comutador blindado, sendo já fornecidos cada qual com um *encoder* absoluto pré-instalado no interior do próprio motor. Segundo o fabricante, o robô atinge uma velocidade máxima de 2,5 metros/segundo, o que é compatível com o que se verifica nos campeonatos internacionais nas melhores equipes. As dimensões externas dos robôs da Tecnomecânica estão rigorosamente de acordo com as regras da FIRA, podendo ser utilizados em jogos de "3 contra 3" robôs ou de "5 contra 5" robôs.

Na seqüência desta série, trataremos sobre um elemento fundamental no projeto "Futebol de Robôs", que é o *software* utilizado.

PARA SABER MAIS

Sobre os *transceivers* de 433,92 MHz e 418 MHz
http://www.radiometrix.com/

Sobre micromotores
http://www.minimotor.ch

Fabricantes de robôs
http://www.tecnomecânica.com.br
http://www.yujinrobot.com/
http://www.solbet.com.br/

Sobre os projetos brasileiros de "Futebol de Robôs"
http://www.ufpr.br/futrobos/
http://www.verlab.dcc.ufmg.br/mirosot/pt/
http://www.lti.pcs.usp.br/robotics/futepoli/



TRABALHANDO COM MATERIAIS ALTERNATIVOS

José Francci Júnior

Apesar da boa aceitação dos leitores com os artigos da série "Trabalhando com Plásticos", existem várias situações – tempo, custo, disponibilidade, etc., que dificultam – além daqueles que pretendem se dedicar mais à pesquisa eletrônica e informática propriamente ditas, sem se verem "obrigados" a execução de construções complexas, com acabamento esmerado de objetos, muitas vezes, de curta vida útil. Esta nova série tentará dar à vocês, leitores, algumas dicas sobre a utilidade de materiais de fácil acesso, "sucata", etc., ficando por conta da criatividade de cada um a "busca" pelas soluções mais simples e eficientes.

Trabalhando com papelão "poliondas"

Largamente empregado em embalagens industrializadas, o papelão "poliondas" é de facílima obtenção sendo encontrado nas mais diversas medidas. Veja **figura 1**.



Figura 1 - Material e ferramentas.

A "estrutura" do papelão

O papelão "poliondas" constitui-se basicamente de um sanduíche de três lâminas de papel: duas planas e uma terceira ondulada no meio, colocadas entre si; isso implica num efeito de "fibras" e como nas madeiras temos uma menor ou maior resistência estrutural conforme o sentido e a direção dos esforços. Atente para as **figuras 2**, **3**, **4** e **5**.



Figura 2 - Vista em sentido transversal às fibras.

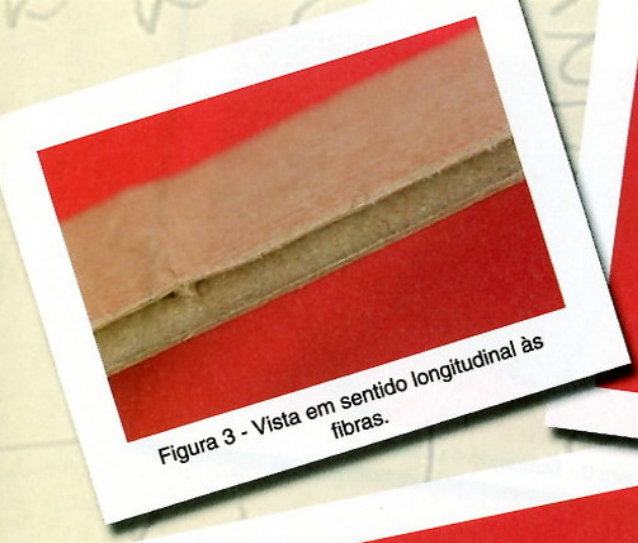


Figura 3 - Vista em sentido longitudinal às fibras.

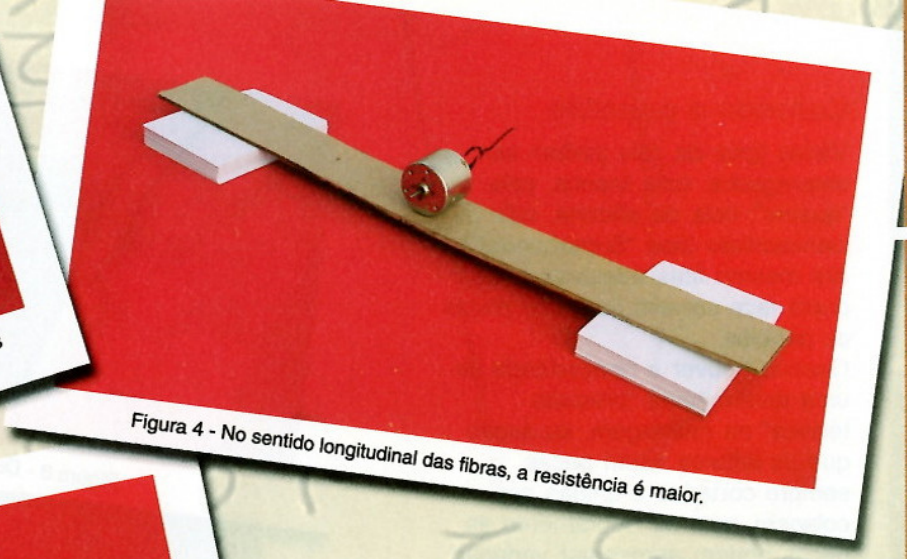


Figura 4 - No sentido longitudinal das fibras, a resistência é maior.



Figura 5 - No sentido tranversal das fibras, a resistência é menor.

Cortando o papelão

Devemos sempre cortar o papelão com estilete ou faca bem afiados, nunca com tesoura, pois ela "esmaga" a peça antes de cortar, estragando o acabamento. É digno de nota o fato de que esse tipo de papelão remove o "fio de corte" das ferramentas com facilidade e rapidez, portanto, se vamos executar um trabalho extenso é interessante termos à mão o material necessário à afiação das mesmas. Observe a **figura 6**.



Figura 6 - Cortando o papelão.

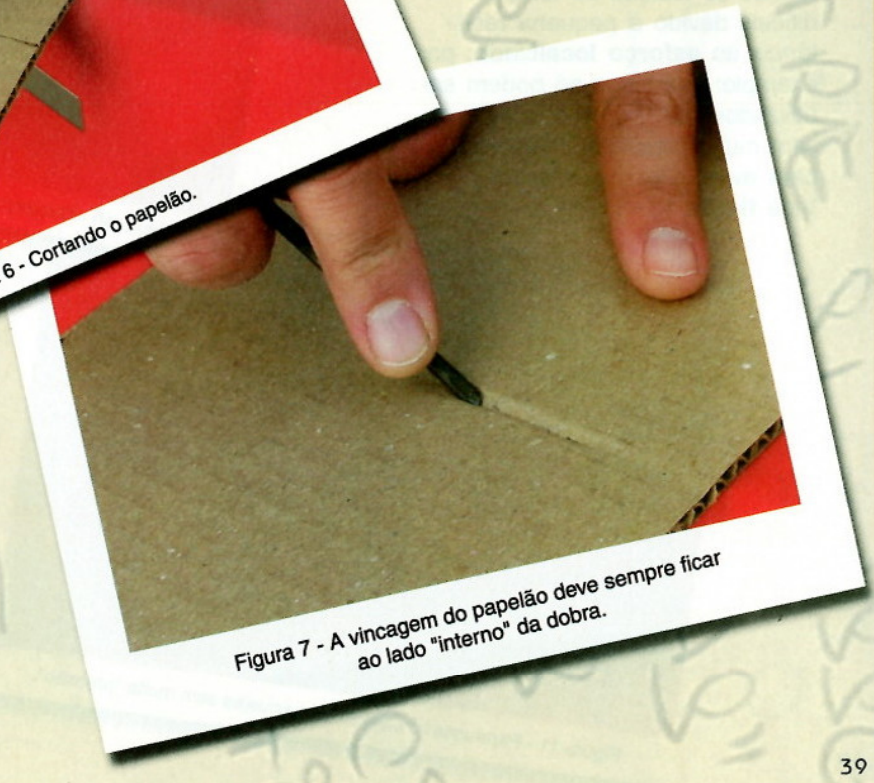


Figura 7 - A vincagem do papelão deve sempre ficar ao lado "interno" da dobra.

Dobrando o papelão

O papelão "poliondas" aceita dobragem com facilidade em ambos os sentidos das fibras, mas é necessário "vincá-lo" previamente com uma ferramenta de ponta obtusa ou arredondada para se obter uma dobra regular. Acompanhe as **figuras 7** e **8**.

Colando o papelão

Vários tipos de cola podem ser empregados: cola branca, cola "quente", cola de silicone, cola de cianoacrilato (tipo "Superbonder"). Até mesmo fitas adesivas e fitas dupla face servem para unir peças de papelão.

Quando houver a necessidade de uma determinada "precisão técnica" na montagem, eu sugiro que os leitores usem peças sempre cortadas e unidas por colagem, pois no dobramento as peças podem apresentar variação de medida.

Um bom método para se obter uniões firmes e precisas é usando-se cola de cianoacrilato (tipo "Superbonder") e eventual reforço com fita crepe,

Observe as **figuras 9** e **10**.



Figura 8 - Dobrando o papelão.



Figura 9 - Devemos colar o papelão sempre no lado interno da união.

O papelão e outros materiais

Quaisquer outros materiais podem ser afixados ao papelão através de colagem ou com o uso de fitas adesivas, ou dupla face. Outras formas de fixação se tornam difíceis devido à pequena resistência ao **esforço localizado,** por exemplo: parafusos só podem ser utilizados com porca e arruela sem muita pressão no "aperto" para evitar o esmagamento.

Vide **figura 11**.



Figura 10 - O reforço com fita é executado no lado externo da união.



Figura 11 - Parafusos só com porca e arruelas sem muita "pressão".

Conclusão

O papelão pode ser usado principalmente na execução de caixas e chassis simples, mas é muito limitado quanto à resistência a esforços mecânicos contínuos e também é muito sensível às intempéries, principalmente a umidade, que causa sua desagregação; portanto, eu aconselho o uso em trabalhos de expectativa de vida útil curta.

Introdução à Mecânica

Wilson Roberto dos Santos

É difícil imaginar a vida sem a mecânica e os seus resultados. Dispositivos mecânicos são usados pelo homem desde os primeiros tempos. A própria história da evolução humana é demarcada pelo avanço no uso dos processos de conformação mecânica. O primeiro período da História foi denominado Pedra Lascada, depois veio a Pedra Polida e em seguida os Metais. Fica clara a evolução da forma como o homem aprendeu a fabricar seus utensílios por meio de diferentes processos mecânicos. Mecânica é essencialmente a ciência que estuda os movimentos e as forças que os produzem.

Para discorrermos sobre mecânica devemos identificar o que realmente abordaremos, pois aprender mecânica não significa apenas saber como consertar automóveis. A mecânica é muito abrangente e compreende inúmeras técnicas utilizadas para projetar e produzir peças e componentes, inclusive para os próprios carros, além de motos, aviões, navios, robôs e etc.

Ao contrário do que se possa imaginar, o mecânico não é um sujeito sujo de graxa até na foto da carteira de identidade e com uma estopa pendurada no bolso do macacão. Atualmente, o chão-de-fábrica das indústrias mecânicas precisa ser limpo e organizado e os mecânicos são, sobretudo, profissionais aptos a projetar, desenhar uma peça em um computador, operar uma máquina-ferramenta, programar uma máquina CNC ou um robô, e também construí-los. Além de conhecer processos de produção, sistemas pneumáticos e hidráulicos.

Vamos entender como se utiliza a mecânica para obter resultados práticos, seja em seu projeto pessoal ou na indústria. O primeiro passo é conhecer os sistemas de medidas e os instrumentos de medição.



UMA BREVE HISTÓRIA

Desde os primórdios de nossa civilização o homem precisou medir as coisas, fossem elas as distâncias, os pesos, etc. Como até certa época não existiam padrões, a medida era baseada em órgãos humanos, tais como braças, pés e polegadas. Por exemplo, para medir um objeto qualquer os homens colocavam um polegar depois do outro ao longo do que se pretendia medir, e quando se chegava ao final era declarada a medida dizendo: este objeto tem "x" polegadas. Quando se tratava de medidas maiores, o instrumento de medição passava a ser o pé e, assim, as coisas continuavam para outros "instrumentos de medição" sobre os quais poderemos falar futuramente.

Bem, a questão é que esse sistema gerava uma tremenda confusão, porque uns tinham "dedões" e outros tinham "dedinhos", a diferença no tamanho dos pés também era grande. Como fazer?

Primeiramente criaram os padrões locais, onde um desenho da medida de uma polegada, um pé ou uma braça, ficavam esculpidos em pedras nas entradas das cidades ou em paredes dos castelos, para que os comerciantes soubessem qual era a medida padrão de negócios naquela região. Sem muito sucesso.

Tentaram então basear as medidas nós órgãos dos reis (veja a ilustração da abertura deste artigo), dos faraós ou chefes, ou seja, o polegar desses superiores era a medida padrão para as polegadas e os pés deles é que serviam de referência para as medidas em pés. Porém, havia reis, faraós e chefes grandes e pequenos também.

Em 1789, os franceses resolveram acabar com a "bagunça" e em 1793 criaram o metro. Metro vem do grego *métron,* que significa "o que mede".

Como definiram o tamanho do metro? Fácil! Os franceses calcularam a décima milionésima parte da quarta parte do meridiano terrestre (ufa!), ou seja, dividiram o comprimento de um meridiano por 4.000.000.

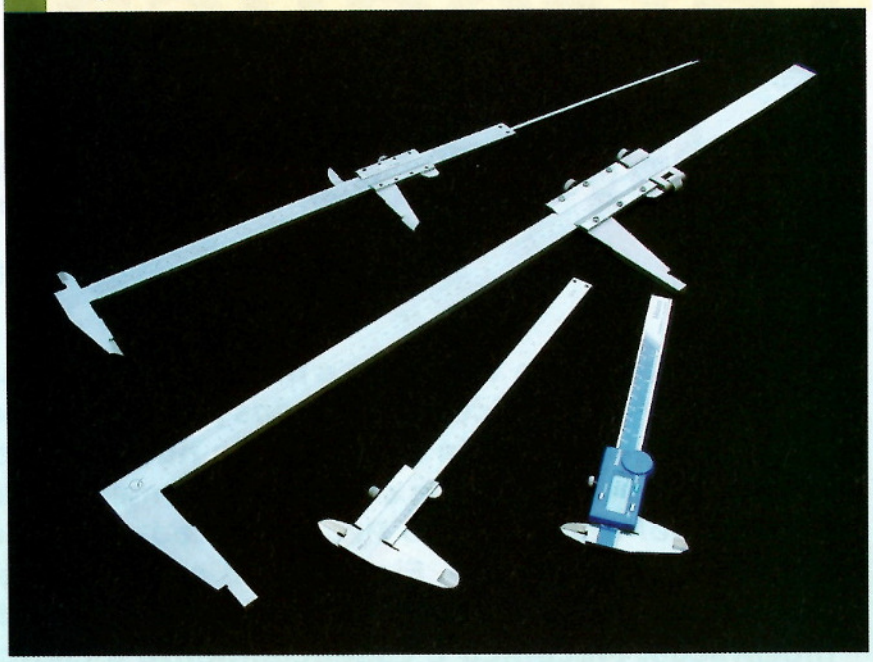
Para evitar cálculos todas as vezes que se falava em "metro" (naquele tempo não existiam calculadoras), decidiram construir um padrão físico para a medida. Esse padrão foi uma barra de platina medindo um metro de comprimento.

Mas os estudiosos não descansaram. Em 1799 fizeram outra barra de platina e em 1889 mais uma, mas, para evitar distorções a medida padrão seria considerada correta quando essas barras estivessem à temperatura de zero graus. (Só para lembrar, os metais se dilatam com o calor e contraem com o frio).

Como se não bastasse a medida do meridiano e as barras de platina (dentro de um *freezer*), resolveram melhorar ainda mais, pois havia necessidade de medidas mais precisas.

Foi quando em 1960, definiram que o comprimento de um metro era *igual a 1.650.763,73 comprimentos de onda, no vazio, da radiação correspondente à transição entre os níveis 2p e 5d do átomo de criptom-86.* (Caramba, tudo isso para falar "um metro"!)

Figura 1 - Paquímetros.



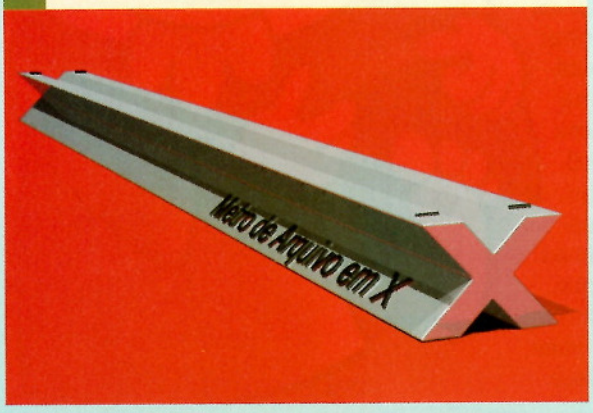
E não acabou aí.

A partir de 1983, um metro passou a corresponder à fração 1/300.000.000 da distância percorrida pela luz no vácuo durante um segundo. Este é o metro que usamos até hoje.

SISTEMAS DE MEDIDAS

Com a criação do metro, foi criado o Sistema Métrico Decimal, que vigorou até 1960 (não é coincidência não, é o mesmo ano da definição do metro pelo átomo de criptom-86), quando foi criado um sistema mais sofisticado, o SI – Sistema Internacional de Unidades, que substituiu o Sistema Métrico Decimal. Em 1962 o Brasil adotou este sistema e desde 1988 é obrigatório o seu uso em todo o território nacional.

Figura 2 – Metro de arquivo em X.



AS MEDIDAS

O sistema utilizado em nosso país para medir é decimal, porém as medidas em polegadas, ou seja, o Sistema Inglês, estão sempre presentes. Isto se deve ao fato de países como os EUA e a própria Inglaterra utilizarem o sistema métrico baseado em órgãos humanos, como a polegada e o pé, até os dias atuais. Certamente, há um padrão fixo para estas medidas e curiosamente nós o traduzimos em centímetros.

Possivelmente você já ouviu alguém pedindo: Quero uma chave de meia!! Alguém inadvertido poderia imaginar uma chave destrancando uma meia! Mas, trata-se de uma chave de "meia polegada".

Mais comum ainda é o tubo de 3/4" usado em instalações hidráulicas nas casas e edifícios. Por que não arredondaram a medida e passaram a chamá-lo de tubo de 19,0 milímetros? (3/4" = 19,05mm)

Ocorre que já existe um padrão que é complicado mudar e 0,05mm faz muita diferença. Na **figura 1** você

pode ver a foto de alguns tipos de paquímetro, instrumento utilizado para medir em polegadas e milímetros, enquanto que na **figura 2** tem um padrão de metro em X.

Milímetros e Polegadas

Na mecânica brasileira as medidas são expressas em milímetros. É o nosso padrão oficial. Milímetro, só pra lembrar, é a milésima parte de um metro. Na **tabela 1**, veja algumas subdivisões do metro, sempre dividindo por dez (afinal, por esta razão o sistema é decimal) e o equivalente em milímetros.

Mas, e quando lemos no manual da impressora que ela tem definição de 360 PPP, ou seja, Pontos Por Polegada? (em inglês DPI, que significa *Dot Per Inch*)

Pois bem, para sabermos quanto isto representa em milímetros, vamos saber a quantos milímetros equivale uma polegada:

1 polegada equivale a 25,4 milímetros e escreve-se: 1" = 25,4 mm.

Aspas (") depois do valor de uma medida linear significa que a medida está expressa em polegadas.

Sendo assim, se uma impressora tem resolução de 360 DPI e queremos saber a quanto isto equivale em milímetros, basta dividir "a quantidade" de DPI pelo equivalente em milímetros de uma polegada, ou seja:

360 : 25,4 = 14,173229

Portanto, sua impressora tem uma resolução superior a 14 pontos por milímetro. Você acha que 14 pontos dentro de um milímetro é muito apertado?

Espere até chegarmos aos instrumentos manuais de medição com os quais é possível medir até 99 pontos dentro de um milímetro.

Tabela 1 – Subdivisões do metro.

Medida	Milímetros	Fração decimal do metro
Metro	1.000	1
Decímetro	100	0,1
Centímetro	10	0,01
Milímetro	1	0,001
Décimo de milímetro	0,1	0,0001
Centésimo de milímetro	0,01	0,00001
Milésimo de milímetro	0,001	0,000001

Tabela 2 – Comparação entre frações de polegada, frações decimais e equivalência em milímetros.

Polegada e frações ordinárias	Frações decimais	Equivalência em milímetros
1/8	0,125	3,20
3/16	0,1870	4,75
¼	0,25	6,35
5/16	0,3125	7,94
3/8	0,3750	9,53
1/2"	0,5	12,7
5/8"	0,6250	15,88
3/4"	0,75	19.05
13/16"	0,875	20,64
7/8"	0,8125	22,23
1	1	25,4

Por falar nisso, acostume-se desde já a ver o milímetro como algo muito grande, pois em mecânica os erros mais grosseiros são dez vezes menores que um milímetro.

Frações de Polegadas

E quando algo mede menos de uma polegada?

Esse valor é expresso em frações de polegada. A norma é que nas medidas expressas em polegadas, a fração seja "ordinária" e não fração decimal.

Na **tabela 2**, temos a equivalência entre a polegada e as frações de polegada, sua redução em frações decimais e o equivalente em milímetros, de algumas medidas muito utilizadas em mecânica leve.

Os equivalentes em milímetros são o resultado do cálculo dessas frações, por exemplo: Quando dizemos que um parafuso tem diâmetro de 1/4", estamos automaticamente indicando que seu diâmetro é 6,35 mm, pois

25,4 : 4 = 6,35 mm.

Da mesma forma, um eixo com diâmetro de 1 1/2" (uma polegada e meia) tem 38,1mm, pois calculamos:

25,4 x 1,5 = 38,1

E ainda, algo com 1/64" (um, sessenta e quatro avos de polegada) mede 0,396875 mm, porque:

25,4 : 64 = 0,396875

Ângulos

Sabemos que uma circunferência tem 360º (trezentos e sessenta graus).

Seja em que desenho for e independentemente da unidade de medida padrão, os ângulos são medidos em graus, minutos e segundos. Por que exatamente 360? Por que não 300?

Há aproximadamente 2.100 anos, um matemático chamado Hiparco dividiu a circunferência em um múltiplo de 60 que, por ter muitos divisores, era indicado como excelente base para contagens. (60 é divisível por 1, 2, 3, 5, 6, 10, 12, 15, 20, 30 e 60).

Ele dividiu a circunferência em 360 partes e a estas partes denominou "arco de 1 grau". Depois dividiu este 1º (um grau) por 60 e a estas subdivisões chamou "arco de 1 minuto". Para aprimorar mais ainda, dividiu as subdivisões por 60 novamente e a estas subdivisões das subdivisões, denominou "arco de 1 segundo. Observe a **figura 3**.

Portanto, não há diferença entre polegadas e milímetros quando falamos de ângulos, pois eles são divididos em graus, minutos e segundos.

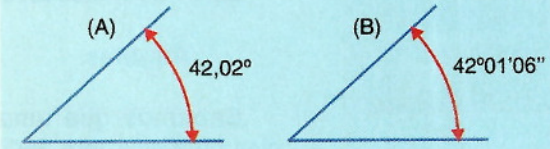
Escritos da seguinte forma:

1º = Um grau

Figura 3 - Hiparco subdividindo a circunferência.



Figura 4 - Indicação de ângulo na forma decimal (A) e em graus, minutos e segundos (B).



1' = Um minuto

1" = Um segundo (observe aqui as aspas indicando segundos e não polegadas).

Porém, a representação pode ser decimal, como pode ser visto na **figura 4**. Mas, para que servem tantas frações, milímetros, subdivisões?

A PRECISÃO MECÂNICA

As medidas são importantes, mas a precisão é o segredo da qualidade mecânica.

O que é a precisão mecânica?

Preciso é algo que está exatamente como deveria ser. Por exemplo, se você precisa de um furo com 10,0 mm, este furo não pode ter 9,9 mm e tampouco 10,1 mm. Essas diferenças podem não parecer muito grandes, mas quando se fala em mecânica 0,1 mm (um décimo de um milímetro) "é quase um metro".

O grau de precisão é indicado pela **tolerância** permitida para a medida especificada, ou seja, o **erro máximo** permitido para tal medida. As medidas e a tolerância são determinadas pelo projetista.

Pois bem, agora já está claro que a precisão na mecânica é importante e que a tolerância determina o tamanho do erro ou alteração permissível para uma medida.

Em outras palavras, a tolerância é uma variação das medidas para mais ou para menos que o projetista sabe que não afetará a qualidade da peça ou sua função.

Mas por que permitir variações de medida, se a precisão é tão importante? Boa pergunta, não é?

A razão para se permitir uma variação que não afete o desempenho é, na grande maioria das vezes, puramente econômica.

Explicando: Em um processo de usinagem, uma medida demora muito mais para ficar "0,0" (zero-zero, como se costuma falar no jargão mecânico para medidas exatas) do que se for permitida uma variação. Alguns trabalhos exigem que a peça seja iniciada em uma máquina e "ajustada" em outra que permita o trabalho mais preciso.

Por exemplo, precisamos de uma placa de aço com as seguintes dimensões: Largura x Comprimento, Espessura = 50,0 x 50,0 x 20,0 sem tolerância especificada e com um furo no centro, cujo diâmetro é 10,0 mm . Na **figura 5** veja um desenho da placa sem indicação das tolerâncias. O operador irá trabalhar provavelmente com o menor erro de sua máquina, geralmente da ordem de 0,05 mm e depois mudará para uma outra máquina que lhe permitirá diminuir ainda mais o erro, provavelmente para 0,02 mm.

Isto demorou mais horas para ficar pronto, e foram necessárias duas máquinas e dois operadores para fazer a peça. Custo de mecânica é $/hora/máquina (significa que se cobra um valor fixo sobre cada hora de uso de uma determinada máquina para fazer sua peça), portanto, ficou mais caro.

Se o objetivo era apenas fazer um peso de papel, poderia ter-se especificado uma tolerância "imensa", da ordem de 0,2 mm e pronto, o custo cairia em torno de 80%. Na **figura 6**, veja um desenho da placa com indicação de tolerância. Notou a diferença?

Mesmo assim, para um peso de papel, 0,2 mm é extremamente preciso, heim!

CONCLUSÃO

Pois bem, até aqui abordamos algo sobre medidas e a importância de observar a tolerância e o grau de precisão. Nos artigos futuros, discorreremos sobre a forma correta de medir, o uso de instrumentos manuais de medição e os códigos mecânicos para ajustes de furos e pinos, além de falarmos sobre a composição química dos diferentes tipos de aço e outras ligas mais utilizadas em mecânica.

Figura 5 – Desenho da placa sem indicação de tolerância.

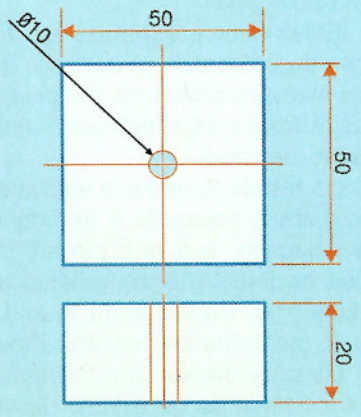
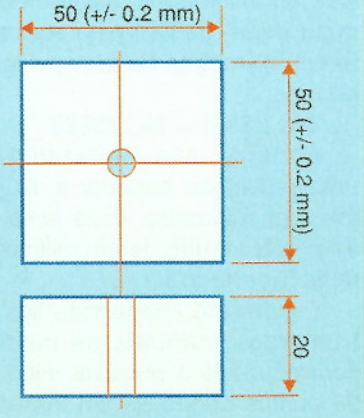
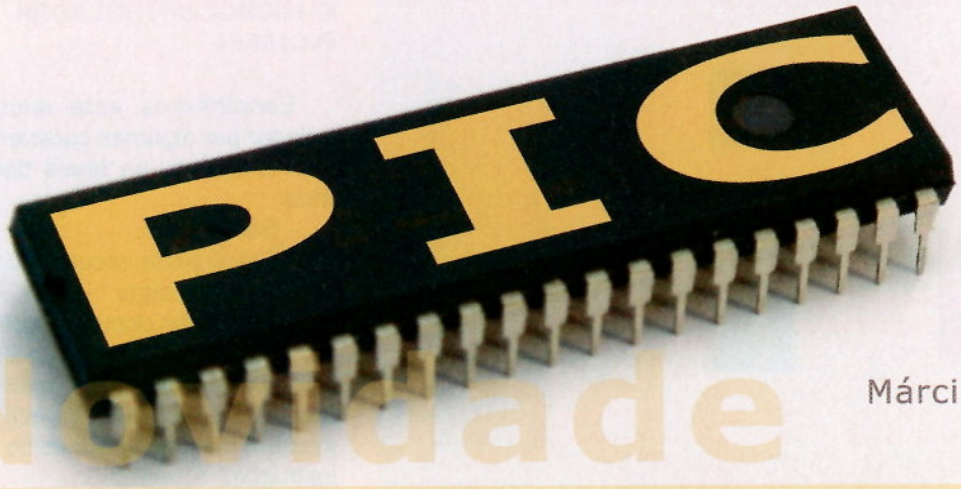


Figura 6 – Desenho da placa com indicação de tolerância.



Microcontroladores



Novidade

Márcio José Soares

Os microcontroladores estão presentes nos aparelhos que utilizamos no nosso dia-a-dia. Essa presença vai desde nossos aparelhos de televisão ou de som, teclados de computadores, celulares, "mouses", refrigeradores, máquinas de lavar, carros, brinquedos, etc. Os microcontroladores permitiram a otimização dos recursos eletrônicos, melhorando a qualidade e o custo final dos produtos.

Esta série de artigos pretende auxiliar o leitor a "entrar" no maravilhoso mundo dos microcontroladores e, basicamente, enfocará o aprendizado no microcontrolador PIC 16F84 da Microchip. Com este pequeno, mas versátil microcontrolador, o leitor poderá realizar projetos na área da Robótica e Automação. As aplicações são muitas e de acordo com a criatividade de cada um, infinitas. Convém lembrar que o Basic Stamp I e II e o Basic Step I foram desenvolvidos com microcontroladores PIC da Microchip.

Nesta primeira etapa, falaremos um pouco a respeito da estrutura do Microcontrolador, citando algumas de suas características, assim como demonstraremos alguns pontos importantes. Acreditamos, porém, que a teoria deve ser sempre acompanhada da prática, e, a propósito, estamos preparando uma surpresa para nosso leitor. Aguardem!

MICROCONTROLADORES

Na década de 80, quando a Intel lançou o seu microcontrolador 8051, ela não poderia prever a revolução que então se iniciava na Eletrônica. Muitos fabricantes como Motorola, Hitachi, National, e outros, trataram logo de lançar o seu. Vários preferiram apenas "clonar" o microcontrolador Intel, outros não. Na **figura 1** vemos alguns exemplares de microcontroladores

Um microcontrolador é, basicamente, a integração de um microprocessador com outros periféricos. Enquanto a maioria dos microprocessadores possui apenas a CPU (Unidade de Processamento Central), ALU (Unidade de Lógica e Aritmética), linhas para dados, linhas para endereços e linhas para controle e acesso de memória externa, alguns microcontroladores possuem tudo isso e mais os periféricos para comunicação serial, *timers*, *Watchdog Timer*, Osciladores, I/O's (portas para o acesso ao mundo exterior), etc.

Podemos, então, entender um microcontrolador como um "cérebro" (CPU) com "braços" (I/O's). Na **figura 2** temos a estrutura básica de um microcontrolador e através dela poderemos entender melhor o que isso significa.

CPU

A CPU ou "Unidade de Processamento Central" é responsável

Figura 1 - Microcontroladores.

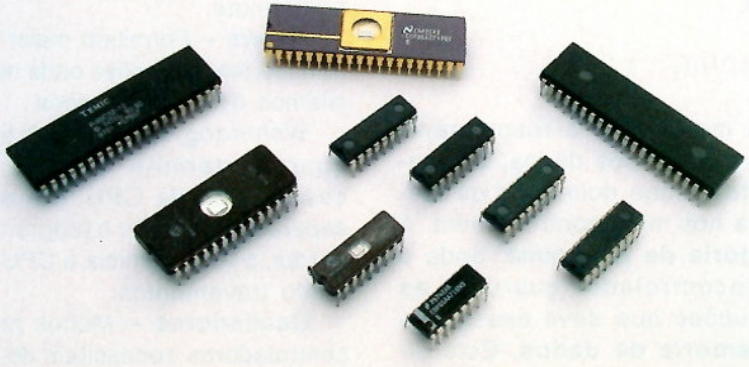
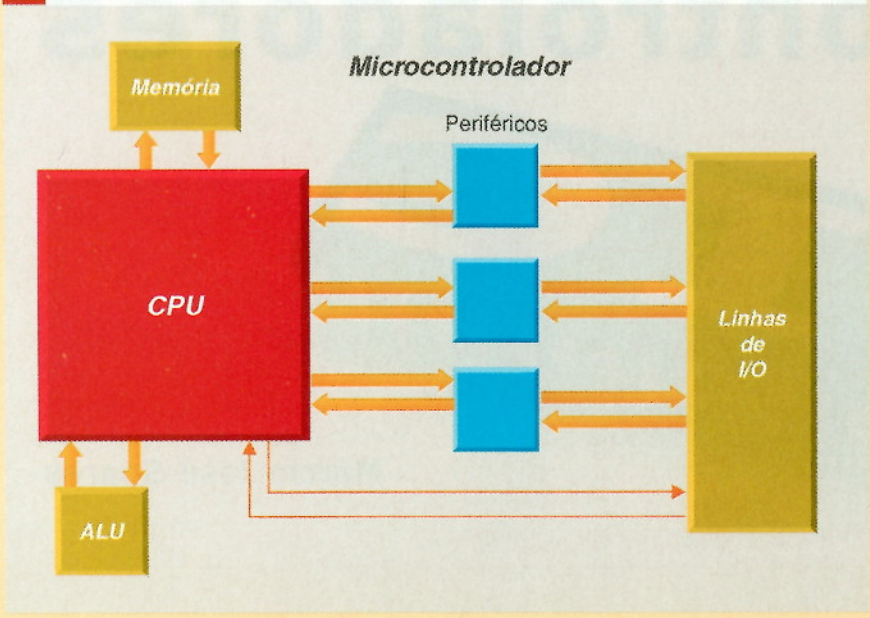


Figura 2 - Estrutura básica de um microcontrolador.



por todo o processamento de dados do microcontrolador. É ela que interpreta os comandos, faz a leitura de dados e ativa as portas de I/O's ou periféricos, se necessário.

ALU

A Unidade de Lógica e Aritmética realiza todos os cálculos que envolvam registros e/ou lógica para tomada de decisões. Ela é ligada e controlada pela CPU em um microcontrolador.

I/O's

As linhas de I/O de um microcontrolador são responsáveis pela "entrada" e "saída" do mesmo. São estes os "braços" do microcontrolador. Podemos colher um determinado dado através delas, e responder também por meio dessas linhas.

MEMÓRIA

A memória é a responsável pela guarda dos dados. Basicamente, temos dois tipos de memória nos microcontroladores. A **memória de programa**, onde o microcontrolador guardará as instruções que deve executar e a **memória de dados**. Esta última pode ser acessada pelo usuário de maneira que lhe permita escrever ou ler um determinado dado sempre que necessário.

PERIFÉRICOS

Os periféricos são circuitos que permitem maior flexibilidade ao microcontrolador e variam de acordo com o tipo e modelo. Geralmente, escolhemos nosso microcontrolador de acordo com nossas necessidades.

Os periféricos mais comuns nos microcontroladores são:

USART's – portas para comunicação serial RS232/RS485 muito utilizadas.

A/D – portas para conversão Analógico/Digital.

Portas I2C – portas para comunicação "2 fios". Permitem interligar mais microcontroladores para a realização de tarefas mais complexas.

Timers – Permitem maior flexibilidade em operações onde necessitamos de temporizações.

Watchdog Timer – O "cão de guarda" permite vigiar o processamento da CPU. Ele opera separadamente. Se o programa se perder, o WDT reinicia a CPU, evitando travamentos.

Osciladores – Muitos microcontroladores necessitam de poucos componentes internos para realizarem seu "clock" (oscilador), pois possuem circuitos internos que facilitam nosso trabalho.

O MICROCONTROLADOR PIC16F84

Escolhemos este microcontrolador por algumas características interessantes, as quais destacamos:

- Baixo custo.
- É facilmente encontrado em lojas especializadas.
- Excelente documentação.
- Serve como porta de entrada para a maioria dos microcontroladores Microchip, já que estes mantêm a compatibilidade no *set* de instruções e outras.

Porém, sua maior vantagem é a gravação. Ela pode ser feita com um pequeno circuito eletrônico, de custo muito baixo, através da porta paralela ou serial de qualquer PC. Em breve, apresentaremos ao leitor um projeto de um gravador para o PIC16F84 para que possamos realizar nossos "exercícios".

As características principais do PIC16F84 são:

- 1024 bytes de memória FLASH, que pode ser gravada várias vezes.
- 68 bytes de memória RAM.
- 64 bytes de memória EE PROM.
- 13 pinos de I/O configuráveis individualmente.
- *Timer* (temporizador) de 8 bits com "divisor escalar".
- Interrupções "externa e interna".
-Arquitetura RISC com *set* de instruções de fácil aprendizado, com apenas 35 instruções.
- *Watchdog Timer* (Cão de guarda).
- Possibilidade para 4 tipos de osciladores, configuráveis via software.
- Entre outras.

ARQUITETURA

A Microchip foi uma das precursoras do uso da tecnologia RISC (Computador com *Set* Reduzido de Instruções) em microcontroladores. Essa tecnologia é baseada na ar-

quitetura HARVARD, que separa a memória de dados da memória de programa. Assim temos um *bus* (via) para dados e outro para programa.

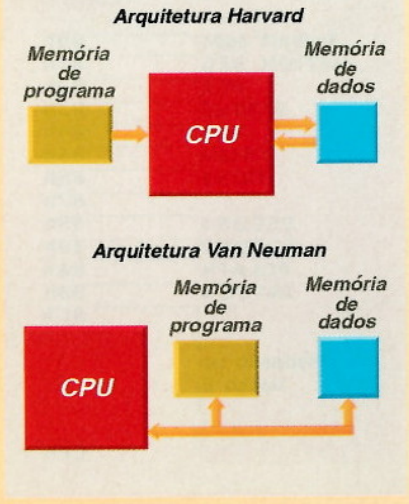
A PIC16F84 tem um *bus* (via) de dados RAM com 8 bits (1 byte) e um *bus* (via) de programa com 14 bits . Isso permite que o microcontrolador carregue na CPU um byte que contém toda uma linha de programa (operador + operando) em um único ciclo de máquina. Isso aumenta a velocidade do *chip*, permitindo uma melhor performance se comparado a outros microcontroladores que não utilizam essa arquitetura .

Os microcontroladores que não utilizam a tecnologia RISC, usam a tecnologia CISC (Computador com *Set* de Instruções Complexas), baseada na arquitetura "Van Neuman". Nessa arquitetura, a memória de dados e a memória de programa empregam o mesmo *bus* (via). Assim, o microcontrolador necessita de mais ciclos para carregar uma linha de programa (operador e o operando). Na **figura 3** temos um demonstrativo entre as duas tecnologias. Nossos PCs utilizam a tecnologia CISC, enquanto os *"Power PC's"* usam tecnologia RISC.

PINAGEM

A pinagem do PIC16F84 pode ser vista na **figura 4**. Trata-se de um encapsulamento DIP de 18 pinos. Na **tabela 1** podemos ver o descritivo de cada pino, sua função e uso.

Figura 3 - Tecnologias RISC e CISC.



MEMÓRIA

A memória do PIC16F84 para programa tem 1K x 14 bites, que a princípio pode parecer pequena, mas como o PIC utiliza a tecnologia RISC, garante uma excelente otimização da área de programa. Na **figura 5** temos o mapa de memória de programa.

PC – *Program Counter*. Este registro é controlado pelo microcontrolador e guarda a posição atual do programa. A cada linha de programa lida, o PC é incrementado em uma posição.

***Stack Level* 1 ... *Stack Level* 8** – Nestes registradores são guardados os endereços de retorno para o microcontrolador, quando utilizamos

Figura 4 - Pinagem do PIC16F84.

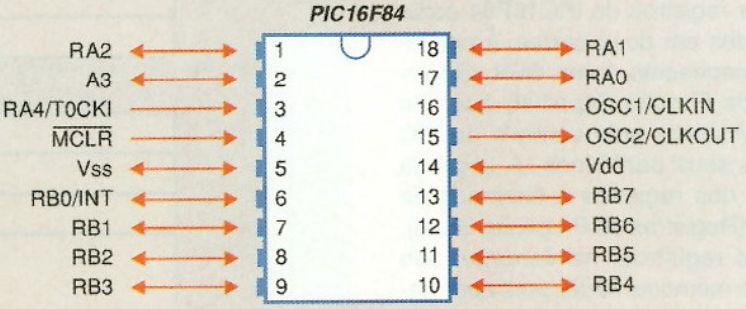


Tabela 1 - Descritivo, função e uso de cada pino.

Nome Pino	Nº	Tipo	Estrutura	Descrição
OSC1/CLKIN	16	Entrada	Shimitt Trigger/CMOS	Entrada osciladora
OSC2/CLKOUT	15	Saída		Saída osciladora
MCLR	4	Entrada/ Programação	Shimitt Trigger	*Reset* e entrada para voltagem de programação
RA0	17	Entrada/Saída	TTL	Portas de entrada e saída
RA1	18	Entrada/Saída	TTL	
RA2	1	Entrada/Saída	TTL	
RA3	2	Entrada/Saída	TTL	
RA4/TOCKI	3	Entrada/Saída	Shimitt Trigger	Porta de entrada e saída ou entrada do *clock* externo para *timer* e contador
RBO/INT	6	Entrada/Saída	TTL/ Shimitt Trigger	Porta de entrada e saída. O port B possui *pull-ups* internos programáveis
RB1	7	Entrada/Saída	TTL	Porta de entrada e saída.
RB2	8	Entrada/Saída	TTL	
RB3	9	Entrada/Saída	TTL	
RB4	10	Entrada/Saída	TTL	Permite interrupção se alterar o nível
RB5	11	Entrada/Saída	TTL	
RB6	12	Entrada/Saída	TTL/Shimitt Trigger	
RB7	13	Entrada/Saída	TTL/Shimitt Trigger	
Vss	5	Alimentação		Terra
Vdd	14	Alimentação		Positivo +5V

a instrução "CALL" (chamada de subrotina com retorno).

Reset Vector – Sempre que nosso microcontrolador for "*resetado*", ele apontará para essa posição de memória.

Interrupt Vector – Quando tivermos uma interrupção (interna ou externa), o microcontrolador apontará para este endereço.

User memory space – Memória de programa. Área dedicada ao nosso programa.

REGISTROS

Os registros do PIC16F84 estão divididos em duas partes. A primeira é conhecida como SFR (Registros de Função Especial) que são responsáveis pelo controle do PIC e dos seus periféricos. A segunda parte dos registros é chamada de GPR (Registros de Propósito Geral). Esses registros, na verdade, são nossa memória RAM, pois será neles que criaremos nossas variáveis de programa, quando necessário. Na **tabela 2** podemos observar esta divisão.

Começaremos agora a demonstrar os registros. Observem que cada um deles tem o tamanho de 1 *byte*, ou 8 bits. Os registros podem ter o estado de cada bit alterado individualmente quando descritos como "Escrita/Leitura" e só poderão ser lidos quando indicados por "Leitura". Veremos o uso de cada registro, mais a frente nesta série, com "exercícios" práticos.

REGISTRO STATUS

Este registro guarda o estado atual da Unidade Lógica Aritmética, do "Reset" e do banco de registros selecionado. Assim, sempre que necessitarmos realizar uma operação de soma, subtração ou operação lógica de comparação, nós o usaremos. Acompanhe a **tabela 3**.

IRP – Este registro não é utilizado pelo PIC16F84, sempre lido como "0".

RP1, RP0 – Registros de seleção do banco de registros. Estes bits permitem-nos trocar de banco de memória. Desse modo, podemos acessar o registro desejado.

Figura 5 - Mapa de memória de programa do PIC16F84.

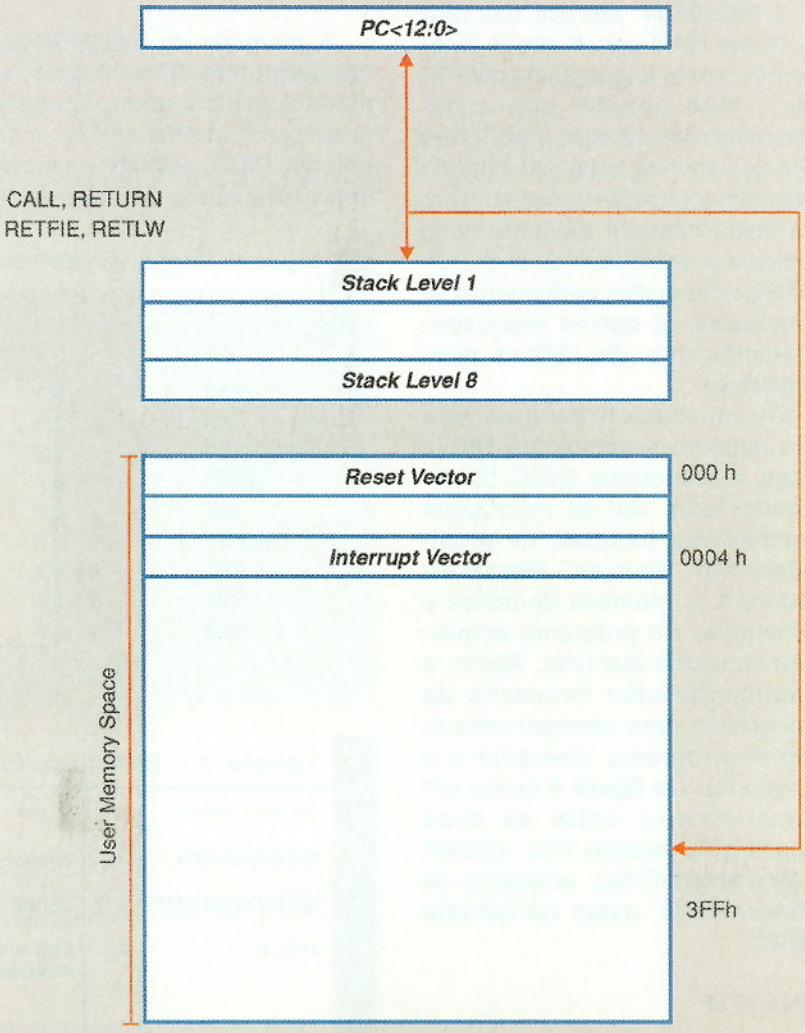


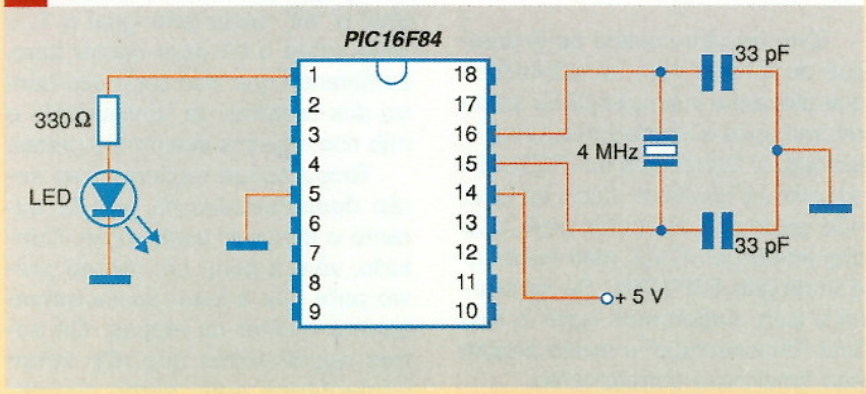
Tabela 2 - Mapa dos registros/RAM

Banco 0		Banco 1	
00h	Indirect addr.	Indirect addr.	80h
01h	TMRO	OPTION_REG	81h
02h	PCL	PCL	82h
03h	STATUS	STATUS	83h
04h	FSR	FSR	84h
05h	PORTA	TRISA	85h
06h	PORTB	TRISB	86h
07h			87h
08h	EEDATA	EECON1	88h
09h	EEADR	EECON2	89h
0Ah	PCLATH	PCLATH	8Ah
0Bh	INTCON	INTCON	8Bh
0Ch	68 bytes de uso geral	Mapeado no banco 0	8Ch
4Fh			CFh

Tabela 3 - Registro status.

Escrita leitura	Escrita leitura	Escrita leitura	Leitura	Leitura	Escrita leitura	Escrita leitura	Escrita leitura
IRP	RP1	RP0	T0	PD	Z	DC	C
bit 7			1 byte				bit 0

Figura 6 - Exemplo prático.



- 00 = Banco 0 (00H – 7FH)
- 01 = Bano 1 (80H – FFH)

TO – *Time-out* – Este registro é muito usado para nos indicar o estado atual do PIC. Podemos saber se houve alguma "anomalia" durante nosso programa, lendo-o.

- 1 = Após "*power-up*" (ligar), instrução CLRWDT (limpar Cão de Guarda) ou instrução SLEEP (dormir)
- 0 = Um "time-out" do "WDT" ocorreu.

PD – *Power-down* – O PIC possui uma proteção para subtensão. Sempre que esta ficar abaixo da tensão normal de operação, a PIC irá parar a execução do programa e informar isso através desse bit.

- 1 = Após "*power-up*" (ligar) ou instrução CLRWDT (limpar Cão de Guarda)
- 0 = execução da instrução SLEEP (dormir)

Z – bit Zero – Zero Aritmético ou Lógico

- 1 = O resultado da operação aritmética ou lógica é igual a zero.
- 0 = O resultado da operação aritmética ou lógica não é zero.

DC – Dígito Carry/borrow – Em lógica binária, durante nossos "cálculos" utilizaremos muito este bit, neste registrador.

- 1 = Aconteceu um transbordamento do 4º bit menos significativo.
- 0 = Não ocorreu o transbordamento do 4º bit menos significativo.

C – *Carry* – Assim como o bit descrito anteriormente, este também terá grande uso durante nossos "cálculos".

- 1 = Aconteceu um transbordamento do 7º bit mais significativo.
- 0 = Não ocorreu um transbordamento do 7º bit mais significativo.

REGISTRO OPTION REG

Este registro pode ser visto na **tabela 4** podemos configurar a interrupção externa através de RB0, a interrupção do *timer*, o divisor para o *timer* e os resistores de "*pull-up*" do PORTB. Atente para a **tabela 4**.

RBPU – Habilita/desabilita "pull-up's" do PORTB

- 1 = "pull-up" desabilitados
- 0 = "pull-up" habilitado

INTEDG – Seleção de borda para interrupção

- 1 = Interrupção ativa na subida de borda em RB0/INT
- 0 = Interrupção ativa na descida de borda em RB0/INT

T0CS – Captura do *clock* do *timer* (temporizador)

- 1 = Transição em RA4/T0CKI
- 0 = Instrução interna, oscilador interno

T0SE – Seleção de borda para *clock*

- 1 = Incrementa no nível alto para baixo na transição em RA4/T0CKI
- 0 = Incrementa no nível baixo para alto na transição em RA4/T0CKI

PSA – Seleção da Divisão Escalar do *Timer*

- 1 = Divisão ligada ao WDT
- 2 = Divisão ligada ao TMR0

PS2...PS0 – Divisão para contagem do *timer* e WDT (Cão de Guarda). Veja a **tabela 5**.

REGISTRO INTCON

Este registro controla as várias interrupções. Também podemos, através deste registro, saber exatamente qual interrupção foi solicitada. Observe a **tabela 6**.

GIE – Bit de habilitação global das interrupções

- 1 = Habilita todas as interrupções
- 0 = Desabilita todas as interrupções

EEIE – Habilita interrupção de "fim de escrita" para EEPROM

- 1 = Habilita interrupção de "fim de escrita"
- 0 = Desabilita interrupção de "fim de escrita"

T0IE – Habilita interrupção para "over-flow" do *Timer*

- 1 = Habilita interrupção pelo *timer*
- 0 = Desabilita interrupção pelo *timer*

INTE – Habilita interrupção pelo pino RB0/INT

- 1 = Habilita interrupção pelo pino RB0/INT
- 0 = Desabilita interrupção pelo pino RB0/INT

RBIE – Habilita interrupção na mudança de estado no PORTB

Tabela 4 - Registro OPTION REG.

Escrita leitura RBPU	Escrita leitura INTEDG	Escrita leitura TOCS	Escrita leitura T0SE	Escrita leitura PSA	Escrita leitura PS2	Escrita leitura PS1	Escrita leitura PS0
bit 7			1 byte				bit 0

Tabela 5

Valor	Proporção no TMR0	Proporção no WDT
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

- 1 = Habilita interrupção na mudança de estado no PORTB
- 0 = Desabilita interrupção na mudança de estado no PORTB

T0IF – *Flag* para leitura da interrupção "over-flow" para o *timer*
- 1 = Houve interrupção pelo *timer*
- 0 = Não houve interrupção pelo *timer*

INTF – *Flag* para leitura da interrupção pelo pino RB0/INT
- 1 = Houve interrupção pelo pino RB0/INT
- 0 = Não houve interrupção pelo pino RB0/INT

RBIF – *Flag* para leitura da interrupção na mudança de estado no PORTB
- 1 = Houve interrupção na mudança de estado no PORTB
- 0 = Não houve interrupção na mudança de estado no PORTB

REGISTRO PCL

Este registro contém o endereço da instrução a executar. Ele é incrementado automaticamente e contém os 8 bits menos significativos do PC (Program Counter).

REGISTRO STACK

Este registro contém os endereços de retorno para as sub-rotinas que utilizam a instrução CALL ou de retorno após a execução de uma interrupção. Notem que no PIC16F84 só temos 8 níveis de *stack*, ou seja, não podemos executar mais que oito instruções CALL sem executar a instrução RETURN. Se ignorarmos isso, poderemos fazer o PIC ficar "desorientado" e nosso projeto não funcionará corretamente.

REGISTROS INDF E FSR

Estes registros permitem endereçar indiretamente o banco de memória RAM. Isso é muito útil quando queremos, por exemplo, escrever ou ler dados seqüencialmente na memória. O registro FSR deve conter o endereço indireto de memória que desejamos ler ou escrever. No registro INDF encontraremos o dado do endereço apontado indiretamente por FSR.

REGISTROS PORTA E PORTB

Estes registros contêm o estado atual dos "port's" de I/O. Podemos através de instruções acessá-los individualmente ou de forma "inteira". Sem dúvida alguma, esses registros são os mais utilizados no PIC. Sempre que desejarmos escrever ou ler um pino de I/O, nós o empregaremos.

Podemos notar que acessamos os "PORT's" do PIC através destes registros. Ou seja, sempre que desejarmos "Ler" ou "Escrever" algo nos "PORT's" o faremos através destes registros.

Tabela 6 - Registro INTCON.

Escrita leitura GIE	Escrita leitura EEIE	Escrita leitura T0IE	Escrita leitura INTE	Escrita leitura RBIE	Escrita leitura T0IF	Escrita leitura INTF	Escrita leitura RBIF
bit 7			1 byte				bit 0

REGISTROS TRISA E TRISB

Estes registros permitem configurar os pinos de I/O como entrada ou saída. Isto é feito de acordo com nossas necessidades. TRISA configura o "PORTA" e TRISB configura o "PORTB". Para configurar um pino de I/O qualquer como entrada, fazemos o "bit" deste pino igual a 1, e para saída o bit deve conter zero. Lembrando que isso será feito dentro dos registros de configuração e não nos registros diretos das portas.

Todos esses registradores serão demonstrados na prática durante o decorrer desta série. Contudo, vale a pena um estudo prévio para que o leitor possa acompanhar melhor as etapas. Os outros registradores que não foram comentados aqui, serão discutidos no futuro, pois estão ligados diretamente aos periféricos do microcontrolador.

I/O's

Na PIC16F84 temos 13 pinos de I/O que podem ser configurados individualmente como entrada ou saída através dos registros "TRISA" e "TRISB". A leitura e escrita também podem ser feitas individualmente através dos registros "PORTA" ou "PORTB".

Os pinos de I/O do "PORTA" têm características "TTL", exceto o pino "RA4", que se comporta como um "Schimitt Trigger". Isto porque esse pino pode receber um sinal de "clock" externo para incrementar o *timer* do PIC. Quando isto não é necessário, ele tem o mesmo comportamento dos outros. Temos no "PORTA" 5 pinos de I/O.

Os pinos do "PORTB" também têm características "TTL", exceto o pino "RB0" que pode assumir características "Shimitt Trigger" quando configurado para receber uma interrupção externa. Temos no "PORTB" 8 pinos de I/O.

Os pinos do I/O da PIC16F84 suportam um dreno de corrente de até 25 mA, mais que suficiente para acender um LED. Notem que este dreno é por pino e não por "PORT".

Será por meio destes pinos que controlaremos o "mundo exterior": LEDs, relés, motores, *buzzer's*, etc.

Código-fonte 1 - Exemplo.

```
P;***************************
;*  Revista Mecatronica Facil
;*  programa LED.ASM
;*  pisca led com PIC16F84A
;*  Marcio Soares
;***************************

;   list p=P16F84           ;listagem
    radix dec               ;valores decimal
     include <P16F84.INC>  ;inclue arquivo

;*************************************
;*  Variaveis necessarias para o programa
;*************************************

PICRES    equ    0x00    ;vetor de reset
CHAR      equ    0x0C    ;var caracter
TEMP      que    0x0D    ;var de uso geral
T1        que    0x0E    ;var para timer
T2        que    0x0F    ;var para o timer
LED       equ    2       ;nr do port do led

;*****************************************
;*define memoria de programa e vetor de
reset
;*****************************************

    org   PICRES   ;reset

meureset:             ;cria label meureset
    goto inicio    ;desvia end 0x04 -int

;*********************************
;*endereço e rotina de interrupção
;*********************************

    org   4   ;toda int aponta p/ aqui
              ;não há ints, retorno apenas
    retfie    ;retorno de interrupção

;
;***********************
;*inicio do programa
;***********************

inicio:
    movlw   0x00          ;ajuste   bit INTCON
    movwf   INTCON        ; INTCON com 0
                          ;desabilita as int

    clrf    PORT          ;limpo registro do
                          port A
    clrf    PORTB         ;limpo registro do
                          port B

    bsf     STATUS,RP0    ;seleciona banco
    movlw   0x00          ;ajusta os bits
    movwf   TRISA         ;A como saida

    movlw   0x00          ;ajusta os bits
    movwf   TRISB         ; B como saida
    bcf     STATUS,RP0    ;volta ao banco
                          ;0... (padrão do
                          reset)

;********************
;*subrotina pisca led
;********************

piscaled:
    bsf     PORTA,LED     ;acende o led
    call    tempo         ;aguarda tempo
    bcf     PORTA,LED     ;apada o led
    call    tempo         ;aguarda tempo
    goto    piscaled      ;volta para fazer
                          ;infinitamente

;*****************************************
;*subrotina para temporizar, sem o uso do
;*temporizador interno
;*****************************************
tempo:          ;aguarda 1 segundo no total
    movlw   0xff          ;carrega W com 255
    movwf                 ;carrega T1 com W

car_1:
    movlw   0xff          ;carrega W com 255
    movwf   T2            ;carrega T2 com 255

dec_1:
    decfsz  T2,1          ;decrementa T2
    goto    dec_1         ;255 x T1 vezes
    decfsz  T1,1          ;decrementa T1
    goto car_1            ;volta a carregar T2
    retlw   0x00          -;retorna 0 em W

;
;*****************
;*  fim do programa
;*****************

    end
```

EXEMPLO PRÁTICO

Apesar do leitor ainda não poder gravar um microcontrolador PIC, é válido darmos um exemplo, para que o leitor consiga entender um pouco sobre o uso dos pinos da PIC, sua configuração básica de trabalho e ainda ver um pequeno programa comentado.

O exemplo escolhido, fará um LED piscar em intervalos de 1 segundo (**figura 6**). Observe atentamente o programa comentado e não se preocupe com as instruções (linguagem). Nas próximas partes dessa série faremos uma abordagem sobre elas. Veja **código-fonte 1**.

CONCLUSÃO

O mundo do microcontroladores é realmente fascinante, porém ele exige daqueles que se propõem a "conquistá-lo" um pouco de dedicação e paciência. Muitas dúvidas surgirão no decorrer desta leitura, mas todas serão sanadas nas próximas edições, quando começaremos a ver alguns exemplos e projetos práticos que serão amplamente comentados. Na próxima parte dessa série abordaremos o uso dos periféricos deste "pequeno notável". Até lá!

Linguagem LOGO para Robótica e Automação – 6ª parte

O LOGO controlando aplicações do Basic Step

Luiz Henrique Corrêa Bernardes

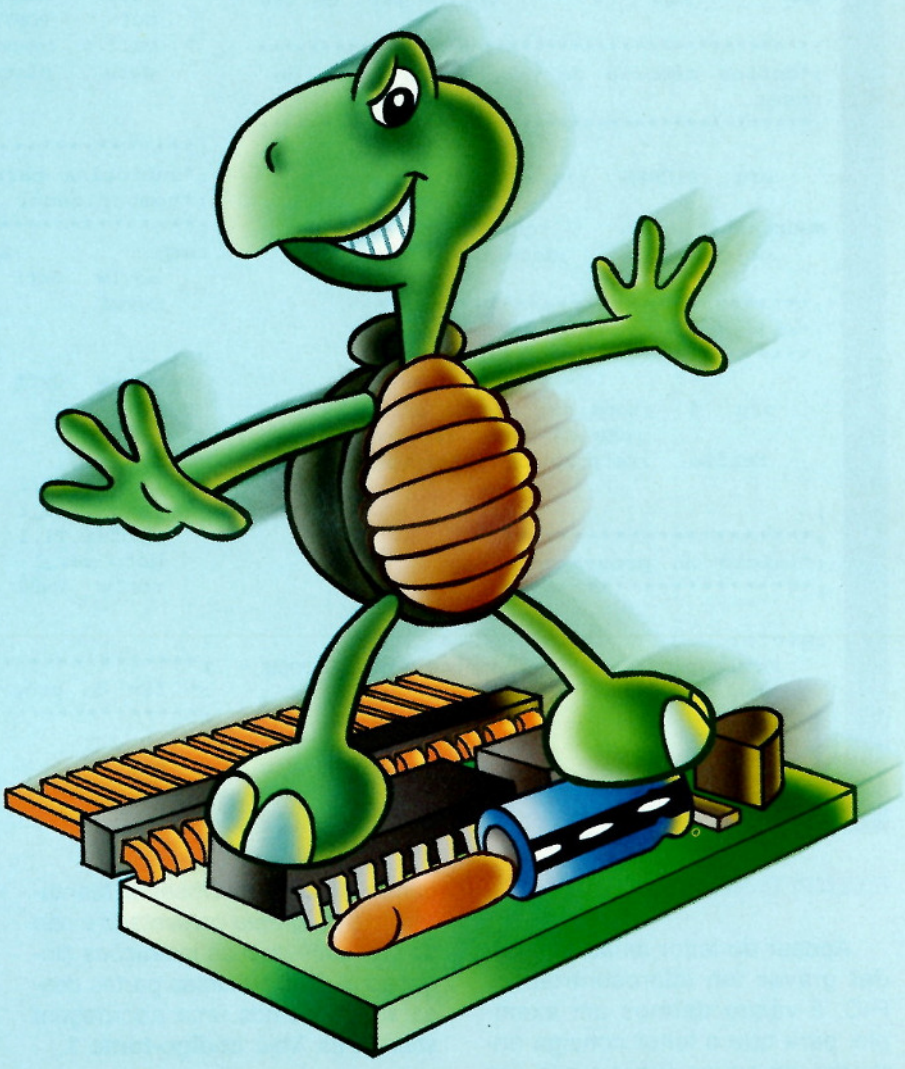
Este artigo explora as possibilidades de conexão entre o PC e o BASIC Step, utilizando programação LOGO no PC e TBASIC no BASIC Step.

No artigo anterior mostramos as noções básicas de uso da porta serial RS-232 do PC; neste artigo vamos colocar um elemento mecânico, um servomotor, para ser controlado pelo LOGO. Visto que a Revista vem publicando uma série de artigos do BASIC Step, achamos interessante a idéia de juntar o LOGO com BASIC Step.

Um programa em LOGO utilizando a porta paralela não consegue controlar um servo diretamente. Essa dificuldade é devida à precisão e freqüência do sinal que tem que ser gerado. Pensando nisso é que entra o BASIC Step, ele irá controlar essa temporização do servo. Dependendo do dado que o LOGO enviar o BASIC Step, será gerado um sinal com a temporização e freqüência adequadas.

Para entender melhor, analise o diagrama de blocos da **figura 1**. Veja que temos o PC com o programa LOGO que irá fazer a interface do usuário. O LOGO envia dados pela serial até o BASIC Step, que faz o papel de um sistema de controle. No mesmo, roda um programa em TBASIC que recebe o dado pela serial e o converte para geração de pulsos, os quais serão enviados para o servo que os converterá em um posicionamento mecânico.

Complicado? Como dissemos lá no começo desta série, se você olhar como um todo poderá achar complicado. Mas, analisando cada elemento em separado fica muito mais simples. Depois de entendido, é só ligar as peças e ver tudo funcionando!



Então, mãos a obra! Vamos começar pelo servomotor. Para o nosso uso didático, escolhemos um servomotor do tipo de aeromodelo (**figura 2**) que, classicamente, tem

3 fios. Dois de alimentação de +5 VDC e um de sinal. As cores e posições dos fios podem variar, dependendo do tipo e modelo. Mas, como exemplo, veja a **figura 3** que mostra em detalhes o conector do servo.

Para controlar o posicionamento do servo temos que gerar pulsos no seu fio de controle. A temporização varia de modelo e fabricante de servo. No nosso caso, as freqüências utilizadas estão detalhadas na **figura 4**.

Ótimo! Agora que o servo já não é um grande mistério para nós, vamos para o BASIC Step. Aqui vamos fazer várias observações. Primeiro: o objetivo do artigo não é ensinar o uso do BASIC Step em seus mínimos detalhes. Vamos apenas dar uma idéia genérica para seu uso na nossa aplicação. O BASIC Step é um microcomputador que armazena e executa um programa em TBASIC. Analise a **figura 5**: nela, você pode ver que ele tem uma memória não volátil do tipo EEPROM (a mesma que a sua televisão utiliza para memorizar os ajustes) onde são armazenados os programas. Tem também um microcontrolador onde está o interpretador TBASIC, e nele também estão os 8 pinos de I/O (*Input /Output* – entrada e saída). Para ser programado, o TBASIC necessita de um ambiente de programação que é um PC, e um software que irá editar, compilar e fazer a carga do programa (*download*) no BASIC Step (**figura 6**) através de um cabo serial. Para informações detalhadas, consulte o fabricante (**http://www.tato.ind.br**) e os vários artigos publicados na Mecatrônica Fácil.

O esquema elétrico do nosso projeto está detalhado na **figura 7**. Nós empregamos a placa StepLab da Tato para facilitar a montagem

Importante!

Caso o leitor necessite de mais informações sobre o funcionamento de servos, consulte os vários artigos que a Mecatrônica Fácil já publicou sobre o assunto.

Figura 1 - Diagrama de blocos do programa.

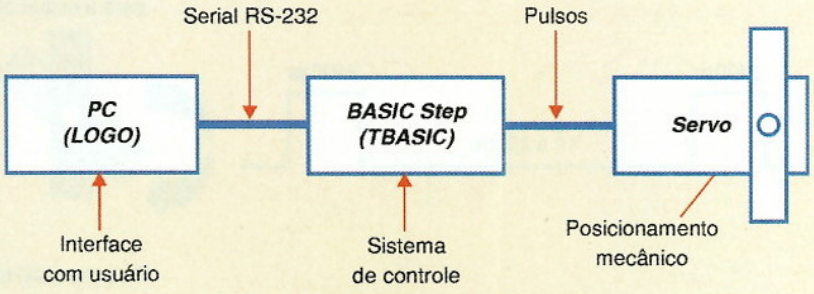


Figura 2 - Servomotor.

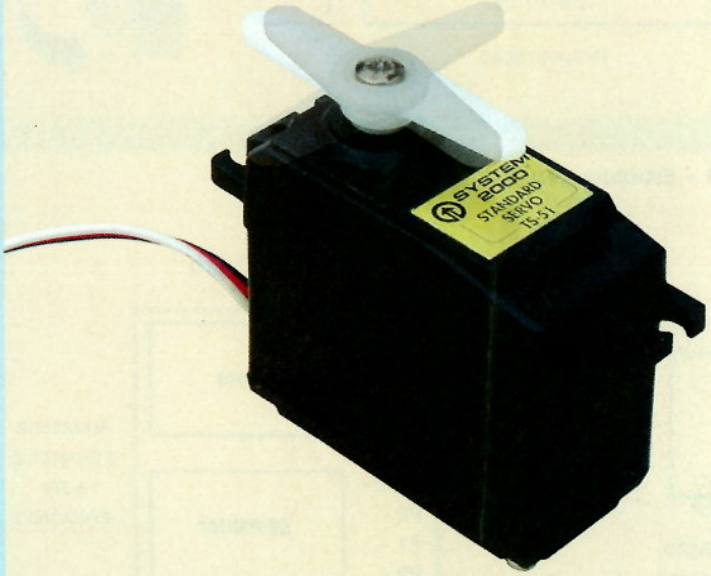


Figura 3 - Conector do servo.

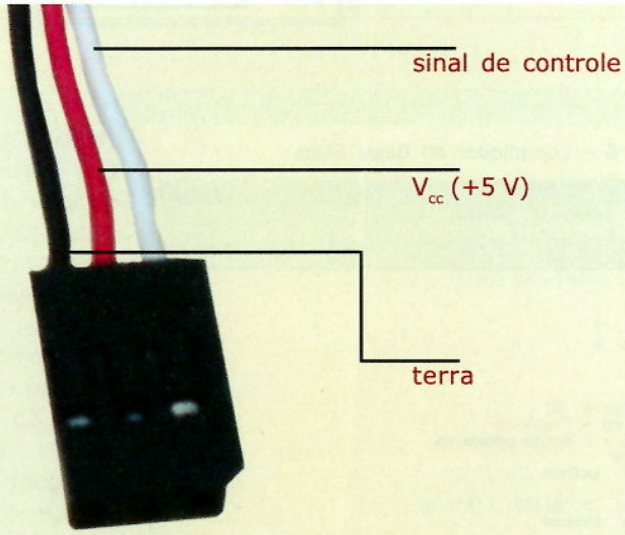


Figura 4 - Freqüências utilizadas para a rotação do servo.

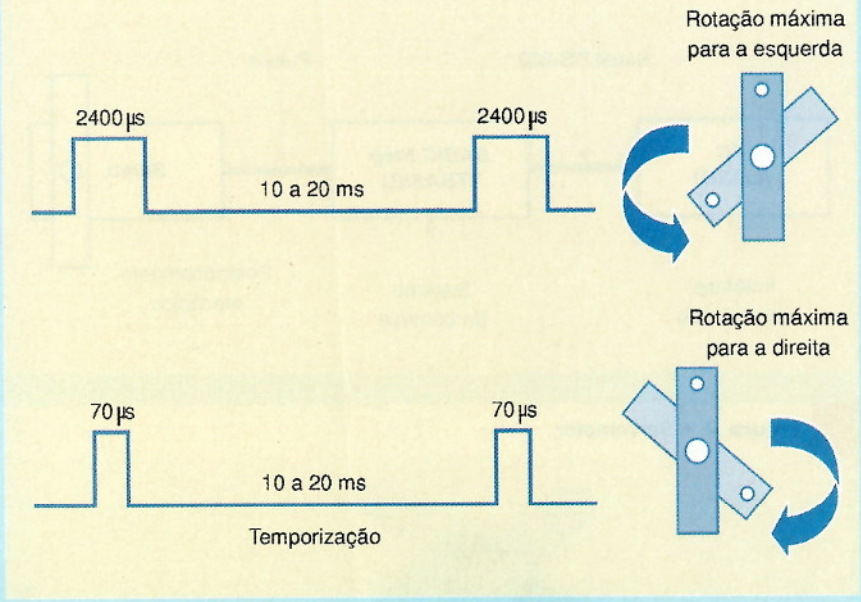


Figura 5 - Estrutura básica do Basic Step.

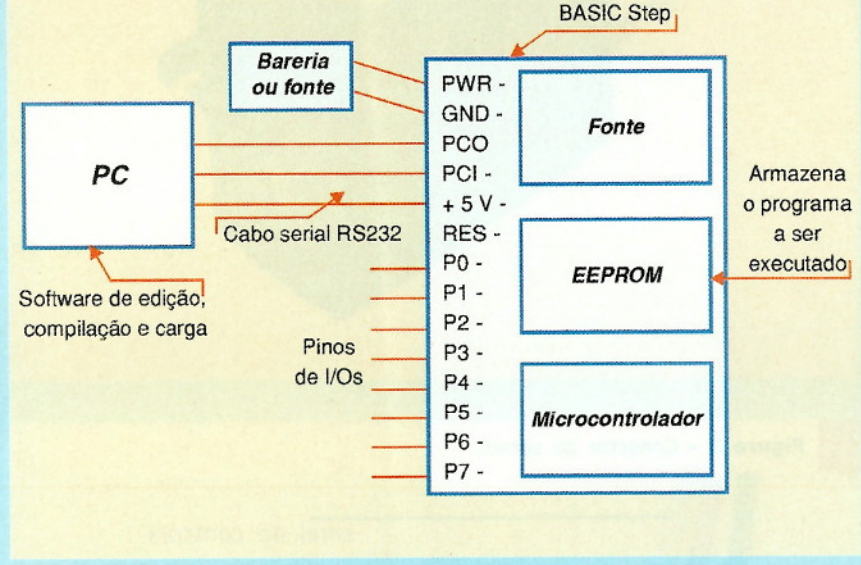
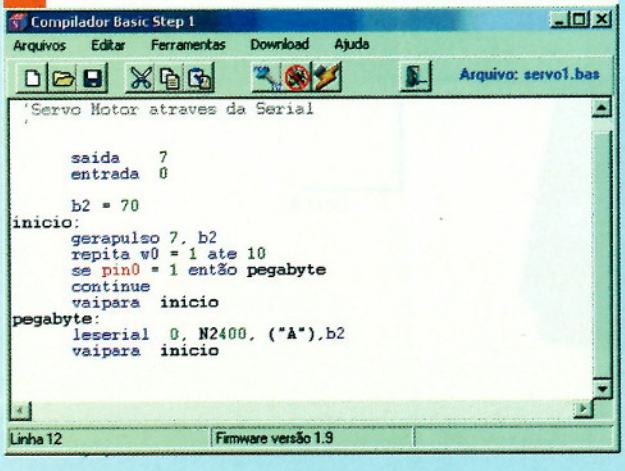


Figura 6 - Compilador do Basic Step.



(**figura 8**). Note que utilizamos um resistor de 10 kΩ para ligá-lo diretamente com o PC. Esse resistor vai adequar as tensões do RS-232 do PC para o nível TTL do BASIC Step.

Agora vejamos o que o BASIC Step tem que executar, ou seja, o que vamos programar nele! Analise o fluxograma na **figura 9**. Nele, você percebe que temos duas grandes funções: uma de gerar pulsos para o servo e a outra de ler a porta serial. Aí é que precisamos colocar algumas ressalvas. O BASIC Step não possui interrupção ou multiprocessamento, ou, em outras palavras, ele consegue fazer uma coisa de cada vez e sempre seqüencialmente. Como o nosso servo precisa que a freqüência de pulsos seja gerada constantemente (senão o nosso servo fica com posicionamento instável!) e também tem que ficar lendo a serial o tempo todo, dividimos a programação em duas partes, ou seja, ele faz um pouquinho de cada função e já que ele faz isso rapidamente, a impressão é que ele está fazendo ambas as funções simultaneamente. Veja a listagem em TBASIC no **código-fonte 1**.

A instrução **gerapulso 7,b2** é a responsável por gerar o nível alto

Código-fonte 1

```
saída 7
entrada 0
b2 = 70
início:
gerapulso 7,b2
repita w0 = 1 ate 10
se pin0 = 1 então pegabyte
continue
vaipara   início
pegabyte:
leserial 0,N2400,("A"),b2
 vaipara   início
```

do pulso de acionamento do Servo (**figura 4**) ,já o *looping* gerado pelas instruções (ver **código-fonte 2**).

Vai gerar o nível baixo do pulso. Note que temos a instrução **if pin0 = 1 then pegabyte**, que é a responsável por ver se está chegando um *byte* pela serial. Se o pino pin0 do BASIC Step estiver em 1, significa que está chegando um *byte*, ou seja, o famoso "start bit" (recorde das noções no artigo anterior). Vale salientar, novamente, que os níveis de tensões do RS-232 são inversas em relação ao nível lógico.

Então se estiver chegando um *byte*, o programa vai ser desviado para executar a instrução:

Figura 7 – Esquema elétrico.

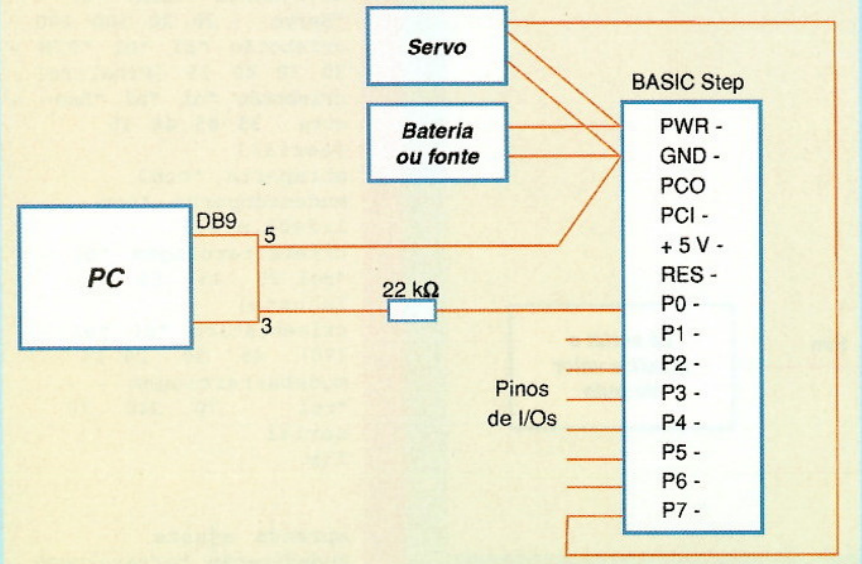
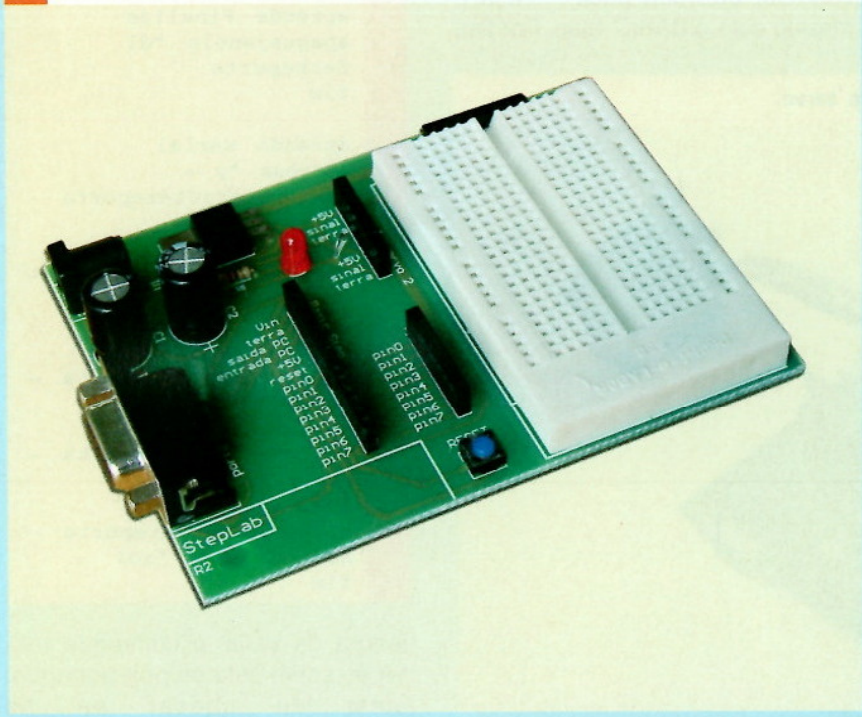


Figura 8 – Placa StepLab.



leserial 0, N2400, ("A"),b2. Aqui ajustamos a leitura para 2400 bps, invertida N2400. Isso se deve a que ligamos o BASIC Step diretamente ao PC através de um resistor pois, se tivéssemos utilizado um conversor de RS-232 para TTL (um MAX232 por exemplo), teríamos que utilizar o parâmetro T2400. O outro parâmetro **("A")** faz com que a instrução fique lendo a serial até chegar um caractere A (0x41 em hexadecimal); depois disso ele vai pegar outro *byte* que contém o valor que será utilizado na instrução de gerar pulso.

Pronto, mais um grande passo foi dado. Já sabemos como funciona o servo e como o BASIC Step o controla. Agora, só falta vermos a parte do LOGO que, diga-se de passagem, é a mais simples.

Fizemos um programa bem simples, mas eficiente, e que mostra alguns elementos novos de programação para você. Veja a interface que utilizamos na **figura 10**. Nela temos dois botões, um para desligar a aplicação e outro para executar o comando para o servo. Temos também uma barra de rolagem onde ajustamos o valor do pulso que queremos enviar para o BASIC Step e que rotaciona a nossa "tartaruguinha" no centro da Terra. Então, carregue as funções mostradas no **código-fonte 3** no ambiente do LOGO.

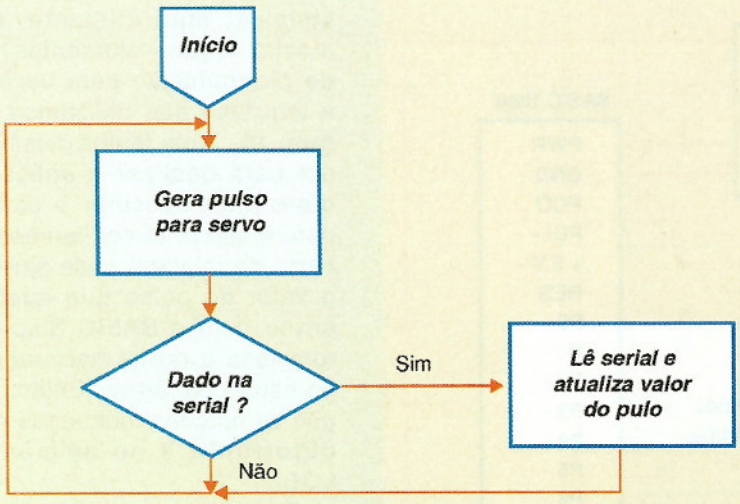
De grande novidade temos a barra de rolagem que é definida pela instrução **criebarrarolagem,** sendo que toda vez que ela for ajustada, a função **ajuste** será executada.

Dentro da função **ajuste** temos a instrução **Mudedireção** que irá fazer o ajuste da direção da nossa "tartaruguinha" e a instrução **barrarolagem** que dará o valor para a mudança do estático que informa o valor da barra de rolagem.

Os comandos de abertura de serial e configuração já foram discutidos no artigo anterior. Toda vez que o botão **Executa** for pressionado, a função serial será executada. De novo temos a função **barrarolagem** que irá fornecer o dado que será enviado para o BASIC Step. Mas se você analisar a função **serial** em detalhes, deverá surgir uma pergunta: Por que estão sendo enviados tantos caracteres "0" e mais uma temporização, se eu só precisava enviar um caractere "A" seguido do dado para temporização do servo? A resposta é: para fazer o sincronismo da serial na instrução **leserial** do BASIC Step. Como não conseguimos ler permanentemente a serial no BASIC Step, estamos sujeitos a perder dados ou pegá-los pela metade, ocasionando uma falha no posicionamento ou travamento do programa do BASIC Step.

Ufa! A parte teórica acabou, agora é só ligar tudo e ver o nosso motor funcionar sob nosso contro-

Figura 9 – Fluxograma do programa.

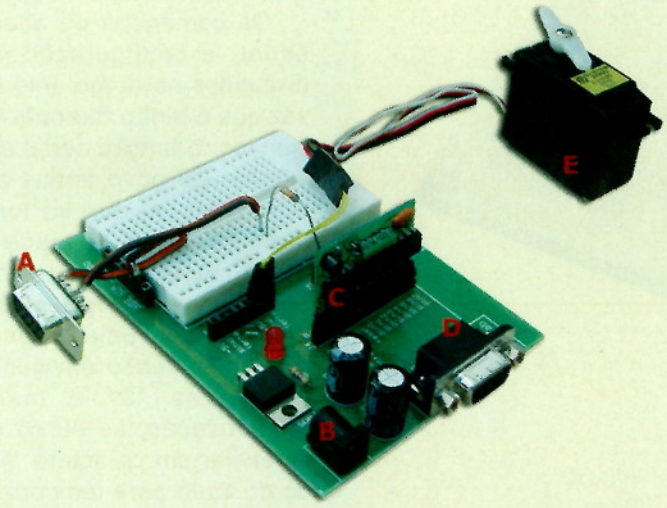


Código-fonte 2

```
repita w0 = 1 ate 10
if pin0 = 1 then pegabyt
continue
```

teza de que a serial do PC esteja funcionando, faça um programa de teste (último artigo) e use um conector de "loop back". Faça o mesmo com o BASIC Step. Por fim,

Figura 10 – Interface de controle do servo.



A - Serial para controle do LOGO
B - Conector de alimentação
C - Basic Step
D - Serial do programador
E - Servo

le. Não é legal? O que? Não funcionou? Como dizemos sempre, não se preocupe! Isso é normal, são muitos detalhes e algum pode ter escapado do controle. Revise com calma todos os passos. Tenha certeza também que o servo esteja funcionando, comente a linha **se pin0 = 1 então pegabyte** e fique alterando o valor da instrução **b2 = 70** entre 70 e 240. Com isso, você elimina a leitura da serial e consegue testar se servo está se posicionando, além de checar se as temporizações estão adequadas.

Feito tudo isso, deve estar funcionando! Agora que você já sabe como acionar um servo através do LOGO, é arregaçar as mangas e começar aquele projeto que você sempre sonhou em fazer e tinha dificuldade em ter um elemento mecânico de posicionamento.

Boa sorte, e quando ele estiver pronto não deixe de enviar-nos!

Até a próxima e bons estudos!

Código-fonte 3

```
aprenda servo
criejanela "main "d1 ~
"Servo    20 20 100 100
criebotão "d1 "b1 "FIM ~
30 70 40 15 [Finaliza]
criebotão "d1 "b2 "Exe- ~
cuta  30 45 40 15
[serial]
abraporta "com1
mudemodoporta "com ~
1:2400,n,8,1
criebarrarolagem "d1 ~
"rol 25  15  50 25 ~
[ajuste]
crieestático "d1 "e1 ~
[70]  45  30  25 15
mudebarrarolagem ~
"rol      70  240  70
serial
fim


aprenda ajuste
Mudedireção barrarolagem ~
"rol
mudeestático "e1 ~
barrarolagem "rol
fim

aprenda Finaliza
apaguejanela "d1
fecheporta
fim

aprenda serial
atribua "y ~
escrevacaractereporta ~
ascii "0
atribua "y ~
escrevacaractereporta ~
ascii "0
espere 5
atribua "y ~
escrevacaractereporta ~
ascii "0
atribua "y ~
escrevacaractereporta ~
ascii "A
atribua "y ~
escrevacaractereporta ~
barrarolagem "rol
fim
```

Controles PWM de potência

Newton C. Braga

Conforme vimos na parte anterior desta série, uma das desvantagens dos controles lineares está no fato de não ser possível manter o controle sobre o torque e a velocidade em baixas rotações. Os motores tendem a partir com "soquinhos" e também não é muito simples compensar as variações de velocidade que ocorrem quando o motor tem que fazer mais força. Uma maneira de se obter um controle muito mais preciso sobre torque e velocidade de motores de corrente contínua ou até de outras cargas, é através de circuitos que empregam a tecnologia PWM ou *Pulse Width Modulation* (Modulação de Largura de Pulso). Com estes circuitos podemos manter o torque mesmo em baixas velocidades, garantindo assim partidas suaves para os motores, ainda que carregados, o que os torna ideais para aplicações em Robótica, Mecatrônica e inclusive em aplicações industriais. Neste artigo vamos analisar o princípio de funcionamento dos controles PWM e dar alguns circuitos práticos.

Recordando o que foi visto no artigo anterior, os motores de corrente contínua são dispositivos cuja potência depende da tensão que lhes é aplicada e também da intensidade da corrente que circula através deles.

Quando carregamos um motor, sua velocidade diminui, e com isso a intensidade da corrente aumenta, elevando-se assim a potência que ele consome, conforme mostra a curva característica da **figura 1**.

Por outro lado, sob carga constante, a potência consumida aumenta com a tensão de um modo mais ou menos linear, juntamente com a corrente e a sua velocidade de rotação; observe a **figura 2**.

Concluímos então que a maneira mais simples de se controlar a velocidade de um motor de corrente contínua é modificando-se a corrente que passa através dele por meio de algum tipo de dispositivo externo. Este tipo de controle em que variamos linearmente a corrente aplicada numa carga ou a tensão, é denominado "controle linear" de potência.

No entanto, os motores de corrente contínua têm uma característica de inércia que impede que eles respondam a tensões muito baixas. Abaixo de certo valor de tensão que lhes seja aplicado, eles simplesmente não têm torque suficiente para partir e permanecem parados, conforme ilustra a **figura 3**.

Isso faz com que os controles lineares tenham uma resposta desigual em sua faixa de operação; veja exemplo na **figura 4**.

O resultado desta resposta é que não conseguimos fazer com que eles partam de modo suave, mas sim aos "trancos" e não obtemos, com eficiência, um controle preciso de baixa rotação.

Podemos resolver esse problema com um tipo de circuito que não controla a corrente no motor de forma constante, mas sim através de

Figura 1 - Curva característica de um motor CC.

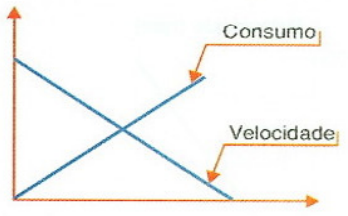
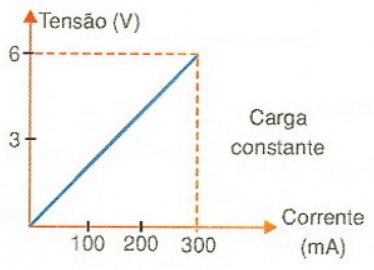


Figura 2 - Curva V x I para um motor CC.



pulsos que mantêm um controle sobre a média da corrente.

Este tipo de controle é que passaremos a analisar a partir de agora.

O CONTROLE PWM

Como podemos manter o controle sobre a corrente média de um motor de corrente contínua sem diminuir a tensão que lhe seja aplicada, uma vez que é a diminuição da tensão que nos leva ao problema de controle em baixas rotações?

Para esta pergunta existe uma resposta simples.

Podemos variar a intensidade média da corrente no motor se o alimentarmos com pulsos e controlarmos a duração desses pulsos.

Para que o leitor entenda melhor, vamos explicar detalhadamente o processo.

Em lugar de alimentarmos o motor com uma corrente contínua, o que fazemos é usar um elemento qualquer que ligue e desligue rapidamente de modo a produzir pulsos retangulares com a duração e o espaçamento iguais, conforme mostra a **figura 5**. Com isso, a tensão de cada pulso se mantém igual à máxima da fonte, mas seu valor médio, aplicado ao motor, será apenas metade do valor de entrada.

Em outras palavras, se a tensão de entrada for de 6 V, o motor recebe pulsos de 6 V, mas se comporta como se em média recebesse uma alimentação de 3 V, e através dele circulará uma corrente média que corresponde à metade da máxima, que é aquela que circula quando ele recebe 6 V.

O motor, nessas condições, irá rodar com metade de sua velocidade máxima.

Para alterar sua velocidade podemos alterar os pulsos aplicados de duas formas.

Se aumentarmos a duração dos pulsos, ou seja, mantivermos o elemento que liga e desliga mais tempo ligado do que desligado, o motor receberá alimentação por um período mais longo e na média podemos dizer que ele terá uma alimentação correspondente a uma tensão maior; observe a **figura 6**.

Nestas condições o motor gira com mais velocidade.

Para diminuir a velocidade do motor, basta reduzir a largura dos pulsos, ou seja, manter o elemento de controle menos tempo ligado e mais tempo desligado, conforme ilustra a **figura 7**.

Os pulsos aplicados ao motor ainda serão de 6 V, mas na média, como sua duração é pequena eles correspondem a uma tensão menor, e com isso a corrente no motor também será menor, com conseqüente diminuição da velocidade.

Veja o leitor que se pudermos controlar a largura dos pulsos numa faixa de valores que vá de 1% a 99%, por exemplo, teremos um excelente controle da velocidade do motor. Não podemos ter 0% ou 100% por motivos óbvios: ou paramos os pulsos com 0 V ou com 6 V!

O importante nesse tipo de controle é que em toda a faixa de velocidades o motor recebe a tensão máxima, e com isso o torque não se altera: mesmo com velocidades muito pequenas, a corrente pelo breve instante em que o pulso está presente é suficiente para tirá-lo da imobilidade mantendo o torque.

Como controlamos a velocidade através da largura dos pulsos, ou seja variamos ou modulamos a largura dos pulsos, o processo de controle recebe o nome de modulação de largura de pulsos, ou do inglês *Pulse Width Modulation*, que abreviado resulta na sigla PWM.

Na prática, o elemento usado para ligar e desligar a corrente pode ser uma chave (os contatos de um relé, por exemplo), ou ainda um componente semicondutor como um transistor comum (bipolar), um tran-

Figura 3 - Característica de inércia de um motor CC.

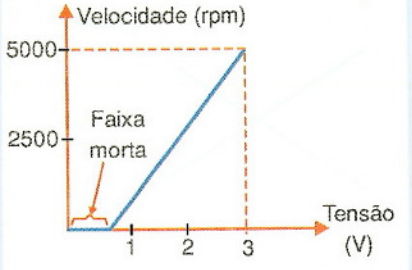


Figura 4 - Resposta desigual dos controles lineares.

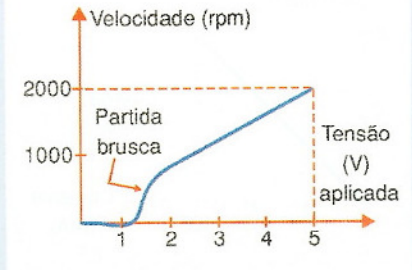


Figura 5 - Pulsos retangulares com duração e espaçamento iguais.

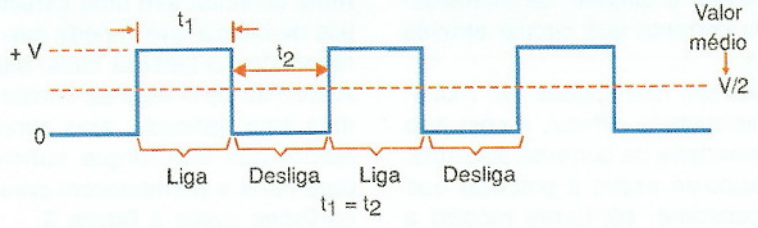
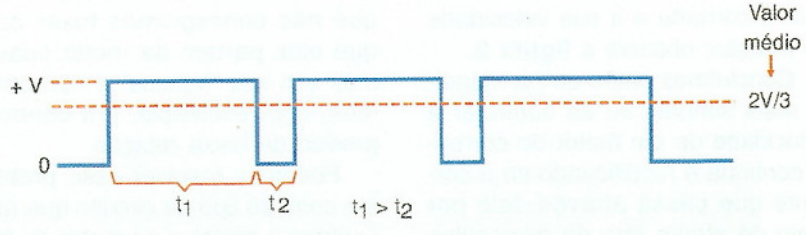


Figura 6 - Aumentando a duração dos pulsos, o motor terá uma alimentação maior.



sistor de efeito de campo (MOSFET de potência) ou um SCR.

VANTAGENS E DESVANTAGENS

Conforme observamos no artigo desta série na edição anterior, os controles lineares têm várias desvantagens como, por exemplo, a de não se conseguir em um controle preciso em baixas rotações com a manutenção do torque.

Uma outra desvantagem está no fato de que o transistor ou outro elemento linear usado no controle faz as vezes do reostato e dissipa calor. Assim, quando o motor está na faixa média de rotações, o elemento de controle dissipa quase tanta potência quanto ele próprio, o que significa produção de calor e perdas que podem não ser interessantes em muitas aplicações; atente para a **figura 8**.

Entretanto, quando usamos um transistor ou outro semicondutor para ligar e desligar a corrente produzindo pulsos, esses problemas não ocorrem.

Quando o transistor está desligado, e portanto a corrente é zero (entre os pulsos), não há dissipação de calor. Da mesma forma, quando o transistor liga e vai à saturação para conduzir a corrente máxima, sua resistência é praticamente zero e a potência que ele dissipa é quase nula.

Na prática, quando o transistor passa de um estado a outro, ou seja, liga e desliga, temos uma variação da corrente que significa uma dissipação de uma certa potência, mas ainda assim ela é muitas vezes menor do que num circuito linear, conforme mostra a **figura 9**.

Em outras palavras, o rendimento de um circuito de controle de potência PWM é muito maior do que um equivalente linear. Podemos empregar transistores de menor dissipação para controlar cargas muito maiores!

A desvantagem dos controles PWM está na comutação rápida dos transistores que podem ligar e desligar milhares de vezes por segundo, dependendo do tipo de aplicação.

A transição rápida de estado desses componentes gera transientes e sinais de altas freqüências que são responsáveis por interferências eletromagnéticas (EMI).

Essas interferências podem afetar aparelhos que operem com sinais de rádio e que estejam nas proximidades, como ilustra a **figura 10**.

Para evitar estas interferências pode ser necessário empregar filtros ou outros recursos que evitem sua propagação.

Esse mesmo efeito pode causar instabilidade em circuitos de controle do mesmo aparelho, os quais devem ser providos de recursos para que se tornem imunes à EMI.

TIPOS DE PWM

Na prática, podemos ter dois tipos de controle PWM que são utilizados em condições diferentes, e que o leitor que faz projetos de Mecatrônica poderá optar para seu esquema.

a) Controle PWM Anti-Fase

Neste tipo de controle o sinal aplicado ao motor oscila entre o positivo e o negativo; observe a **figura 11**.

Assim, temos pulsos positivos e pulsos negativos aplicados ao motor alternadamente numa freqüência que pode variar entre poucos hertz a milhares de hertz, dependendo do tipo de modo.

Se a largura dos pulsos for igual ao espaçamento, ou seja, se os pulsos positivos tiverem a mesma duração que os pulsos negativos, a média de tensão aplicada ao motor será zero e ele permanecerá parado. Na realidade, ele irá apenas oscilar na freqüência do sinal aplicado.

Todavia, se aumentarmos a duração dos pulsos positivos diminuindo ao mesmo tempo a duração dos pulsos negativos, teremos a predominância de tensões positivas aplicadas no motor e ele passará a girar no sentido que esses pulsos determinam, conforme vemos na **figura 12**.

Quanto maior for a largura dos pulsos positivos em relação aos negativos, maior será a tensão média positiva no motor e maior sua velocidade no sentido que ela determinar.

Da mesma forma, se a largura dos pulsos negativos for maior que a dos

Figura 7 - Reduzindo a largura dos pulsos, diminui a velocidade do motor porque sua alimentação diminui.

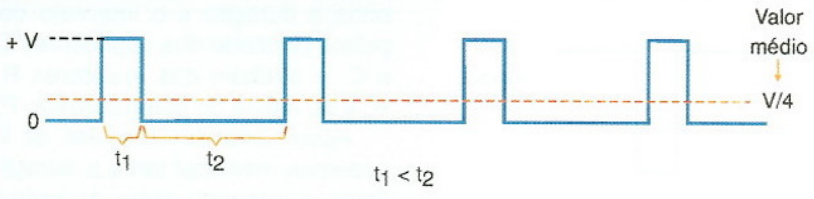


Figura 8 - Na faixa média da rotação, o controle dissipa quase tanta potência quanto o motor.

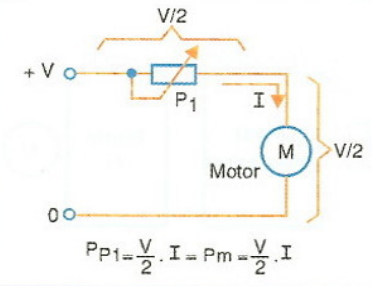
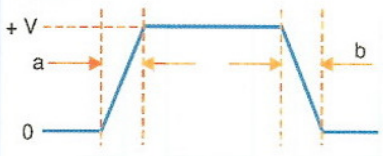
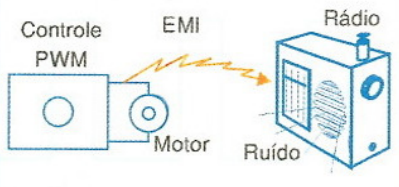


Figura 9 - O transistor dissipa pouquíssima potência quando muda de estado.



* Nas faixas a e b, o transistor dissipa potência.

Figura 10 - EMI (interferência eletromagnética) gerada pela comutação no controle PWM.



pulsos positivos, predominará a tensão negativa na média no motor e ele passará a girar no sentido oposto. Tanto maior for a largura dos pulsos negativos em relação aos positivos, maior será a velocidade do motor neste sentido. A **figura 13** dá uma idéia do que ocorre.

Se bem que este tipo de circuito tenha a vantagem de possibilitar um controle de velocidade de um motor de corrente contínua em ambos os sentidos, ele tem uma desvantagem: quando o motor está parado, ele recebe praticamente a potência máxima, continuando assim a dissipar energia convertendo-a em calor. Veja que a corrente prossegue circulando com intensidade máxima nos dois sentidos, mesmo estando o motor parado!

Do mesmo modo, qualquer que seja a velocidade do motor em um sentido ou no outro, a potência aplicada se mantém constante, pois na média, as larguras dos dois pulsos somadas se mantêm!

Por estas características, esse tipo de controle só se aplica no controle de motores de potências muito baixas, onde não existem problemas de dissipação de calor tanto pelo motor quanto pelo próprio circuito de controle.

b) Controle Polaridade/Intensidade

Neste tipo de controle temos circuitos separados para a velocidade e para a polaridade que determina o sentido de rotação do motor.

Conforme indica o diagrama de blocos da **figura 14**, temos um oscilador que gera o sinal retangular para o controle de velocidade e uma ponte H que determina o sentido de rotação do motor.

O oscilador pode ser um multivibrador astável de qualquer tipo e a ponte H pode ser uma das que vimos em artigo anterior desta série.

A vantagem desse circuito está no fato de que a potência aplicada ao motor, que é a potência que também o circuito absorve, depende da velocidade. Para menores velocidades, a potência diminui, não havendo o desperdício que acontece nos controles lineares e anti-fase.

A grande maioria dos controles práticos de velocidade/sentido PWM usa esta configuração.

CIRCUITOS PRÁTICOS

Vamos analisar alguns circuitos práticos simples que podem ser usados como controles PWM em projetos de Robótica.

a) **Transistorizado**

A configuração básica de um controle PWM simples consiste de um oscilador que gere um sinal retangular, cujo ciclo ativo pode ser variado, e uma etapa de potência para alimentar a carga.

Por ciclo ativo entendemos a relação em porcentagem entre a duração do pulso e a duração completa do ciclo do sinal; observe a **figura 15**.

Desse modo, se a duração do pulso é igual ao intervalos entre pulsos, temos um ciclo ativo de 50%.

Um circuito prático para esta finalidade empregando somente transistores é mostrada na **figura 16**.

Esse circuito é ideal para cursos técnicos, uma vez que permite entender bem a operação das diversas etapas com componentes discretos, pois não faz uso de circuitos integrados.

Temos, então, um multivibrador astável com dois transistores NPN onde a duração e o intervalo dos pulsos depende dos capacitores C_1 e C_2 e também dos resistores R_1, R_2 e do ajuste do potenciômetro P_1.

Assim, através do ajuste de P_1 podemos modificar tanto a duração como o intervalo entre os pulsos gerados por esse circuito.

Figura 11 - Sinal aplicado ao motor pelo Controle PWM anti-fase.

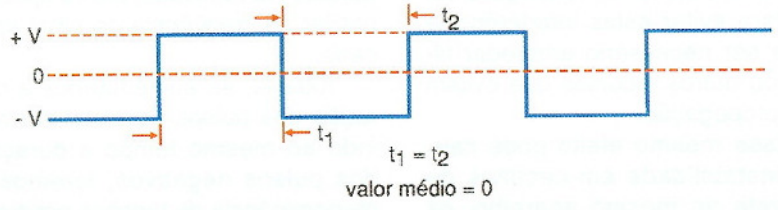


Figura 12 - Aumentando a duração dos pulsos positivos e diminuindo a dos pulsos negativos, maior será a alimentação positiva do motor.

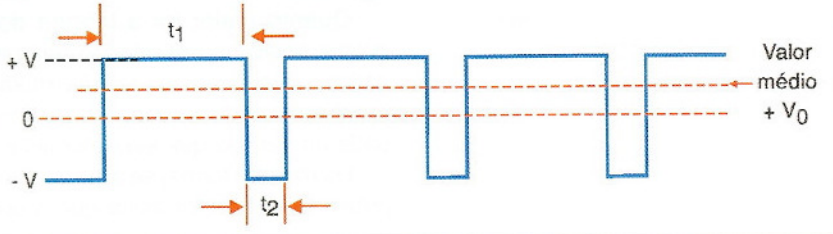


Figura 13 - A largura dos pulsos negativos é maior que a dos pulsos positivos. Agora, maior será a alimentação negativa do motor.

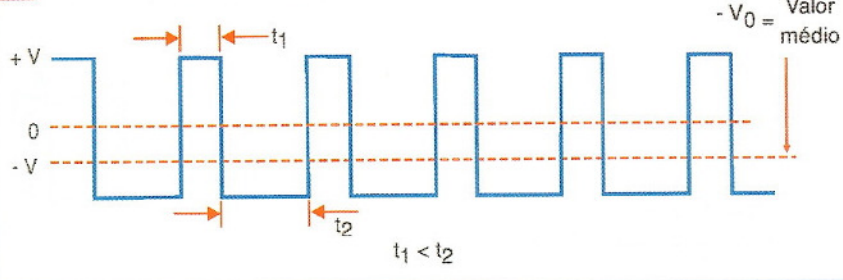
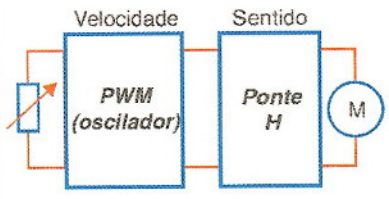


Figura 14 - Controle PWM tipo Polaridade/Intensidade.



Neste tipo de circuito, os transistores conduzem alternadamente, já que ele não possui um estado estável e a velocidade com que acontece esta troca depende justamente dos componentes citados.

Como ele não tem potência suficiente para alimentar o motor de corrente contínua, usamos uma etapa adicional de amplificação com dois transistores. No coletor do segundo transistor é, então, ligado o motor.

Na **figura 17** mostramos como montar este circuito de uma forma simples para experimentação usando uma ponte de terminais.

Uma técnica alternativa de melhor aparência e também confiabilidade é a que faz uso de uma matriz de contatos.

Motores de 3 a 12 V com correntes de até 500 mA podem ser controlados com este circuito. Os capacitores usados devem ser experimentados na faixa indicada, pois dependendo do motor, podem ocorrer vibrações em baixas rotações com valores mais baixos.

O transistor de potência Q_4 deve ser dotado de um pequeno radiador de calor. Este radiador consiste de uma chapinha de metal dobrada em U que é presa ao corpo do componente por um pequeno parafuso com porca.

b)Com o circuito integrado 555

O circuito integrado 555 permite a montagem de um excelente controle PWM para motores pequenos, como já descrevemos em artigo da edição anterior de uma forma mais completa usando o LEGO como base.

Contudo, vamos repetir a configuração para que os leitores não precisem voltar àquela edição para consulta.

Na **figura 18** temos um circuito em que o 555 funciona como um oscilador retangular cujo ciclo ativo pode ser ajustado em P_1.

Neste circuito, P_1 atua sobre o tempo em que a saída vai ao nível baixo, ou seja, atua sobre a separação dos pulsos e não duração.

Para termos a inversão do efeito, basta colocar no circuito um transistor PNP que conduza com os pulsos negativos. Dessa forma, temos a inversão dos pulsos, conforme ilustra a **figura 19**, obtendo-se os efeitos desejados.

Com este circuito é possível fazer com que um motor de corrente contínua de 3 a 12 V gire com velocidades que vão desde 2 ou 3 rotações por minuto até a máxima, praticamente sem perda de torque. A corrente máxima do motor deve ser limitada a 500 mA.

O capacitor determina a banda de freqüências de controle e deve ser experimentado na faixa indicada, pois depende do tipo de motor usado. Com alguns motores podem ocorrer vibrações e não acontecer a partida nas baixas rotações com os valores menores de capacitores.

Para mais detalhes, sugerimos a leitura do artigo das revista anterior).

c) Com o LM350T

Um controle mais potente empregando um circuito integrado que pode fornecer correntes de até 3 ampères é dado a seguir.

Na **figura 20** temos um controle PWM em que um circuito integrado 4093 é usado como oscilador retangular com ciclo ativo controlado pelo potenciômetro P_1.

Neste circuito, uma das quatro portas do 4093 é utilizada como oscilador e as três outras como um amplificador digital.

Lista de Material 1 (circuito transistorizado)

- **Q_1, Q_2 - BC548 ou equivalentes -** transistores NPN de uso geral
- **Q_3 - BC558 ou equivalente -** transistor PNP de uso geral
- **Q_4 - TIP31 -** transistor NNP de potência
- **D_1 - 1N4002 ou equivalente -** diodo de silício
- **R_1, R_4 - 1 kΩ x 1/8 W -** resistores (marrom, preto, vermelho)
- **R_2, R_3 - 2,2 kΩ x 1/8 W -** resistores (vermelho, vermelho, vermelho)
- **R_5 - 2,7 kΩ x 1/8 W -** resistor (vermelho, violeta, vermelho)
- **R_6 - 47 Ω x 1/8 W -** resistor (amarelo, violeta, preto)
- **C_1, C_2 - 100 nF a 1 µF -** capacitores cerâmicos ou poliéster (ver texto)
- **P_1 - 47 kΩ -** potenciômetro
- **S_1 -** Interruptor simples
- **M_1 -** Motor de 3 a 6 V até 500 mA
- **B_1 - 3 a 12 V -** pilhas ou bateria
- **Diversos -** ponte de terminais, fios, solda, suporte de pilhas, etc.

Figura 15 - Configuração básica de um Controle PWM.

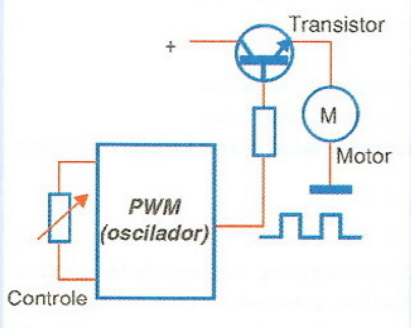


Figura 16 - Controle PWM transistorizado.

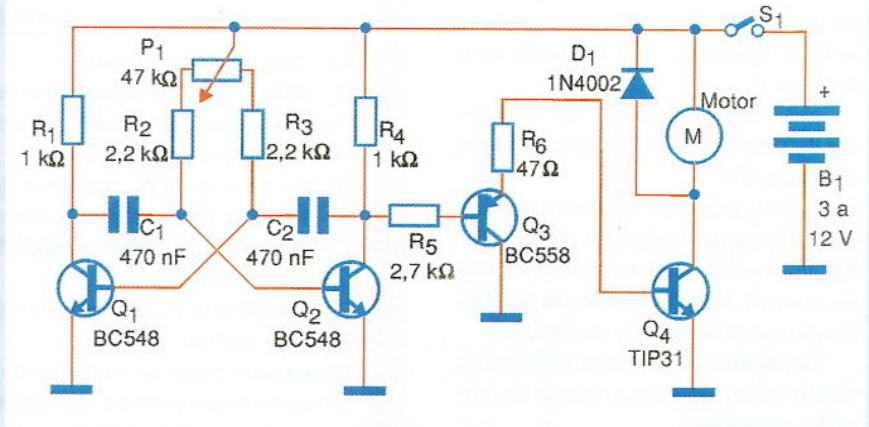
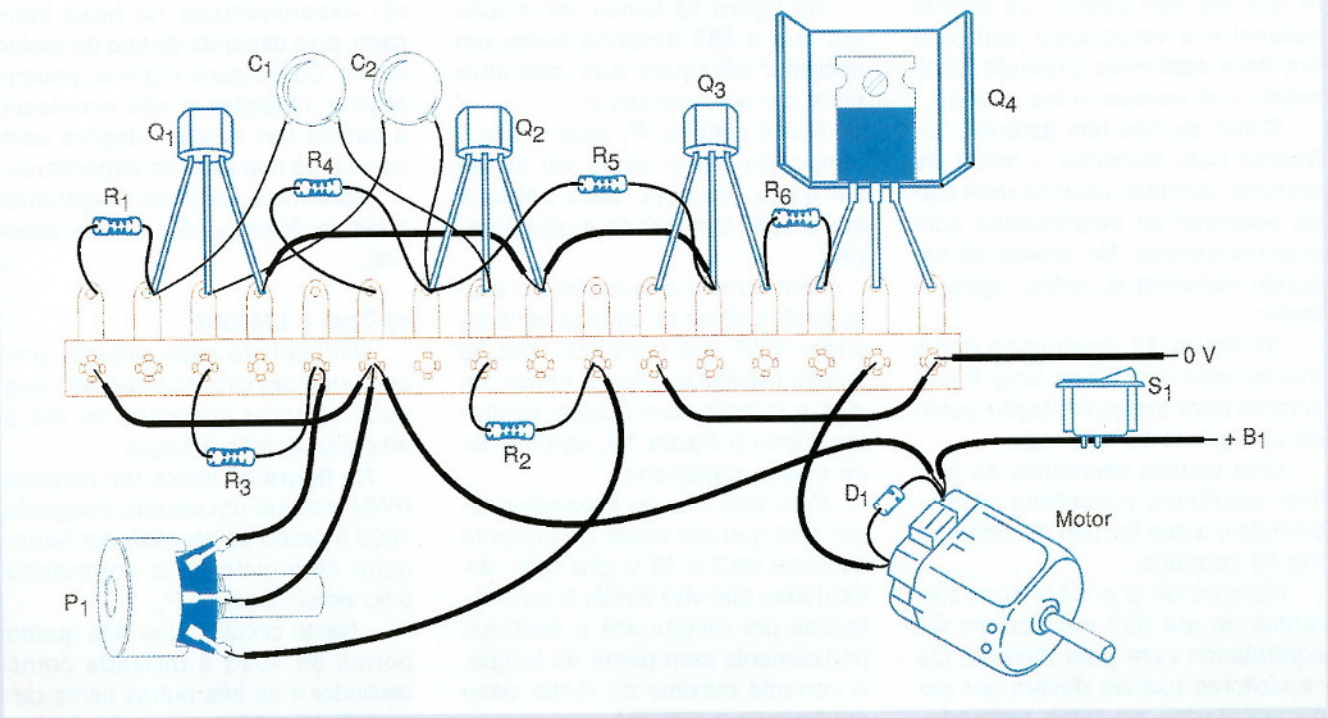


Figura 17 - Montagem do Controle PWM em pontes terminais.



Quando a saída do oscilador vai ao nível alto, as três portas invertem este sinal e o amplificam de modo que ele apareça no nível baixo na saída. Quando o nível da saída do oscilador é baixo, a tensão na saída dos três amplificadores vai ao valor máximo, igual ao da fonte.

Este sinal, formado por níveis baixos (zero volts) e altos (tensão da fonte) é aplicado ao terminal de controle do circuito integrado LM350T, que consiste num regulador de tensão para 3 ampères de corrente de saída.

Quando o nível aplicado na sua entrada é baixo, na sua saída aparece uma tensão de 1,25 V que corresponde ao zener interno.

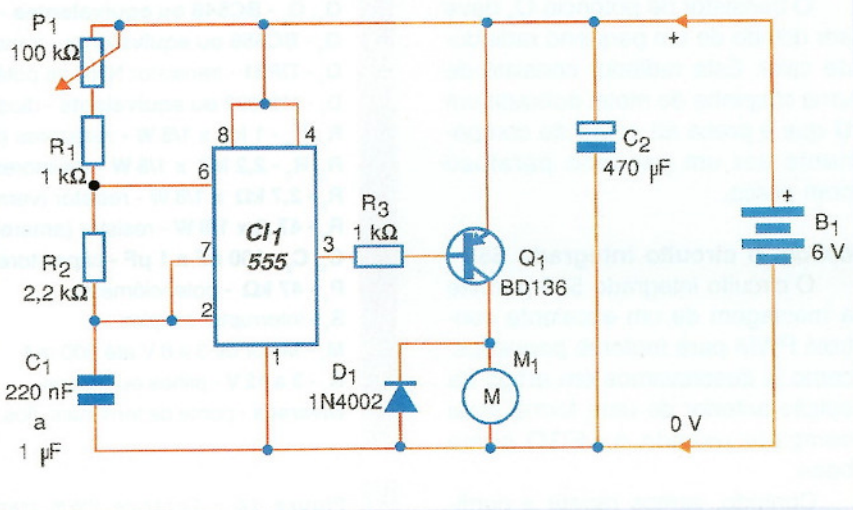
Quando o nível é alto, a tensão na saída será a do zener mais a tensão aplicada, que é fixada pelo *trimpot* P_2.

A tensão na saída do circuito oscila entre estes dois valores; veja a **figura 21**.

Em P_2 pode-se ajustar a tensão dos pulsos aplicados ao motor conforme o seu tipo e de maneira mais ou menos independente da tensão usada para alimentar o circuito.

Na **figura 22** temos a montagem deste controle numa placa de circuito impresso.

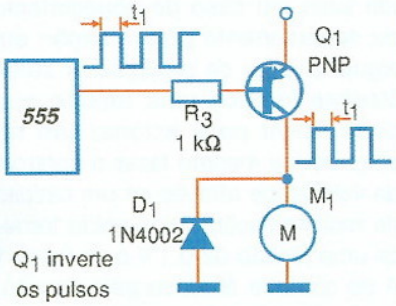
Figura 18 - Controle PWM usando o timer 555.



Lista de Material 2 (circuito integrado 555)

CI_1 - 555 - circuito integrado
Q_1 - BD136 ou equivalente - transistor PNP de média potência
D_1 - 1N4002 ou equivalente - diodo de silício
P_1 - 100 kΩ - potenciômetro
R_1, R_3 - 1 kΩ x 1/8 W - resistores (marrom, preto, vermelho)
R_2 - 2,2 kΩ x 1/8 W - resistor (vermelho, vermelho, vermelho)
C_1 - 220 nF a 1 µF (560 nF ou 680 nF - recomendados para o LEGO) - capacitor cerâmico ou poliéster
C_2 - 470 µF x 12 V - capacitor eletrolítico
B_1 - 6 V - 4 pilhas (ver texto)
Diversos - placa de circuito impresso ou matriz de contato, botão para o potenciômetro (opcional), suporte de pilhas, fios, solda, etc.

Figura 19 - Invertendo os pulsos c/ um transistor PNP.



lógicos obtidos nas portas de comunicação de um computador.

Esses circuitos integrados são usados principalmente no controle de motores de corrente contínua em máquinas industriais ou equipamentos onde se necessita de um controle preciso desses motores.

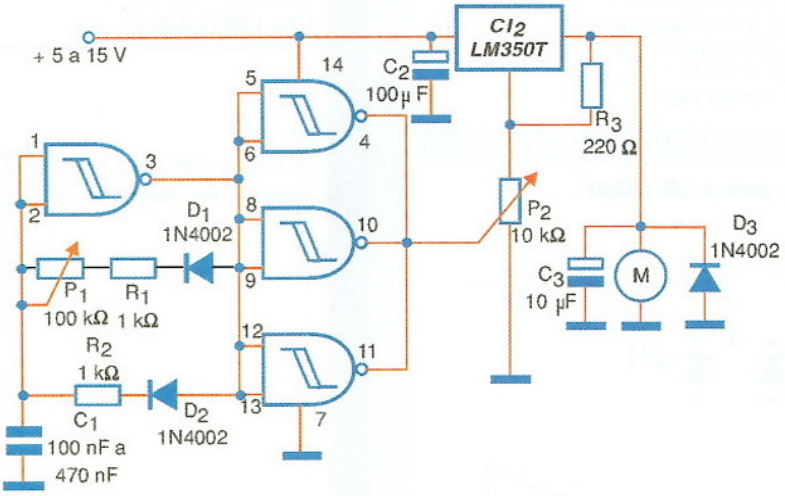
e) Circuito com o LM18201

Um primeiro circuito integrado específico para aplicações em controle PWM de motores de corrente contínua é o LM18201 da National Semiconductor, que é fornecido num invólucro SIL de 11 pinos com recursos para montagem em dissipador de calor, conforme mostra a **figura 23**.

Esse circuito integrado pode alimentar motores com até 55 V e corrente de até 3 A, possuindo internamente uma ponte H com o circuito equivalente ilustrado na **figura 24**.

Este circuito integrado pode interfacear com circuitos tanto do

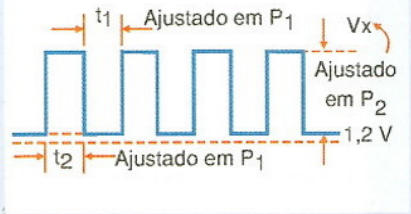
Figura 20 - Controle PWM com CI 4093 como oscilador.



Lista de Material 3 - com o LM350T

CI_1 - 4093 - circuito integrado CMOS
CI_2 - LM350T - circuito integrado regulador de tensão
D_1, D_2, D_3 - 1N4002 - diodos de silício
P_1 - 100 k W - potenciômetro
P_2 - 10 k W - *trimpot*
C_1 - 100 nF a 470 nF - capacitor cerâmico ou poliéster
C_2 - 1000 µF x 25 V - capacitor eletrolítico
C_3 - 100 µF x 16 V - capacitor eletrolítico
R_1, R_2 - 1 k W x 1/8W - resistor (marrom, preto, vermelho)
R_3, 220 W/1/8W (vermelho, vermelho, marrom)
M_1 - Motor de 6 a 12 V até 3 ampères
Diversos - placa de circuito impresso, radiador de calor, fios, solda, botão para o potenciômetro, etc.

Figura 21 - Sinal na saída do regulador LM 350 T.

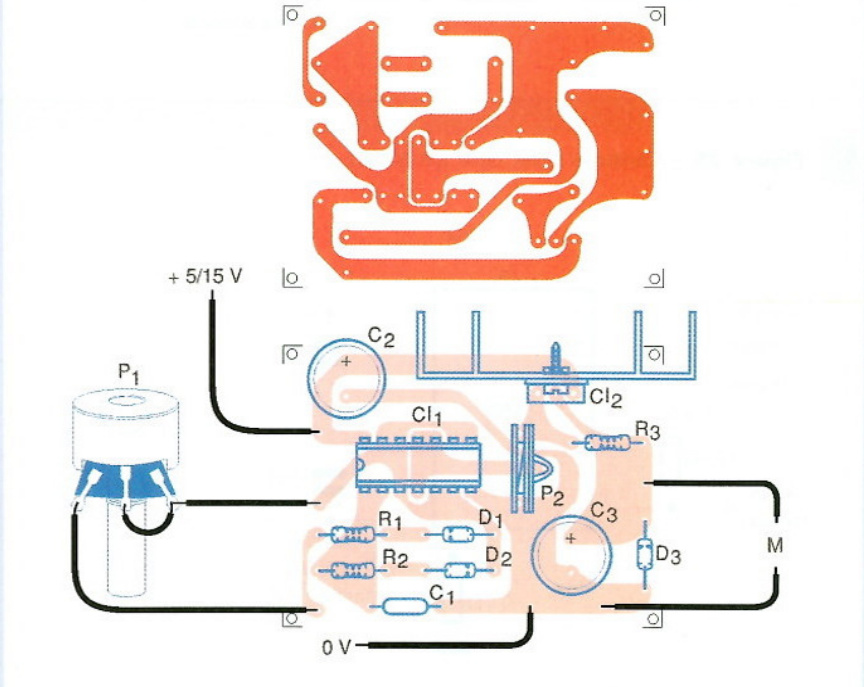


O circuito integrado LM350T deve ser dotado de um radiador de calor.

d) Usando circuitos integrados dedicados

Existem circuitos integrados que já são formados por todos os elementos para a elaboração de um controle PWM, alguns bastante sofisticados com recursos lógicos tais como os que permitem a regulagem da velocidade pela rotação do próprio motor, a parada forçada (*break*) e até mesmo o controle por níveis

Figura 22 - Montagem do Controle PWM em placa de c. impresso.



modo anti-fase como sinal/intensidade.

Na **figura 25** apresentamos o circuito típico de aplicação deste controle.

Na entrada "direção" aplicamos o sinal que estabelece o sentido de rotação do motor. Esse sinal de controle atua justamente sobre a ponte H determinando quais dos seus transistores de efeito de campo de potência vão conduzir. Este sinal é compatível tanto com lógica TTL como CMOS de 5 V de alimentação.

A função "*break*" faz com que o motor seja levado a uma parada forçada.

Na entrada PWM aplicamos os pulsos que vão determinar a potência aplicada ao motor. Esses pulsos são compatíveis com lógica TTL ou CMOS.

A saída *Flag* serve para fornecer um sinal em caso de aquecimento ou sobrecorrente para atuação em algum circuito de proteção. A saída *Vsense* fornece uma tensão que pode servir para acionar um tacômetro, ou mesmo fazer o controle da velocidade através de um circuito de realimentação. Essa saída fornece uma tensão de 0,1 V para cada 1 A de corrente drenada pelo circuito.

A seguinte tabela verdade especifica as condições de funcionamento do LMD18201.

Figura 23 - CI LM 18201, da National Semiconductor.

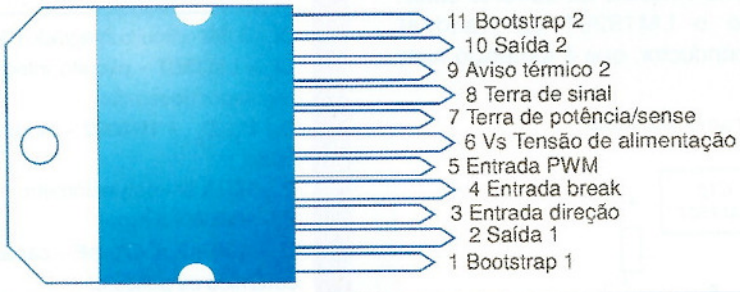


Figura 24 - Circuito interno equivalente para o LM 18201.

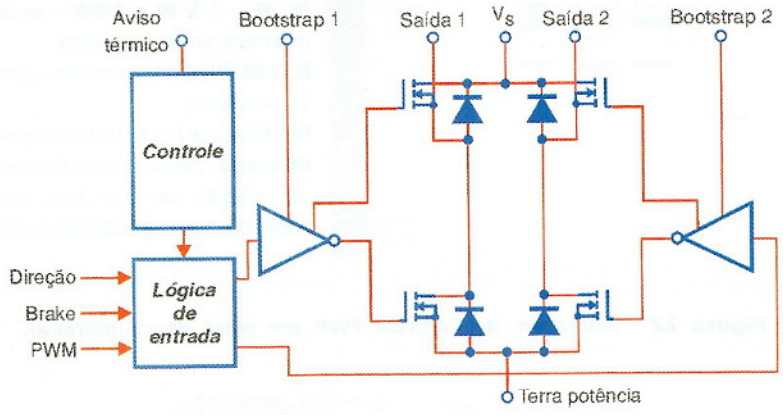
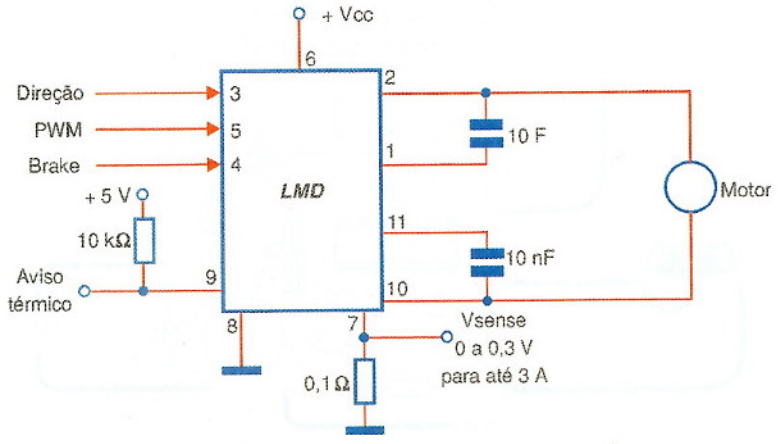


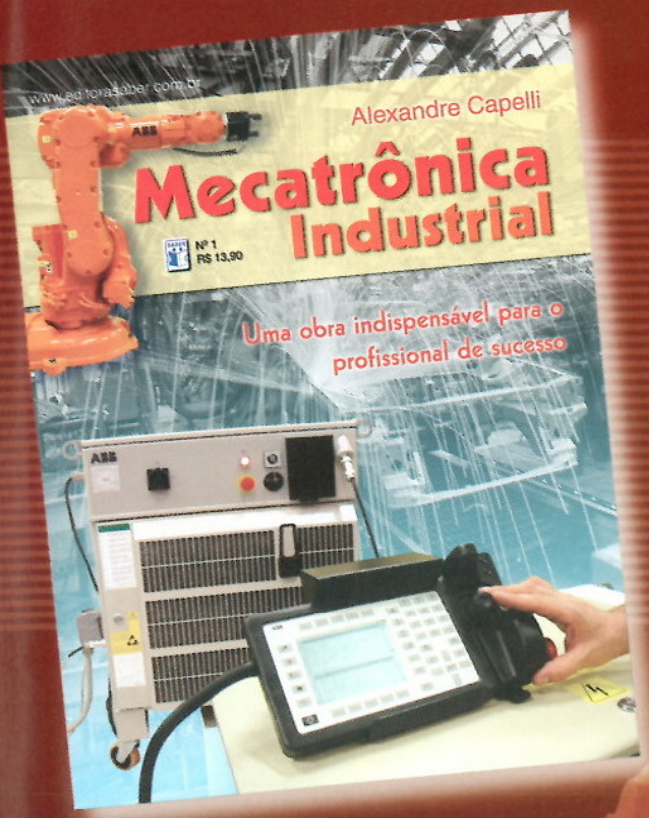
Figura 25 - Controle PWM com LM 18201.



Código-fonte 1
(do LMD18201)

PWM	*Dir*	*Breake*	*Drivers Ativos de Saída* (drena ou fornece corrente)
1	1	0	Fornece 1, Drena 2
1	0	1	Drena 1, Fornece 2
0	X	0	Sornece 1, Fornece 2
1	1	1	Fornece 1, fornece 2
1	0	1	Drena 1, Drena 2
1	x	1	Nenhum

(**Obs.:** a versão LM18200 possibilita o sensoriamento da corrente no motor por percurso de corrente que não passa pelo motor).

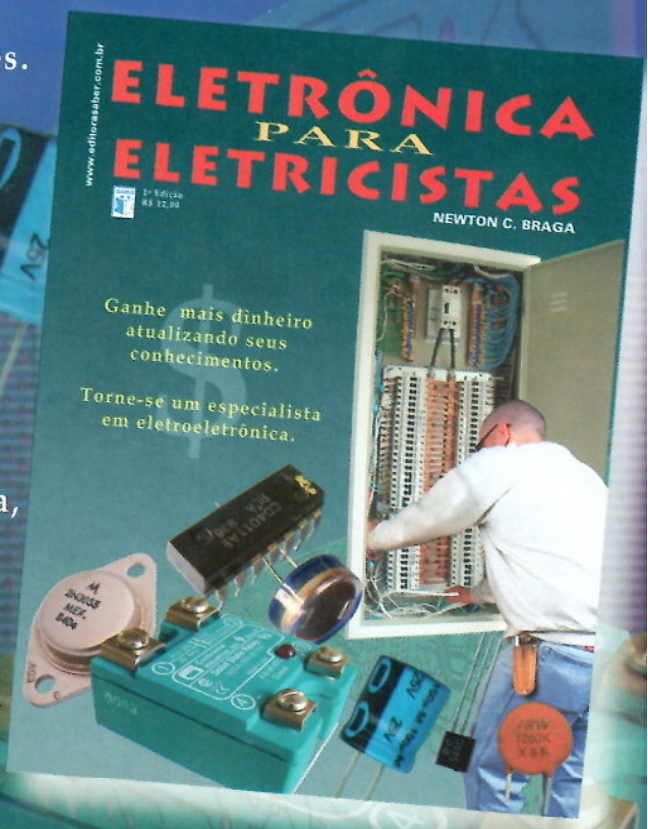


Uma obra indispensável para o profissional de sucesso, escrita pelo engenheiro Alexandre Capelli.
Este livro é muito mais do que uma coletânea atualizada dos seus principais artigos acrescido de material inédito.
É um verdadeiro tratado de automação em chão-de- fábrica (CLP's, eletropneumática, eletrohidráulica, robótica, inversores de freqüência e muito mais).
Tudo isso de forma prática com aplicações reais em campo.
Esse trabalho é um importante passo para sua empregabilidade, seja você: engenheiro (de desenvolvimento, aplicação ou comissionamento), técnico de service, ou estudante.

BANCAS NAS BANCAS NAS BA

(nos estados de São Paulo e Rio de Janeiro)

Até pouco tempo, eletrônica e eletricidade (também chamada de eletrotécnica) eram campos independentes. Enquanto o técnico em eletrônica tinha como foco o reparo de equipamentos de bens de consumo (rádio, tv, vídeo-cassete, etc.), o profissional em eletrotécnica concentrava sua atuação nos sistemas de geração, transmissão, e, principalmente, distribuição da energia elétrica. Hoje, esse cenário mudou. A eletrônica, definitivamente, "invadiu" a maioria dos sistemas e equipamentos. Eletrotécnica e eletrônica, agora, formam um novo campo: a eletroeletrônica. Pensando nisso, criamos essa obra, que aborda de maneira objetiva os princípios básicos da Eletrônica aplicados nos dispositivos encontrados nas instalações elétricas e analisa a maioria deles, explicando o que são, como funcionam e como são encontrados exatamente nas instalações ou nos aparelhos do qual fazem parte.



Construção de uma estação de reciclagem

Mais um projeto educacional utilizando o LEGO dacta. Esta estação é, na verdade, uma esteira que separa objetos metálicos de não metálicos. O principal objetivo deste projeto é mostrar como é possível trabalhar a Educação Ambiental na escola através da Mecatrônica.

Introdução à Mecânica

Finalmente vamos desbravar esse universo que, muitas vezes, constitui um verdadeiro "nó" para quem deseja realizar projetos de Mecatrônica. Neste artigo, são discutidos alguns sistemas de medidas de comprimento e ângulo, bem como alguns dos instrumentos de medidas mais utilizados.

Trabalhando com materiais alternativos

Conheça algumas dicas para executar soluções mecânicas simples utilizando o papelão "poliondas", que é largamente empregado em embalagens industrializadas e, portanto, de facílima obtenção.

Robô Mecha-Medusa MV-949

Conheça este robô andarilho que é acionado pelo som e veja, passo-a-passo, como montar o *kit* e também várias dicas úteis.

Basic Step – 4ª Parte

Aprenda a utilizar o Basic Step para enviar informações ao operador através de um visor de cristal líquido e conheça também as vantagens de empregar uma interface de comunicação serial para ligar o microcontrolador ao *display*.

Os robôs que jogam futebol

Conheça os minúsculos e velozes robôs que jogam futebol que, apesar de parecerem simples brinquedos, são equipamentos muito sofisticados. Saiba quais as características desses robôs e as especificações das competições internacionais.

Microcontroladores PIC

Os microcontroladores estão presentes em grande parte dos aparelhos que utilizamos no nosso dia-a-dia. Neste artigo, você vai conhecer a estrutura básica do microcontrolador e suas características básicas.

Elevador controlado pelo LOGO

Mais um projeto onde apresentamos duas versões: uma em plástico e outra utilizando materiais de baixo custo. A idéia é controlar a subida, descida e parada de um elevador pelo PC através da Linguagem LOGO.

O LOGO controlando aplicações do Basic Step

Neste artigo, o autor discute como controlar um servomotor através do PC usando a linguagem de programação LOGO. E para interfacear esse controle, é utilizado o Basic Step, visto que não é possível controlar o servo diretamente pela porta paralela.

Controles PWM de potência

Uma forma de obter um controle mais preciso sobre o torque e velocidade de motores de corrente contínua ou até de outras cargas, é através de circuitos que empregam a tecnologia PWM.