Motores de passo - LOGO - Futebol de robôs - Basic Step

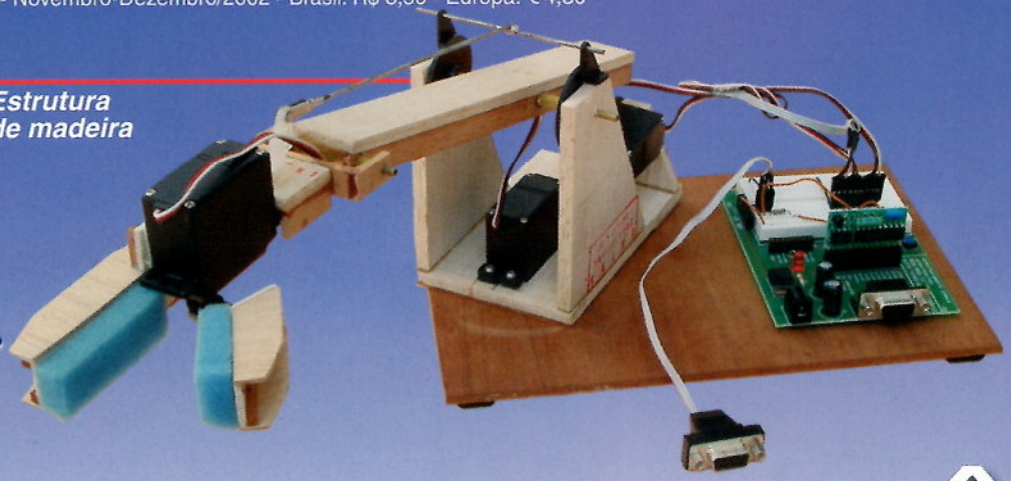
www.mecatronicafacil.com.br

MECATRÔNICA FÁCIL

Ano 2 - nº7 - Novembro-Dezembro/2002 - Brasil: R$ 8,50 - Europa: € 4,30



Aprenda a utilizar o paquímetro



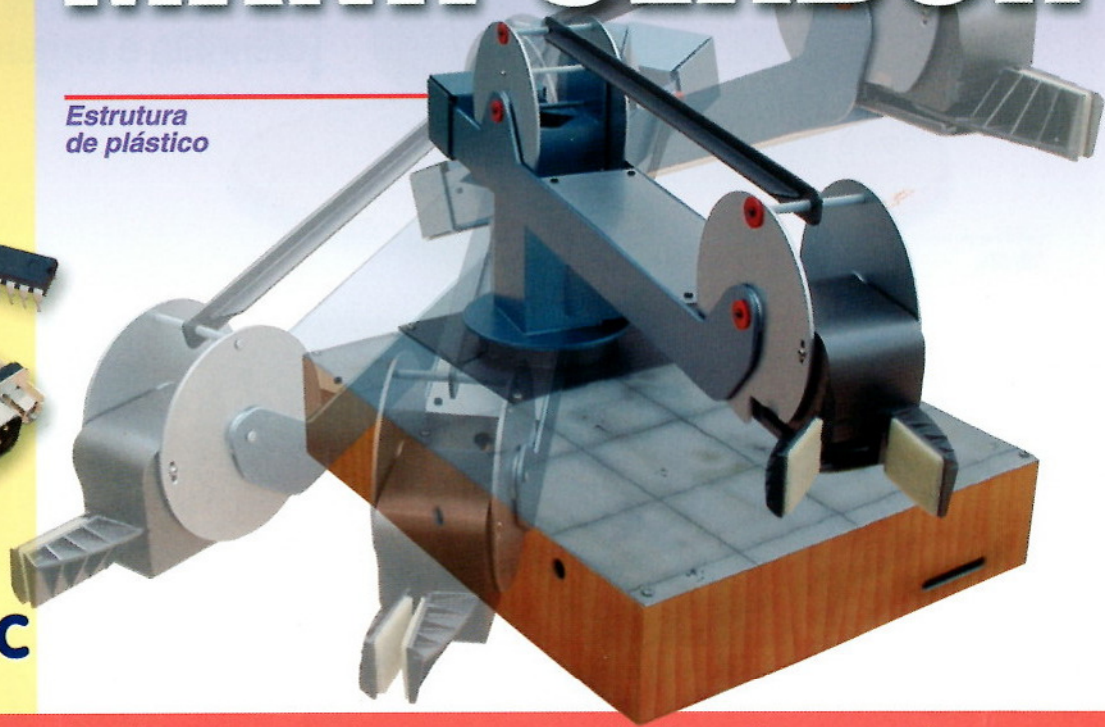
Estrutura de madeira

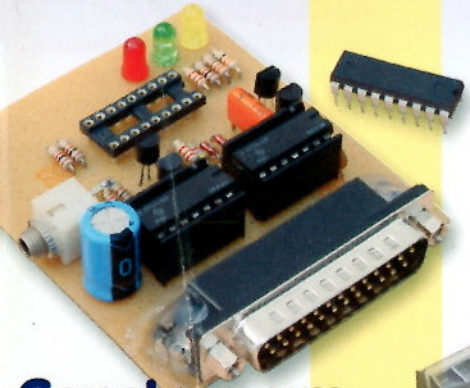


Gere energia com o LEGO Dacta

CONSTRUA UM ROBÔ MANIPULADOR

Estrutura de plástico





Construa um gravador de PIC

ISSN 1676-0980

9 771676 098004 07

Trabalhando com materiais alternativos

FEBRACE 2003
Feira Brasileira de Ciências e Engenharia

12 a 15 de março de 2003
Universidade de São Paulo, campus de cidade de São Paulo



ILUSTRAÇÕES
Drops

Pré-inscrição de Projetos
até 30 de novembro de 2002

Inscrição de Projetos
até 20 de janeiro de 2003

criatividade e inovação
FEBRACE
feira brasileira de ciências e engenharia

Informações:
site: www.lsi.usp.br/febrace
e-mail: febrace@lsi.usp.br
tel.: 0xx11-30915430

Para estudantes da 8ª série do Ensino Fundamental e estudantes do Ensino Médio e Técnico

ORGANIZAÇÃO E REALIZAÇÃO

PATROCÍNIO

APOIO NA REALIZAÇÃO

APOIO CULTURAL

1. O QUE É A FEBRACE?

A FEBRACE é uma feira anual de Ciências e Engenharia que envolve projetos de alunos das escolas públicas e particulares de todo o Brasil, em diversas categorias estabelecidas a partir das Ciências (Exatas e da Terra, Biológicas, da Saúde, Agrárias, Sociais e Humanas) e Engenharia e suas Aplicações.

A FEBRACE tem como principais objetivos estimular novas vocações em Ciência e Engenharia através do desenvolvimento de projetos criativos e inovadores, e aproximar as escolas públicas e privadas das Universidades, criando oportunidades de interação espontânea entre os alunos e professores das escolas com a comunidade universitária (estudantes universitários, professores, funcionários) para uma melhor compreensão dos papéis das Universidades em Ensino, Pesquisa, Cultura e Extensão.

2. QUEM PODE PARTICIPAR?

Estudantes da 8ª série do Ensino Fundamental e de Ensino Médio e Técnico das escolas públicas e particulares de todo o Brasil, não ultrapassando a idade máxima permitida de 21 anos (até o dia 1° de maio de 2003), que submetem seus projetos no prazo estabelecido e que têm seus projetos selecionados para participação.

Os estudantes podem participar individualmente ou em grupos de no máximo três alunos por projeto. Todos os projetos devem ter a participação de um adulto orientador, com mais de 21 anos de idade. Os projetos submetidos são avaliados por especialistas que selecionam os melhores projetos para participarem da FEBRACE.

3. QUAIS SÃO AS PREMIAÇÕES?

Todos os estudantes e professores orientadores de projetos que participam da FEBRACE recebem certificados de participação. Serão selecionados dentre os participantes, durante a FEBRACE, projetos para representarem o Brasil na feira internacional ISEF (International Science & Engineering Fair), os quais terão suas despesas com viagens aos E.U.A. e estadias pagas pela Intel, patrocinadora oficial da FEBRACE e da ISEF. São também convidados a visitarem a FEBRACE e a oferecerem premiações (computadores, PDA's, equipamentos científicos, bolsas de estudo, estágios, visitas técnicas, literatura, certificados), de acordo com seus respectivos critérios de avaliação, sociedades e associações científicas e profissionais, entidades governamentais e da iniciativa privada.

4. QUAIS SÃO AS CATEGORIAS?

OBSERVAÇÃO:
Todo projeto que envolva pesquisas com seres humanos e/ou animais e/ou substâncias controladas e/ou substâncias perigosas deverá ter aprovação prévia do instituto responsável pelo projeto. Esta aprovação deverá ser emitida por cientista(s) qualificados(s), antes de ser iniciada a pesquisa, e deverá acompanhar os demais documentos da inscrição.

Os projetos submetidos à FEBRACE devem pertencer a uma (ou mais) das seguintes categorias:

CIÊNCIAS EXATAS E DA TERRA: Matemática, Probabilidade e Estatística, Ciência da Computação, Astronomia, Física, Química, Geociências, Oceanografia.

CIÊNCIAS BIOLÓGICAS: Biologia Geral, Genética, Botânica, Zoologia, Ecologia, Morfologia, Fisiologia, Bioquímica, Biofísica, Farmacologia, Imunologia, Microbiologia, Parasitologia.

CIÊNCIAS DA SAÚDE: Medicina, Odontologia, Farmácia, Enfermagem, Nutrição, Saúde Coletiva, Fonoaudiologia, Fisioterapia e Terapia Ocupacional, Educação Física.

CIÊNCIAS AGRÁRIAS: Agronomia, Recursos Florestais e Engenharia Florestal, Engenharia Agrícola, Zootecnia, Medicina Veterinária, Recursos Pesqueiros e Engenharia de Pesca, Ciência e Tecnologia de Alimentos.

CIÊNCIAS SOCIAIS APLICADAS: Direito, Administração, Economia, Arquitetura e Urbanismo, Planejamento Urbano e Regional, Demografia, Ciência da Informação, Museologia, Comunicação, Serviço Social, Economia Doméstica, Desenho Industrial, Turismo.

CIÊNCIAS HUMANAS: Filosofia, Sociologia, Antropologia, Arqueologia, História, Geografia, Psicologia, Educação, Ciência Política, Teologia.

ENGENHARIAS: Eletrônica, Eletrotécnica, Mecânica, Química, Civil, de Minas, de Materiais e Metalúrgica, Sanitária, de Produção, Nuclear, de Transportes, Naval e Oceânica, Aeroespacial, Biomédica.

5. COMO SUBMETER UM PROJETO?

5.1 . TEMA:
Escolher um tópico para o projeto de pesquisa.

5.2 . REQUER APROVAÇÃO?
Se o projeto envolve uma das seguintes áreas:
- Pesquisa com seres humanos
- Animais vertebrados
- Recombinação de DNA
- Substâncias patogênicas ou microorganismos
- Tecido animal ou humano ou seus fluidos
- Substâncias ou equipamentos perigosos

Preencher e enviar um **Pedido de Aprovação** ANTES de iniciar a pesquisa. Aguardar o recebimento do código de aprovação do projeto para iniciar a pesquisa.

5.3. PESQUISA:
Conduzir a pesquisa seguindo o Método Científico ou o Método de Engenharia. Projetos em andamento que serão concluídos até o prazo de submissão ou projetos já completados podem também ser submetidos.

5.4. DIÁRIO DE BORDO:
Todo participante da feira deve elaborar, manter e trazer na feira um Diário de Bordo do Projeto.

No Diário de Bordo, o estudante deve registrar as etapas que realizou para desenvolver o projeto, relatando todos os fatos e as datas respectivas. O Diário de Bordo deve abranger o período relativo a todo o desenvolvimento do projeto e deve conter: o registro detalhado e preciso dos fatos, dos passos, das descobertas e das novas indagações, o registro das datas e locais das investigações; as entrevistas conduzidas, o registro dos testes e resultados alcançados.

5.5. PRÉ-SUBMISSÃO:
Preencher e enviar até 30 de novembro 2002 o formulário de **Planejamento de Projeto**.

5.6. SUBMISSÃO:
Preencher o formulário de **Informações do Projeto** e enviar com o **Relatório de Projeto** até o dia 20 de janeiro 2003.

6. MÉTODO CIENTÍFICO

6.1. ENUNCIAR O PROBLEMA OU A AFIRMAÇÃO
- Qual é seu objetivo?
- Qual é a idéia que você está tentando testar?
- Qual é a pergunta científica que você está tentando responder?

6.2. DESENVOLVA UMA HIPÓTESE
- Pense como seu projeto pode demonstrar seu propósito ou objetivo.
- Faça uma previsão dos resultados do experimento.
- Liste os resultados previstos em termos mensuráveis.

6.3. DESENVOLVA UM PROCEDIMENTO PARA TESTAR A HIPÓTESE
- Explique com detalhe como seu experimento será executado e como ele vai testar sua hipótese.
- Identifique as variáveis (elementos do experimento que mudam para testar a hipótese) e os controles (elementos do experimento que não mudam).
- Especifique como as medidas dos resultados vão provar ou refutar sua hipótese.
- Este procedimento deve ser como uma receita: uma outra pessoa deve poder executar o experimento seguindo o procedimento. Teste com um amigo ou parente para verificar que o procedimento está claro e completo.

6.4. MATERIAIS
- Liste os materiais e os equipamentos que serão utilizados.
- Esta lista deve incluir todos os ingredientes do procedimento.

6.5. OBSERVAÇÕES
- Registre sempre no Diário de Bordo do Projeto todas as observações, os dados e resultados. Estes podem ser medidas ou anotações sobre seu experimento.
- Fotografe os resultados de seu projeto ou as fases do mesmo, se possível e se isto ajudar sua análise ou a apresentação de seu experimento na feira.

6.6. ANÁLISE
- Explique suas observações, dados e resultados.
- Liste os pontos principais que você aprendeu.
- Porque você obteve estes resultados? O que seu experimento provou?
- Sua hipótese estava correta? Seu experimento provou ou refutou sua hipótese? Explique em detalhes.

6.7. CONCLUSÃO
- Responda ao problema ou à afirmação elaborados.
- Qual é o valor de seu projeto?
- Dados os resultados de seu experimento, qual seria a próxima pesquisa a ser desenvolvida? Qual seria a próxima pergunta que deveria ser feita?
- Se você tivesse que refazer a pesquisa, o que você mudaria?

7. MÉTODO DE ENGENHARIA

7.1. RECONHECER NECESSIDADES
- Observa o mundo ao seu redor.
- Quais são os problemas de sua comunidade?
- Quais são as necessidades?
- O que poderia ser melhorado?

7.2. DEFINIR O PROBLEMA
- Defina o problema escolhido.
- Descreva o problema em linhas gerais.
- Reduza o problema focando em um aspecto específico.

7.3. PROPOR ALTERNATIVAS DE SOLUÇÃO
- Pense em algumas alternativas de soluções que você encontrou para resolver o problema.

7.4. AVALIAR AS ALTERNATIVAS DE SOLUÇÃO
- Reflita sobre as soluções que você encontrou.
- Avalie as alternativas. O que elas têm de bom, o que elas têm de ruim?
- Qual seria o impacto delas na comunidade?
- Qual seria a dificuldade de implementar estas alternativas?

7.5. SELECIONAR A ALTERNATIVA PREFERIDA
- Escolha uma solução.
- Justifique sua escolha.

7.6. ESPECIFICAR A SOLUÇÃO E COMUNICAR O PROJETO
- Especifique o seu projeto.
- Faça sua pré-submissão a FEBRACE preenchendo e enviando o Planejamento de Projeto.

7.7. IMPLEMENTAR (FABRICAR E DISPONIBILIZAR) A SOLUÇÃO
- Implemente sua solução.
- Teste seu projeto.
- Submeta a FEBRACE redigindo, preenchendo e enviando o Relatório de Projeto.

OBSERVAÇÕES
- Registre sempre no Diário de Bordo do Projeto todas as observações, os dados e resultados. Estes podem ser medidas, descrições ou anotações.
- Fotografe os resultados de seu projeto ou as fases do mesmo, isto pode ajudar a documentar e a apresentar seu projeto na feira.

Informações mais detalhadas no site: www.lsi.usp.br/febrace

Nº 7 - NOVEMBRO/2002

ÍNDICE

Leitor 4

Robonews 6

Medindo com o paquímetro 07

Ao ver alguém utilizando um paquímetro, é curioso saber como se pode ler uma medida tão pequena em uma escala tão repleta de números. Neste artigo, o autor mostra de maneira clara como se utiliza um paquímetro, instrumento de grande importância para medir comprimentos com grande precisão.

Trabalhando com materiais alternativos II 16

Veja, neste artigo, algumas dicas de como trabalhar com garrafas ou recipientes plásticos descartáveis, aproveitando sua forma previamente moldada nos projetos de mecatrônica.

RM-1 - Robô Manipulador 18

Construa um braço mecânico, passo-a-passo, capaz de manipular pequenas "cargas". Todo o controle é feito por um microcomputador, aumentando ainda mais sua possibilidades. Apresentamos sua construção em duas versões: uma com material de fácil acesso e outra com plástico. Assim, o leitor poderá escolher a versão que julgar mais conveniente.

Conversão de energia e transmissão de movimento com LEGO 32

Neste artigo, o autor mostra como utilizar peças do kit da LEGO Dacta associadas a outros materiais eletrônicos para demonstrar a geração de energia elétrica (pela transformação da energia mecânica) a fim de acender pequenas lâmpadas, LEDs ou até mesmo para armazenar esta energia num capacitor.

O software que faz os robôs jogarem futebol 38

Ainda que o aparato eletrônico e mecânico seja essencial, de fato, a criação do futebol de robôs teve muito a ver com a necessidade de fomentar o desenvolvimento de técnicas sofisticadas e avançadas em diversas áreas da Ciência da Computação. Conheça o sistema de software por trás dos robôs-jogadores.

Como gravar um microcontrolador PIC 43

O uso dos microcontroladores é crescente e o estudante/hobista/técnico não pode ficar de fora. Aprenda como gravar um microcontrolador PIC com o auxílio das ferramentas Microchip e do projeto PROGPIC-1, que é um gravador de baixo custo.

Automação Simples 50

O real poder do microcontrolador está em sua habilidade de tomar decisões baseadas em suas "entradas" e, então, manipular o "mundo real" com suas saídas. Neste artigo, o autor explora este conceito propondo uma automação simples utilizando o Basic Step.

Registrando eventos com o LOGO 54

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.

Motores de passo 58

Estes motores são ideais para casos onde a precisão seja um requisito importante do projeto. Saiba como funcionam esses motores e saiba também como utilizá-los em projetos de Mecatrônica.

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
Gabriel dos Santos
José Francci Júnior
Luís Henrique C. Bernardes
Márcio José Soares
Newton C. Braga
Octávio Nogueira
Wilson Roberto dos Santos

ASSINATURAS
e números atrasados
www.mecatronicafacil.com.br
Fone/Fax: (11) 6195-5335
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) 6195-5333

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



A revista "Mecatrônica Fácil" está comemorando 1 ano! Foi um ano de muitos desafios e de muitas vitórias. Desbravamos um mercado carente de publicações no segmento de Mecatrônica trazendo muitos artigos interessantes e úteis. Sabemos que ainda há o que melhorar. Nosso empenho nas próximas edições estará voltado para atender cada vez mais as necessidades dos nossos leitores. Para isso, é fundamental que vocês continuem interagindo conosco através da "Seção do Leitor".

E para comemorar, nada como publicarmos o projeto mais solicitado pelos leitores: o robô manipulador! Muitos o chamam de braço mecânico, ou braço mecatrônico, devido a sua capacidade de se mover e "pegar" objetos. Adotamos o nome de robô manipulador, pois é como equipamentos semelhantes a este (numa escala bem maior!) são conhecidos na indústria. É um projeto complexo, se compararmos com os demais projetos que publicamos até aqui, mas a satisfação de se conseguir montá-lo certamente compensará as eventuais dificuldades.

Dando seqüência à série de artigos sobre os microcontroladores PIC, neste número você vai conferir como gravar este microcontrolador e mais: vai construir o seu próprio gravador! E na série sobre Mecânica, você aprenderá como utilizar o paquímetro, ferramenta de fundamental importância para medir peças com precisão. Nesta edição também, você terá algumas dicas de como trabalhar com garrafas plásticas descartáveis nos seus projetos, conhecerá como funcionam os motores de passo, e muito mais!

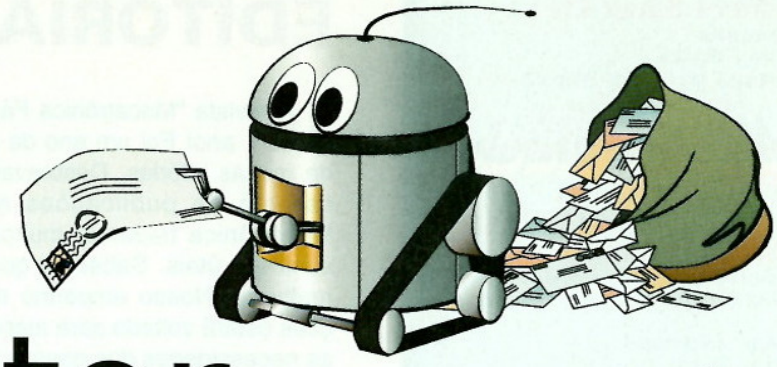
A Editora Saber está apoiando a realização da FEBRACE – Feira Brasileira de Ciências e Engenharia que ocorrerá do dia 12 a 15 de março do próximo ano. Um dos principais objetivos desta feira é estimular a formação científica dos estudantes de 8ª série do Ensino Fundamental e também dos estudantes do Ensino Médio e Técnico. Gostaríamos de reforçar o convite feito pelo LSI a todas as escolas do Brasil para que participem desta feira e que os professores estimulem seus alunos a desenvolverem projetos baseados na metodologia científica e de engenharia para que eles compreendam o real significado destas áreas do conhecimento. Seja um dos seis vencedores que representarão o Brasil na feira internacional que ocorrerá nos Estados Unidos, com despesas de viagens e estadias pagas, e vamos mostrar que no nosso país também se produz ciência de qualidade.

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.

Seção do Leitor



Código-Fonte do Elevador

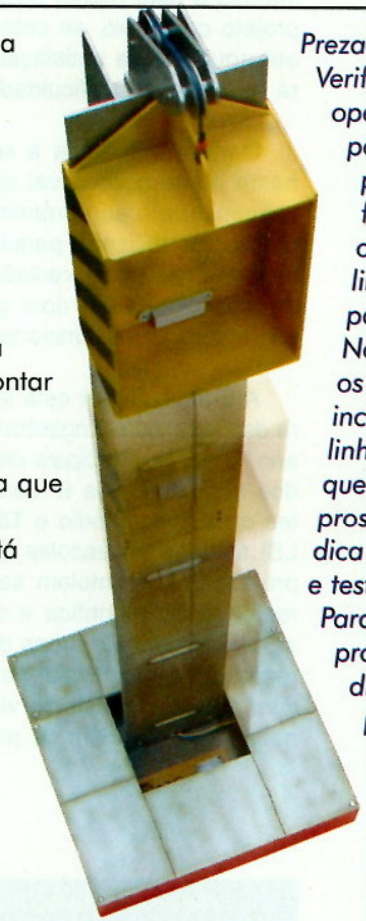
Comprei a edição número 6 da revista e na parte da programação do elevador está acontecendo um erro: quando peço para executar o "elevador", o programa LOGO está sendo terminado, sem que nada aconteça. Eu gostaria que verificassem o código.

Marcelo Sperandio Fanchiotti

Meu nome é **Bruno** e eu comprei a revista Mecatrônica Fácil N° 6, a que ensina a montar um elevador.
Eu não estou conseguindo programar o sistema pelo LOGO. Existe outro programa que eu possa usar, ou outro código-fonte? O problema é que o comando "para" não está funcionando...

Gostaria que me enviassem o código-fonte 1 do programa do elevador, pois foi feito o projeto, a placa eletrônica mas, mesmo digitando o programa, não entram os botões para efetuar o teste.

Rosana César



Prezados Marcelo, Bruno e Rosana,
Verifiquem se o seu sistema operacional e' Win95 ou Win98, pois os comandos da porta paralela no LOGO não funcionam em Win2000 e WinNT ou XP. Para isso, comentem as linhas que escrevem na porta paralela.
Nos seus programas verifiquem se os digitaram corretamente, inclusive os acentos. Algumas linhas são "quebradas" por ~ (til), que indica que o comando prossegue na próxima linha. Outra dica é dividir o programa em partes e testar cada parte em separado.
Para ajudar, o código-fonte do programa em LOGO está sendo disponibilizado no nosso site para download no endereço:
www.mecatronicafacil.com.br\downloads
Boa Montagem!

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

ERRAMOS:

Na edição n° 5, página 43, figura 8, houve um erro na numeração dos pinos. Onde está 6 é 7 e onde está 7 é 6 (foram invertidos).

Elevador controlado pelo LOGO

Olá! Meu nome é Peter, gostaria de esclarecer algumas dúvidas que tive em relação ao elevador da revista número 6 ("Mecatrônica Fácil"). Eu não encontrei exatamente o reed-relé que está na revista, e gostaria de saber se posso utilizar outro (Ampola REED, Tensão de comutação 200 V, 500 mA, resistência residual do contacto 150:mW). Outra dúvida é sobre o relé mencionado, vocês podem me informar, com mais detalhes, que tipo de relé que foi usado. E, por último, gostaria de saber se existe algum lugar aonde eu posso encontrar com facilidade esses pequenos motores utilizados em robótica. Agradeço desde já pela sua atenção!



Caro Peter,
*Você pode utilizar outro reed-relé no seu projeto. Faça o teste com um ímã e veja se funciona. Depois disso, é só colar na haste do elevador! Quanto ao relé, utilizamos um de 6 volts na bobina para ser alimentado com bateria (4 pilhas). A Saber Marketing (**www.sabermarketing.com.br**) dispõe para venda a caixa de redução utilizada no elevador.*

Novos Microcontroladores

Adquiri a revista "Mecatrônica Fácil" número 6, e gostei bastante dos artigos e da forma como os assuntos são abordados. Estou desenvolvendo algumas aplicações com os microcontroladores PIC, e gostaria de sugerir que nas próximas edições, vocês passassem a abordar os modelos 16f627 e 16f628, pois eles foram desenvolvidos pela Microchip para substituir o 16f84, que não está sendo produzido mais e logo estará em falta no mercado, dificultando aos leitores que não têm essa informação, na aquisição dos microcontroladores, bem como não lhes permitindo que utilizem todos os recursos a mais que a nova linha de PICs está oferecendo.
Atenciosamente,
Marcos

Caro Marcos,
Os microcontroladores 16F62X não são substitutos do 16F84, constituindo sim um aprimoramento. A Microchip garante o fornecimento de seus microcontroladores (o antigo 16C54C é encontrado muito facilmente em lojas e distribuidores) e atualmente nas lojas especializadas o mais comum ainda é o 16F84.
A série 16F8X é igual em comandos à 16F62X. A diferença está no número de comandos e registradores da versão mais nova, que é maior. Portanto, todos os que aprenderem a programar com um PIC16F84 estarão aptos a desenvolverem aplicações com qualquer microcontrolador Microchip. Porém, no final da série o autor escreverá sobre os novos modelos, não somente da série 16F62X como também da série 16F87X maior e mais poderosa, promovendo uma migração mais fácil.

Lista de Material do ADR-1

Caro Senhores da Editora Saber,

Meu nome é Diego Santana, e sou assinante da revista "Mecatrônica Fácil". Gosto muito disso, e por esse e outros motivos, eu e meus amigos resolvemos apresentar o robô ADR-1 na Feira de Ciências 2002.
Estamos construindo-o para a feira, que se realizará em 2 meses, em nosso colégio (Colégio Cenecista Presidente Kennedy-PR), porém encontramos algumas dificuldades sobre quais são os componentes usados para a construção?

Procuramos na revista ("Mecatrônica Fácil" nº 5, Julho - Agosto de 2002), mas percebemos que para esse robô não veio acompanhado impresso a lista de componentes. Solicitamos, se possível, a lista de componentes para o ADR-1 urgentemente! Atenciosamente,
Diego Santana

Caro Diego,
Perdoe-nos a falha. A lista de componentes do ADR-1 está disponível no endereço:
www.mecatronicafacil.com.br\downloads.
Esperamos que o seu projeto seja um sucesso.

ROBONEWS

Newton C. Braga

Cristais podem memorizar sons

Cristais ferroelétricos podem memorizar sons.Para provar isso, pesquisadores aplicaram ondas acústicas em cristais de niobato de lítio e observaram que o tom enviado apareceu reproduzido um certo tempo depois. A equipe de pesquisadores da Universidade do Mississipi publicou a descoberta em setembro último, esperando que ela possa ter muitas utilidades.

Uma possível aplicação é a realização de testes não destrutivos em materiais do tipo usado em dispositivos de telecomunicações. As propriedades ferroelétricas são semelhantes ao ferromagnetismo. Aquecendo-se acima de certa temperatura e depois esfriando-se vagarosamente na presença de um campo elétrico, o material se organiza de modo a produzir depois um campo elétrico. Os domínios polarizados de átomos se alinham gradualmente, e criam um campo que pode ser "lembrado" depois que o campo aplicado for desligado.

Trabalhando com um sanduíche de $LiNbO_3$ entre dois eletrodos, o pesquisador Mack Breazeale enviou um sinal acústico através do material por meio de um eletrodo e pretendia captá-lo no outro eletrodo para realizar medidas de intensidade. Mas, para a sua surpresa, um sinal com a mesma freqüência e fase foi captado nos eletrodos 70 microssegundos depois do sinal original. Aparentemente, o material armazenou energia do sinal acústico para depois "soltá-lo" após um retardo. A freqüência usada no experimento foi de 26 MHz.

Robô "Roomba" aspira o pó de sua casa

O novo robô aspirador de pó chamado Roomba,trabalha aspirando o pó de sua casa enquanto você descansa. Analisando a aparência desse robô, ele nada tem a ver com o R2-D2 ou a empregada-robô dos Jetsons, chamada Rosey. Roomba é bem diferente, mas funciona.

Criado pelo laboratório de Inteligência Artificial do MIT (*Massachusetts Institute of Technology*), o iRobt chamado Roomba tem por função aspirar pó. O ponto de destaque, é que trata-se do primeiro robô projetado para conviver com você em sua casa. O atraente é que ele custa aproximadamente o mesmo que um aspirador (199 dólares nos Estados Unidos), porém faz tudo sozinho. Essa é a idéia básica do projeto: um robô aspirador deve ser simples e ter condições de conviver com os humanos. Além disso, ninguém vai querer pagar mais de 1 000 dólares por um aspirador de pó que exige conhecimentos de engenharia para ser manuseado...

Depois de 12 anos de pesquisa, saiu o Roomba que pesa aproximadamente 2,5 kg e mede aproximadamente 35 cm de diâmetro, e se movimenta em espiral pelos ambientes aspirando pó. Ele usa regras heurísticas para navegar e sabe evitar obstáculos. Baseado em robôs "caçadores de minas", ele avalia primeiro o ambiente em que se encontra determinando a posição dos obstáculos para depois realizar sua tarefa percorrendo todos os locais possíveis da sala onde deve aspirar o pó.

A venda deve ser feita pelo *site* www.roombavac.com brevemente.

Medindo com o paquímetro

Wilson Roberto dos Santos

O leitor já deve ter visto alguém tentando medir a boca de um tubo com uma régua de plástico, no clássico (e assustador) estilo: fecha um olho, posiciona bem o outro, leva a peça lá longe, traz perto do olho (como quem procura o foco perfeito) e....vamos lá.....este tubo tem.....2,2 centímetros de boca!!!
Meu Deus!!!! Isto dói!!! Dói ao ouvir, não é?
Claro que dói, tá tudo errado!!!
Primeiro, não se mede secção circular com uma régua.
Segundo, a medida não é dada em *cm*, mas sim em *mm* ou *polegadas*.
Terceiro, o que se pretendia medir no tubo não é a boca, mas sim o "diâmetro", que nada mais é do que a medida linear das extremidades de qualquer "corda" que passe pelo centro de um círculo, ou secção de um elemento cilíndrico. Também poderíamos dizer que se trata da multiplicação do "raio" (que é um segmento de reta que vai de uma circunferência ao seu centro) por dois. Mas chama-lo-emos de diâmetro, que é o mesmo.
O diâmetro, a corda e o raio? Já, já falaremos mais sobre eles.



INSTRUMENTOS DE MEDIÇÃO

O **Sistema Internacional de Medidas** que abordamos na edição anterior, trouxe uma boa base para falarmos de medições, precisão e tolerâncias.

Agora, vamos aprender como é possível medir e trabalhar com tanta precisão quanto desejarmos, sem medos e problemas.

Vamos conhecer e aprender a utilizar nossos aliados fiéis: **os instrumentos de medição.**

A forma correta de medir é sempre utilizando o instrumento de medição adequado à medida que se pretende tomar.

Os instrumentos de medição são projetados para permitir aos usuári-



Figura 1 – Paquímetro com nônio.

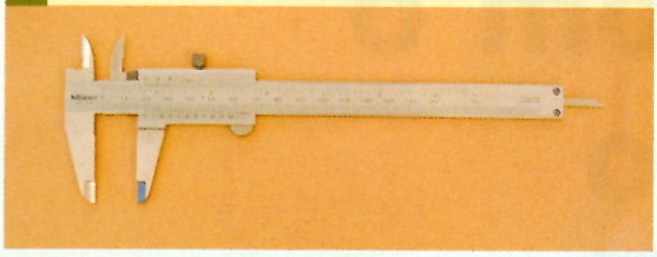
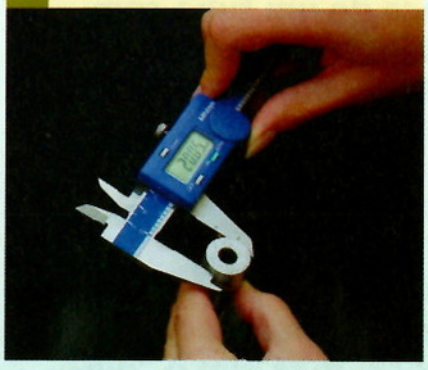


Figura 1A – Paquímetro digital.



os a tomada e / ou aferição de medidas com precisão, portanto **são** também **instrumentos de precisão** e como tais devem ser tratados e mantidos adequadamente.

Existem muitos instrumentos de medição, e o primeiro sobre o qual iremos falar é o **paquímetro**.

CONHECENDO O PAQUÍMETRO

Certamente você já deve ter visto alguém utilizando um paquímetro e talvez tenha ficado curioso para saber como se pode ler uma medida tão pequena em uma escala tão repleta de números.

Pois bem, chegou a hora de acabar com essa dificuldade, vamos entender definitivamente como se utiliza um paquímetro.

Há vários tipos de paquímetros, desde os simples, os digitais, para grandes dimensões, paquímetros de profundidade, de espessura, alguns construídos para finalidades específicas, articulados, fixos e diversos outros. Veja exemplares nas **figuras 1 e 1-a**.

Há também diferenças na resolução dos paquímetros.

Resolução é a menor medida que se pode medir/aferir com um instrumento de medição.

Os paquímetros mais comuns têm resolução de 0,05 mm. Há paquímetros com resolução de 0,02 mm e os especiais fabricados por algumas poucas empresas que têm resolução de 0,01 mm.

Em nosso trabalho utilizaremos um paquímetro de 150,0 mm, com precisão de 0,05 mm, adequado para a maioria das operações mecânicas de precisão média.

O Paquímetro ou *"Calibre de Vernier"* é o mais versátil instrumento manual de medição. É fabricado em aço inoxidável, titânio ou outros materiais não oxidantes, com medidas muito precisas. Sua escala é gravada em baixo relevo de modo a evitar o desgaste dos números com o uso e, normalmente, as gravações são salientadas em preto para facilitar a leitura.

O sistema de gravação a *laser* é largamente utilizado na fabricação de paquímetros.

Com o paquímetro podemos medir dimensões lineares, internas, externas, profundidade, espessuras e, obviamente, o diâmetro.

Enfim, qualquer dimensão linear pode ser medida com o uso de um paquímetro, desde que não exceda a sua resolução nominal.

Na **figura 2** são mostrados os componentes do paquímetro.

Figura 2 – Componentes do paquímetro.

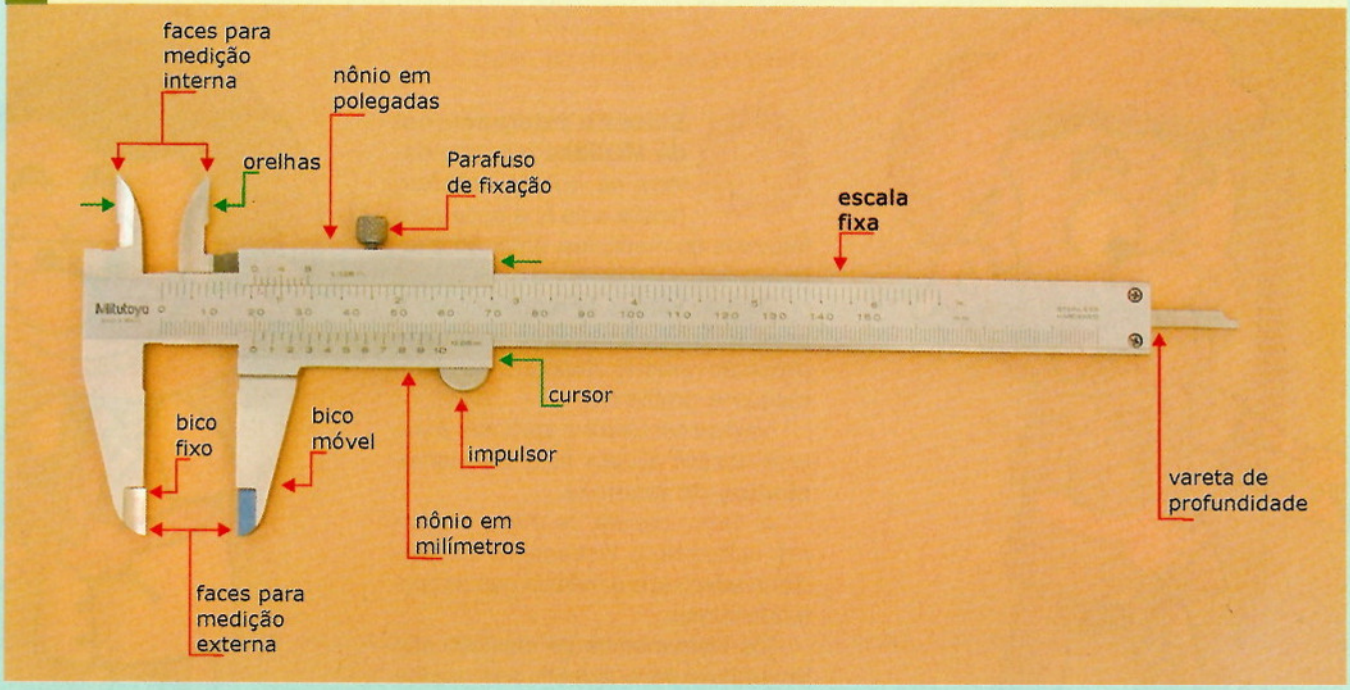
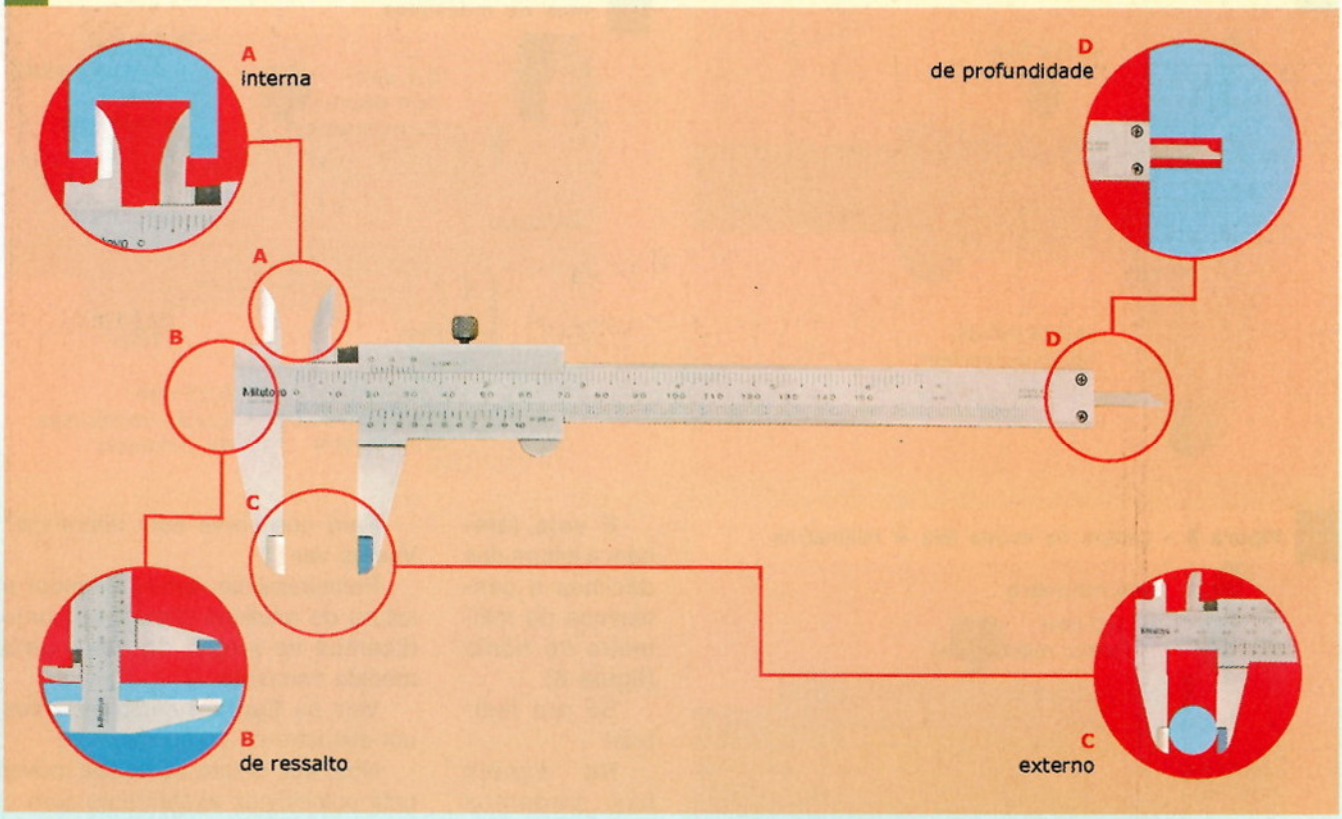


Figura 3 – Possibilidade de medida com um paquímetro.



Quando há necessidade de maior resolução e precisão, entram em cena outros instrumentos, entre eles o **Micrômetro**, que veremos na próxima edição.

Na **figura 3** estão descritas as quatro formas de utilização do paquímetro, e é importante observar que ao movermos o *vernier* tudo se move em conjunto. Sejam as orelhas para medição interna (a), os paralelos para medição de ressaltos (b), os bicos para medidas externas (c) ou a haste para medidas de profundidade (d), todas as dimensões são simultâneas, pois o deslocamento é de todo o conjunto, visto que todos estes componentes estão interligados.

É PAQUÍMETRO OU CALIBRE DE VERNIER?

Tanto faz!

Paquímetro vem de *paqui + metro*. O metro você já sabe, vem de *metron* que significa "o que mede", e *paqui*, vem do grego *"Pachys"* que significa espesso ou grosso. Então, paquímetro é essencialmente um medidor de espessura ou grossura e, portanto, um tipo de calibrador.

É chamado "Calibre de Vernier" (lê-se cálibre) em homenagem ao matemático e cartógrafo francês Pierre Vernier. Ele inventou em 1631 um instrumento para medir ângulos, distâncias e alturas, baseado em uma escala móvel deslizante, aplicando na prática o conceito do "Nônio", criado por João Pedro Nunes por volta de 1542.

Por esta razão, alguns denominam Nônio e outros Vernier, mas os dois estão corretos.

Vernier não foi o inventor do paquímetro, mas foi o primeiro a aplicar o conceito do nônio em um instrumento prático.

O conceito de Nônio criado por Pedro Nunes foi tão útil que ultrapassou barreiras e foi considerado perfeito para tomada de medidas precisas em muitos segmentos, inclusive na mecânica, note que o utilizamos até hoje tal qual foi concebido!

O PRINCÍPIO DE FUNCIONAMENTO *(princípio do nônio)*

Há dois tipos de escalas muito utilizadas em paquímetros: nônio com escala de 10 traços e nônio com escala de 20 traços.

Vamos trabalhar com o nônio de 20 traços, pois é mais legível e ameniza os erros por paralaxe.

Nônio (ou vernier) é uma escala (móvel) secundária acoplada a uma escala (fixa) principal que permite obter medidas menores do que a menor divisão desta escala principal.

Xí! Menor do que o menor .....parece difícil!

Sem medo, vamos explicar.

Como pode ser visto na **figura 4**, o paquímetro possui duas escalas, uma fixa e outra móvel.

Na escala fixa a menor divisão é 1,0 mm, a do nônio (por construção) é 9/10 da escala fixa. Conseqüentemente, quando os zeros das duas escalas coincidem, a distância entre o 1° traço da escala fixa e o 1° traço do nônio é de 0,05 mm, entre o

Figura 4 – Exemplo de nônio de 20 traços.

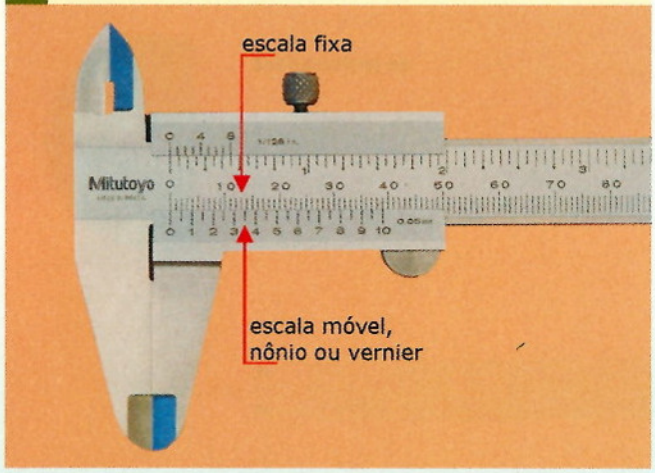


Figura 6 – Leitura no nônio = décimos e centésimos de milésimos.

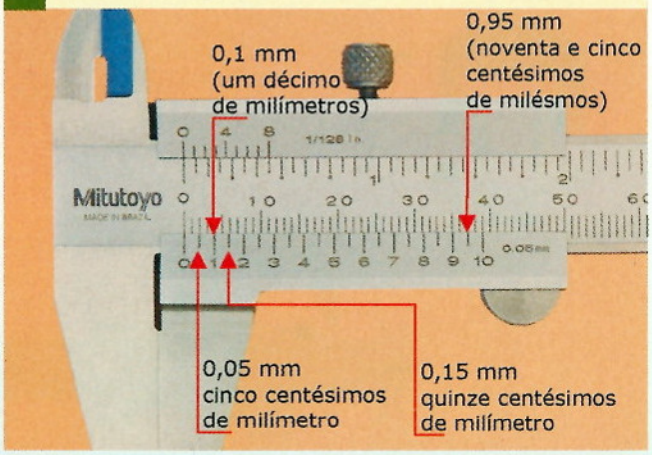
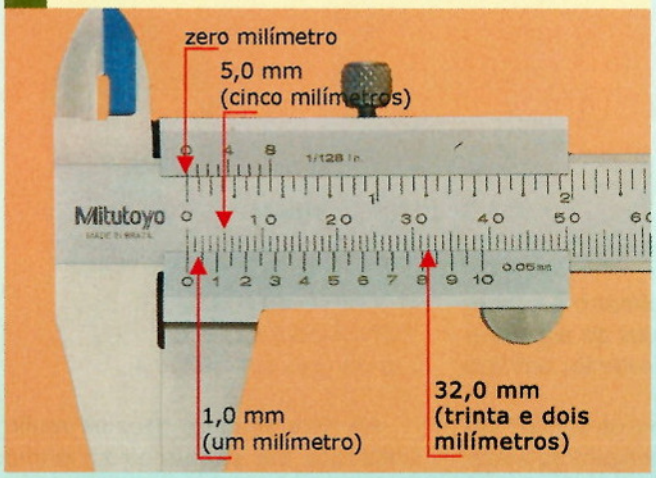


Figura 5 – Leitura na escala fixa = milímetros



1° traço da escala fixa e o 2° traço do nônio é de 0,1mm e assim por diante até 0,95 mm. Isso permite que se avalie até 0,05 mm, pois a diferença entre a escala e o nônio é de 0,05 mm.

Em outras palavras: *Na escala fixa lemos os milímetros e no nônio lemos as frações de milímetro.*

Este é o princípio do Nônio de nosso amigo português João Pedro Nunes, aplicado ao calibre de Pierre Vernier!!!

Como o próprio nome diz, o nônio de 20 traços, possui vinte divisões, onde se pode ler uma escala de 0 a 10.

Cada intervalo entre os traços simboliza 0,05 mm (cinco centésimos de milímetro).

Observe na **figura 5**, a leitura dos milímetros na escala fixa.

E veja também a leitura dos décimos e centésimos de milímetro no nônio (**figura 6**).

Só pra lembrar!

Na escala fixa podemos ler os milímetros, enquanto que na escala móvel lemos os décimos e centésimos de milímetro.

A leitura da medida é feita lendo-se primeiro os milímetros e depois as suas frações.

Primeiras Medidas

Os dois "zeros" das escalas fixa e móvel, devem coincidir perfeitamente quando o paquímetro estiver fechado, ou seja, com os bicos de leitura encostados.

Exatamente como está na **figura 7**, a seguir.

Observe que nenhum outro traço coincide com qualquer traço da escala fixa, apenas os dois "zeros".

Observou? Interessante, não é?

Apenas um traço da escala móvel poderá coincidir com qualquer um dos traços da escala fixa no momento da tomada da medida, não será possível que haja dois traços coincidentes.

Para que serve esta diferença? Vamos ver.

Primeiramente vamos entender a leitura de medidas cheias, ou como dizemos no jargão da área, uma medida zero/zero.

Veja na **figura 8** como medimos um diâmetro de 20,00 mm.

Note que o zero da escala móvel está coincidindo exatamente com o traço que demarca 20,0 mm na escala fixa, portanto o diâmetro que estamos medindo tem 20,0 mm .

Vamos agora medir outro diâmetro, porém desta vez de 20,05 mm. Observe na **figura 9**.

Veja a leitura no nônio:

Note que o zero da escala móvel está "depois" do traço que demarca os 20,0 mm da escala fixa, logo a peça que está sendo medida tem **mais** do que 20,0 mm.

Mas, quanto mais do que 20,0 mm tem esta peça?

Para encontrar a medida exata comece a correr os olhos da esquerda para a direita sobre o nônio até que um traço deste esteja coincidindo com um traço da escala fixa.

Em nosso caso, o traço coincidente do nônio é o que demarca 0,05 mm, portanto, somamos os 20,0 mm da escala fixa + 0,05 mm do nônio = 20,05 mm.

Vamos a mais um exemplo. Veja a **figura 10**.

Note que o zero do nônio está "depois" do traço que indica 3,0 mm, portanto a peça tem **mais** do que 3,0 mm.

Figura 7 – Coincidência dos dois "zeros" das escalas.

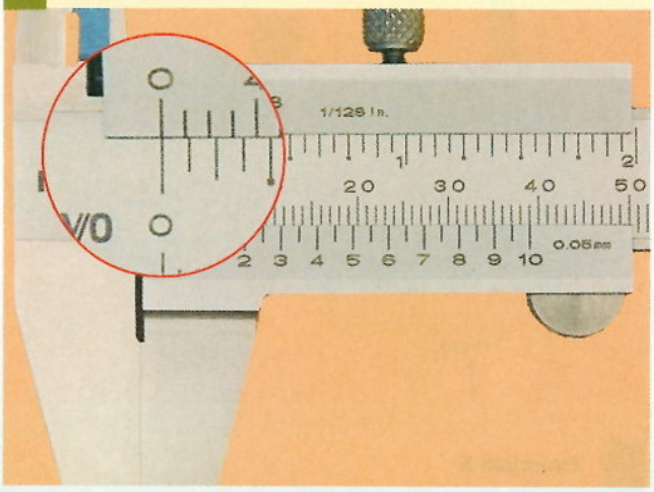


Figura 10 – Medida de 3,20 cm.

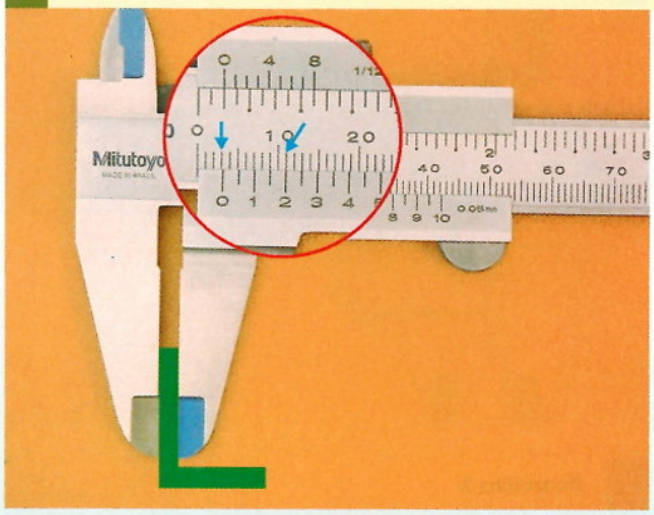


Figura 8 – Medida de diâmetro de 20 mm.

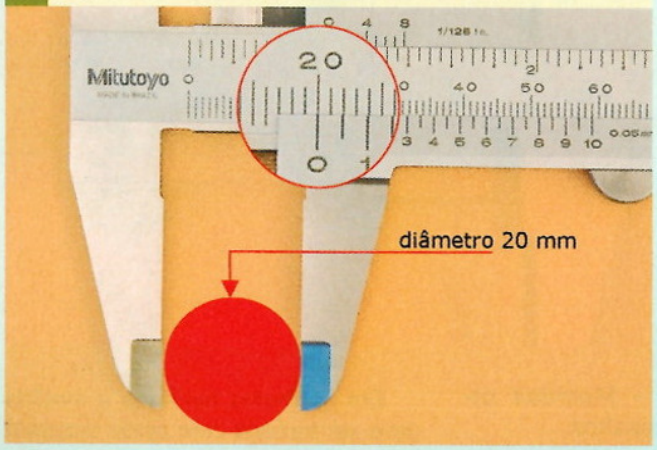


Figura 11 – Medida menor que 1,0 mm (0,55 mm).

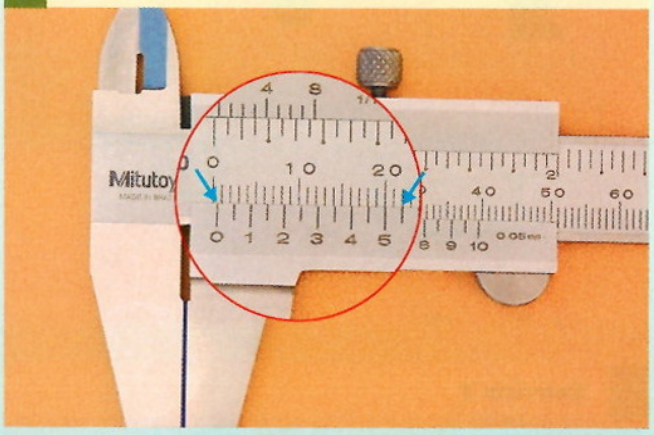
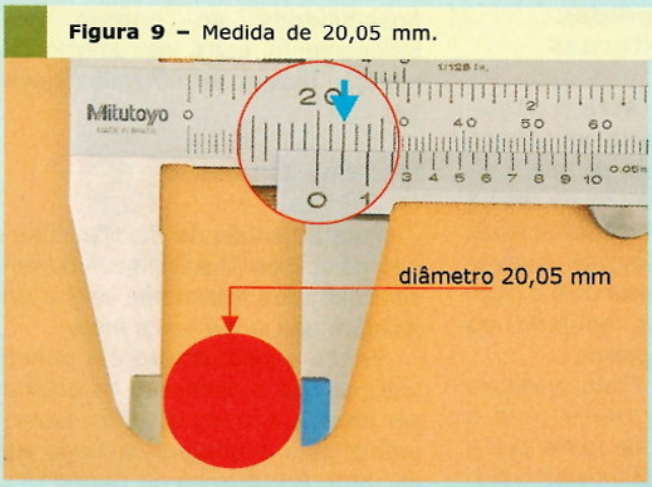


Figura 9 – Medida de 20,05 mm.



Quanto mais?

Vamos novamente buscar o traço coincidente, sempre da esquerda para a direita.

Em nosso caso o traço coincidente é o que indica 0,2 mm (dois décimos de milímetro). Assim, somaremos 3,0 mm (da escala fixa) + 0,2 mm (da escala móvel) = 3,20 mm

Está ficando interessante?

Vejamos mais um exemplo (**figura 11**):

E agora?

O zero da escala móvel nem mesmo chegou a um milímetro!

Isto significa que o objeto medido tem **menos** do que um milímetro.

Sua medida será encontrada usando a mesma técnica de busca dos traços coincidentes, e neste caso o traço coincidente é o que demarca os 0,55 mm (cinqüenta e cinco centésimos).

Portanto, nossa peça tem apenas 0,55 mm.

Simples, não é?

Vamos exercitard os conhecimentos adquiridos.

Faça a leitura dos **exercícios 1 a 5** da próxima página e anote, e depois confira nas respostas corretas.

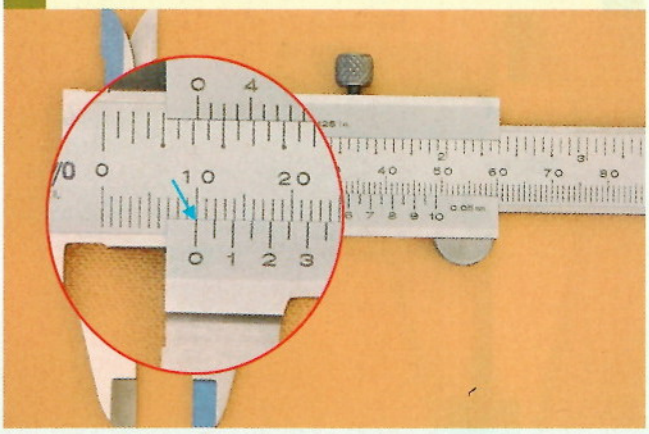
TÉCNICAS DE MEDIÇÃO

Agora que você já sabe como ler as medidas em um paquímetro, vamos aprender a medir com ele.

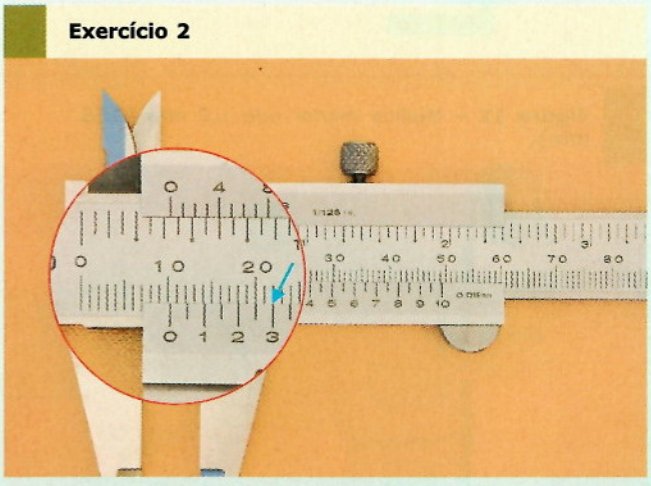
Para medir corretamente é importante observar que:

1. Os encostos, orelhas e haste de leitura devem estar limpos.

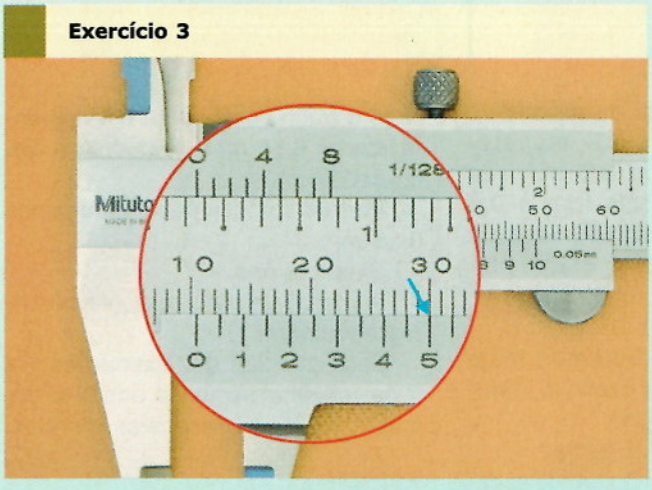
Exercício 1



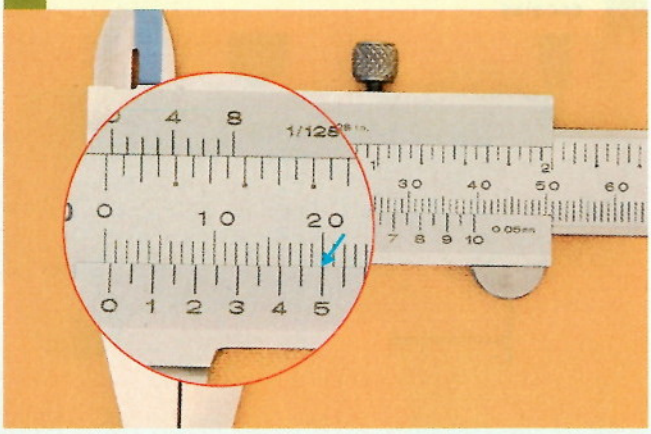
Exercício 2



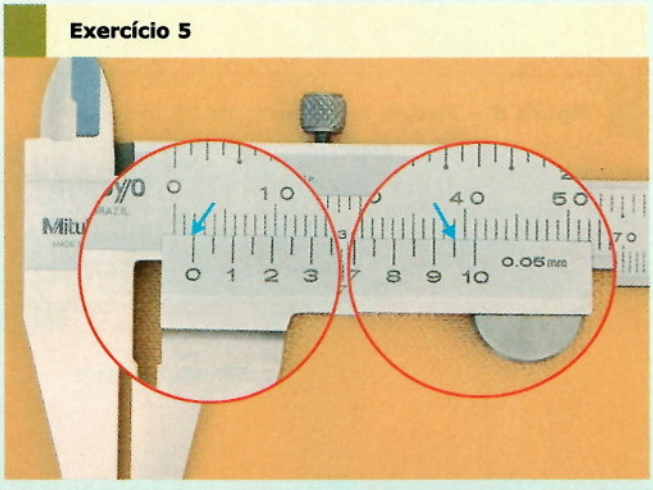
Exercício 3



Exercício 4



Exercício 5



2. A peça deve ser posicionada corretamente e o paquímetro também.

Veja agora as técnicas de medição para:

- Medidas externas
- Medidas internas
- Medidas de profundidade
- Medidas de ressaltos.

Medidas externas

Abra o paquímetro um pouco mais do que a dimensão da peça a ser medida e mova o cursor suavemente com o polegar apoiado no impulsor.

Faça a leitura de frente para o paquímetro e abra-o de forma que a peça seja retirada sem "raspar" o bico do paquímetro. Atente para a **figura 12**.

A peça a ser medida deve ser colocada o mais próximo possível da base do bico fixo, para evitar desgastes por uso nas pontas (**figura 13**).

Em nenhuma hipótese a medida deve ser tomada com apoio incorreto dos bicos ou orelhas; em caso de impossibilidade do correto posicionamento, outro instrumento de medição deverá ser utilizado (**figura 14**).

Medidas Internas

Para a tomada de medidas internas, as orelhas do paquímetro devem ser colocadas totalmente dentro da abertura que se pretende medir.

Feche o paquímetro um pouco mais do que a dimensão da peça a ser medida e mova o cursor suavemente com o polegar apoiado no impulsor. Acompanhe na **figura 15** .

As orelhas do paquímetro devem ser posicionadas paralelamente às paredes do furo a ser medido. Observe a **figura 16**.

Nos casos em que se precisar medir furos redondos ou seja, diâme-

Figura 12 – Medidas externas.

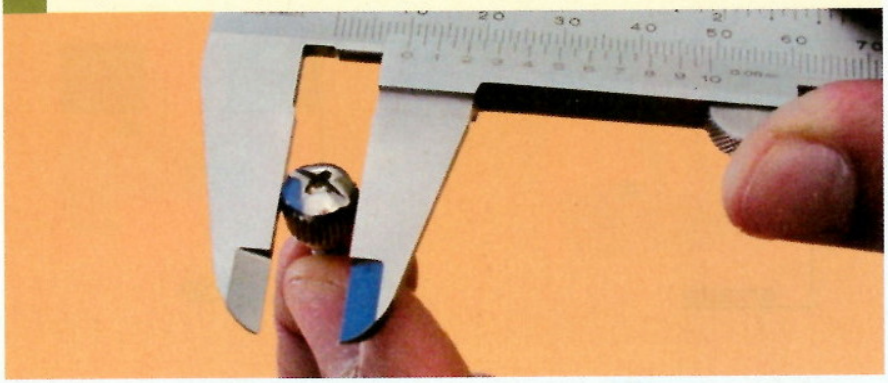
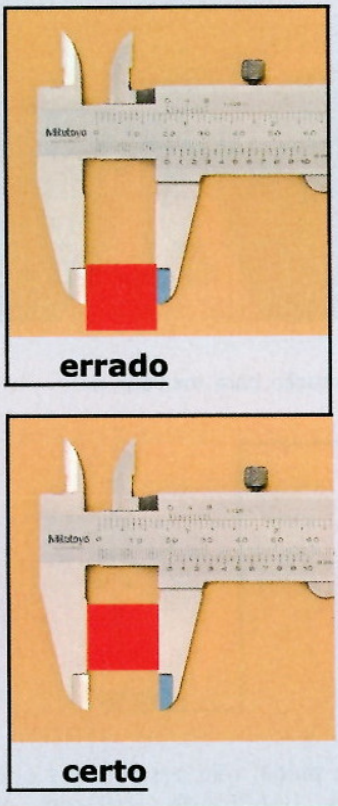


Figura 13 – Posição correta da peça para medidas externas



tros internos, é importante que as superfícies planas de medição nas orelhas coincidam com a linha de centro do furo a medir. Veja **figuras 17** e **18**.

Medidas de profundidade

Nos casos de medidas de profundidade é importante que o paquímetro esteja apoiado corretamente de forma que não fique inclinado.

Feche o paquímetro completamente e mova o cursor suavemente com o polegar apoiado no impulsor. Veja as **figuras 19, 20** e **21**.

Quando a medida de profundidade for crítica, deve-se optar pelo uso de um paquímetro de profundidade, com apoio adequado e mais largo. Observe a **figura 08**.

Medidas de ressaltos

Nos casos de medidas de ressaltos, deve-se apoiar a parte do paquímetro construída especificamente para este fim na superfície de referência da peça e mover o cursor suavemente até obter a medida desejada (**figura 23**)

Atenção à inclinação! É importante para que não se meça incorretamente o ressalto.

O uso da haste de profundidade é inadequado para este tipo de medida, conforme mostra a **figura 24**.

CUIDADOS COM O PAQUÍMETRO

Instrumentos de medição de uso comum como os paquímetros, micrômetros e outros, são objetos pessoais.

Em uma empresa é correto que cada funcionário tenha seu paquímetro e dele cuide corretamente.

O cuidado com o paquímetro pode evitar muitos dissabores.

Imagine que, se o instrumento estiver danificado ou tenha sofrido uma queda, mesmo que pequena, a tomada de uma medida pode ser errada e com isto muito trabalho pode ser perdido. Costuma-se dizer quando uma peça está fora de medida (e isto pode ser 0,03 mm de diferença) que a peça está "morta" ou seja, é um refugo.

Muitos problemas acontecem na indústria e mesmo em laboratórios de controle de qualidade por má ma-

Figura 14 – Não incline o paquímetro ao tomar uma medida.

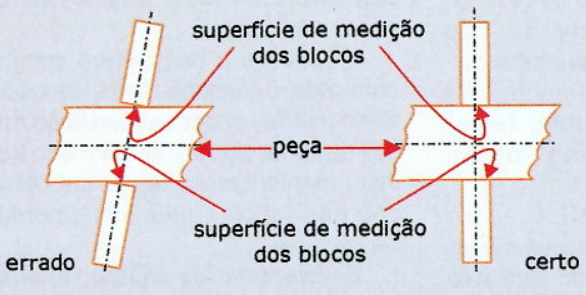


Figura 15 – Medidas internas.



Figura 16 – As orelhasdevem estar paralelas às paredes do furo a ser medido.

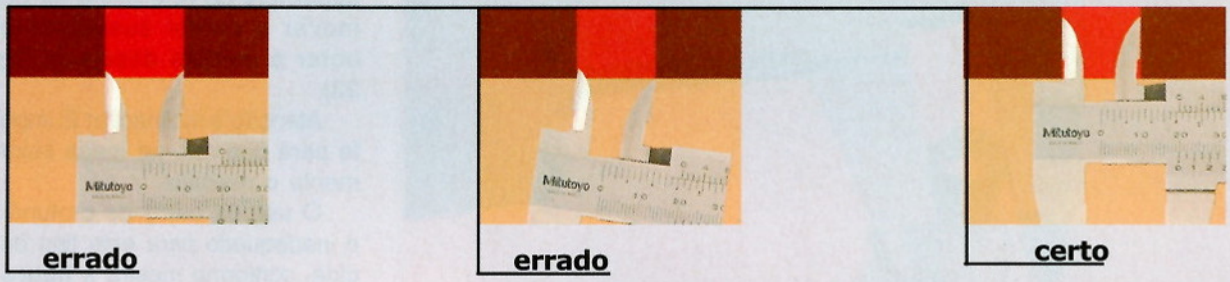


Figura 17 –Cuidados com inclinações ao tomar medidas.



Figura 18 – Orientação para medir corretamente furos

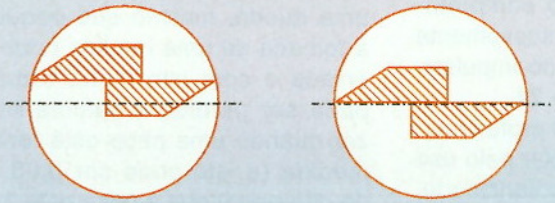


Figura 19 – Medidas de profundidade.

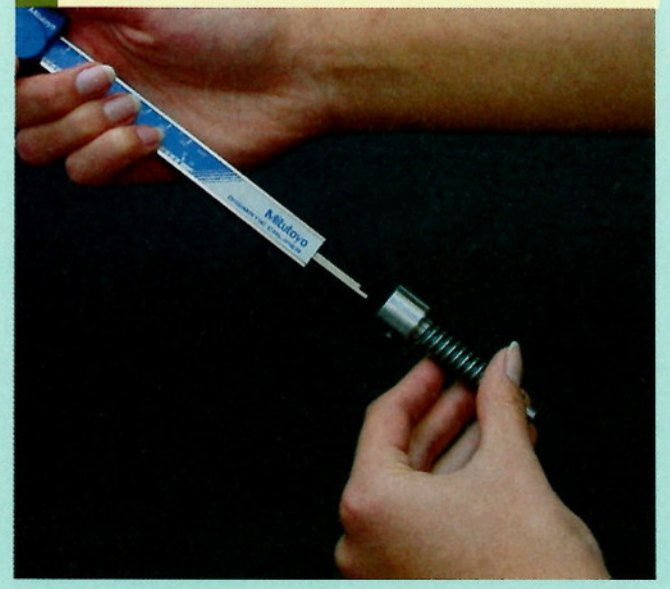


Figura 20 – Medidas de profundidade.



Figura 21 – Orientação para medidas de profundidade



nutenção ou maus cuidados com os instrumentos de medição.

Portanto, cuide de seu paquímetro corretamente, ele é uma ferramenta importante.

Não use para outra finalidade que não seja medir, não permita que outros sem habilidade utilizem seu paquímetro.

Os seguintes cuidados podem prolongar enormemente a vida de seu paquímetro e manter a qualidade de suas medições.

1. Maneje o paquímetro sempre com todo o cuidado, evite choques. Caso o utilize sobre bancadas ou mesas duras de aço ou mesmo ferro fundido, mantenha um tapete de borracha para apoiar seus instrumentos de medição.

2. Em caso de queda, mesmo

Figura 22 – Paquímetro de profundidade

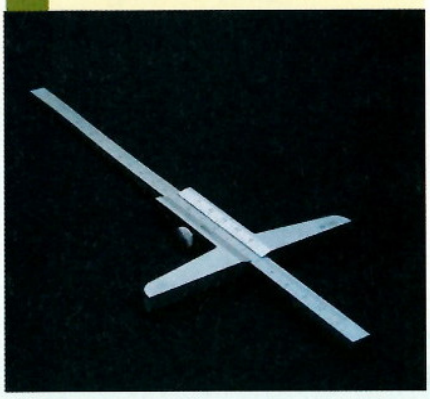


Figura 23 – Medida de ressalto.



Figura 24 – Orientação para medir corretamente.

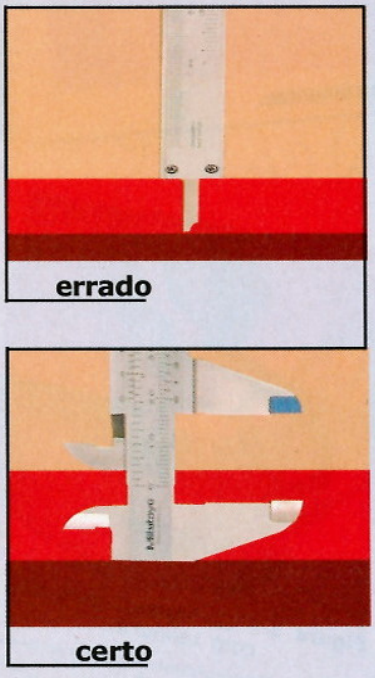
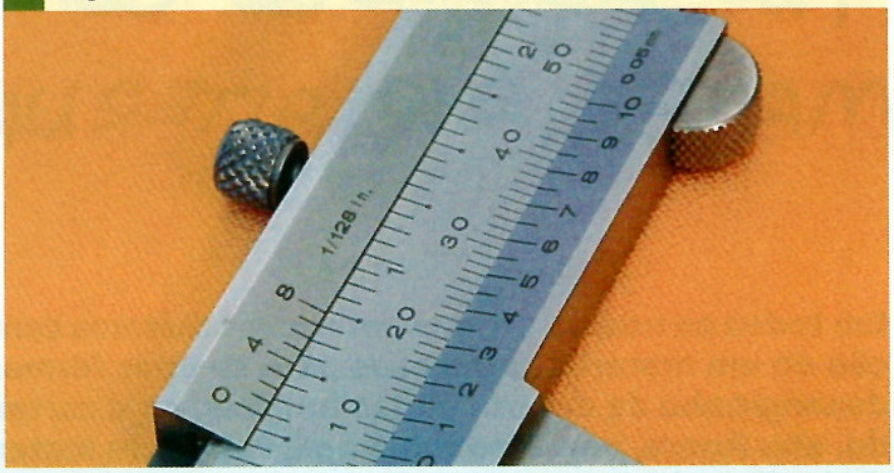


Figura 25 – Evitando erro de leitura por paralaxe.



pequena, envie seu paquímetro para aferição.

3. No contato dos bicos com a peça deve-se ter o cuidado de posicionar o paquímetro corretamente.

4. Nunca guarde seu paquímetro com outras ferramentas e não permita que outras ferramentas sejam utilizadas próximas de seu paquímetro, elas podem causar danos graves e nem sempre aparentes.

5. Limpe e guarde seu paquímetro no estojo apropriado após o uso, aplique óleo fino em "spray" para manter a superfície sem umidade.

ERROS DE LEITURA

Algumas informações complementares são importantes para o usuário que começa a medir com paquímetro.

Além do que já foi citado, há dois fatores que merecem especial atenção para que não ocorram tomadas de medidas incorretas e erros de leitura.

Paralaxe

Dependendo do ângulo de visão do operador pode ocorrer erro de leitura por paralaxe, pois conforme esse ângulo pode haver coincidência de um traço do nônio com outro incorreto da escala fixa.

Isto ocorre com mais freqüência nos nônios de dez traços (lembra-se que mencionamos no início?)

Como se pode observar, a escala móvel é posicionada sobre a escala fixa e uma pequena inclinação é comum para aproximar os traços no ponto de concordância para minimizar essa diferença de altura.

Mesmo assim, os traços gravados no nônio ficam mais elevados e "próximos" dos olhos do leitor do que os traços da escala fixa

Para evitar qualquer erro de leitura por paralaxe o paquímetro deve ser lido de frente, em linha perpendicular ao eixo dos olhos, assim não haverá ângulos para confundir nossa visão e os erros de leitura serão minimizados.

Acompanhe a **figura 25**.

Pressão de medição

Ao posicionar o paquímetro para medição não se deve exercer pressão demasiada no impulsor, pois isto além de prejudicar a leitura danifica o paquímetro.

As folgas deslizantes entre a escala fixa e o nônio são reguladas e ajustadas rigorosamente.

Nunca desmonte seu paquímetro para limpeza. Envie para assistência técnica que o fará corretamente e manterá a leitura de seu instrumento confiável.

Agora já sabemos utilizar um paquímetro na escala métrica. Na próxima edição aprenderemos como medir com um paquímetro em polegadas e conheceremos outro instrumento muito utilizado em mecânica: O Micrômetro.

TRABALHANDO COM MATERIAIS ALTERNATIVOS II

José Francci Júnior

Em todos os nossos artigos anteriores tivemos como "enfoque" a utilização de um material como base para se criar formas e/ou peças para o desempenho de determinadas funções, seja cortando, dobrando, colando, etc. Agora, veremos como aproveitar um material previamente moldado, ficando a sua utilização condicionada à forma já estabelecida. Estamos falando em trabalhar com garrafas ou recipientes plásticos descartáveis.

Trabalhando com recipientes plásticos descartáveis

Talvez o maior exemplo da "praticidade" dos plásticos na vida do homem moderno, os recipientes plásticos são encontrados em inclassificáveis formas, tamanhos e em quantidades que os tornam os principais pesadelos dos ambientalistas. Podemos utilizá-los em nossos projetos para vários fins, principalmente como coberturas ou invólucros preferencialmente sem grandes esforços estruturais. Observe a **figura 1**.

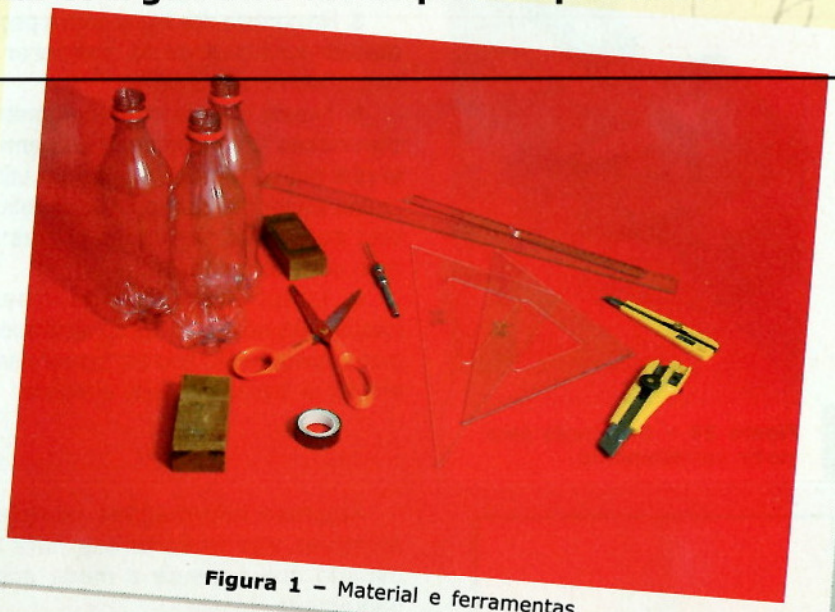


Figura 1 – Material e ferramentas.

Cortando os recipientes

Como as formas dos recipientes são, em geral, arredondadas, a principal dificuldade está na "marcação" da linha de corte de modo que este seja executado uniforme e simetricamente.
Podemos fazer a marcação com fita adesiva, colando-a e alinhando-a até a medida desejada e executando o corte final com tesoura ou estilete. Veja as **figuras 2 e 3.**



Figura 2 – Marcação da linha de corte com fita adesiva.



Figura 3 – Cortando o recipiente com tesoura.

Também podemos fazer a "marcação" apoiando o estilete sobre um suporte na medida que pretendemos cortar, e "girar" ou "correr" o recipiente contra a ponta do mesmo, criando um risco de preferência para um corte final que poderá ser feito com tesoura ou o próprio estilete. Note que neste caso devemos apoiar as peças sobre uma superfície bem plana. Atente para as **figuras 4, 5 e 6.**
Observem os leitores que nas fotos que ilustram esse artigo, usamos sempre o mesmo tipo de garrafa de refrigerante descartável, mas as técnicas se aplicam a qualquer recipiente plástico.



Figura 4 – Marcando o recipiente no sentido transversal.



Figura 5 – Marcando o recipiente no sentido longitudinal.



Figura 6 – Cortando o recipiente com o estilete.



Figura 7 – Cúpula plástica colada com cola de silicone.



Figura 8 – Cúpula plástica parafusada sobre chapa de plástico.

Fixando as peças de recipientes plásticos em outros materiais.

Os tipos de plásticos utilizados na manufatura de recipientes varia muito, mas em geral, eles são muito difíceis ou impossíveis de serem colados porque são muito flexíveis e extremamente resistentes ao ataque químico de qualquer natureza. Porém, ainda nos restam duas opções: cola plástica com pistola quente e cola de silicone poderão ser utilizadas com sucesso. É interessante fazer alguns testes com o material antes do corte e da fixação final. Veja **figura 7.**

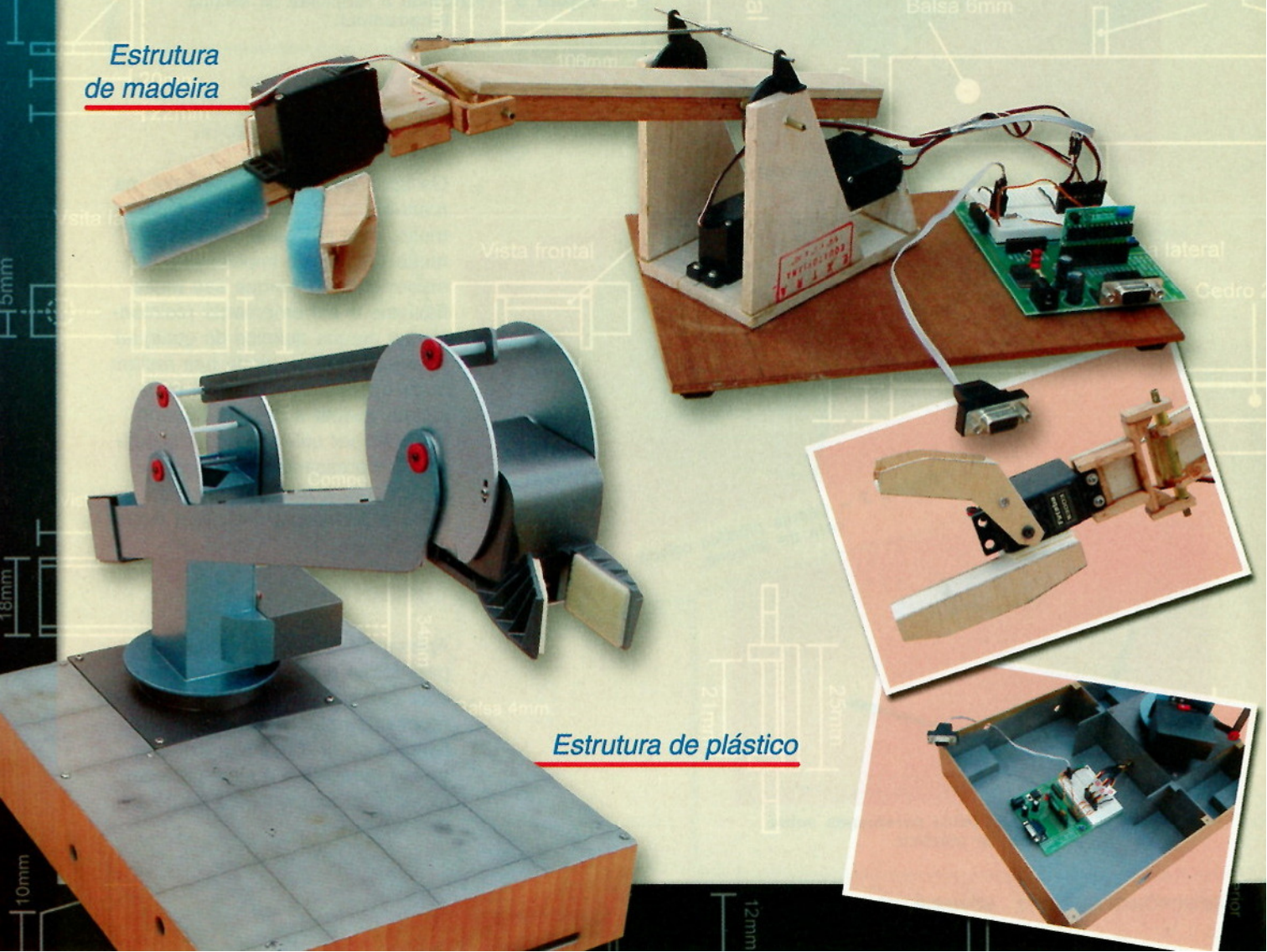
Conclusão

Apesar de limitados na utilização pela forma e medidas e também com certa dificuldade na fixação, os recipientes plásticos ou as peças feitas a partir deles são extremamente duráveis e podem proteger circuitos e componentes de pó e umidade.

RM-1 - Robô Manipulador

Márcio José Soares

Muitas vezes, quando falamos em robótica, aquela imagem de um braço mecânico manipulando pequenos objetos presentes a nossa volta vem em nossas mentes. Pensando nisso desenvolvemos o robô manipulador RM-1, fácil de montar e operar (através do software LOGO ou outro qualquer). Os materiais utilizados na montagem são de fácil acesso e como sempre, temos duas versões: uma em material de sucata e outra em plástico, visando assim atender a todos os nossos leitores.



Os robôs manipuladores estão presentes hoje nas indústrias. Sua precisão e velocidade são alguns dos fatores que viabilizam o seu uso em diversas áreas. Em muitas destas, eles são quase que indispensáveis, pois desenvolverão funções que põem em risco a vida humana.

O técnico/engenheiro de qualquer curso de Mecatrônica tem em seu currículo algumas "horas/aula" sobre o assunto "braços mecatrônicos" (Robótica). Nestes cursos a montagem de um pequeno "braço", às vezes, se faz necessária e muitos necessitam de algumas dicas para a construção do mesmo.

Nesta edição, propomos a construção de um robô manipulador que utiliza servos motores do tipo empregado em aeromodelos (modelismo), ao invés de motores comuns ou mesmo motores de passo. Essa escolha se deve a fatores como:

• A precisão obtida com os servos motores (podemos controlar sua posição relativa em "graus"); o custo dos mesmos, que apesar de parecer alto ainda é menor se comparados aos motores de passo com seus "encoders"; e circuitos de posição, etc.

• A simplicidade do circuito: com apenas um microcontrolador Basic Step, podemos controlar até 6 servos (com pequenas alterações no programa), sem a necessidade de complexos circuitos de posição, *drivers* para motores, etc.

• A velocidade de montagem. Seguindo as dicas, tanto para o protótipo em material alternativo (sucata) quanto às dadas para a construção da versão plástica, o leitor poderá finalizar seu projeto em poucas semanas.

Figura 1 - Circuito do RM-1.

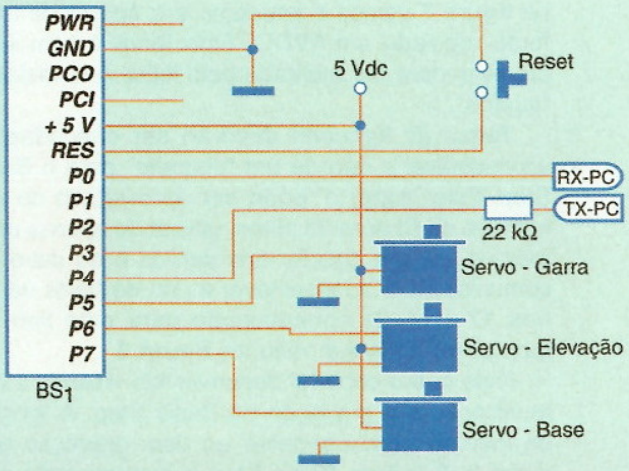


Figura 2 - Comandos para o RM-1.

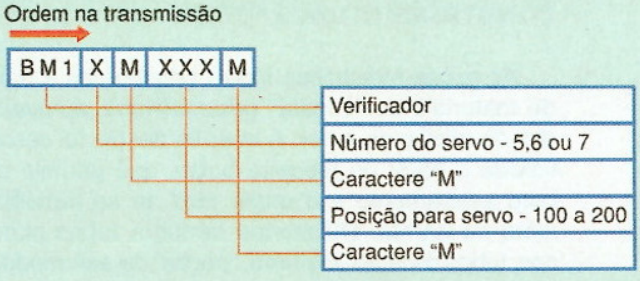
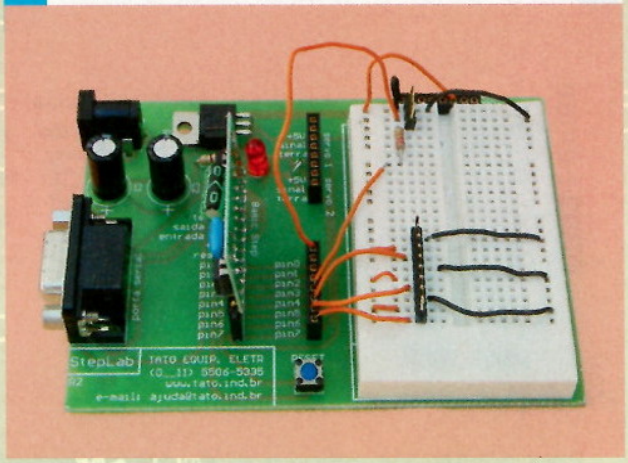


Figura 3 - Montagem na placa Step LAB.



CIRCUITO DE CONTROLE

Os movimentos do RM-1 são controlados por três servos comuns (tipo *standard*), utilizados em aeromodelismo.

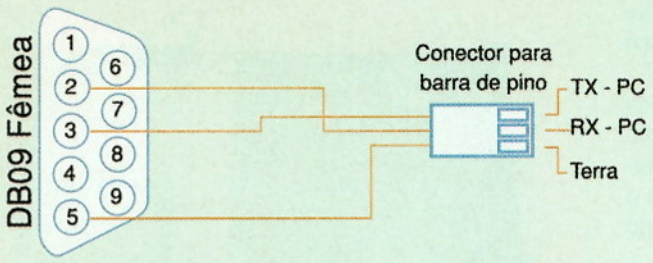
Na **figura 1** temos o circuito elétrico (controle) do RM-1. Para controlar os servos utilizamos um Basic Step I, que se comunica com um PC através da porta serial RS-232. O leitor poderá obter maiores informações sobre comunicação serial na edição nº5 de Mecatrônica Fácil, na série LOGO.

O Basic Step recebe os dados (**figura 2**) da porta serial, decodifica-os e os transfere para os servos. Isso é refletido em movimento para o braço mecânico. Estes dados são enviados para o Step através da linguagem LOGO, que pode ser obtida gratuitamente através do *site:* **www.nied.unicamp.br**. Os leitores mais experientes em programação poderão, se desejarem, desenvolver seus próprios programas de controle em outras linguagens , de acordo com suas necessidades e conhecimentos.

MONTAGEM DO CIRCUITO DE CONTROLE

Para esta montagem temos duas possibilidades: a primeira utilizando a placa Step LAB da Tato Ind. (**www.tato.ind.br**) e a segunda com placa de circuito impresso. O leitor poderá optar pela primeira como teste e somente então partir para a montagem definitiva, ou montar o circuito de acordo com os componentes existentes em sua bancada (no caso apenas o STEP 1, sem a placa Step LAB).

Figura 4 - Cabo adaptador para Step LAB.



Montagem com a placa Step LAB

Para o leitor que possui o *kit* de desenvolvimento Basic Step com a placa Step LAB a montagem é bem simples, pois necessitamos apenas de alguns pedaços de fio encapado rijo para as ligações, um resistor de 22k (limitador) e algumas barra de pinos para a ligação dos servos e do canal serial. Na **figura 3** apresentamos a montagem utilizando a placa Step LAB. A alimentação do circuito será fornecida pela própria fonte da placa. A gravação do Basic Step também será facilitada. Siga atentamente o circuito para realizar as ligações.

Figura 5 - Montagem em placa de circuito impresso.

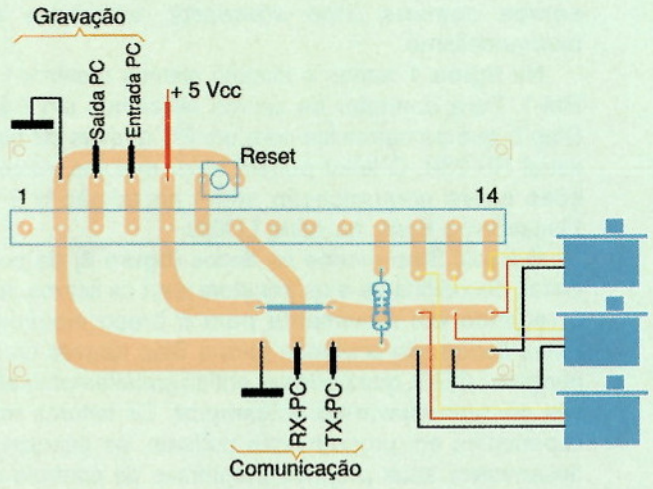
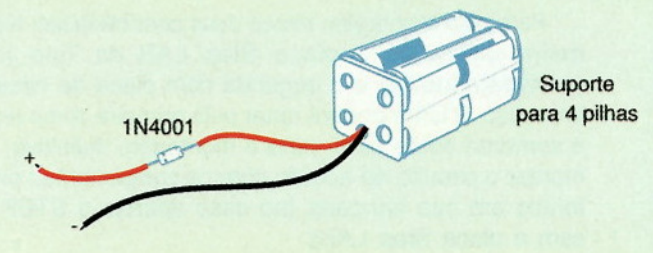


Figura 6 - Alimentação com pilhas.



O servo da base do braço deverá ser ligado a porta P7, o servo de elevação à porta P6 e o servo da garra a porta P5. O canal serial é ligado às portas P0 e P4. Para interligar a placa Step LAB ao PC, utilizaremos o cabo de gravação fornecido com a placa. Porém devemos providenciar um pequeno "adaptador" (**figura 4**) para o "pront-o-board" de nossa placa, pois o conector DB09 presente na placa serve apenas para gravação do Basic Step.

Montagem com circuito impresso

A montagem também poderá ser realizada com uma placa de circuito impresso comum ou mesmo do tipo universal, e na **figura 5** o leitor tem um exemplo do "lay-out" da placa para a montagem do circuito de controle.

A alimentação deve fornecer 5VDC máximos para o circuito e poderá ser feita através de 4 pilhas (6VDC) com um diodo em série para reduzir a voltagem para 5VDC (alimentação padrão do Basic Step). O tamanho das pilhas determinará o tempo máximo de operação do braço. Para tempos "médios" é aconselhável o uso de pilhas médias ou grandes, preferencialmente alcalinas. Na **figura 6** temos um exemplo de ligação com pilhas e o uso do diodo recomendado.

Para tempos maiores, ou até mesmo "infinitos", na **figura 7** vemos o esquema e o *lay-out* de uma fonte regulada em 5VDC. Esta fonte é bem simples e poderá ser montada pelo leitor sem maiores "sustos".

Todas as ligações deverão ser checadas. É aconselhável o uso de um "soquete" para o Basic Step. Este "suporte" pode ser aproveitado de um soquete de CI para 28 pinos, utilizando apenas uma metade. Para a ligação dos servos e do cabo de comunicação é aconselhável o uso de barra de pinos. O cabo de comunicação para este tipo de montagem pode ser visto na **figura 8**.

Note que o cabo foi desenvolvido visando a comunicação e a gravação do Basic Step. A ligação do mesmo deve obedecer ao uso: gravação nos pinos 2, 3 e 4 do Basic Step e comunicação nas portas P0, P4 e terra. O uso de um resistor de 22k para o sinal de TX do PC é necessário para evitar danos ao Basic Step.

CONSTRUÇÃO MECÂNICA

As peças mecânicas foram construídas a partir de materiais de "sucata" (alternativos), aproveitados da oficina do autor. A maioria destas foi desenvolvida a partir de madeira balsa, que permite um bom acabamento e é muito fácil de se trabalhar, além de barata. Utilizamos também tubos plásticos, tubos de latão, espuma, "peças" de aeromodelo, parafusos, cola, etc.

Figura 7 - Circuito e lay-out da fonte.

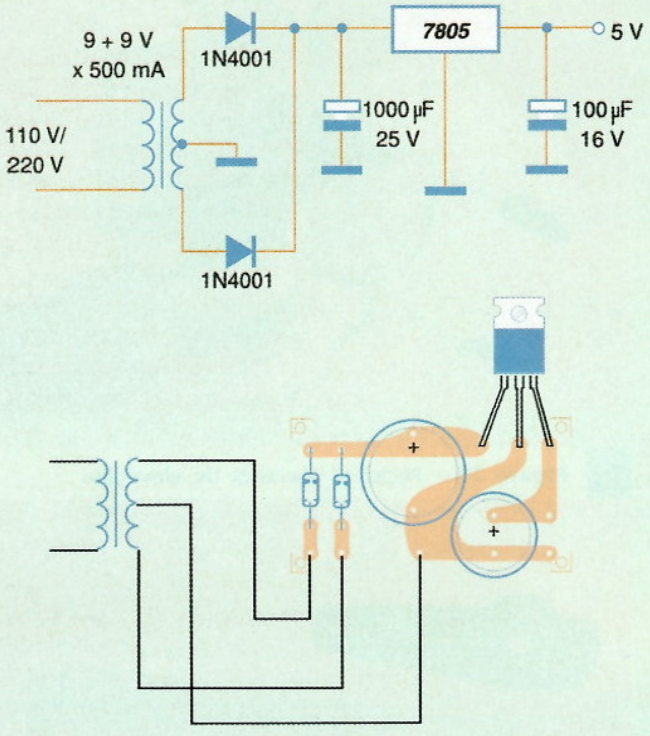
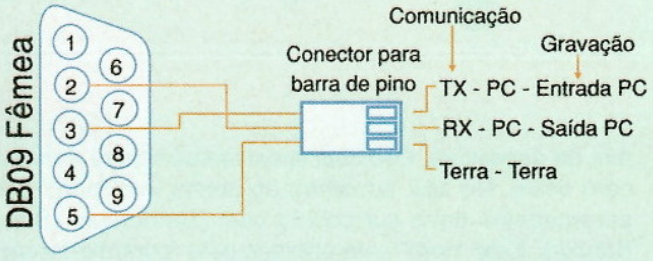


Figura 8 - Cabo de comunicação.



Base móvel

Ela foi montada em madeira balsa de ¼ de polegada (6mm) de espessura. Para realizar as furações e cortes, utilize um estilete ou serra fina e uma pequena furadeira. A balsa é uma madeira macia e o trabalho com a mesma é simples. ***Caso o leitor não tenha experiência no uso de estiletes, serras e furadeiras deverá pedir a ajuda a uma pessoa mais experiente.*** A união das partes deve ser feita com cola tipo epóxi.

No endereço:
www.mecatronicafacil.com.br/downloads
você encontrará os desenhos das peças de madeira com suas respectivas medidas para facilitar a construção.

Figura 9 - Montagem da parte fixa da garra.

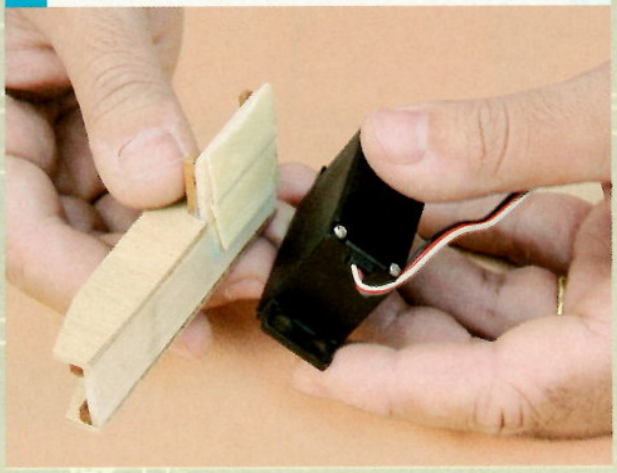


Figura 10 - Montagem da parte móvel da garra.

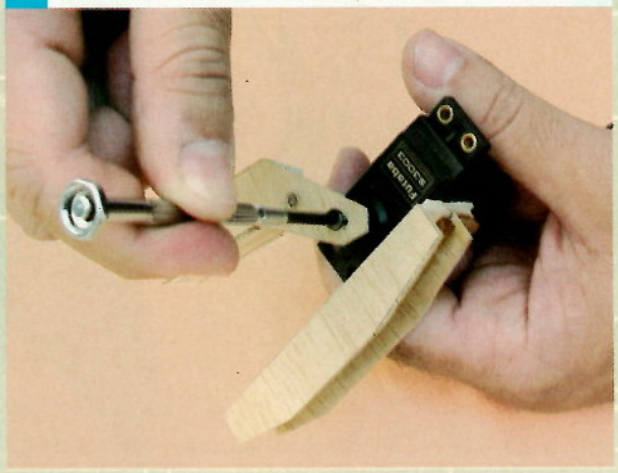


Figura 11 - Peças da parte móvel da garra.

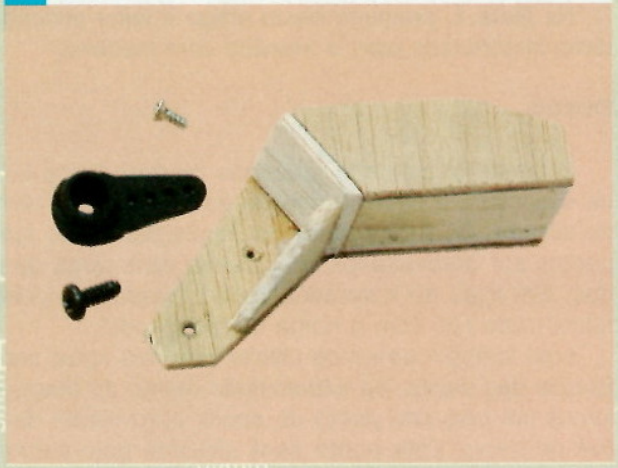


Figura 12 - Espuma para garra.

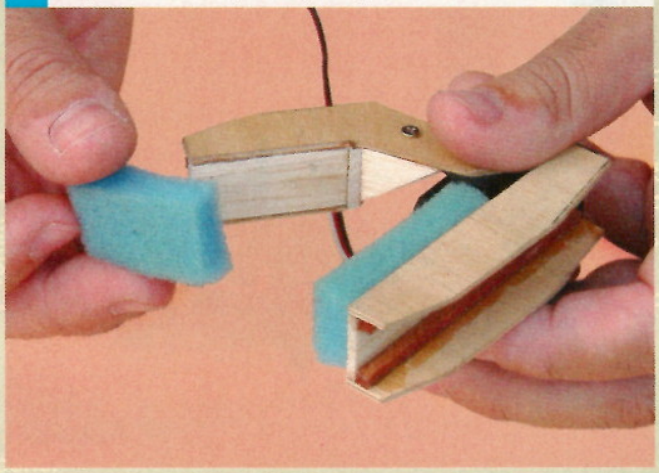


Figura 13 - Montagem do servo de giro.

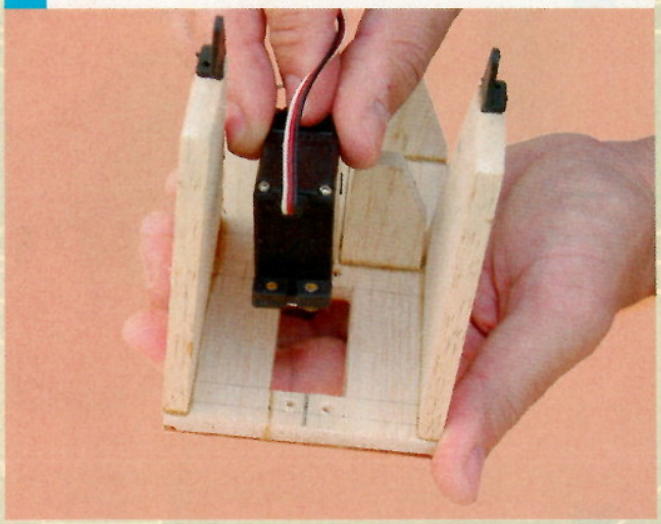


Figura 14 - Montagem do servo de elevação.

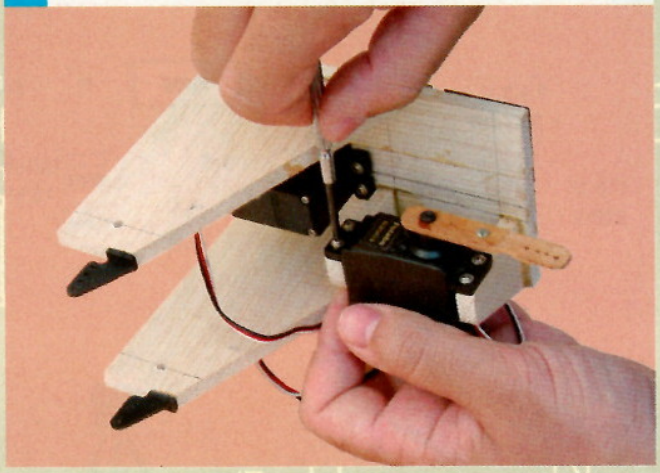
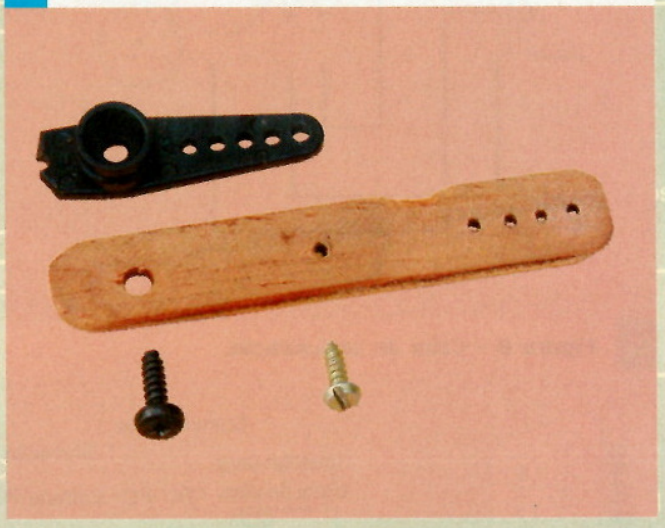


Figura 15 - Peças p/ alavanca de elevação.



Na **Nota 1**, presente neste artigo o leitor encontrará dicas importantes para o trabalho com madeira.

Braço

O braço foi montado utilizando madeira balsa 6 mm de espessura e cedro 2 mm de espessura com 10 mm de largura. Os tubos plásticos empregados nas juntas podem ser aproveitados de canetas sem carga ou outros. Em lojas de aeromodelismo é possível encontrar tubos deste tipo com o nome de "push-rods".

Aqui também devemos utilizar cola tipo epóxi para a fixação das peças. Na extremidade menor do braço, colamos um pequeno ponto de apoio, aproveitado de um *link* de servo. Este ponto será utilizado pelo servo de elevação.

Punho

O punho foi montado com cedro 2 mm de espessura com 10 mm de largura, além de retalhos de balsa com 6 mm de espessura. Use cola epóxi também para a montagem deste. No alto, ao centro do punho, um "horn" para aeromodelos deve ser colado com cianocrilato (Super Bonder). Este "horn" será utilizado pelo "controle mecânico do punho".

Garra

A garra foi montada com madeira compensado de 1,5 mm de espessura e retalhos de balsa com 2 e 4 mm de espessura.

Para a montagem da garra, o leitor poderá optar por aplicar cola epóxi ou mesmo cianocrilato (Super Bonder).

Base fixa

Esta base segura todo o conjunto e foi preparada a partir de um pedaço de compensado com 4 mm de espessura, 160 mm de largura e 240 mm de comprimento. Ela recebe apenas um furo para a inserção de um *link* de servo na parte de baixo da mesma.

MONTAGEM DO CONJUNTO

Garra

Começaremos por montar a garra. A parte fixa deve ser montada na lateral do servo com auxílio de "cola quente" ou fita dupla face (**figura 9**). A parte móvel da garra é fixa ao servo com o auxílio de um "link" de servo (**figura 10**). Este *link* é fornecido junto com o servo. Utilize um pequeno parafuso extra para melhorar a rigidez mecânica. Observe a **figura 11**, onde mostramos a parte móvel com suas peças. O leitor também poderá colar espuma comum à garra com o uso e fita dupla face, para melhorar o desempenho da mesma, como demonstrado na **figura 12**.

Figura 16 - Fixando o punho ao braço.

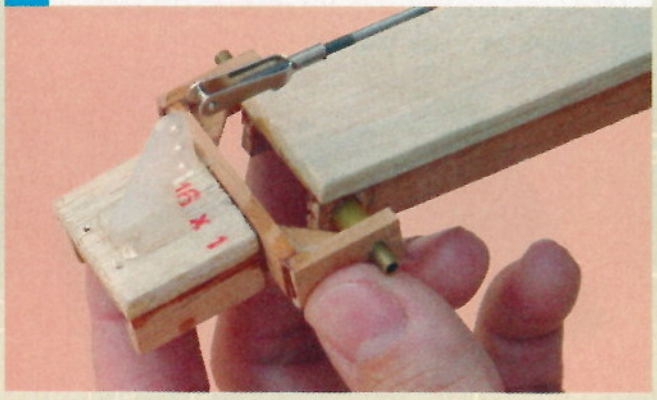
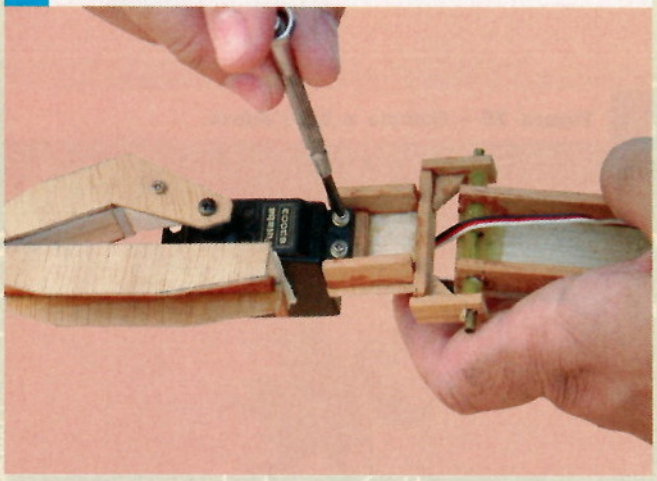


Figura 17 - Fixando o servo ao punho.



Base móvel

A seguir montaremos a base móvel. A fixação do servo de giro na base móvel deve ser feita com parafusos para madeira, na medida dos servos utilizados. Eles também não devem ser grandes demais, para não ultrapassar a espessura da base móvel. Veja a **figura 13**.

Notem que a base móvel também segura o servo responsável pela elevação do braço, e a fixação deste também deve ser feita com o uso de parafusos de madeira apropriados (**figura 14**).

O servo de elevação precisa ter sua "alavanca" de comando aumentada. Para isso utilizamos cedro com 2 mm de espessura com largura de 10 mm. A **figura 15** traz as partes necessárias para a alavanca de elevação.

Figura 18 - Fixando o braço a base móvel.

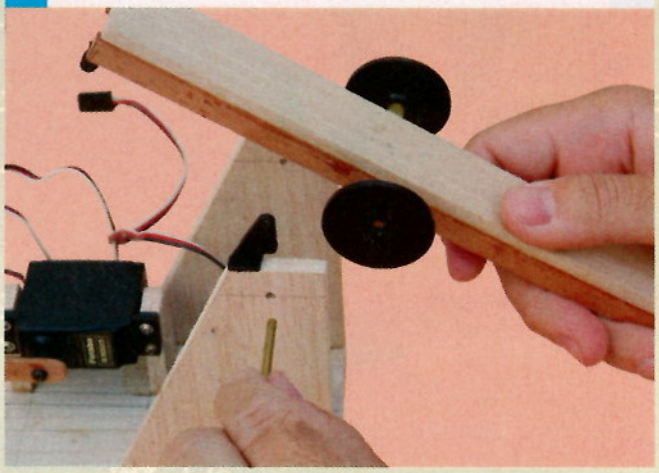
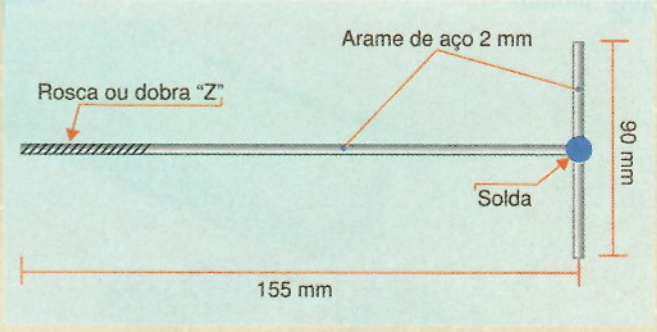


Figura 19 - Esquema para controle do punho.



Punho

Fixe o punho ao braço, utilizando um tubo de latão com espessura externa igual a espessura interna do tubo plástico usado na construção do braço, como indicado na **figura 16**.

Fixe o servo da garra ao punho. Utilize para isso dois parafusos de madeira. Note que o servo será fixado de "cabeça-para-baixo", conforme a **figura 17**.

Braço

Agora fixaremos o braço à base móvel (**figura 18**). Utilizaremos para isso um tubo de latão. Aqui também este tubo deverá ter sua espessura igual à espessura interna do tubo plástico. As "rodas" demonstradas nesta figura são *links* redondos para servos.

Precisamos agora fixar o controle do punho. Este controle é mecânico e permite que o punho fique sempre paralelo a base. Ele foi feito com arame de aço com ros-

Figura 20 - Fixando o controle de punho a base.

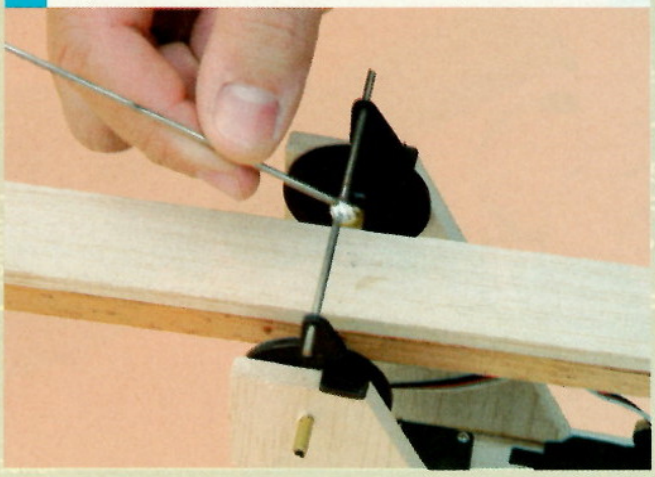


Figura 21 - Fixando o controle de punho ao punho.

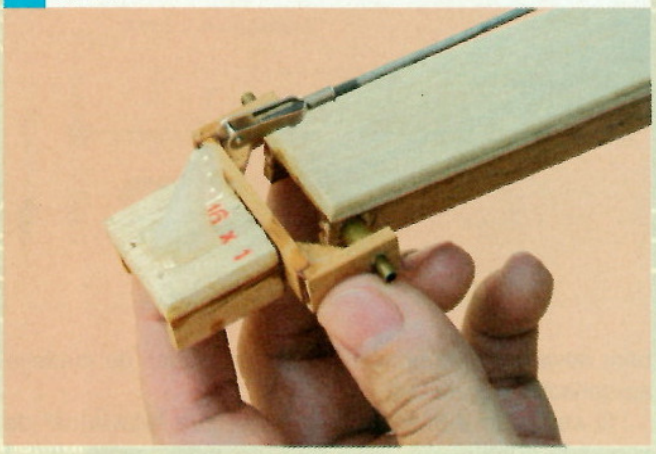


Figura 22 - Peça de ligação entre alavanca e braço

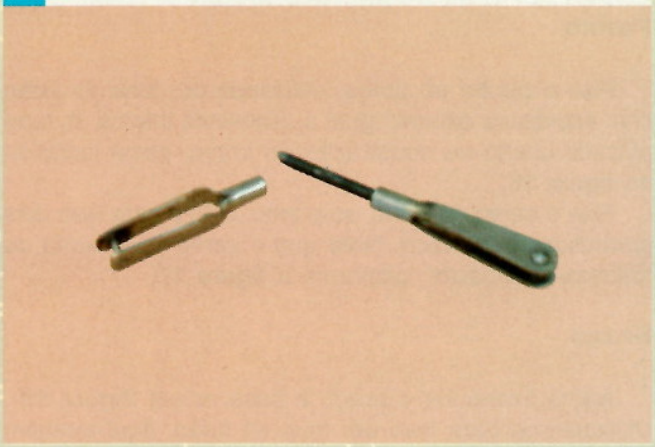


Figura 23 - Ligação da alavanca e braço.

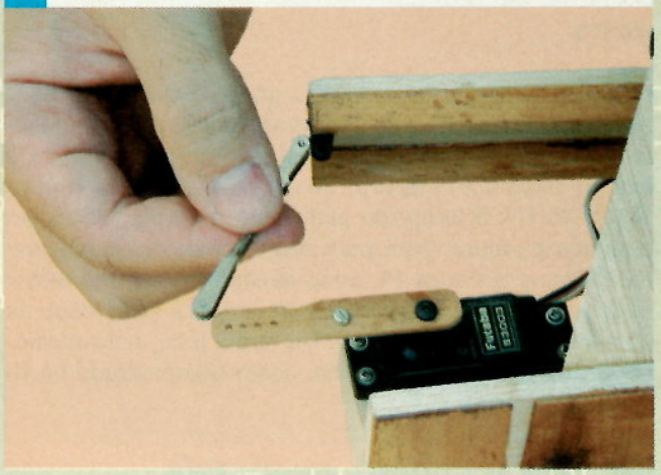


Figura 24 - Link "estrela".

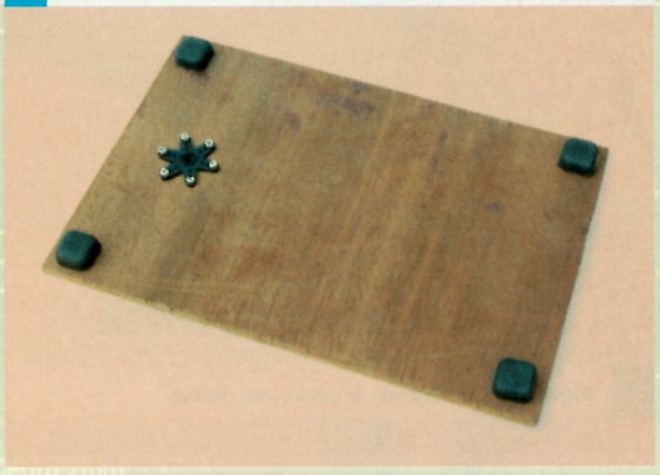
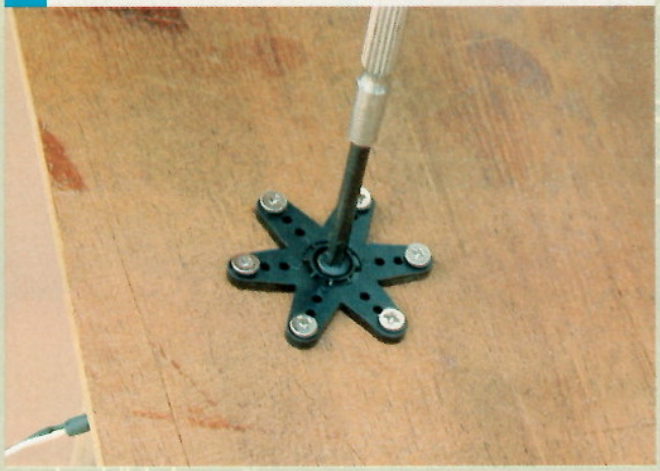


Figura 25 - Fixando a base móvel.



ca em uma das pontas. Ele é facilmente encontrado em casas especializadas. Na **figura 19** temos o diagrama com as medidas e formas do mesmo.

A fixação do mesmo na base móvel é feita utilizando-se "horns" do tipo pequeno para aeromodelos. Estes "horns" deverão ser posicionados na base móvel, de maneira que seus furos fiquem perpendiculares ao eixo central do braço. Isto é muito importante para a correta operação do mesmo. A fixação do controle do punho pode ser vista na **figura 20**.

Na outra extremidade colocamos um *link* de aeromodelo para fixar ao "horn" do punho. Regule através da rosca do arame para que o punho e o braço fiquem alinhados, quando conectar o *link* ao "horn de aeromodelo" colado no punho. Veja a operação na **figura 21**.

Agora, já podemos fixar a alavanca de elevação ao braço. Para isso desenvolvemos uma peça aproveitando apenas a extremidade com rosca do arame de aço e dois *links* para aeromodelo. Veja a **figura 22**. Essa peça permite uma regulagem precisa da altura do braço. Na **figura 23** temos a instalação da mesma na alavanca e braço.

Montagem final

Fixado o braço, resta-nos apenas montar a base móvel à base fixa. Para isso devemos parafusar o servo de giro a base fixa. Utilizamos um *link* para servo tipo "estrela" parafusado na parte de baixo da base fixa (**figura 24**). Não devemos apertar muito este parafuso, pois a base móvel irá girar sobre a base fixa. Sendo assim teremos um pequeno atrito. Não sobrecarregue o servo. A **figura 25** mostra esta operação.

LIGAÇÕES ELÉTRICAS

Agora já podemos ligar os servos a placa de controle. Porém antes devemos "alongar" o fio de ligação do servo da garra. Para isso o leitor necessita de um fio triplo, que pode ser aproveitado de uma cinta de conexão com "disk-driver's" ou "HD's". O comprimento da extensão deverá ser feito observando-se o livre movimento do braço. É melhor pecar por excesso do que por falta.

Corte o fio do servo próximo ao conector do mesmo e solde o "alongador". Tome cuidado para não inverter as ligações. Use termocontrátil para isolar, evitando possíveis curtos e melhorando o acabamento. Veja a **figura 26**.

Ligue o servo da garra a porta P5, o servo de elevação à porta P6 e o servo de giro (base móvel) a porta P7 conforme visto na **figura 27**. Ligue o cabo de comunicação a placa (**figura 28**), e pronto (**figura 29**)!

PROGRAMAÇÃO

Para programar o Basic Step, digite o programa "RM_1.BAS" no "Compilador Basic Step". Verifique os erros e envie o programa para o Step. O leitor notará que o braço executará alguns movimentos, buscando ficar com a base no centro, o braço ajustado na metade de seu curso e a garra fechada.

No endereço:
www.mecatronicafacil.com.br/downloads
você poderá fazer o *download* dos códigos-fonte.

Figura 26 - "Alongando" o cabo do servo.

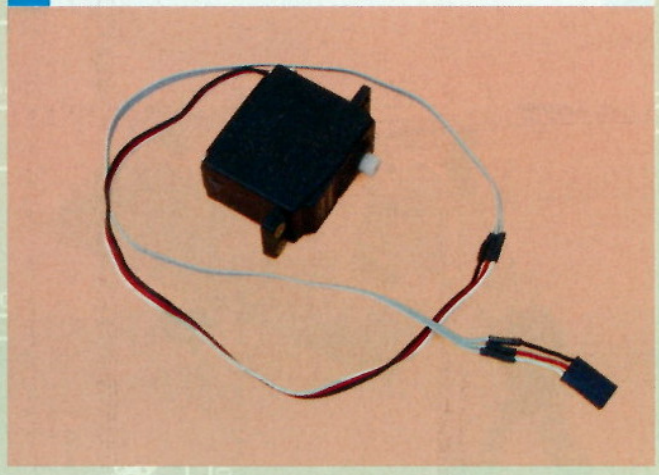


Figura 27 - Ligando os servos a placa de controle.

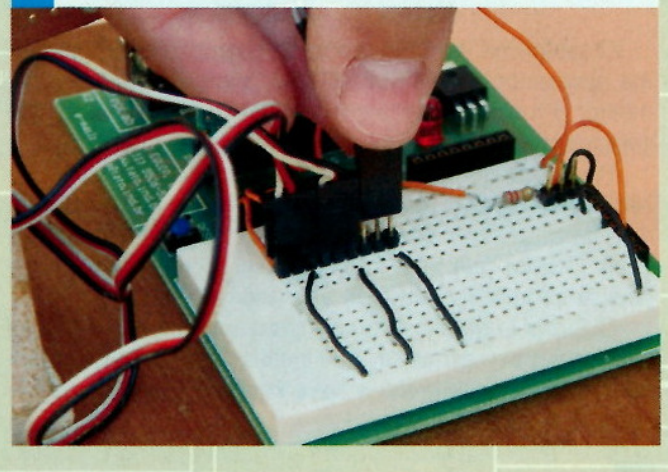
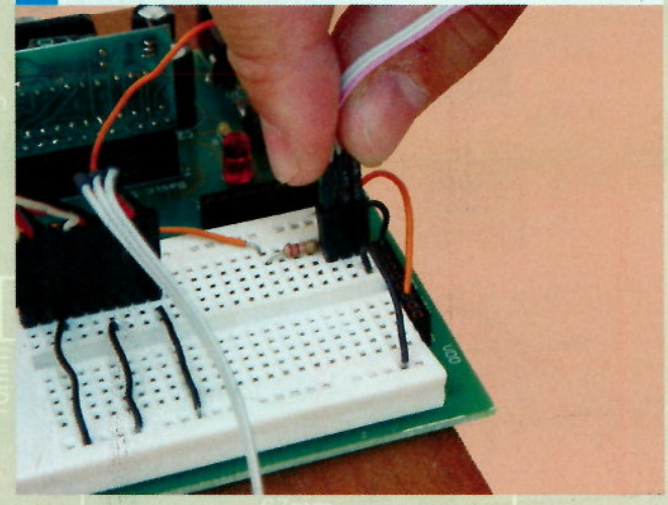
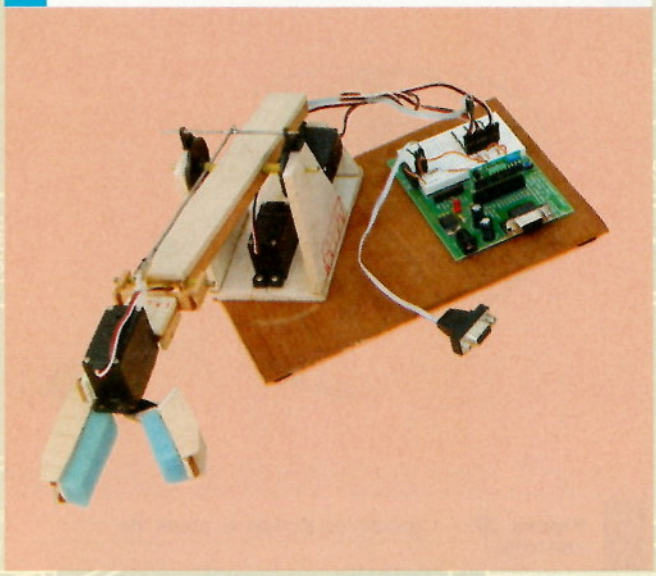


Figura 28 - Ligando o cabo de comunicação.



Talvez seja necessário ajustar o braço. Ajuste tudo movendo as partes para que obedeçam as configurações demonstradas. Lembre-se de soltar os parafusos para isso!

Figura 29 - O RM-1 pronto!

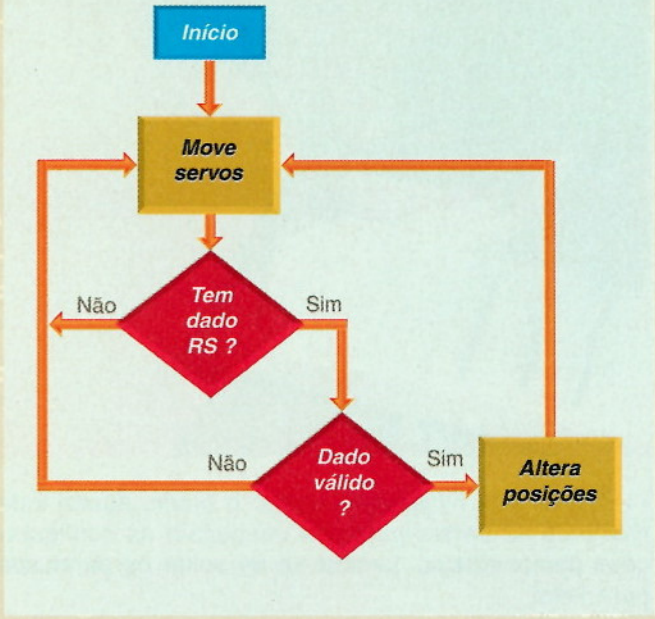


O programa do STEP roda de acordo com o fluxograma apresentado na **figura 30** e o leitor poderá entender melhor o funcionamento de cada linha do programa, observando os comentários inseridos no mesmo.

Para o controle do braço optamos pelo Super Logo, do Nied de Campinas. Esta linguagem é muito interessante e alvo de uma série de artigos aqui na revista Mecatrônica Fácil. O leitor que tiver interesse em acompanhar a série poderá adquirir os números anteriores da revista.

O Super Logo é distribuído gratuitamente no site **http://www.nied.campinas.br**. Sua instalação é idêntica a qualquer software comercial e bem intuitiva. Digite o programa RM_1.LGO no Logo.

Figura 30 - Fluxograma para o RM-1.BAS.



PROVA E USO

Com tudo devidamente conferido, é hora do teste. Conecte o cabo de gravação do Basic Step ou o cabo desenvolvido pelo leitor (dependendo do caso) à porta serial 1 (COM 1) ou porta serial 2 (COM 2). O programa RM_1.LGO foi desenvolvido para rodar na porta COM 1. Se o leitor desejar usar a porta COM 2, altere as linhas de programa como segue:

```
abraporta "com2
mudemodoporta "com2:2400,n,8,1
```

Esta alteração permitirá o uso da porta COM 2 pelo LOGO. Ligue o RM-1 e execute o programa. As opções do programa são suficientes para operar o braço (**figura 31**). Existem comandos (botões) totais e comandos parciais (barras de rolagem). O leitor também poderá gravar os seus comandos para posteriormente ver o braço executá-los, independentemente do teclado do micro.

Para isso, *clique* em "Abre Arquivo". O LOGO criará um arquivo chamado RM1.DAT onde guardará os comandos salvos. Execute um comando qualquer no RM-1. *Clique* em "Salva Comando". E assim sucessivamente. Para cada comando, um *clique* no botão para salvar o comando. Após salvar todos os comandos você já pode executar os comandos salvos. *Clique* em "Executa arquivo". O braço repetirá todos os comandos gravados. Ao final ele irá parar, aguardando "novas ordens". O arquivo RM1.DAT conterá os comandos e poderá ser sempre executado.

Na **figura 32**, o leitor poderá observar através do fluxograma o funcionamento do programa RM_1.LGO. O leitor mais experiente em programação poderá também desenvolver um programa diferente para outras operações ou até mesmo inserir novos comandos, ou modificar os já existentes no programa apresentado. As possibilidades são infinitas.

Com seu RM-1 pronto o leitor poderá pintá-lo, melhorando muito o acabamento do mesmo. As cores e padrões ficam por conta de cada um.

Figura 31 - Tela do programa.

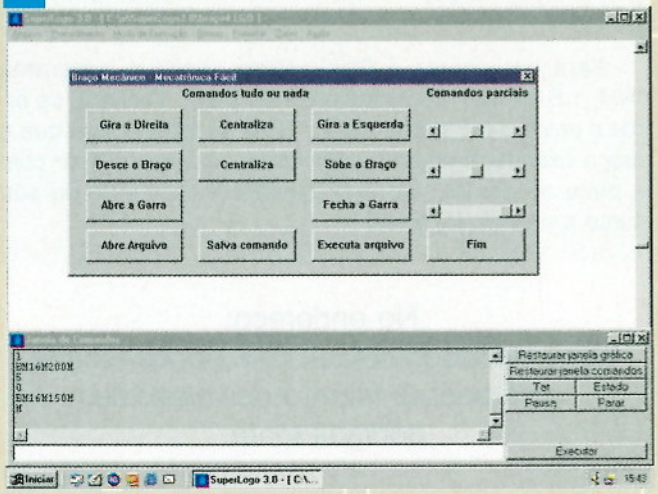
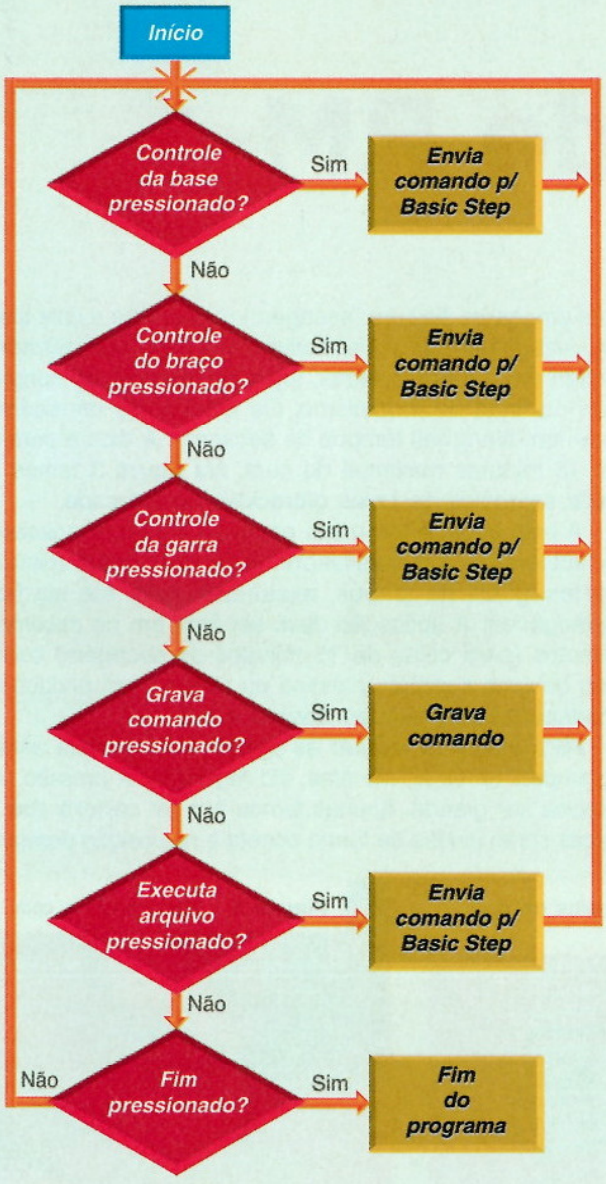


Figura 32 - Fluxograma p/ BM_1.LGO



CONCLUSÃO

Um braço mecânico, muitas vezes, pode parecer complexo, mas não é! Aconselhamos uma leitura cuidadosa do texto apresentado e uma análise das fotos deste artigo. Faça isso várias vezes, se necessário. Você notará que tudo o que lhe parece "difícil", ficará mais claro a cada revisão. Esperamos que todos os que se proponham a montar o RM-1, tenham sucesso. Boa montagem!

LISTA DE COMPONENTES

Para o RM-1 com Step Lab

1 – Basic Step 1
3 – Servos para aeromodelo Standart Futaba ou compatível
1 – resistor 22 KΩ x 1/8 Watt

Cabo adaptador

1 – DB09 fêmea
20 cm Fio triplo

Diversos

1 – barra de pinos, fios rijos para ligação, pés de borracha, etc.

Para o RM-1 sem Step Lab

1 – Basic Step 1
3 – Servos para aeromodelo Standard Futaba ou compatível
1 – resistor 22 KΩ x 1/8 Watt
1 – chave normalmente aberta tipo "push-button"

Cabo de comunicação Gravação

1 – DB09 fêmea
1,5 – metros de cabo com 3 vias

Fonte

1 – 7805 (regulador de voltagem)
1 – transformador 9+9 volts x 500mA
2 – diodos 1N4001
1 – capacitor eletrolítico 1000µFx25V
1 – capacitor eletrolítico 100µFx16V
1 – Caixa para fonte

Diversos

1 – placa de circuito impresso virgem ou padrão, barra de pinos, pés de borracha, rabicho, fio duplo p/ fonte, etc.

Peças de aeromodelo (comuns as duas versões)

2 – *horns* tipo pequeno
1 – *horn* tipo médio
2 – arames com rosca
3 – *links* para aeromodelos

Dicas para construção do RM-1

Nosso protótipo do RM-1 foi construído em madeira balsa e outros materiais. A balsa é fácil de manusear, mas são necessários certos cuidados.

Corte – O corte da madeira pode ser feito com estilete e régua ou serra fina, dependendo da espessura da madeira. Espessuras de até 2 mm podem ser trabalhadas com estiletes. Espessuras maiores requerem o uso da serra fina. É recomendável fazer os traços com caneta ou lápis antes para uma melhor orientação do corte.

Outra dica para um corte preciso é obedecer à orientação dos grãos da madeira. Com um estilete, o corte no sentido dos grãos é facilitado. O corte na transversal deste sentido, deve ser feito preferencialmente com a serra fina. As **figuras 1 e 2** demonstram o corte com estilete e serra fina.

Colas – Podemos utilizar cola branca, muito comum em papelarias, o cianocrilato (Super Bonder) ou a cola epoxi. Esta última oferece excelente rigidez mecânica e tem um tempo de cura (secagem) melhor que a cola branca, mas maior que o cianocrilato. Ela pode ser facilmente obtida em supermercados, lojas de material de construção ou lojas de modelismo. Os fabricantes oferecem a cola em diferentes tempos de secagem. A dica é para as de 15 minutos máximos de cura. Na **figura 3** temos alguns exemplos de colas oferecidas no mercado.

A cola epóxi é fornecida em duas partes: o adesivo e o "acelerador" ou catalisador. Para usar basta misturar partes iguais de ambos, misturando bem até ela ficar homogênea. A aplicação deve ser feita em no máximo 3 minutos (para colas de 15 minutos de secagem) com o uso de uma espátula plástica ou mesmo um pedaço de madeira (sobra). Veja as **figuras 4 a 8**.

Para ajudar a segurar as partes, você poderá utilizar "grampos" de varal, alfinetes, etc (**figura 9**). A pressão não precisa ser grande. Apenas temos que ter certeza que as peças serão unidas de forma correta e na posição desejada.

Figura 1 - Cortando a madeira com o estilete.



Figura 2 - Cortando a madeira com a serra.

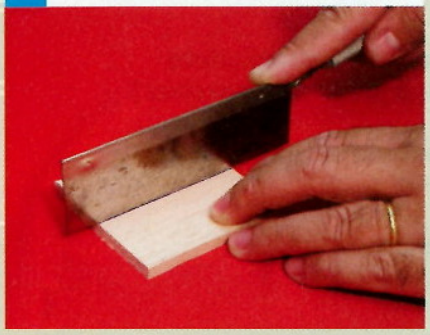


Figura 3 - Exemplos de cola epoxi.



Figura 4 - Colocando a cola num recipiente.



Figura 5 - Misturando a cola.

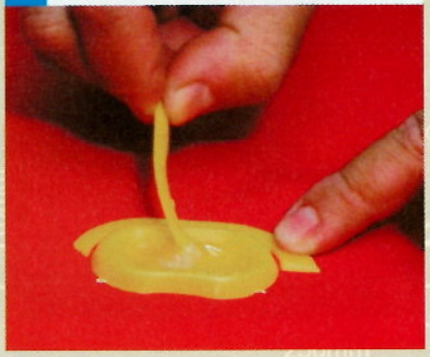


Figura 6 - Cola pronta.

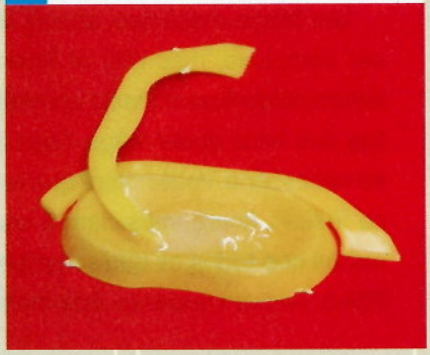


Figura 7 - Passando a cola na superfície a ser colada.

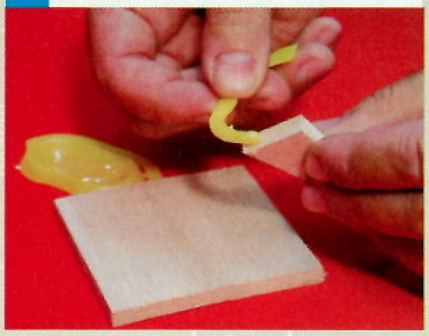


Figura 8 - Fixando a peça.

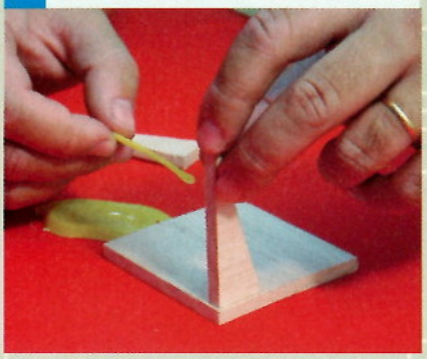
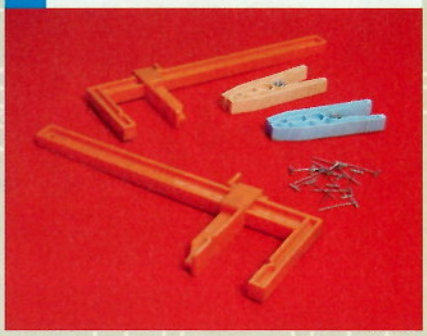


Figura 9 - Grampos e alfinetes.



Substituições

Todas as peças de aeromodelos podem ser substituídas. Nosso objetivo foi demonstrar que podemos aproveitar várias peças, de diversas áreas, para a construção de nossos robôs. Devemos estar sempre atentos a possibilidade do emprego de novos materiais.

***Link* de aeromodelo** – os *links* de aeromodelo podem ser substituídos por arames com terminação em "Z". Esta adaptação não permite regulagens de distância. Para fazer este arame siga os passos demonstrados nas **figuras 10 a 12.**

Horn – O *horn* de aeromodelo pode ser montado pelo leitor com madeira compensado de 2 mm, plástico ou qualquer material rijo que o leitor disponha. Na **figura 13** temos um exemplo de "horn" comercial.

Arames de aço – o arame de aço pode ser obtido em casas de aeromodelismo, ou mesmo aproveitado de um aro de bicicleta. Ambos possuem rosca em uma única ponta. O corte destes pode ser feito com minifuradeiras com disco de corte fino, ou serra de metal, ou mesmo através de alicates do tipo universal com corte. Na **figura 14** temos um exemplo do arame descrito, encontrado em casas de modelismo.

Figura 11 - Segunda dobra do arame.



Figura 13 - "Horn" comercial.



Figura 10 - Primeira dobra do arame.

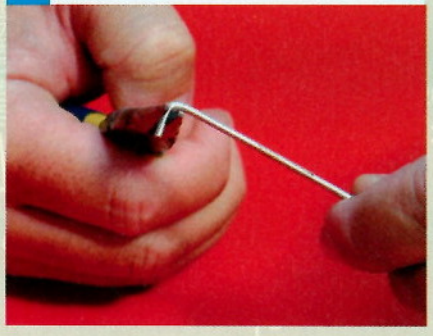
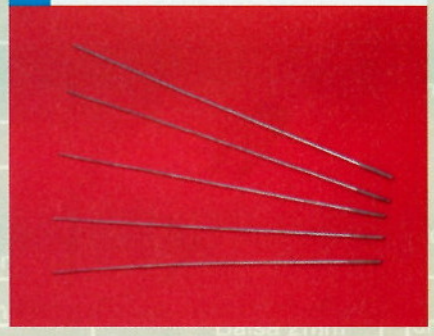


Figura 12 - Última dobra do arame.



Figura 14 - Arame com rosca.



A estrutura de plástico do robô manipulador

José Francci Júnior

Este projeto é a estrutura de plástico mais complexa já idealizada por nós, desde o lançamento da revista Mecatrônica Fácil. Na verdade, se os leitores prestarem atenção, eles vem evoluindo a cada edição.

Os leitores poderão notar também que, independentemente das necessidades da eletrônica e/ou da informática, buscamos soluções sempre baseadas nos mesmos materiais, ferramentas e métodos construtivos apresentados na série de artigos suplementares, intitulados "Trabalhando com plásticos" (Mecatrônica Fácil nº 2, 3, 4 e 5).

O nosso robô manipulador constitui-se de 190 peças de plástico coladas ou unidas entre si por 57 parafusos das mais diversas medidas, e também utilizamos 56 barretes de chumbo para equilibrar e estabilizar todo o conjunto que executa três movimentos independentes acionado por 3 servomotores.

A partir da edição nº 5 passamos a apresentar os projetos com uma solução em plástico e outra em materiais alternativos, procurando dar a vocês uma visão do quão diferentes podem ser os caminhos ao se executar os mecanismos e estruturas. Comparem o robô manipulador de plástico e o de madeira executado pelo próprio autor do projeto de eletrônica e informática (Márcio José Soares). A partir da próxima edição, passaremos a integrar plástico e materiais alternativos em projetos inicialmente mais simples estruturalmente, aumentando ainda mais as possibilidades daqueles que gostam de um desafio à criatividade e habilidade. Até lá, mãos à obra. Idéias a se concretizar não faltarão!

Figura 1 - Detalhe da construção da base. Note os espaços nos cantos para a colocação dos barretes de chumbo.

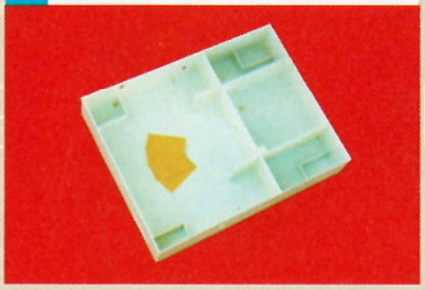


Figura 2 - Instalação do servomotor na base do conjunto.

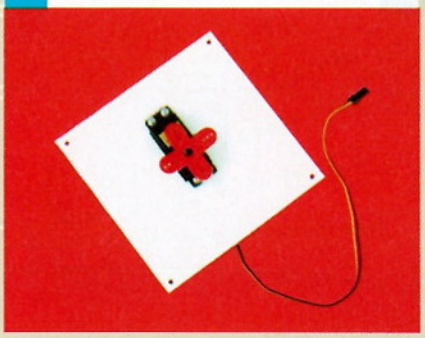


Figura 3 - Construção e instalação do servomotor na garra.

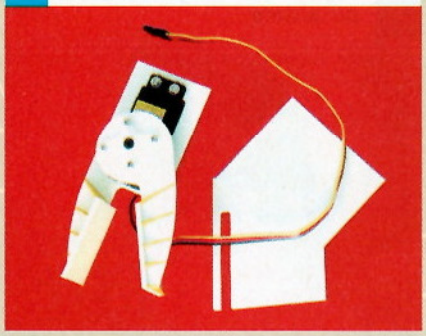


Figura 4 - As peças que compõem a garra.

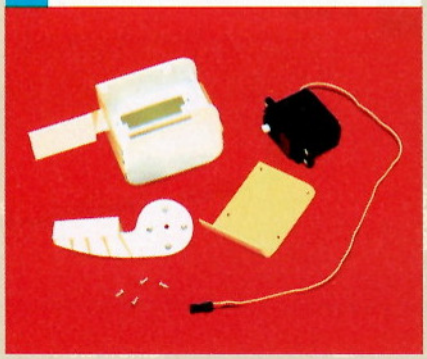


Figura 5 - A garra completa.

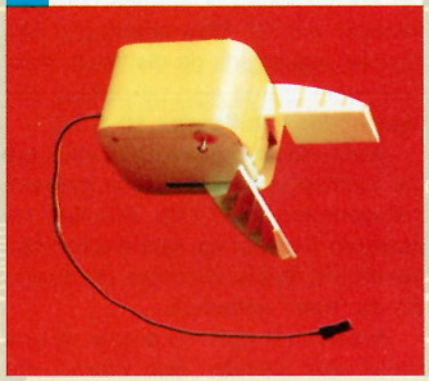


Figura 6 - Construção da longarina e os discos da estrutura principal.

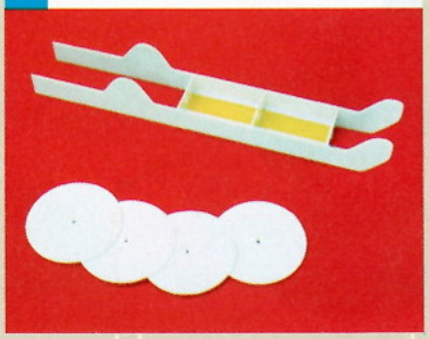


Figura 7 - A longarina já com os barretes de chumbo para equilibrar o conjunto.

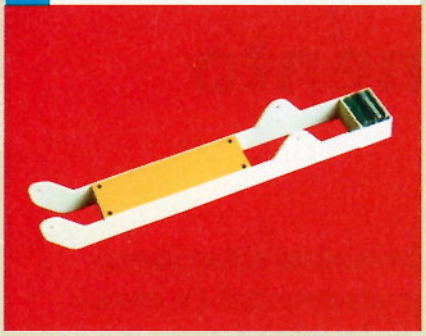


Figura 8 - Os discos, o tirante superior e os pivôs feitos em tarugo de plástico.

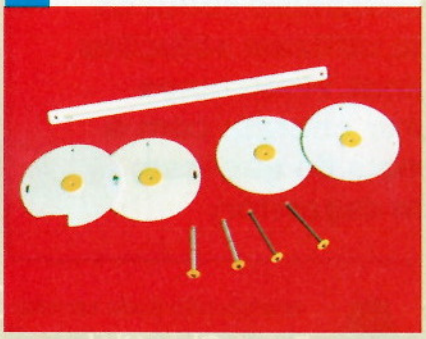


Figura 9 - Construção da torre vertical.

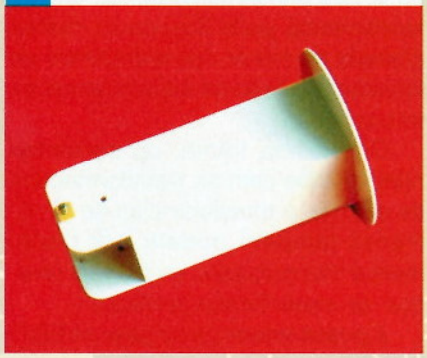


Figura 10 - Detalhe da instalação da torre vertical no servomotor da base.



Figura 11 - A estrutura principal já com a garra, e a torre vertical.

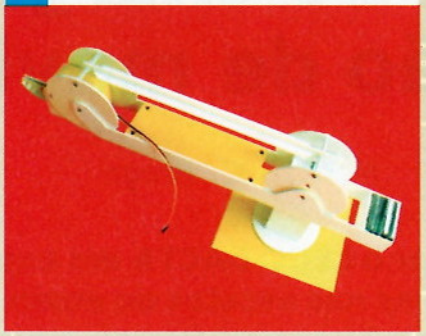


Figura 12 - Detalhe da estrutura principal instalada sobre a base.

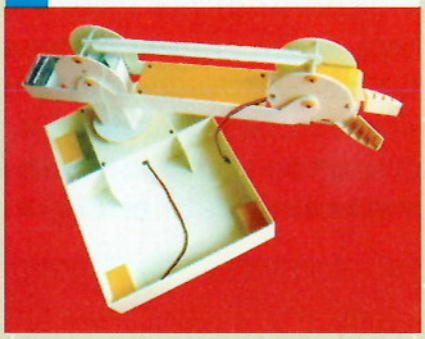


Figura 13 - Conjunto do acionador vertical.

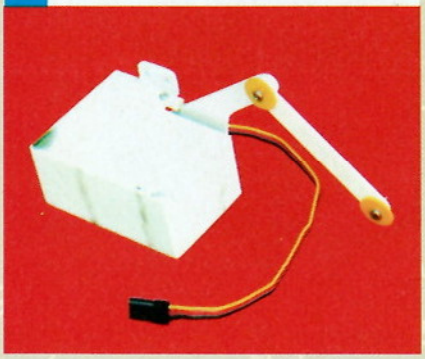


Figura 14 - A torre vertical com o suporte para a instalação do acionador vertical.



Figura 15 - O conjunto torre vertical e acionador vertical.

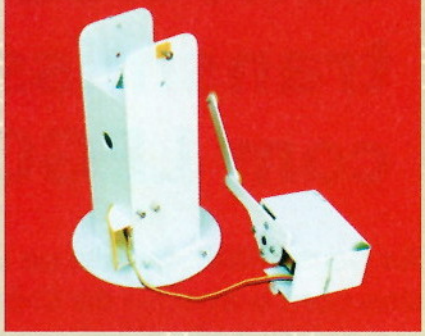


Figura 16 - Detalhe da instalação do acionador vertical na estrutura principal.

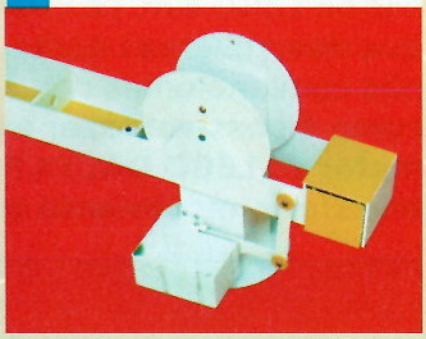


Figura 17 - O robô manipulador RM-1 completo sem pintura.



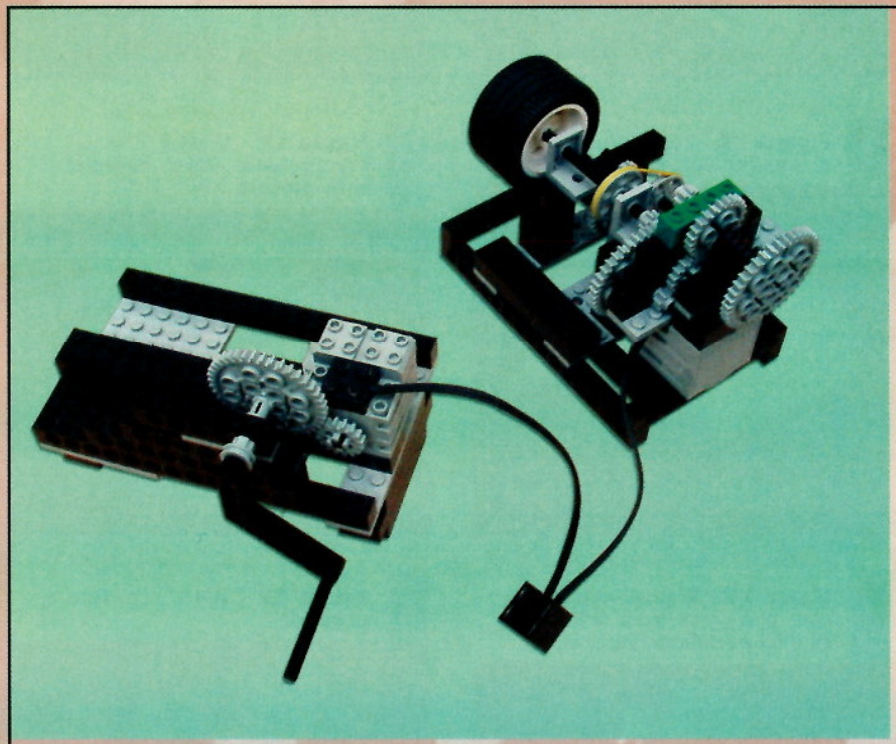
Figura 18 - RM-1 completo e acabado.



Conversão de energia e transmissão de movimento com LEGO

Newton C. Braga

As possibilidades de se usar o LEGO DACTA em experimentos relacionados com o ensino de Ciências nas escolas de ensino fundamental (e mesmo médio) são ilimitadas e não se restringem aos componentes encontrados no conjunto básico. Com componentes comuns, que podem ser adquiridos em casas de materiais eletrônicos, podemos ampliar os recursos do LEGO DACTA criando experimentos muito interessantes como os que descrevemos neste artigo, e até alguns outros que utilizam soluções não convencionais dos componentes do conjunto original. Mostramos como é possível fazer transformações de energia, gerar energia com o motor do LEGO DACTA funcionando como dínamo e, mais do que isso, transferir movimento ou armazenar esta energia num capacitor.









A energia não pode ser criada e nem destruída. O princípio da conservação da energia, que é ensinado já nas primeiras séries do ensino fundamental, pode ficar muito melhor gravado na mente dos estudantes se for acompanhado de experimentos interessantes.

Da mesma forma, os conceitos relacionados com as transformações de energia e transferências de movimento também podem ser muito melhor memorizados se acompanhados de experiências práticas.

A possibilidade de se usar o LEGO DACTA, unindo assim a tecnologia de um *kit* avançado com experiências tradicionais encaixa-se bem nos PCNs dos cursos de Ciências do ensino fundamental e médio.

Os experimentos que descrevemos são bastante simples, mas podem ensinar muito e, mais que isso, podem ser ampliados de modo a serem usados no aprendizado de outros princípios físicos relacionados com a geração, transmissão e armazenamento de energia estudados no ensino médio também, a seguir:

- Como funciona um dínamo - indução eletromagnética
- Como podemos transmitir movimentos a partir de transformações sucessivas de energia
- Como funciona um capacitor
- Como funciona um servomotor
- Curva de descarga de um capacitor
- Cálculo da energia armazenada num capacitor
- Transformações de energia
- Princípio de funcionamento do diodo semicondutor

Analisemos nossos experimentos.

A MONTAGEM

O que vamos fazer inicialmente é montar o motor que acompanha o LEGO DACTA como um dínamo que será acionado por um conjunto de engrenagens e transmissão via correia.

O movimento da manivela fará com que o eixo do motor gire, ocorrendo então a conversão da energia mecânica (que aciona a manivela) em energia elétrica (que aparece nos terminais do motor). A **figura 1** mostra como a montagem do sistema no LEGO DACTA é feita.

A polaridade da tensão que aparece nos terminais do motor depende do sentido de rotação da manivela e isso é importante na determinação do modo de ligação dos componentes seguintes do projeto.

A saída do motor é ligada a um diodo semicondutor e um capacitor, conforme ilustra a **figura 1**.

O diodo é necessário para funcionar como uma válvula de retenção. Quando giramos a manivela, a tensão gerada faz com que circule uma corrente do motor (funcionando como dínamo) para o capacitor. Essa corrente polariza o diodo no sentido direto e pode, portanto, passar sem dificuldades.

No entanto, ao parar de girar a manivela, sem o diodo, a corrente que carregou o capacitor voltaria e imediatamente haveria a descarga deste componente através do motor. Para evitar a descarga, o diodo impede o retorno da corrente, e com isso o motor se mantém carregado, veja a **figura 2**.

Isso significa que, para carregar o capacitor basta girar a manivela até que isso ocorra. Quando pararmos de girá-la, se nada houver ligado ao capacitor, ele se manterá carregado.

No nosso projeto colocamos um capacitor eletrolítico de filtro de 22000 µF ou 0,022 farads, que é

O motor como dínamo

Todos os motores elétricos, que são dispositivos que convertem energia elétrica em energia mecânica (força e movimento), também funcionam como geradores, ou seja, dispositivos que convertem energia mecânica em elétrica.

Isso ocorre quando, ao forçarmos o eixo do motor a girar, o conjunto de bobinas de seu interior corta as linhas de força dos pequenos ímãs que existem em seu interior, conforme mostra a **figura abaixo** .

Os pequenos motores encontrados em brinquedos podem gerar uma boa quantidade de energia elétrica quando forçados a girar, e isso acontece também com o motor usado no LEGO-DACTA.

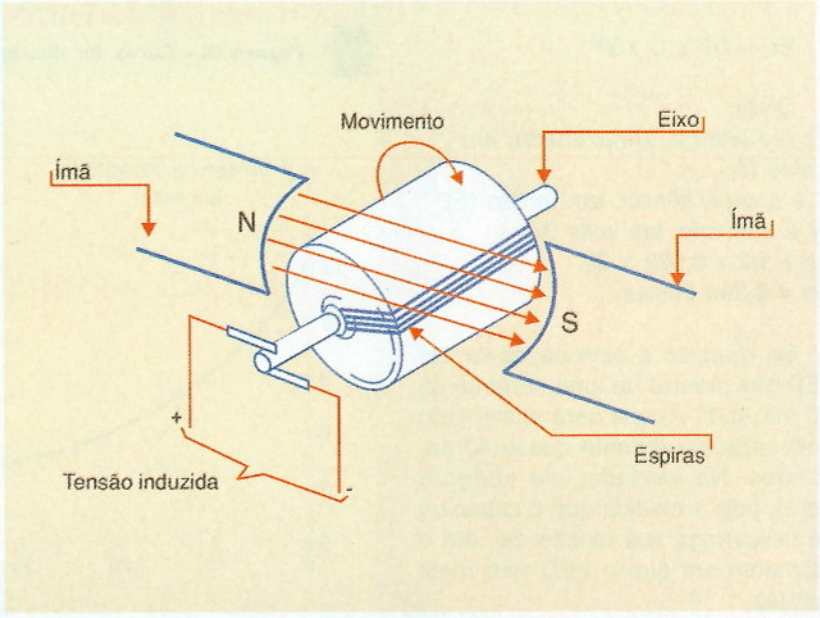
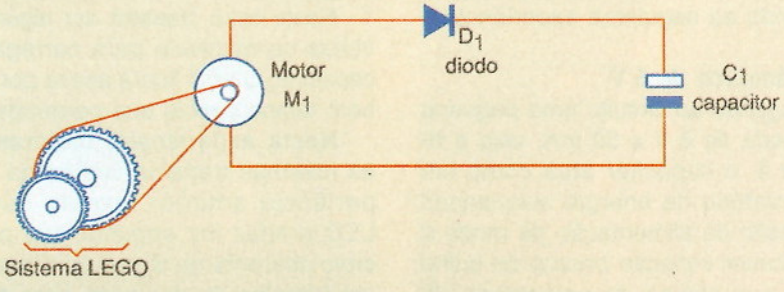


Figura 2 - Ligação da saída do motor.



uma capacidade enorme, a qual até há pouco tempo era difícil de ser obtida. Hoje, capacitores com este valor e até mesmo maiores podem ser obtidos em muitas lojas de componentes de eletrônica a preços relativamente baixos. Na Rua Santa Ifigênia (São Paulo) e travessas, vimos diversos desses capacitores sendo vendidos até como sucata!

Com uma tensão de 6 V, um capacitor desse tamanho pode armazenar uma energia equivalente a:

$$Ec = 1/2 \times C \times V^2$$

Onde:
Ec é a energia armazenada, em joules (J)
C é a capacitância, em farads (F)
V é a tensão, em volts (V)
Ec = 1/2 x 0,022 x 36
Ec = 0,396 Joules.

Se ligarmos a este capacitor um LED que precise de uma corrente de 10 mA (0,01 A), ele será alimentado pelo capacitor durante quase 40 segundos. Na verdade, ele apagará antes, pois à medida que o capacitor se descarrega sua tensão cai, até o momento em que o LED não mais conduz.

Estudar a queda de tensão do capacitor, conforme ilustra a **figura 3** é justamente uma das experiências que podemos fazer.

Na prática, podemos então ligar ao capacitor diversos dispositivos fazendo as seguintes experiências:

a) Lâmpada de 6 V

Ligando ao circuito uma pequena lâmpada de 6 V x 50 mA, veja a **figura 4**, o capacitor atua como um reservatório de energia suavizando a tensão da alimentação de modo a não haver variação brusca do brilho com a mudança da velocidade do dínamo ou mesmo quando paramos por um instante.

Nesta experiência, mostramos a transformação da energia mecânica (movimento da manivela) em energia elétrica (corrente para a lâmpada e capacitor) e depois radiante (luz).

Figura 2 - O diodo impede o retorno da corrente ao motor.

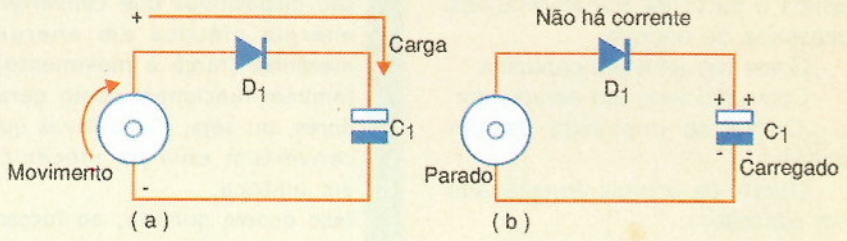
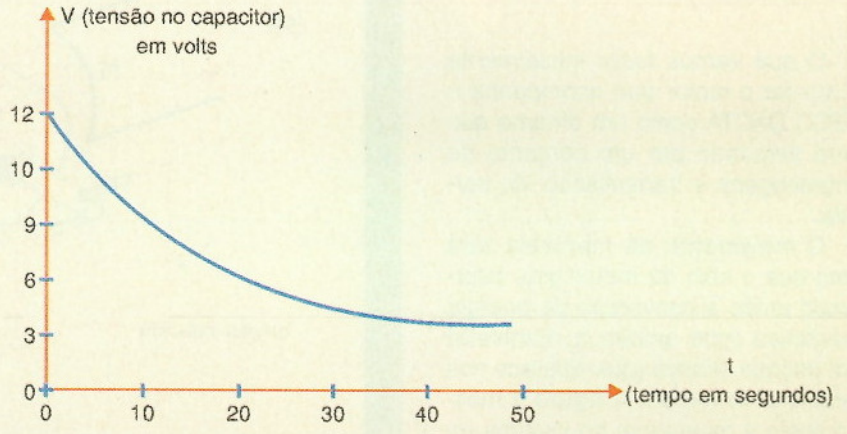


Figura 3 - Curva de descarga típica sobre um resistor (num resistor).



b) LED

Um LED vermelho comum pode ser ligado ao circuito, de acordo com a **figura 5**.

Neste caso, bastará dar algumas voltas na manivela para carregar o capacitor. O LED ficará aceso por um bom tempo depois que pararmos.

Nesta experiência, mostramos as mesmas transformações da experiência anterior, exceto que o LED produz luz segundo um princípio diferente da lâmpada. (Que tal um trabalho de pesquisa para descobrir isso?)

c) Curva de descarga do capacitor

Para essa finalidade, usamos um multímetro comum ligado na saída do capacitor; observe a **figura 6**.

Gira-se a manivela e, depois, anota-se de 10 em 10 segundos a tensão no capacitor, obtendo-se com isso a curva de descarga do capacitor. Com o multímetro ligado diretamente ao capacitor a resistência de descarga será a resistência equivalente ao multímetro na escala escolhida. Por exemplo, um multímetro analógico comum com 1 kohms / volt, na escala de 0-10 V se comporta como um resistor de 10 kohms.

Podemos aumentar a velocidade de descarga do capacitor, ligando em paralelo resistores de valores entre 1 kohms e 5 kohms.

d) Transmissão de movimento

Para esta interessante experiência, vamos usar apenas a montagem da parte de LEGO do Controle PWM

Figura 4 - Usando uma lâmpada de 6V x 50 mA como carga

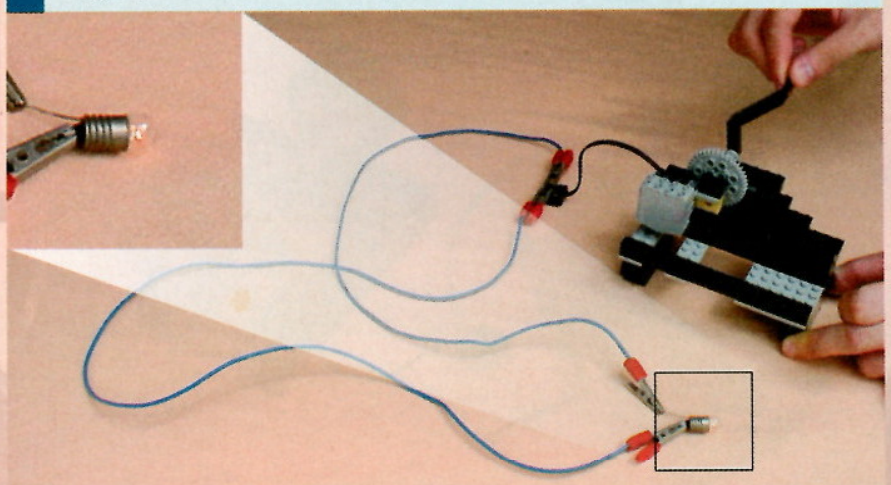


Figura 5 - O resistor usado em série com o LED pode ter valores entre 470Ω e 1,2 KΩ.

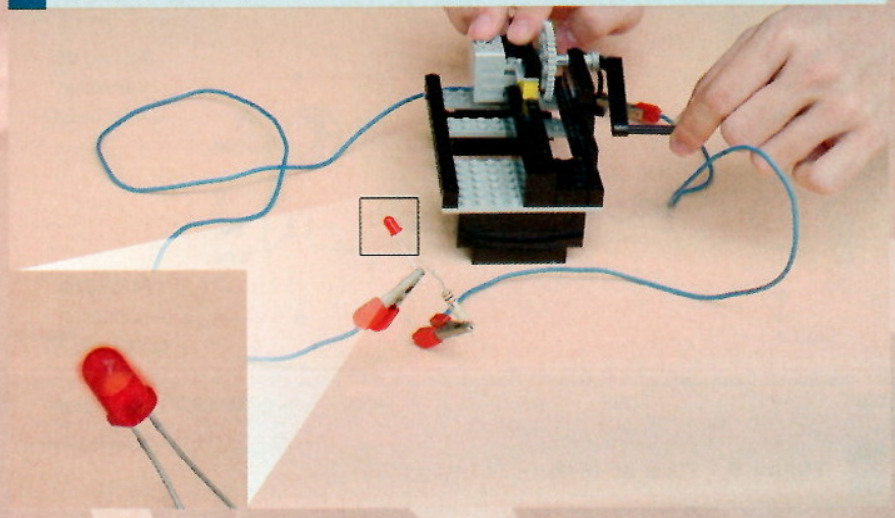
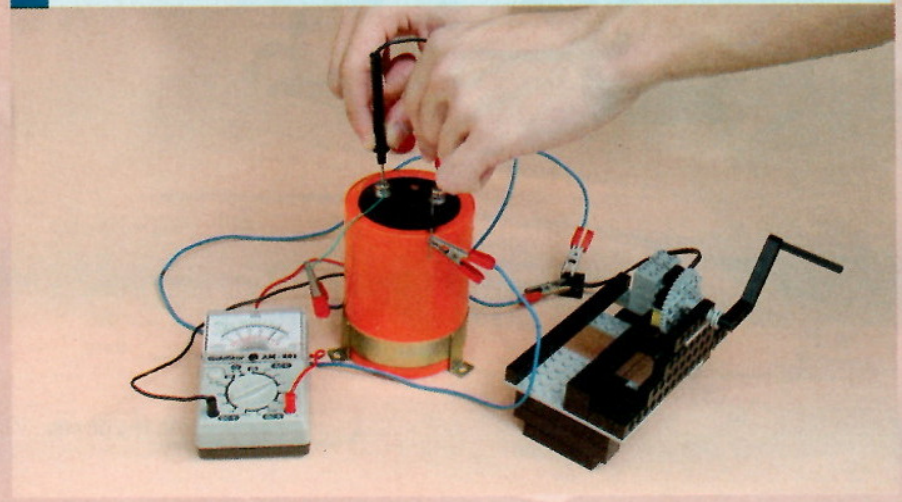


Figura 6 - Estudando a descarga do capacitor com um multímetro analógico de baixo custo.



publicado na revista Mecatrônica Fácil - 5 (Agregando Eletrônica ao Lego).

O que faremos, conforme mostra a **figura 7**, é acoplar o nosso gerador ao sistema mecânico mostrado naquele artigo, interligando-os com os fios disponíveis no LEGO DACTA.

Girando a manivela do Gerador, a energia mecânica necessária a esse movimento será convertida em energia elétrica e transmitida ao receptor, que é justamente o motor do conjunto de redução que, então, girará. Será interessante notar que o movimento de um é acompanhado quase que fielmente pelo outro, e isso também quando invertemos o sentido de rotação da manivela.

Demonstramos, de forma simples, como transmitir movimento através de transformações de energia e até dar idéia para projetos de controles usando o par de motor do LEGO DACTA.

Que tal projetar um elevador controlado a distância pela manivela ou um robô que caminhe com o girar da manivela?

e) Alimentando um rádio

Na **figura 8** exibimos um circuito de rádio experimental que pode ser alimentado pelo capacitor carregado por muito tempo, pois seu consumo é muito baixo. Umas maniveladas a cada 5 a 10 minutos serão necessárias para manter esse rádio em operação.

Trata-se de uma montagem bastante interessante para uma Feira de Ciências ou demonstração.

O rádio tem volume para excitar apenas um fone de ouvido, e exige uma antena externa. Ele não funcionará em locais que tenham elevado nível de interferência como, por exemplo, em lugares com muitas lâmpadas fluorescentes acesas ou ainda dentro de um prédio com uma estrutura de metal muito grande. O ideal é que seja usado em locais abertos, onde a antena (um fio comum de 5 a 20 metros) possa ser esticada.

Na **figura 9** mostramos como este rádio pode ser implementado

numa pequena placa de circuito impresso.

A bobina é formada por 80 a 100 voltas de fio esmaltado 28 AWG num bastão de ferrite de 1 cm de diâmetro e comprimento de 12 a 20 cm. A tomada é feita na vigésima espira a partir do lado da ligação à terra. Para ligar a bobina, raspe as pontas do fio esmaltado antes de soldar.

O fone de ouvido é um pequeno alto-falante de 2,5 a 5 cm colocado numa caixinha ou mesmo um fone comum.

A ligação a terra pode ser feita segurando-se a ponta do fio ou mesmo ligando-o a qualquer objeto metálico de porte.

O capacitor de sintonia pode ser aproveitado de qualquer rádio de ondas médias fora de uso.

Dê algumas maniveladas no gerador para carregar o capacitor, e depois procure sintonizar alguma estação local mexendo no capacitor variável.

Lista de componentes

Para o rádio

Q_1 - BC548 ou equivalente - transistor NPN de uso geral
Q_2 - BC558 ou equivalente - transistor PNP de uso geral
D_1 - 1N34 ou 1N60 - qualquer diodo de germânio
L_1 - Bobina - ver texto
CV - capacitor variável
FTE ou Fone - Fone de baixa impedância ou pequeno alto-falante
C1 - 100 nF - capacitor de poliéster
R_1 - 470 kΩ x 1/8 W - resistor - amarelo, violeta, amarelo
Diversos: placa de circuito impresso, antena, terra, fios, solda, etc.

Para o LEGO DACTA

D_1 - 1N4002 ou 1N4004 - diodo de silício
C_1 - 22 000 µF ou maior x 9 a 50 V - capacitor eletrolítico
Diversos: fios, solda, etc.

Figura 7 - Acoplamento do gerador ao sistema mecânico.

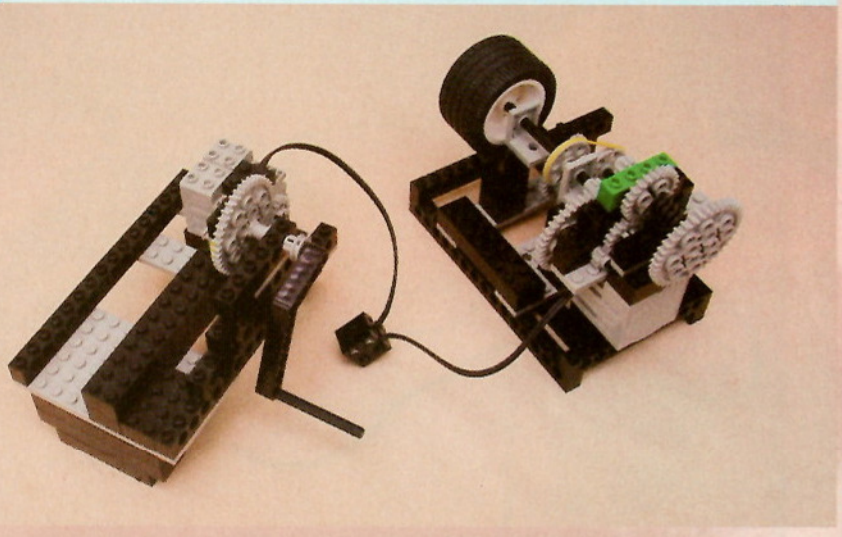


Figura 8 - Rádio AM alimentado "à manivela".

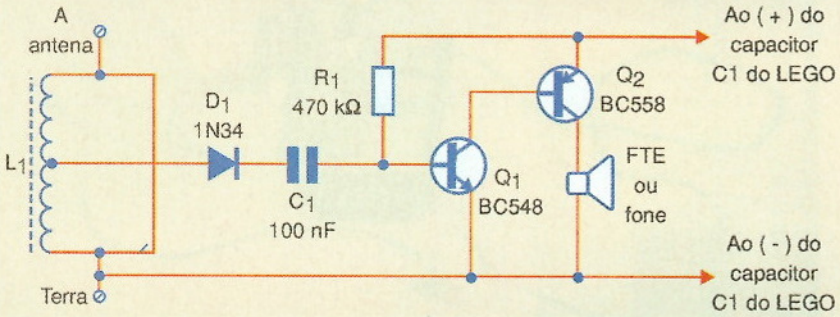
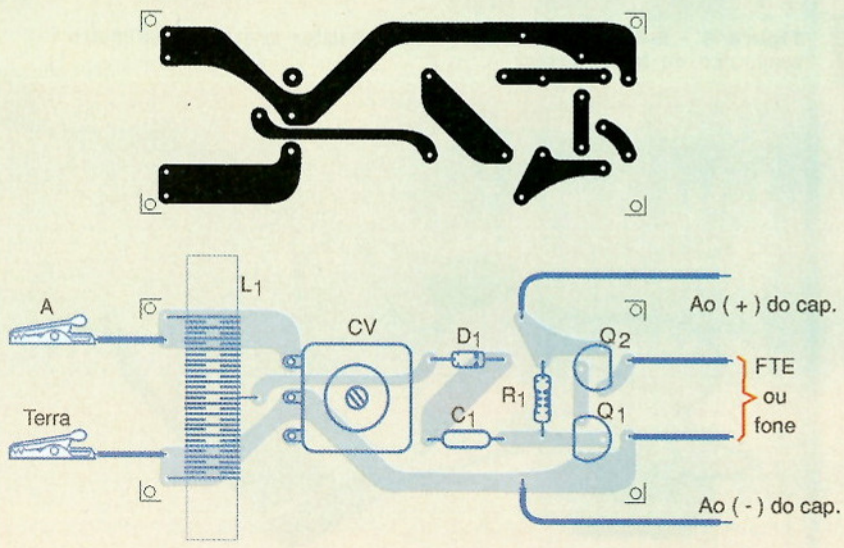


Figura 9 - Placa do rádio.



Muito mais que energia

Prof. Francisco L. de Sousa - francisco@eca.com.br - ECA - Escola Comunitária de Arujá
Programa Educacional Pueri Domus Escolas Associadas

No mundo moderno, a todo instante, podemos testemunhar a conversão de energia elétrica ou de combustíveis fósseis em movimento, mas e o inverso dessa transformação? Poderíamos citar as usinas hidrelétricas e outros exemplos, todos aparentemente distantes do dia-a-dia.
Talvez por isso a montagem deste experimento tenha despertado nos alunos tamanho interesse e surpresa pelos seus resultados. A transformação de energia mecânica, aplicada por eles mesmos, em um motor, gerando energia elétrica que pode ser utilizada para movimentar outro motor ou acender uma lâmpada, envolve muito mais do que a simples construção de um modelo, transcende as "disciplinas" e envolve o aluno, seja qual for sua idade, em uma discussão profunda de conceitos.
Só a constatação do fato (que energia mecânica pode ser convertida em energia elétrica) já gera os primeiros "desequilíbrios" e causa as indagações que levam o aluno a compreender a energia como uma coisa única, desvinculada da idéia de compartimentos, ou seja, o aluno não se concentra mais em decorar os tipos de energia (solar, elétrica, eólica, potencial, mecânica, térmica, etc.), mas se empenha em formular situações para propiciar a conversão de um tipo em outro. Surgem propostas como a de se acoplar uma hélice no motor para aproveitar a energia eólica ou mesmo idéias de sistemas complexos, como a de um dos grupos que levantou a possibilidade de se aproveitar a água que escorre dos telhados para girar pequenas turbinas embutidas no encanamento.
É nessa formulação de hipóteses que surge a possibilidade de unir as diversas disciplinas, já que é exigido do aluno um conhecimento abrangente. Tomemos como exemplo a proposta citada sobre aproveitamento da água da chuva: muitos questionaram sobre onde não chove com freqüência (ótima ponte para trabalhar o Clima e Regiões Geográficas) ou o quanto de energia poderíamos produzir assim (chance para utilizar a Matemática e a Física).
Este tipo de trabalho expande a compreensão dos alunos, de forma multidisciplinar, aguçando sua forma de enfrentar os desafios e problemas que lhes são apresentados.

Visite o site:

www.eca.com.br

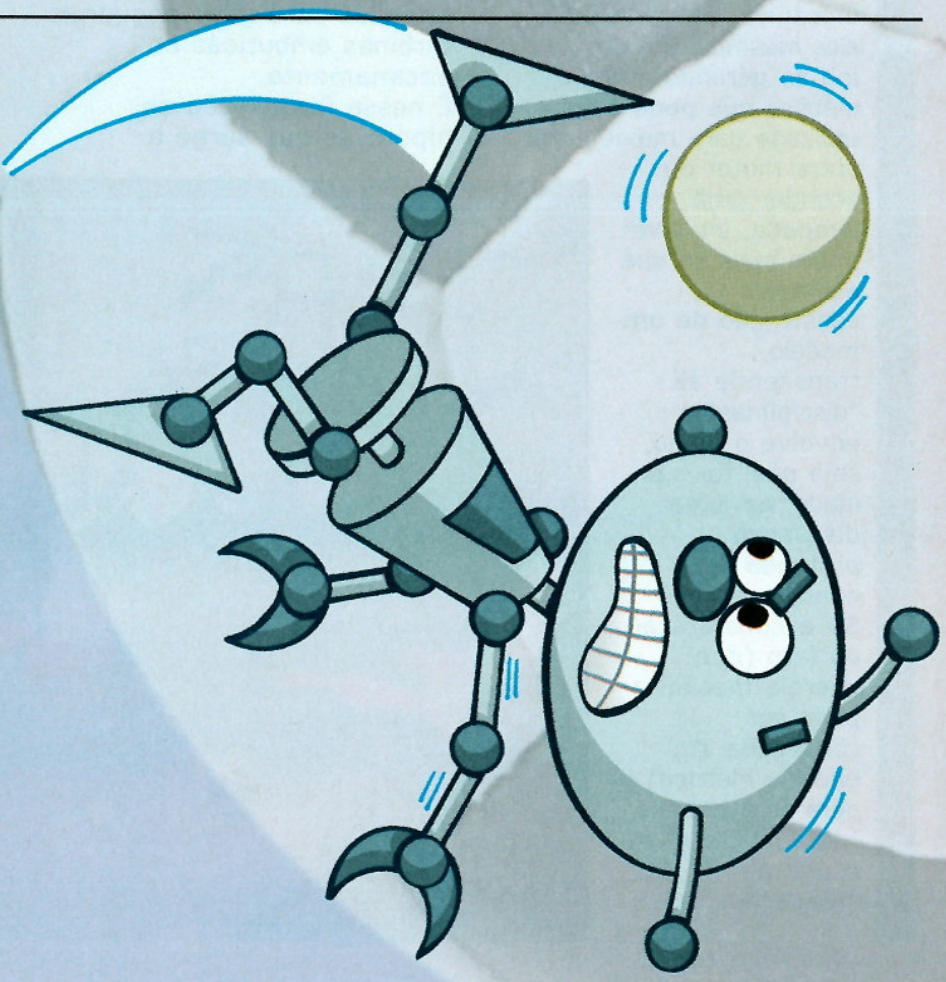


Futebol de Robôs - 4ªparte

O software que faz os robôs jogarem futebol

Ewaldo Luiz de Mattos Mehl
Gabriel dos Santos

Nesse quarto e conclusivo artigo, fechamos a série sobre o projeto Futebol de Robôs discorrendo um pouco sobre o sistema de *software* por trás dos robôs-jogadores. Ainda que o aparato eletrônico e mecânico seja essencial, de fato, a criação do Futebol de Robôs teve muito a ver com a necessidade de fomentar o desenvolvimento de técnicas sofisticadas e avançadas em diversas áreas da Ciência da Computação. Neste artigo, apresentaremos a arquitetura geral do sistema e estaremos apontando quais as áreas da ciência que se beneficiam de desenvolvimentos em cada parte da sistema.



Nos artigos anteriores, foram vistos diversos aspectos do Futebol de Robôs, desde a sua história até os aspectos de *hardware*. Neste artigo, deixaremos o *hardware* de lado para falarmos um pouco de *software*. Começamos respondendo a uma possível pergunta do leitor: se esta é uma revista de Mecatrônica, como se deve abordar a descrição de um *software* como é o usado no Futebol de Robôs, uma vez que esta não é um revista sobre programação? O fato é que o desenvolvimento do *software* é um aspecto essencial do Futebol de Robôs e deve ser pensado como um elemento tão importante para se ter sucesso em um campeonato como os próprios robôs. Com o propósito de se evitar a necessidade de conhecimentos profundos na área de computação, veremos o sistema em uma abordagem mais abstrata, desprovida dos detalhes excessivamente técnicos.

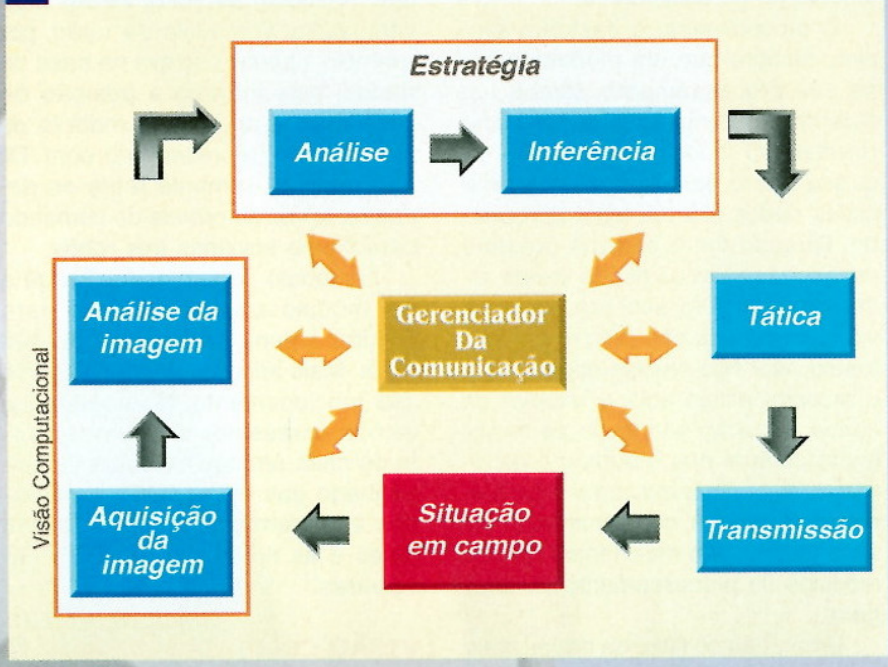
Desde já, para evitar confusões, deve-se fazer um esclarecimento

com relação aos artigos anteriores, onde citamos que existem empresas que comercializam *kits* para Futebol de Robôs. Ao se comprar um desses *kits*, recebe-se geralmente os robôs juntamente com um *software* que permite testar o sistema. Tal *software* é bastante simples e desprovido de vários elementos essenciais para que o sistema seja completamente funcional. Ou seja, se uma equipe for para um campeonato internacional apenas com o *software* que é fornecido junto com estes *kits* de Futebol de Robôs, não poderá esperar que obtenha bons resultados. As equipes internacionais que estão participando do projeto Futebol de Robôs têm desenvolvido cada qual seu próprio *software*, com características bem específicas e tentando cada vez mais introduzir táticas que possam garantir-lhes bons resultados.

O sistema sobre o qual discutiremos aqui foi inteiramente desenvolvido pela nossa equipe para o sistema operacional *Linux*, rodando em um microcomputador tipo PC comum. Conforme comentamos nos artigos anteriores, é importante que o microcomputador tenha uma boa velocidade de processamento, porquê o *software* é bastante complexo. Surge aqui, porém, a primeira questão sobre o programa: porque usar o sistema operacional *Linux* e não o *Windows* da Microsoft? O motivo mais importante é o fato do *Linux* ser um sistema gratuito e aberto. As vantagens de ser gratuito são óbvias. Quanto ao aspecto de ser aberto, a maior vantagem é flexibilidade do sistema, que pode até mesmo ser alterado, se necessário, para se adaptar melhor a aplicação. Ou seja, ao se optar pelo *Windows*, poderíamos ter um ótimo *software*, mas em razão dos detalhes do sistema operacional serem guardados como segredo comercial pela Microsoft, acaba sendo impossível obter-se um desempenho plenamente confiável.

Desde já, podemos observar o primeiro aspecto interessante do *software*: uma vez que o sistema funciona em um microcomputador tipo PC e utilizando um sistema operacional gratuito, pode-se dizer que qualquer pessoa que possua um computador com bom desempenho poderia estar rodando o nosso sistema para jogar Futebol de Robôs. Esse é um ponto importante, pois uma das filosofias do projeto é justamente incentivar a entrada de cada vez mais equipes nas disputas, com o que incentiva-se a pesquisa em Robótica em todo o mundo.

Figura 1 – Diagrama mostrando os vários módulos que compõem o *software* usado para o Futebol de Robôs.



VISÃO GERAL E ARQUITETURA

O processamento do sistema de *software* do Futebol de Robôs pode ser descrito por um ciclo que é executado constantemente até o final do jogo. O ciclo pode ser resumido da seguinte forma:

Primeiramente, a câmera posicionada sobre o campo obtém uma imagem da situação atual do jogo. As informações são coletadas pelo sistema por um módulo do *software* denominado *aquisição de imagem*. De posse da imagem atual, um outro módulo chamado de *análise da imagem* faz uma varredura na imagem em busca de cada um dos robôs e da bola. A saída desse módulo é, portanto, um conjunto de coordenadas que representam a posição dos robôs e da bola. Juntos, os módulos de *aquisição de imagem* e de *análise de imagem* formam o módulo de *Visão Computacional*.

A seguir, entra em ação o módulo de *estratégia*, o qual analisa a situação e determina qual procedimento cada robô do nosso time deve tomar. Esses comandos são passados para o módulo de *tática*, o qual decide como os movimentos determinados devem ser realizados em termos de comandos simplificados.

Finalmente, esses comandos simplificados são transmitidos para os robôs pelo módulo de *transmissão*. Dessa forma, os robôs deverão executar os comandos determinados pelo *software* e se movimentarão pelo campo, gerando uma nova situação, o que implica no reinício do ciclo.

A **Figura 1** ilustra o ciclo descrito.

Note que no centro da **figura 1** existe um módulo chamado *Gerenciador da Comunicação*. Embora o fluxo natural do sistema seja representado pelas setas pretas, os dados gerados pelos módulos não trafegam diretamente para o próximo

módulo. Ao invés disso, a comunicação inter-módulos é realizada pelo módulo de gerenciamento.

O procedimento é bastante simples: sempre que um módulo termina seu processamento, temos um determinado conjunto de dados como resultado. O módulo em questão, requisita que o gerenciador armazene esses dados em um local apropriado. Quando for o caso, o próximo módulo requisita os novos dados armazenados pelo seu precursor. A vantagem dessa abordagem é que o transmissor não precisa esperar que o receptor esteja apto a receber os dados, situação que pode se tornar muito comum nos módulos relacionados à inteligência, uma vez que estes, devido à complexidade de suas tarefas, são mais lentos que os módulos de processamento de imagem.

Uma visão do fluxo de dados mais próxima da realidade pode ser vista na **figura 2**.

No centro vertical da imagem da **Figura 2**, temos representada a região na qual os dados são armazenados, também conhecida como *Base de Conhecimento*. Acima e abaixo, vários módulos diferentes interagem com esta região por intermédio do módulo de gerenciamento. As setas azuis indicam que o módulo em questão escreve na base de dados, enquanto as setas verdes indicam leitura. O módulo de *visão*, por exemplo, apenas escreve na base de dados, pois informa a posição de cada robô e da bola; *o módulo de transmissão* (representado com **TX** na **Figura 2**) somente retira os dados na forma de ordens de comando para serem enviados aos robôs.

O módulo denominado *Gauge* é um módulo auxiliar utilizado para calibrar alguns parâmetros do sistema. Mais adiante, descreveremos seu funcionamento. Na seqüência, vamos apresentar o funcionamento de cada um dos módulos, acompanhado das dificuldades associadas à implementação de cada um deles e as áreas de pesquisa envolvidas.

VISÃO COMPUTACIONAL

O nome desse módulo é também o nome dado a uma área de pesquisa de grande importância dentro da Ciência da Computação. Por *visão* entende-se não apenas o processo de adquirir a imagem, o qual é feito com facilidade por diversas câmeras, mas principalmente o de *interpretar* a imagem, ou seja, determinar o significado dos elementos presentes na mesma.

Um exemplo típico de processamento de um sistema de visão computacional é o isolamento e classificação dos diferentes elementos de uma cena, por exemplo o reconhecimento da face de uma pessoa. Pode parecer uma tarefa simples, mas a quantidade de fatores envolvidos é enorme. Um sistema de reconhecimento de pessoas, como neste exemplo, deve ser capaz de resolver problemas decorrentes de imagens deficientes, e até pequenas diferenças na iluminação como se a pessoa está sorrindo ou séria influenciam muito a imagem e podem modificar completamente um padrão.

O objetivo a longo prazo que deseja-se atingir nessas pesquisas é imitar ou mesmo superar a capacidade humana de visão. Assim como nós somos capazes de identificar um amigo ou parente mesmo que ele ou ela tenha modificado o penteado, as pesquisas de *Visão Computacional* objetivam dar às máquinas não só a capacidade de *ver*, mas também de interpretar corretamente uma imagem.

Dentro do futebol de robôs, nosso alvo é muito mais modesto. O sistema deve apenas determinar onde está cada robô e onde está a bola. Para facilitar o processo de identificação, cada robô possui uma adesivo em sua parte superior com duas cores, uma para especificar o time ao qual pertence e a outra para identificar o robô. Dessa forma, reduzimos nosso problema em encontrar regiões coloridas na imagem entregue pela câmera para o micro-

Figura 2 – Fluxo de informações entre os módulos que compõem o *software*.

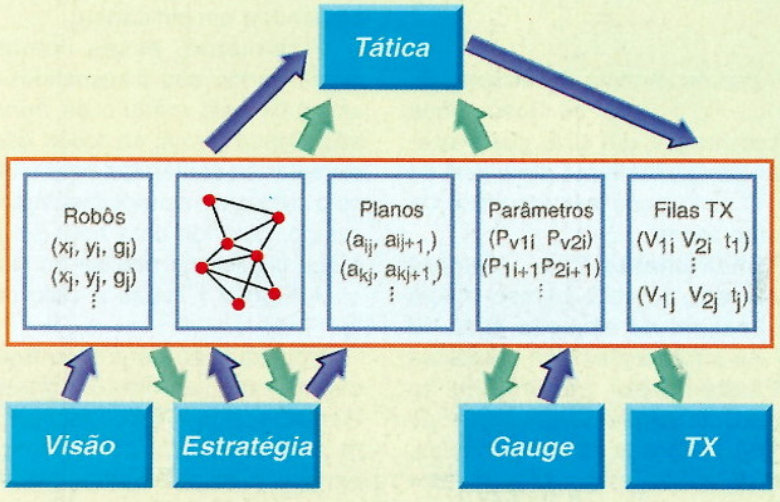
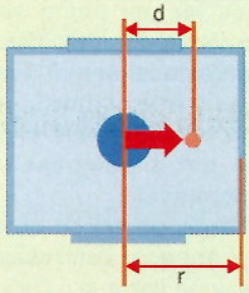


Figura 3 – Esquema de marcas coloridas para identificar os robôs.



computador, e associar o centro da região ao robô.

Na **figura 3** temos um esquema apresentando como as cores são colocadas no robô. O círculo no centro é usado para identificar o robô, também chamado de *marca*, enquanto um círculo menor, denominado *camiseta*, identifica o time.

Ao final do processamento, o módulo de visão determina um conjunto de coordenadas cartesianas representando a posição dos robôs. Por exemplo, se o robô A está na posição x=140 e y=200 e a bola está na posição x=331 e y=592, o sistema teria uma saída semelhante à seguinte: A(140, 200) bola(331, 592)

Cada robô, seja adversário ou não, é associado na verdade a um par de coordenadas cartesianas, uma representando o centro da marca e a outra representando o centro da camiseta. São necessárias duas coordenadas para que seja possível determinar a orientação e a direção do robô. A orientação é determinada pela reta que passa por ambas as coordenadas, enquanto o sentido do robô, ou seja, para onde o robô está "olhando", é determinado pela seta que parte da marca e vai até a camiseta, tal como ilustrado pela seta vermelha na **figura 3**.

ESTRATÉGIA

Reunindo os módulos de *Estratégia* e *Tática*, poderíamos definir um módulo de *Inteligência Computacional*. Enquanto a *Visão Computacional* busca imitar a visão humana, a área de *Inteligência Computacional* (também conhecida como *Inteligência Artificial*) se propõe a um desafio ainda mais complexo: imitar a inteligência humana.

Se as pesquisas na área de *Visão Computacional* ainda estão na infância, na *Inteligência Computacional* mal começamos a engatinhar. Ainda estamos muito longe de entender como funciona a inteligência humana e não há previsão de quanto tempo levaremos para criar um sistema capaz de imitá-la.

O estudo e desenvolvimento na área de Inteligência Artificial é uma das principais razões da existência do Futebol de Robôs. Há pouco tempo, cientistas da IBM anunciaram a criação de um sistema capaz de vencer o melhor enxadrista humano, o *DEEP BLUE*. Realmente, em partidas reais o sistema computacional mostrou-se capaz de derrotar mestres internacionais de xadrez. Para entendermos a complexidade do problema a ser resolvido no Futebol de Robôs, nada melhor que compará-lo portanto ao jogo de xadrez, uma vez que considerou-se que o desenvolvimento do *DEEP BLUE* foi um marco na área (ver **tabela 1**)

Tabela 1 - Comparativo

	Xadrez	Futebol de robôs
Espaço	Discreto: as peças estão em uma das 32 x 32 posições.	Contínuo: os robôs podem estar em qualquer lugar dentro do "campo".
Dinamismo	Estático	Dinâmico
Previsibilidade	Previsível	Imprevisível
Tempo para decidir	Arbitrário	Fração de segundo

A tabela mostra que existem algumas diferenças entre o problemas. O problema do Futebol de Robôs é muito mais complexo, portanto, do que jogar Xadrez, em especial devido à imprevisibilidade e ao dinamismo do jogo, além da dificuldade de termos pouco tempo para tomar uma decisão.

No sistema desenvolvido para o Futebol de Robôs, o processo passa por duas fases: a análise e a inferência. A análise consiste em tentar classificar a situação atual, ou seja, podemos dizer que a análise consiste em entender *o quê* está acontecendo. Uma vez que o sistema determinou o que está acontecendo, é necessário que o sistema tome uma decisão (inferência) para reagir a esta situação.

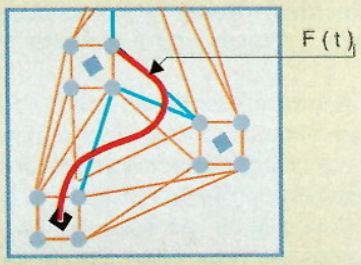
Por exemplo, se a bola está próxima de um dos robôs do "nosso" time, este pode empurrar a bola para o gol. Mas se houver uma obstrução, o sistema deve ser capaz de reagir de forma inteligente, talvez impulsionando a bola para uma região na qual outro robô possa interceptá-la e conduzi-la ao gol livremente. Teríamos, nesse caso, um autêntico "passe de bola", semelhante ao que ocorre com os jogadores humanos (ou melhor, com os *bons* jogadores humanos...).

O objetivo do módulo de *estratégia* é não menos que determinar quais as ações que cada robô deve tomar. Ou seja, a saída do módulo de estratégia é um *plano de ação*, onde a cada robô é associada uma ação que deve ser executada em reação à situação do jogo naquele instante.

TÁTICA

Além da área de *Inteligência Computacional*, o Futebol de Robôs gera uma série de problemas que envolvem as áreas de Geometria Analítica e do Cálculo Diferencial e Integral. Para quem acha que Computação não tem nada a ver com Matemática, temos no Futebol de Robôs uma prova do contrário!

Figura 4 – Cálculo da trajetória do robô.



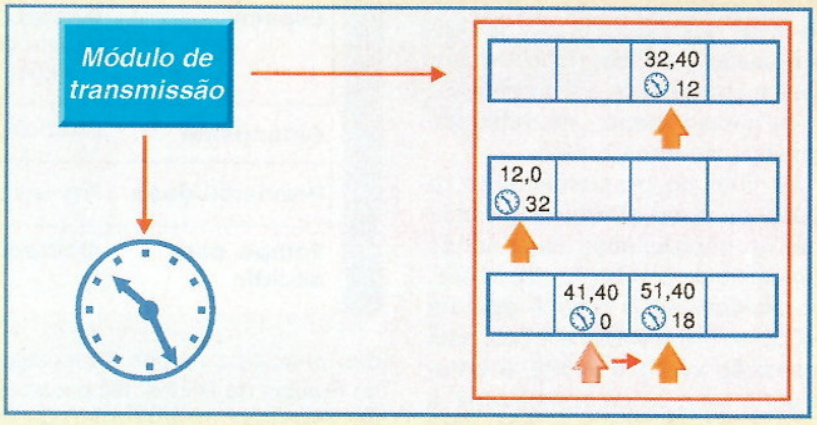
Conforme vimos, o módulo de *estratégia* determina quais ações devem ser realizadas, mas não há a preocupação em *como* elas serão realizadas. O objetivo do módulo de tática é justamente determinar como a ação deve ser realizada, desde a decisão de qual o caminho a ser tomado até a determinação do ângulo no qual o robô deve estar para que, ao atingir a bola, esta siga para a direção que o módulo de estratégia desejava.

Algoritmos para encontrar caminho e para determinar equações de curvas são bastante aplicados em Robótica. A inteligência entra na escolha dos pontos pelo qual o robô deve passar para chegar ao seu objetivo sem colidir. Observe na **figura 4**, o quadrado preto representa a posição de saída do robô. O primeiro passo é determinar um caminho para que o robô chegue ao destino sem colidir com os adversários. As linhas de cor azul na figura 4 representam o caminho encontrado. Como não é interessante que o robô ande apenas em linha reta, até porque haveria um desperdício de tempo, é necessário determinar a trajetória real do robô. Em nossa abordagem utilizamos os pontos do caminho encontrado para determinar uma curva, representada por uma função F(t), pela qual o robô deve passar. Uma questão importante é que os robôs na realidade só entendem comandos muito simples, na forma de velocidade de rodas (por exemplo, velocidade 30 na roda 1 e 40 na roda 2). Dessa forma, não podemos mandar uma curva para o robô seguir. Portanto, a tática é também responsável por determinar um conjunto de comandos do seguinte modo:

Velocidade Roda 1 / Velocidade Roda 2 / Tempo de Execução

A interpretação do comando é muito simples: o robô deverá manter as velocidades especificadas por um determinado tempo. Ou seja, a tática deve definir um conjunto de comandos de velocidade que, quando executados, farão com que o robô realize a curva determinada. Para ajudar na determinação desses comandos, o módulo de *tática* utiliza um conjunto de parâmetros calculados por um módulo chamado *Gauge*, o qual calibra o sistema.

Figura 5 – A formação de uma "fila" de comandos.



TRANSMISSÃO

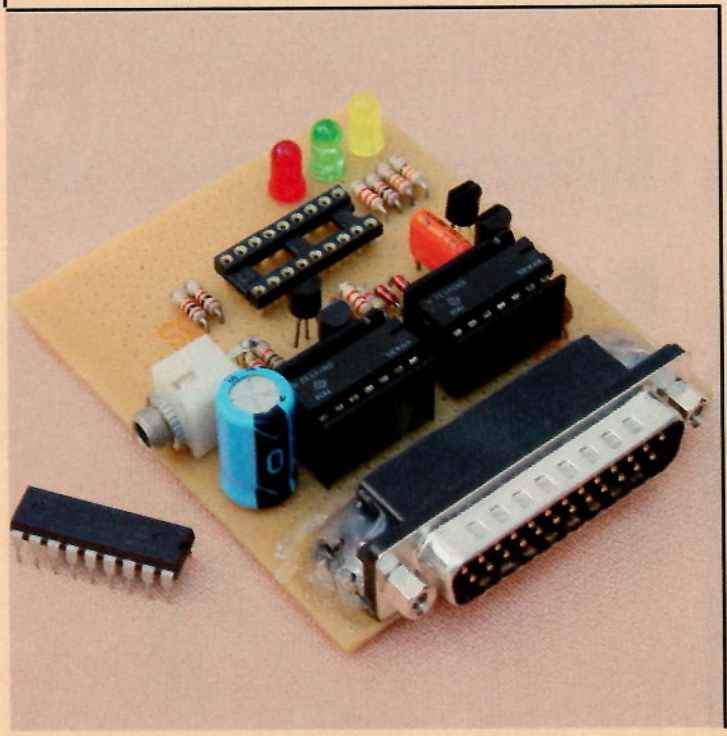
Finalmente, o módulo de *transmissão* tem duas responsabilidades: enviar os comandos para os robôs e controlar o *tempo*. Os robôs por si só não entendem o parâmetro de tempo descrito anteriormente. O módulo de transmissão mantém uma "fila" de comandos para cada robô, a qual contém os comandos gerados pela tática. Para controlar quando deve ser enviado o próximo comando, a cada unidade de tempo o sistema subtrai uma unidade do comando na fila. Quando o tempo chega a zero, o comando é retirado da fila e o próximo comando é atendido. A **figura 5** apresenta uma ilustração do processo. Podemos observar três filas no diagrama. Note que na última fila o comando **41,40** chegou ao tempo zero, e portanto há uma indicação de que o próximo comando (**51,40**) será executado.

CONCLUSÃO

Conforme pode-se perceber, desenvolver o *software* para o Futebol de Robôs é uma tarefa com grande grau de complexidade. Longe no entanto de se ficar preocupado ante tal dificuldade, trata-se de um desafio apaixonante para quem se envolve profundamente com o desenvolvimento do *software*. Nesse ponto, reside um dos elementos mais importantes do projeto Futebol de Robôs: o *software* necessita elementos de cálculo e de geometria que freqüentemente os estudantes consideram aborrecidos e sem aplicação no mundo real. Ao contrário, ao ver um robô executar uma curva precisa desviando-se de obstáculos comprova-se que a Matemática, a Geometria e a Física são ciências vivas, que a aridez dos livros nos faz esquecer que estão presentes na descrição do movimento de uma bola de futebol, assim como podem levar um homem à Lua. Após envolver-nos no projeto Futebol de Robôs, adquirimos uma nova visão da complexidade que está por trás de tarefas aparentemente simples, como a ação de chutar uma bola. É possível que um dia tenhamos um robô capaz de brincar com os nossos filhos ou netos e, quando esse dia chegar, a Robótica terá demonstrado ter atingido um grau extremamente avançado, para o qual o projeto Futebol de Robôs terá dado uma importante contribuição.

Como Gravar Microcontrolador PIC

Márcio José Soares



Na primeira parte desta série, discorremos sobre os registradores do PIC, "partes" importantes em qualquer microcontrolador. Agora veremos algo mais prático com o projeto de um pequeno gravador, montado com componentes baratos, e que possibilitará gravar e verificar um microcontrolador PIC16F84A. Também faremos uma abordagem sobre o uso do MPASM, *assemblador* da Microchip (http://www.microchip.com). Com a soma destas duas ferramentas, o leitor poderá começar suas experiências práticas.

A Microchip disponibiliza gratuitamente seu *assemblador* MPASM, assim como ambientes de desenvolvimento como o MPLAB. Para executar alguns exemplos práticos um *assemblador* e o gravador já são o bastante.

Na página da Microchip é possível fazer o *download* do MPASM. O endereço: "http://www.microchip.com/1010/pline/tools/archive/other/61/index.htm"

Trata-se de um *link* direto para o *download* do arquivo **"asm 21500.zip"**. Lembrando que a Internet está em constante mutação, caso o leitor não consiga acessar o arquivo através deste *link*, deverá tentar fazê-lo através dos instrumentos de busca da página da Microchip: *"http://www.microchip.com"*.

De posse do arquivo, o leitor deverá descompactá-lo com o uso do "WinZip", **http://www.winzip.com**. Crie um subdiretório com o nome "c:\gravador" e copie os arquivos do MPASM para ele.

O leitor contará com dois *assembladores*: um para Windows 95/98 o ***mpasmwin.exe,*** que pode ser visto na **figura 1**, e outro para DOS o ***mpasm.exe***, demonstrado na **figura 2**. O uso de ambos é similar, porém como o programa do gravador foi desenvolvido para ambiente MS-DOS, para permitir que qualquer leitor possa utilizá-lo, preferimos aqui tratar do *assemblador* da Microchip para MS-DOS para compatibilizar os ambientes.

Para compilar um arquivo .ASM no MPASM-DOS, bastará ao leitor executar a seguinte linha de comando no MS-DOS: "Mpasm.exe / p16F84 arquivo.asm"

O *assemblador* irá compilar o arquivo.ASM (**figura 3**) orientado para o PIC16F84 e criará outros dois arquivos importantes. Um deles é o .HEX, que será utilizado pelo nosso gravador. O outro é o .ERR que terá a listagem do programa com as mensagens de erro, mensagens de "cuidado" (*warning*) e mensagens comuns. As mensagens de erro impedem a criação do arquivo .HEX, enquanto as mensagens de "cuidado" e mensagens comuns não o fazem. Verifique todas as mensagens e faça as correções no arquivo .ASM se necessário.

A Microchip fornece o manual do compilador para *download* e é aconselhável que o leitor faça um estudo sobre o mesmo. Será importante conhecer algumas diretivas do

Figura 1 – Mpasm para Windows.

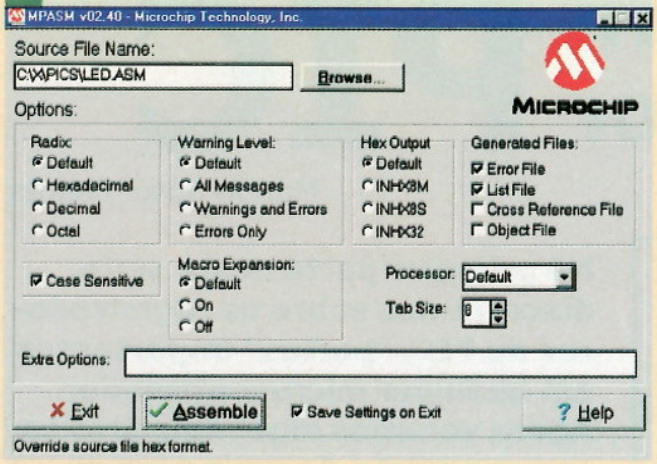


Figura 2 – Mpasm para DOS.

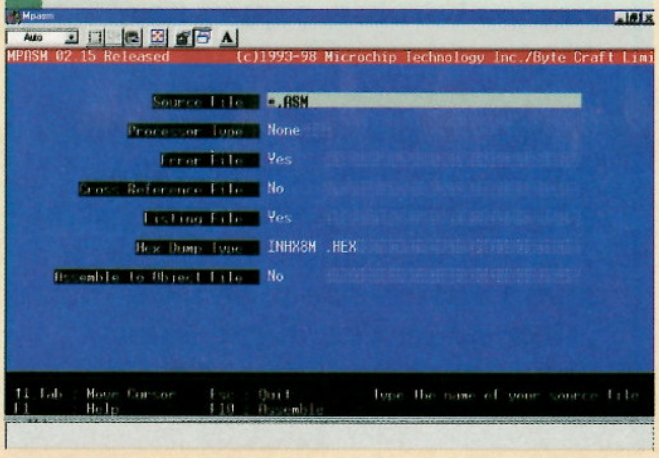


Figura 3 – Resultado da compilação.

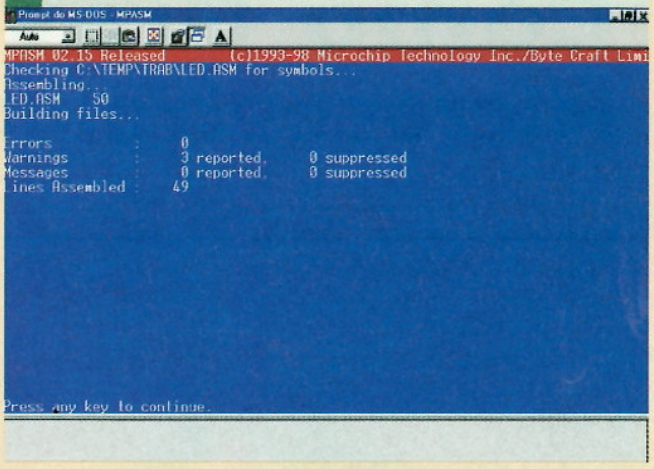
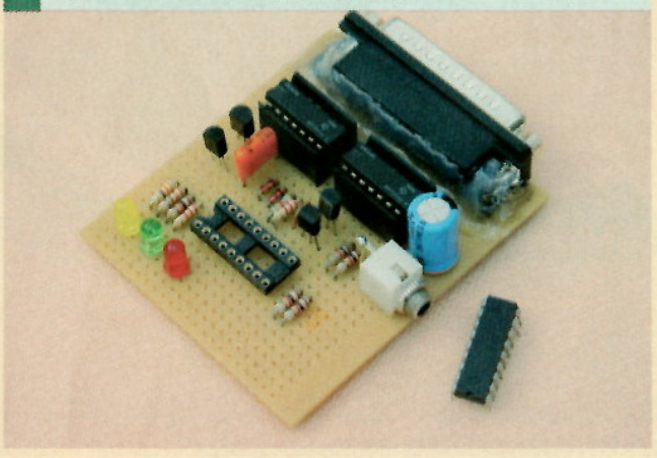


Figura 4 – Protótipo do gravador.



compilador no futuro. O *assemblador* tem um pequeno "help", basta passar o parâmetro "/h" durante a chamada do programa. Verifique no *site* da Microchip, **http://www.microchip.com** sobre o *download* o mesmo.

GRAVADOR

Acreditamos que a prática permite assimilar melhor a teoria. Pensando assim, desenvolvemos um pequeno gravador experimental, o PROGPIC1, que possibilitará gravar e verificar um microcontrolador PIC16F84A. Este gravador é ligado à porta paralela de um microcomputador PC e necessita do programa PROGPIC.EXE, que poderá ser obtido gratuitamente no *site* **http://www.mecatronicafacil.com.br.**

Montagem do gravador

O circuito do gravador pode ser visto na **figura 5**. Nosso protótipo foi montado em uma placa de circuito universal e fio "wire-wrapping" para as ligações. O uso de soquetes para os CIs é recomendável. Para o PIC ele é indispensável e deve, preferencialmente, ser do tipo torneado. Os LEDs devem ser de cores diferentes para facilitar a visualização durante a operação do gravador. Usamos um LED vermelho comum para a indicação "LIGADO" (alimentação), um LED verde comum para a indicação "PIC LIGADO" (VDD) e um LED amarelo comum para indicar a conexão VPP (14V) ao PIC. Os reguladores de voltagem 78L05 e 78L12 podem ser substituídos pelo tipo "K" de maior corrente e mais comum, caso o leitor tenha dificuldades em encontrar os tipos recomendados. Os reguladores são associados de forma a se obter 5 volts e 14 volts para gravação.

Nota

Este gravador tem caráter experimental e foi desenvolvido para operação com máquinas PC com DOS ou Windows 95/98 no "Prompt do MS-DOS". Todos os testes foram realizados em um Pentium MMX 166 Intel, um Pentium II 200 Intel e um K6II 400 AMD, apresentando funcionamento satisfatório.
O gravador depende de rotinas de temporização, que poderão ter um comportamento anormal em máquinas com *clocks* superiores a 500 MHz.

Para os diodos, não utilize diodos retificadores, apenas diodos de sinal. Estes são mais rápidos. É recomendável o uso de capacitores cerâmicos de 100 nF para o desacoplamento dos CIs. Ligue-os entre o positivo e o terra de cada CI.

O capacitor eletrolítico tem polaridade e esta deve ser observada. Os diodos também e uma inversão neste ponto poderá fazer com que o gravador não funcione. Todos os resistores são de 1/8 de watt para dissipação.

A fonte de alimentação do gravador deverá fornecer de 16 V a 18 V DC com uma corrente mínima de 300 mA. No gravador, existe um "plug" P_2 mono para a ligação desta fonte, porém algumas fontes comerciais utilizam outros tipos de "plug's". Neste caso, o leitor deverá compatibilizar o "plug" da placa com o da fonte, ou vice-versa. Na **figura 6** temos o circuito e o *lay-out* de uma fonte, que poderá facilmente ser montada pelo leitor.

Utilizamos um cabo de extensão para impressoras paralelas para fazer a ligação do PC à placa. Ele é comum em lojas de informática. Entretanto, o leitor poderá também montar o mesmo utilizando o esquema da **figura 7**.

Confira cuidadosamente todas as ligações feitas na placa, fonte e cabo. Utilize um multímetro ou "testador de continuidade" para essa tarefa.

O DB25 utilizado na placa deverá ser fixado com "cola quente" ou outro adesivo para melhorar a rigidez mecânica, evitando que a placa e as ligações se danifiquem ao ligar o cabo.

Teste e uso do gravador

Para testar o gravador, o leitor precisará do programa PROGPIC.EXE e de um arquivo HEX. Este arquivo HEX é gerado pelo MPASM durante a compilação e contém o código do programa a ser gravado na PIC. ***É aconselhável que o leitor instale o programa do gravador no mesmo subdiretório do compilador MPASM para facilitar ao máximo o trabalho de compilação/gravação.***

Figura 5 – Circuito do gravador.

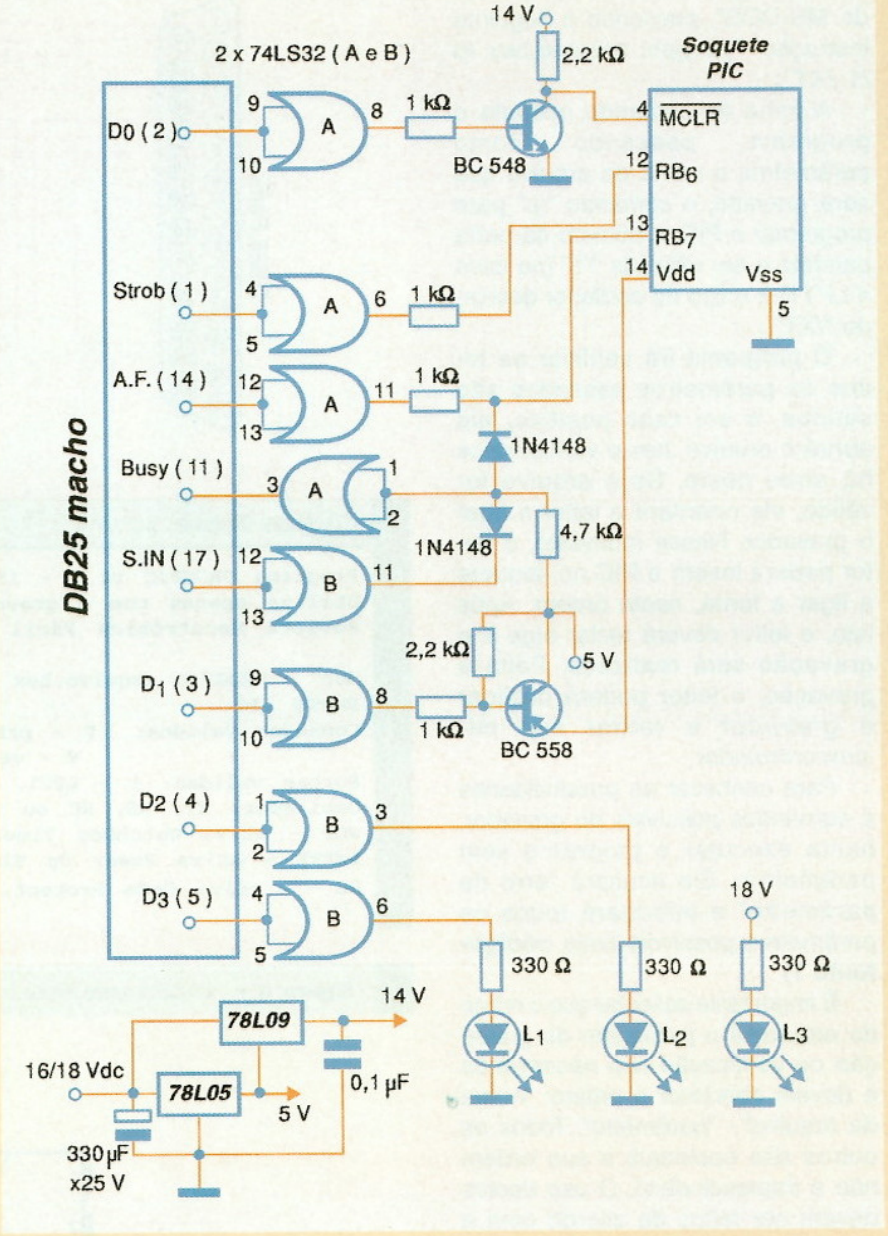
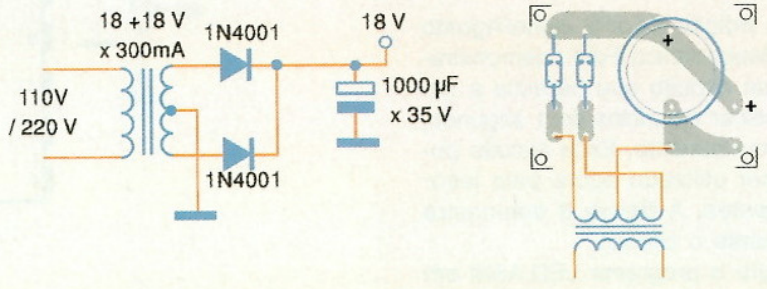


Figura 6 – Esquema para a fonte.



Conecte o cabo de comunicação entre o PC e a placa gravadora. Execute o programa a partir do "prompt do MS-DOS", passando a seguinte instrução: **"progpic arquivo.hex /p /1 /XT"**.

A linha de comando executa o programa passando como parâmetros o nome do arquivo que será gravado, o comando "/p" para programar o PIC, o número da porta paralela a ser utilizada "/1" (no caso a LPT1) e o tipo do oscilador desejado "/XT".

O programa irá verificar se todos os parâmetros passados são válidos, e em caso positivo, ele abrirá o arquivo .hex e verificará se há erros neste. Se o arquivo for válido, ele orientará o leitor a ligar o gravador. Nesse momento, o leitor poderá inserir a PIC no soquete e ligar a fonte, nesta ordem. Após isto, o leitor deverá teclar algo e a gravação será realizada. Feita a gravação, o leitor poderá desligar o gravador e retirar seu microcontrolador.

Para conhecer as possibilidades e comandos possíveis do gravador, basta executar o programa sem parâmetros. Ele acusará "erro de parâmetro" e informará todos os parâmetros possíveis (veja **código-fonte 1**)

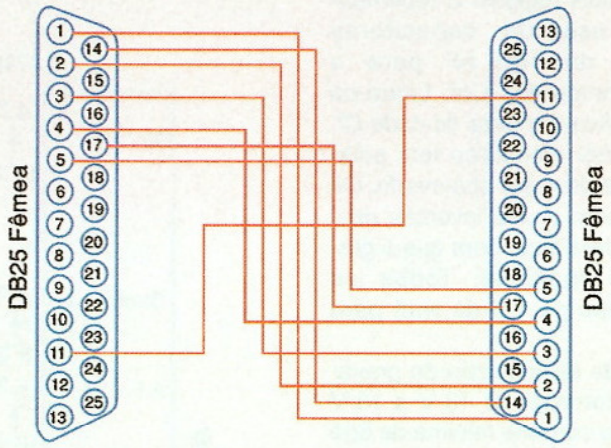
É importante salientar que o nome do arquivo e o parâmetro de gravação ou verificação são necessários e devem obedecer a ordem: "nome do arquivo" - "parâmetro". Todos os outros são opcionais e sua ordem não é imprescindível. O uso destes devem ser feitos de acordo com a operação desejada.

PRIMEIRO PROJETO

Na edição de nº5 Julho/Agosto 2002 Mecatrônica Fácil, demonstramos um circuito que permitia a um LED piscar em ciclos de 1 segundo, aproximadamente. Esse circuito poderá ser utilizado agora pelo leitor para testes. A **figura 8** demonstra novamente o circuito.

Digite o programa LED.ASM em um editor de textos como o "NotPad"

Figura 7 – Esquema do cabo.

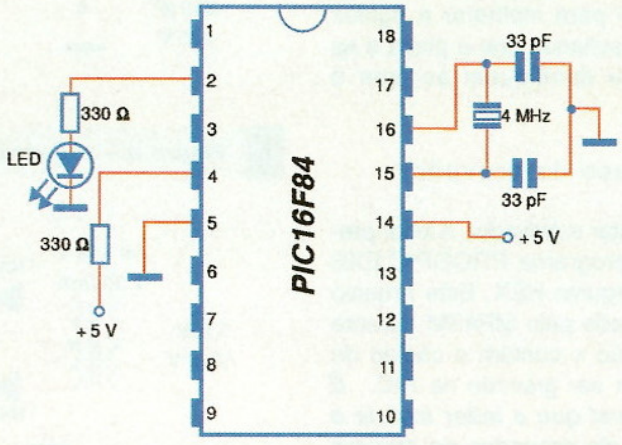


Código-fonte 1:

```
Programa PROGPIC v1.0 - 19/07/2002 - Márcio José Soares
Utilize apenas com o gravador PROGPIC-1 publicado na
Revista Mecatrônica Fácil

Uso : PROGPIC arquivo.hex /comando /porta /oscilador /WDT /
PWRTE /PC
Comandos validos:   P - programa PIC
                    V - verifica PIC
Portas validas: 1 - LPT1, 2 - LPT2..Default = LPT1
Oscilador: XT, HS, RC ou LP.....Default = XT
WDT - ativa WatchDog Timer.....Default = desativado
PWRTE - ativa Power Up Timer.....Default = desativado
PC -  ativa Code Protect.....Default = desativado
```

Figura 8 – Circuito para teste.



do Windows ou o "Edit" do DOS e grave-o com o mesmo nome. Verifique o arquivo digitado e somente então execute o MPASM:**"Mpasm. exe /p16F84 led.asm"**

Caso o arquivo não possua erros, ele será compilado sem problemas e o arquivo LED.HEX com as instruções para o gravador será gerado. Se o arquivo apresentar erros, o compilador indicará a existência destes e o arquivo LED.HEX não será gerado. O arquivo LED.ERR contém a lista de erros encontrados durante a compilação.

O leitor poderá agora gravar o PIC. Execute o programa PROGPIC a partir do "Prompt do MS-DOS" :**"progpic led.hex /p /1 /XT"**

Siga as instruções do programa e aguarde a gravação. Caso o leitor deseje verificar a gravação, execute novamente o programa mas desta vez passe o parâmetro "/V": **"progpic led.hex /v /1 /XT"**

O programa irá comparar o arquivo LED.HEX com a memória interna no PIC. Se esses forem diferentes, o programa mostrará o primeiro endereço onde encontrou a diferença e terminará. Esse comando será muito utilizado quando desejarmos saber qual programa está gravado no PIC.

Após o término da gravação/verificação, o leitor deverá retirar o PIC do gravador e inseri-lo no circuito de testes. Este circuito poderá ser montado em um "pront-o-board" conforme foi demonstrado na **figura 9** ou mesmo em uma placa universal. Verifique o circuito de testes e certifique-se que tudo está correto. Alimente o circuito com 5 volts DC. ***Tome cuidado para não inverter a polaridade da alimentação ou ultrapassar a voltagem recomendada. Isso poderá danificar irremediavelmente o PIC.***

Caso o leitor não possua uma fonte regulada em 5VDC, poderá realizar a alimentação do circuito de testes com o uso de quatro pilhas mais um diodo retificador tipo 1N4001 para reduzir a voltagem. Na **figura 10** temos um esquema que mostra como utilizar pilhas para alimentar o circuito.

Figura 9 – Montagem do circuito teste no "pront-o-board".

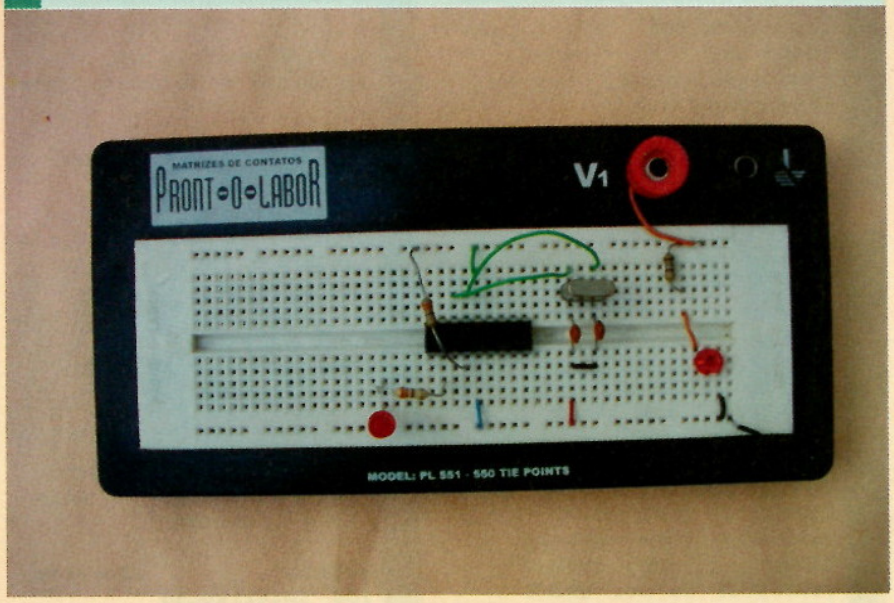
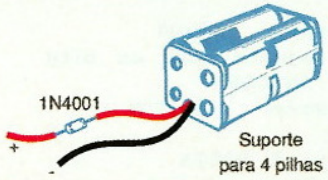


Figura 10 – Uso de pilhas para gerar 5VDC.



Ao ligar o circuito, o LED deverá piscar em intervalos de 1 segundo, comprovando o funcionamento do circuito e do programa.

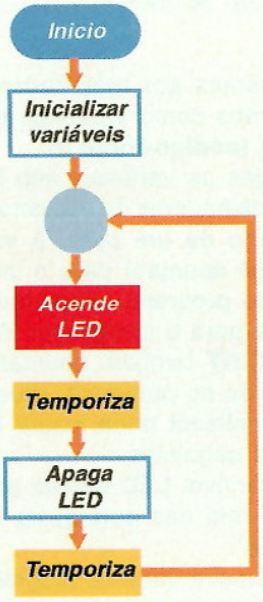
Como funciona o programa

Vamos analisar o programa, e assim levar ao leitor o entendimento sobre o mesmo. Na **figura 11**, podemos observar o fluxograma de operação do programa.

Na primeira parte, temos nossas definições, conforme **código-fonte 2**.

Informamos ao compilador para gerar as listagens para o microcontrolador PIC16F84, determinamos que o padrão para os valores "literais" será decimal e incluímos um arquivo. Esse arquivo traz as definições sobre os endereços de portas, registradores, etc. O leitor mais curioso poderá abrir esse arquivo para conhecê-lo melhor. Use o "EDIT" do DOS ou "NotePad" do Windows. ***Tenha cuidado para não alterar nada!!!***

Figura 11 – Fluxograma do programa.



Na segunda parte de nosso programa temos a inicialização de nossas variáveis. Notem que da-

Código-fonte 2 :

```
list p=P16F84    ;listagem
radix dec        ;padrão-
;>valores decimal se não
;informado outro
;include <P16F84.INC>
;inclui arquivo
```

Código-fonte 3 :

```
;
;variáveis
;

PICRES      equ   0x00
;vetor de reset
PICINT      que   0x04
;vetor de interrupções
CHAR        equ   0x0C
;variável para caracter
TEMP        equ   0x0D
;variável de uso geral
T1          equ   0x0E
;variável para timer
T2          equ   0x0F
;variável para timer
T3          que   0x10
;variável para timer
LED         equ    3
;i/o do led
```

mos nomes aos registradores que usaremos como variáveis no programa (**código-fonte 3**)

Todas as variáveis são ligadas aos registradores de uso geral e têm tamanho de um *byte*. A variável **PICRES** apontará para o início da área de programa, servindo como diretiva para o compilador. A variável **PICINT** também apontará para uma área do programa, informando ao compilador onde inserir as instruções seguintes.

A variável **LED** aponta para um bit de uma das portas, no caso a "PORTA".

A terceira parte do programa define nosso vetor de "reset" e nosso vetor de interrupções. Sempre que ligarmos o PIC, ele apontará para o endereço de "reset". O vetor de interrupção é utilizado pelo PIC para desvio. Sempre que nosso PIC sofrer uma interrupção, ele desviará o processamento para este endereço, retornando após o término das instruções para interrupção com a instrução **retfie** (**código-fonte 4**).

Código-fonte 4 :

```
;
;define memória de
;programa e vetor de reset
;

org       PICRES   ;reset

meureset:
goto      inicio   ;desvia do
;endereço 0x04 -
;interrupção

;
;endereço e rotina de
;interrupção
;

org       PICINT   ;toda
;interrupção aponta para
;este endereço

retfie             ;retorno
;de interrupção
```

Código-fonte 5 :

```
inicio:

movlw     0x00
;ajuste para os bits
;INTCON
movwf     INTCON

clrf PORTA
;limpa port A
clrf PORTB
;limpa port B

bsf       STATUS,RP1
;seleciona banco 1 para
;options e tris

movlw     0x00
;ajusta os bits em A
;como saída
movwf     TRISA

movlw     0x00
;ajusta os bits em B como
saída
movwf     TRISB

bcf       STATUS,RP0
;volta ao banco 0...
;(padrão do reset)
```

Após o endereço de "reset" temos um desvio para nossa rotina principal, evitando que o PIC "entre" no vetor de interrupção. Mais à frente nesta série, será abordado o uso das interrupções no PIC.

Código-fonte 6 :

```
piscaled:

bsf   PORTA,LED
;acende o led
call tempo
bcf   PORTA,LED
;apada o led
call tempo
goto piscaled
;faz infinitamente
```

Após a inicialização dos vetores e variáveis, iniciamos nossas rotinas. No programa- exemplo a rotina principal tem o nome de **início** (**código-fonte 5**)

Nesta rotina são "zerados" os bits de **INITCON**, registro responsável pela ativação ou desativação das interrupções do PIC. Como o programa não usará qualquer interrupção, desativaremos qualquer interrupção. A rotina também limpa as portas de I/O. Em seguida, temos que informar ao PIC como desejamos que os bits de cada porta sejam configurados. No caso, ajustaremos ambas as portas como saída. Para isso, bastará configurar os registradores **TRISA** e **TRISB** com "zeros". Entretanto, esses registradores estão no banco de memória "1" e teremos que mudar de banco para acessá-los. Para tanto utilizaremos o registro **STATUS**. Após carregar os registros **TRISA** e **TRISB**, deveremos retornar ao banco zero.

A rotina **início** continua através da rotina **piscaled**, que será nosso "loop" eterno (**código-fonte 6**).

A rotina liga o bit 3 da porta "A" através da instrução **bsf PORTA,LED** e chama a sub-rotina de temporização. Neste momento, o LED ligado à porta ficará ligado durante 1 segundo. No retorno da sub-rotina o "Program Counter" do PIC contém o endereço posterior à linha de chamada da sub-rotina. Agora, o LED é desligado através da instrução **bcf PORTA, LED**. A sub-rotina de temporização é novamente chamada para mais 1 segundo de espera. No retorno a instrução **goto piscaled** será execu-

tada. Isso causa um desvio no programa que apontará agora para o endereço de memória **piscaled**. O leitor notará que o PIC ficará ascendendo e apagando o LED "eternamente".

A última sub-rotina de nosso programa é a rotina de temporização. Apesar do PIC ter um temporizador interno de precisão, optamos por demonstrar como fazer um temporizador via software (**código-fonte7**).

A rotina inicia as variáveis **T3** com 0x05 hexadecimal (5 decimal). As variáveis **T1** e **T2** recebem **0xff** hexadecimal (255 decimal). Após isso, a rotina decrementará 5 x 768 x 255 vezes. O PIC, operando com um cristal de 4 MHz, tem um ciclo de máquina igual a 1 microssegundo (0,000001 segundo). A instrução **decfsz** consome 1 ciclo de máquina no decremento e 2 ciclos para executar o desvio. A instrução **goto** consome 2 ciclos de máquina. Então, para decrementar **T2,** temos (3 ciclos x 255) + 2 = 767. O programa decrementa **T1** 255 vezes e **T3** 5 vezes. O programa realiza esta operação 255 x 5, portanto, nosso calculo é: **Tempo = 255 x 767 x 0,000001**

Temos aí um tempo de aproximadamente 977 milissegundos (0,97 segundos). Este tipo de temporização não é preciso, pois depende do "clock". Como experiência o leitor poderá mudar o cristal para um novo valor como, por exemplo, 2 MHz. O leitor notará que o tempo dobrará entre um ciclo e outro do LED.

No final do programa, encontramos a instrução **END**, que na verdade é uma diretiva do compilador e informa que o programa finaliza naquele ponto. Esta diretiva é necessária, pois sem ela o compilador acusará o erro.

CONCLUSÃO

Nas próximas edições pretendemos trazer mais informações práticas sobre o microcontrolador PIC16F84, o seu *set* de instruções e um novo projeto ligado à área da Mecatrônica/Automação. Até lá!

Código-fonte 7:

```
tempo:

movlw   0x05      ;carrega
                  ;W com 5
movwf   T3        ;carrega
                  ;T3 com 5

car:

movlw    0xff     ;carrega
                  ;W com
                  ;255
movwf   T1        ;carrega
                  ;T1 com W

decfsz   T3
goto    car_1
retlw    0x00

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
goto     car      ;volta a
                  ;carregar
                  ;T1
```

Lista de componentes

Para o gravador

1 – PIC16F84A-04/P
2 – CIs 74LS32
1 – regulador 78L05
1 - regulador 78L09
1 – transistor BC548 (NPN)
1 – transistor BC558 (PNP)
2 – diodos de sinal 1N4148
1 – LED comum redondo vermelho
1 – LED comum redondo verde
1 – LED comum redondo amarelo
5 – resistores de 1kW - 1/8Ω
2 – resistores de 2k2 - 1/8Ω
2 – resistores de 330W - 1/8W
1 – resistor de 4K7 - 1/8Ω
3 – capacitores 100 nF cerâmicos
1 – capacitor 0,1µF poliéster
1 – capacitor 330 µFx25V eletrolítico

Diversos:
2 – soquetes para CI de 14 pinos
1 – soquete torneado p/ CI de 18 pinos
1 – *plug* P2 fêmea
1 – DB25 macho para inserção em placa
1 – placa universal 8 x 7 cm

fio "wire-wrapping" para ligações, cola quente, solda, etc

Para a fonte

1 – transformador com tensão de primário de acordo com a rede e secundário com 18+18 volts com 300 mA.
2 – diodos 1N4001
1 – Capacitor eletrolítico 1.000 µF X 35 volts
1 – *plug* P2 macho

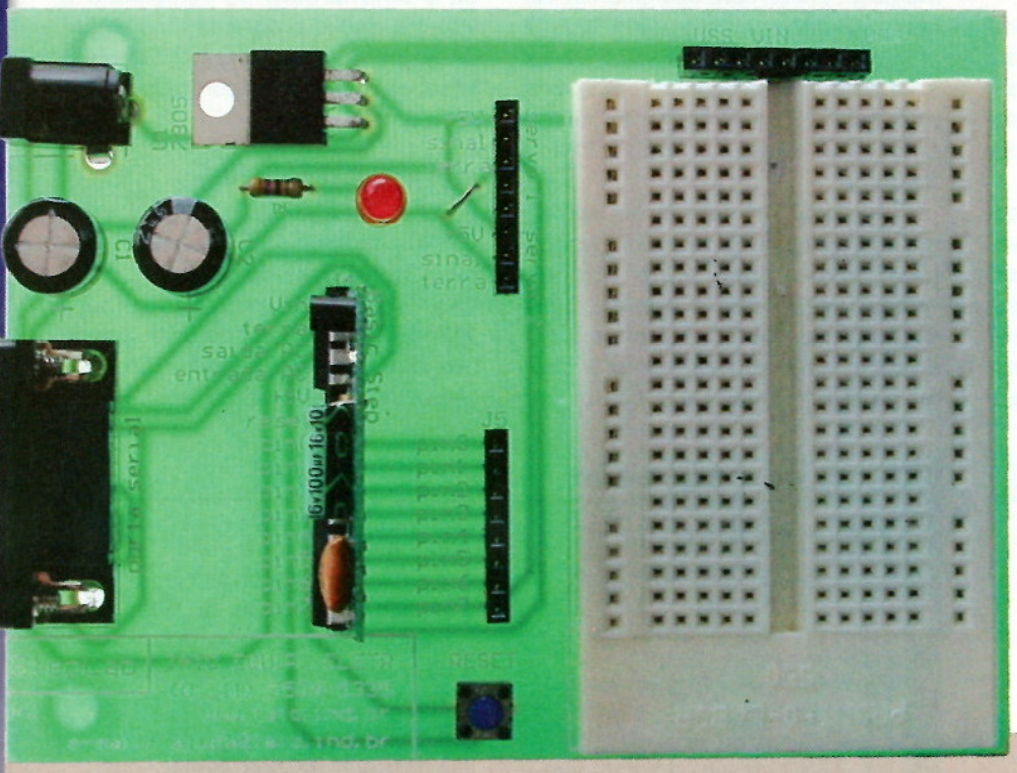
Diversos:
Placa circuito impresso para montagem, fio duplo para ligação do *plug*, caixa em tamanho compatível, etc.

Para o circuito de teste

1 – *Pront-o-board* ou placa universal
1 – cristal 4MHz
2 – capacitores 33 pF cerâmicos
1 – LED comum redondo vermelho
2 – resistores 330Ω - 1/8W

Diversos:
Fios para as ligações, suporte de pilhas, diodo retificador, etc

Automação simples



Basic Step

5ª parte

Octávio Nogueira

No Experimento #3 nós usamos um servo (um tipo especializado de motor) para demonstrar como um microcontrolador pode manipular um dispositivo mecânico no "mundo real". O programa que nós escrevemos (e enviamos para o BASIC Step executar) controlou a rotação e a posição do servo.

O programa fez o servo girar entre duas posições extremas. Isto foi um exemplo de como um microcontrolador pode fazer um dispositivo móvel operar. Entretanto, no Experimento #3 o BASIC Step estava "cego". Tudo que o servo fez foi responder ao nosso código. Lembre-se de que o real poder do microcontrolador está em sua habilidade de tomar decisões baseado em suas "entradas" e, então, manipular o "mundo real" com suas saídas.

Neste experimento nós iremos fazer exatamente isto. Sim, nós iremos mover o servo novamente, mas apenas se as condições apropriadas ocorrerem nas entradas. Você poderia pensar neste experimento como um equivalente em pequena escala da porta de um supermercado. A porta está fechada a maioria do tempo até que alguém (ou alguma coisa) se aproxime, então a porta se abre automaticamente. Aparentemente nós não precisamos fazer nada para a porta se abrir. Nós não estamos apertando nenhum botão, apenas estar próximo da porta é suficiente para fazê-la abrir. Esta é uma forma básica de automação.

Alguns dos sensores que são usados para este tipo de aplicação são bastante sofisticados, e outros muito simples, todavia, todos eles têm algo em comum, eles detectam um evento e geram um sinal para o microcontrolador para que ele pos-

Lista de material:

- **1 -** BASIC Step 1
- **1 -** StepLab
- **1 -** Cabo de Programação
- **1-** LED
- **1-** Resistor de 470 ohms, ¼ watt
- **1-** Resistor de 10k ohms, ¼ watt
- **1-** Foto-resistor
- **3 -** Fios de conexão
- **1 -** Computador PC com Windows 95/98 e uma porta serial livre
- **1 -** Compilador BASIC Step

Figura 16 – As orelhasdevem estar paralelas às paredes do furo a ser medido.



Figura 17 –Cuidados com inclinações ao tomar medidas.

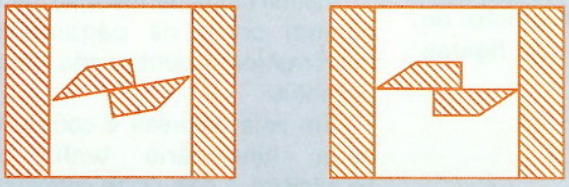


Figura 18 – Orientação para medir corretamente furos

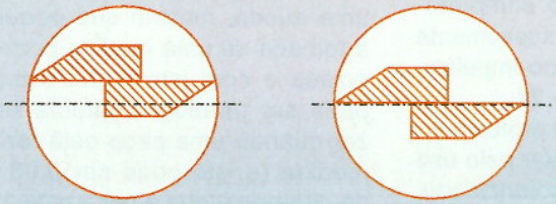


Figura 19 – Medidas de profundidade.

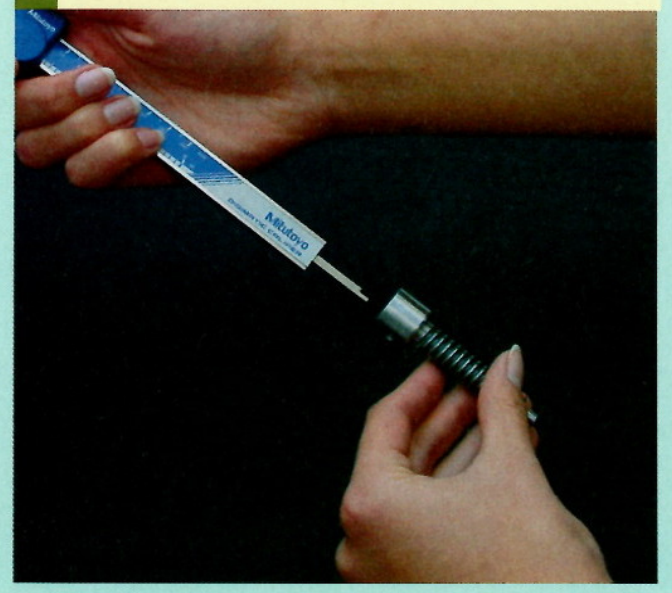


Figura 20 – Medidas de profundidade.



Figura 21 – Orientação para medidas de profundidade



nutenção ou maus cuidados com os instrumentos de medição.

Portanto, cuide de seu paquímetro corretamente, ele é uma ferramenta importante.

Não use para outra finalidade que não seja medir, não permita que outros sem habilidade utilizem seu paquímetro.

Os seguintes cuidados podem prolongar enormemente a vida de seu paquímetro e manter a qualidade de suas medições.

1. Maneje o paquímetro sempre com todo o cuidado, evite choques. Caso o utilize sobre bancadas ou mesas duras de aço ou mesmo ferro fundido, mantenha um tapete de borracha para apoiar seus instrumentos de medição.

2. Em caso de queda, mesmo

Não se preocupe se o servo se mover por um instante, isto ocorre devido a inicialização de seu circuito interno.

Ligue o seu PC e *clique* no ícone do Compilador BASIC Step. Você deve estar agora rodando o programa "Compilador BASIC Step". Este é o programa que nós criamos para ajudá-lo a escrever e transferir os seus programas para o microcontrolador BASIC Step.

Digite o seguinte programa (**código-fonte 1**)

Como você já deve saber, *clique* no ícone e transfira o programa para o BASIC Step.

Apenas mais um programa pisca-pisca? Bem, sim e não. Note que não existe o comando "**output 2**" no programa. Se você ler o arquivo de ajuda, irá descobrir que os comandos "**high**" e "**low**" automaticamente fazem com que o pino seja configurado como saída.

Isto nos economiza alguma digitação, e o mais importante, ele economiza **espaço de programa** no BASIC Step. Nós não precisamos nos preocupar com falta de espaço na memória pois os nossos programas são pequenos, mas à medida que você começar a criar programas mais complexos e maiores, é um bom hábito exercitar as suas habilidades e criar programas de "alta qualidade". Não apenas você estará menos sujeito a ficar sem espaço de memória, mas o seu programa irá executar mais rápido.

Modifique o programa para que ele fique assim (**código-fonte 2**)

Execute o programa e *clique* no ícone.

Você agora terá não apenas o LED piscando, mas também uma "tela informativa" no monitor do seu PC mostrando alternadamente "1" e "0".

Vamos analisar e ver o que está acontecendo.

symbol n = bit0

Declara a variável n, de tamanho 1 bit.

este_lugar

Um rótulo para marcar lugar.

n=0

Algo novo. Nós estamos dizendo que o valor de "n" é zero.

low 2

Faz com que o pino pin2 vá para o nível baixo, acendendo o LED.

debug n

O comando "debug" soa familiar. Você se lembra que "debugar" significa remover os erros do seu programa? Bem, a linguagem TBASIC possui um comando chamado "**debug**", que pode realmente ajudá-lo a encontrar todas as pequenas falhas de seu programa.

Normalmente, nós mandamos o programa do PC através do cabo de programação para o BASIC Step. Isto tem sido (até agora) em apenas um sentido.

Debug é um comando muito especializado que permite ao BASIC Step mandar informações ("dados") de volta pelo cabo e mostrar no monitor do PC. Desta maneira, nós podemos "olhar dentro do BASIC Step" e ver os dados com que ele está trabalhando. Neste caso nós colocamos o valor "0" em "n" no comando anterior. Quando o comando *debug* é encontrado, ele mostra o valor de "n" na janela de debug do PC.

Na verdade, o comando *debug* envia todas as suas variáveis para o PC, então temos a possibilidade de ver as variáveis b0 a b13 e w0 a w7.

pause 1000

Fácil neste ponto, certo?

high 2

Apaga o LED

Código-fonte 3:

```
symbol n = bit0
este_lugar:
n=pin0
debug n
pause 100
goto este_lugar
```

n=1

Aqui nós estamos alterando o valor de "n" para 1.

debug n

Nós estamos mandando o valor de "n" de volta para o PC. Como o valor de "n" mudou, o comando *debug* mostra a mudança na janela do PC.

pause 1000

Sim, nós sabemos o que este comando faz.

goto este_lugar

Vá e repita tudo novamente.

Agora alguns de vocês podem estar pensando "Por que nós precisamos ver o valor de "n" em nosso PC? Como o nosso programa está *setando* o valor de "n", nós já sabemos qual é o seu valor. Por que fazê-lo mostrar no PC?"

É verdade, o valor de "n" está **codificado** dentro do programa. Não há maneira de variar o seu valor. Porém, quando a variável é *setada* por eventos externos, por exemplo, quando uma entrada muda devido a um botão ser pressionado, o comando *debug* nos permite ver a mudança no dado de entrada e então ver se o nosso pro-

Código-fonte 4:

```
symbol x = w1
symbol n = bit0
output 6
feche_a_porta:
for x = 1 to 100
pulsout 6,100
pause 10
next
pause 10
veja_pessoas:
n=pin0
if n = 1 then abra_a_porta
pause 100
goto veja_pessoas
abra_a_porta:
for x = 1 to 100
pulsout 6,200
pause 10
next
pause 10
n=pin0
if n = 0 then
feche_a_porta
goto abra_a_porta
```

grama está reagindo corretamente ou não.

Vamos tentar. Modifique o programa (**código-fonte 3**).

Agora ao invés do valor de "n" estar codificado no programa, nós estamos fazendo "n" ser igual ao valor que estiver em "pin0". Como pin0 está ligado ao foto-resistor, quando a quantidade de luz que chega a ele muda, a tensão de entrada em pin0 também muda.

Mova a sua mão sobre o foto-resistor (não é necessário tocar, apesar de não danificá-lo). O valor no monitor de seu PC deve mudar entre "0" e "1" dependendo de quanta luz o foto-resistor estiver recebendo. Certifique-se de que as luzes estão acesas na sala.

Esta parte do circuito é chamada "divisor resistivo" e nós o iremos explorar em detalhes no próximo experimento. É suficiente dizer neste ponto que, à medida que o foto-resistor muda sua resistência (devido a variação de luminosidade), a tensão varia em pin0. Esta variação de tensão é um sinal analógico e como o pino de entrada do microcontrolador somente reconhece valores binários (digitais), quando a tensão atingir certo ponto, o valor que pin0 vê pode ser "1" (+5 volts) ou "0" (0 volt).

O que nós acabamos de fazer foi construir um interruptor que não precisa ser "apertado"! É um sensor que reage automaticamente ao nível de iluminação externo.

Vamos fazer algo divertido agora ...

Modifique o programa para o **código-fonte 4**.

O que você acha que o programa irá fazer?

Não se sinta desencorajado pelo fato do programa estar ficando um pouco longo. Apenas vamos quebrá-lo em pedaços menores e fáceis de entender.

Esta é a inicialização do programa, nós já sabemos o que estas três instruções fazem (**código-fonte 5**).

Nós usamos esta rotina no Experimento #3. Ela manda uma série de pulsos que fazem com que o servo se mova para uma de suas posições extremas (**código-fonte 6**).

Código-fonte 5:

```
symbol  x  =  w1
symbol  n  =  bit0
output  6
```

Código-fonte 6:

```
feche_a_porta:
for  x  =  1  to  100
pulsout  6,100
pause  10
next
pause  200
```

Código-fonte 7:

```
veja_pessoas:
n=pin0
if  n=1  then  abra_a_porta
pause  100
goto  veja_pessoas
```

Esta parte do programa simplesmente verifica se o foto-resistor detectou uma "sombra". Se sim, ele faz com que o programa vá para a rotina que irá "abrir a porta" (**código-fonte 7**)

Se o nosso servo fosse grande o suficiente e conectado à porta da frente do supermercado, esta rotina iria abri-la.

Isto é para segurança. Enquanto as pessoas estiverem paradas perto da porta, mantenha a porta aberta. (Você não deseja esmagar muitos consumidores, eles iriam começar a fazer suas compras em outro lugar!). Veja **código-fonte 9**.

Código-fonte 8:

```
abra_a_porta:
for  x  =  1  to  100
pulsout  6,200
pause  10
next
pause  200
```

Código-fonte 9:

```
n=pin0
if  n=0  then  feche_a_porta
goto  abra_a_porta
```

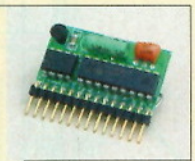
Saber como "automatizar" certas tarefas podem reduzir o risco de muitos tipos de trabalho. A automação na indústria automobilística melhorou significativamente muitas fases do processo de montagem. Pintura e soldagem, por exemplo, são feitos agora por robôs automatizados, dando uma melhor qualidade a um custo menor e com menos riscos para os trabalhadores.

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

Linguagem LOGO para Robótica e Automação – 7ª parte

Registrando eventos com o LOGO

Luiz Henrique Corrêa Bernardes

Em sistemas de automação e robotizados, é muito importante o registro de eventos para análise ou estatística . Neste artigo, vamos explorar um pouco esse assunto mostrando como fazer registros de eventos utilizando o LOGO.

Muito conhecido como "LOG" (pronuncia-se "logue"), são registros de eventos que ficam armazenados em memória não volátil (Disco Rigido, disquete, memória Flash, papel, entre outros) . Nos sistemas de automação ou robotizados, esses registros de eventos indicam ciclos dos equipamentos, números de peças pro-

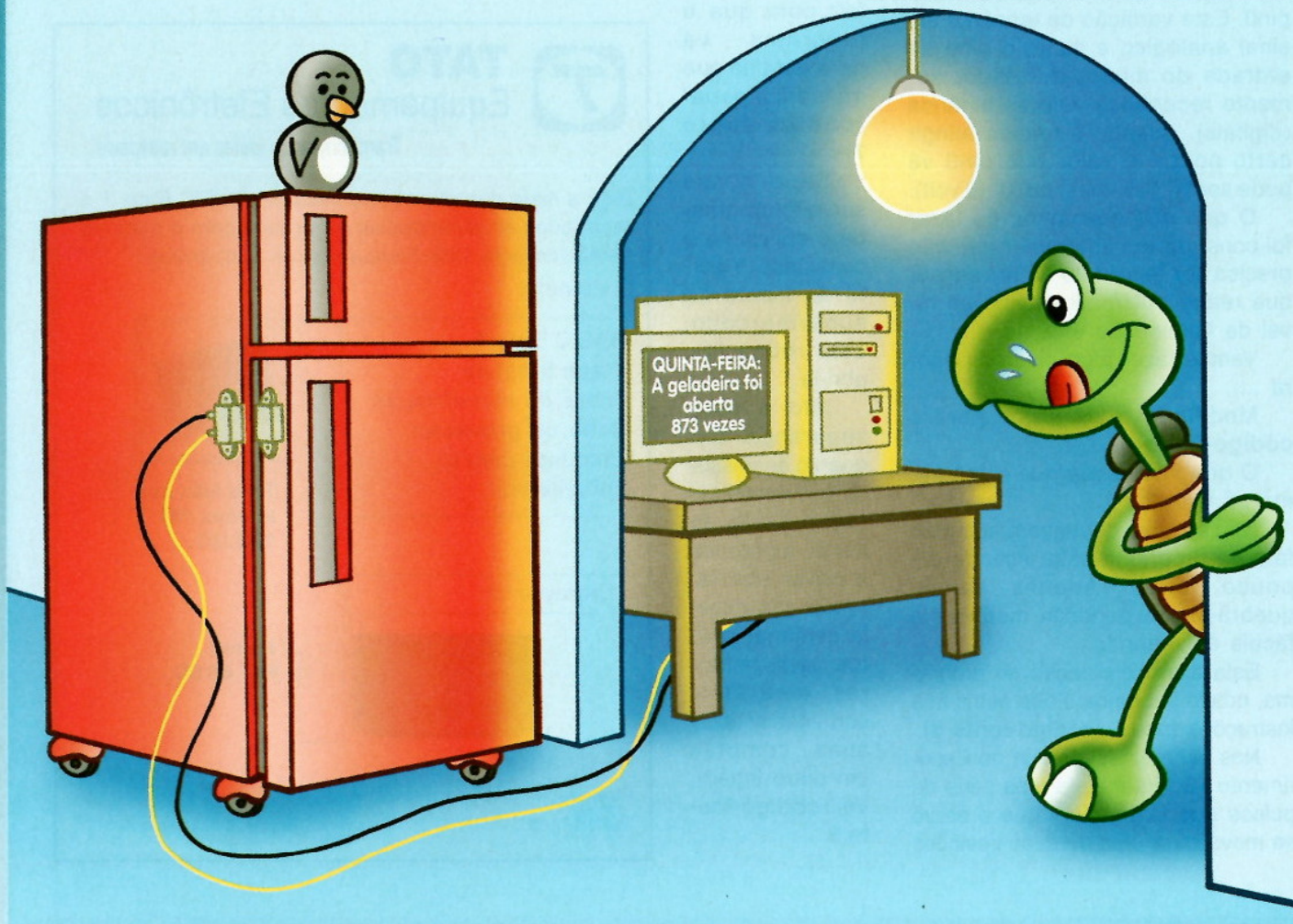


Figura 1 – Hidrômetro.



duzidas, horas de funcionamento, horário de eventos de anomalias tais como falhas e quebras, entre outros.

Tendo esses registros, podemos fazer uma análise para verificar se tudo está ocorrendo corretamente ou em que freqüência estão surgindo as falhas.

Podemos ainda fazer uma análise estatística de um processo produtivo, vendo o número de peças produzidas e as horas de funcionamento do equipamento.

Para fazer uma analogia bem simples com o que estamos falando, podemos usar o hidrômetro de nossa residência. É aquele aparelho que fica na entrada de água de nossa residência (**figura 1**) e que registra o consumo de água. Ele vai acumulando o consumo e mostrando o número cor-

Código-fonte 1

```
aprenda log
abraparaadicionar "c:\\log.txt
mudeescrita "c:\\log.txt
escreva diahora
mudeescrita [ ]
fechearq "c:\\log.txt
escreva [registro adicionado]
fim
```

Figura 2 – Resultado obtido no bloco de notas.

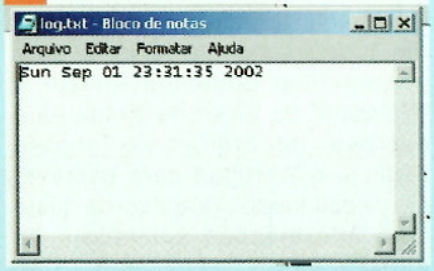


Figura 3 – 2º resultado obtido.

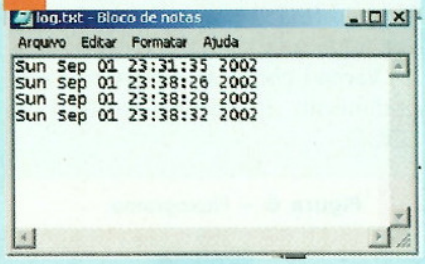
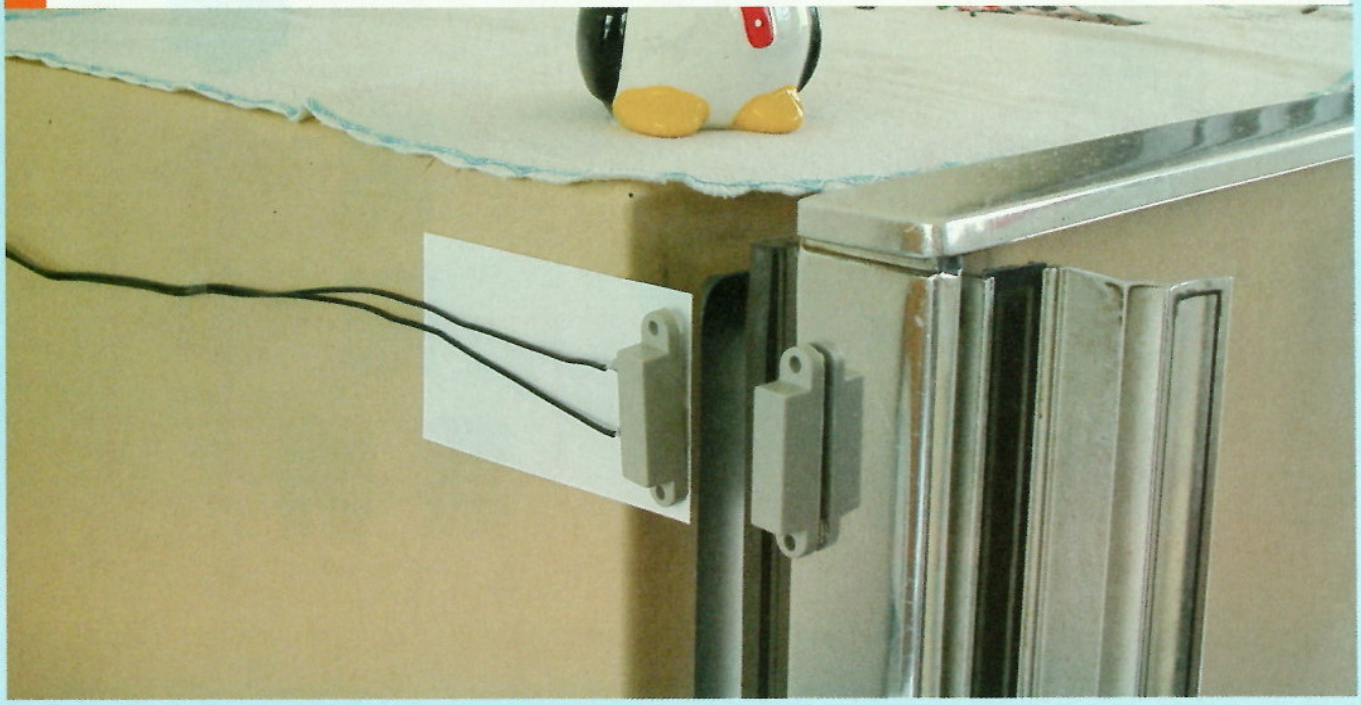


Figura 4 – Sensor magnético com reed-relé.



disco rígido (*Hard Disk* ou simplesmente HD) em arquivos que, freqüentemente, são no formato texto (ou conhecidos como **txt**). Atualmente, quando nos referimos ao arquivo texto, sempre associamos a um arquivo gerado por um processador de texto (Word ou Wordpad, da Microsoft, por exemplo). Mas, esses processadores de texto geram um formato especial incorporando dados (fonte de caracteres, formato do papel, ilustrações, etc.). Devemos salientar que um arquivo texto é diferente de um arquivo feito por um processador de texto, e para exemplificar podemos utilizar o "Notepad" ou Bloco de Notas para escrever um arquivo no formato texto, e o Wordpad para escrever um "documento" (arquivo de texto com informações adicionais de formatação) . Fizemos questão de salientar essa diferença porque no LOGO iremos gerar arquivos no formato texto para registrar os nossos eventos.

Vamos começar editando o procedimento do **código-fonte 1** no LOGO:

Antes de executar, vamos analisar alguns comandos novos e ver como funciona o nosso procedimento.

A linha **abraparaadicionar "c:\\log.txt** irá abrir o arquivo "log.txt" que está no diretório c: . Caso o arquivo não exista, ele será criado. Note que devemos utilizar duas barras invertidas (\\) para indicar o que seria somente uma barra no *prompt* do DOS.

A linha **mudeescrita "c:\\log.txt** direciona o comando "escreva" do terminal (janela do LOGO) para o arquivo que está aberto (no caso **log.txt**)

O comando **diahora** retorna o dia e a hora corrente do sistema no formato de lista. Por exemplo, na linha de comando execute "mostre diahora" e você verá que o sistema retorna algo parecido com [Sun Sep 01 23:19:31 2002] (nesse caso o sistema está ajustado para o formato em inglês)

Então, a linha **escreva diahora** irá escrever a lista de retorno do comando **diahora** no final do arquivo log.txt .A linha **mudeescrita [ ]** direciona novamente o comando "escreva" para o terminal do LOGO.

Figura 5 – Esquema elétrico de ligação do sensor.

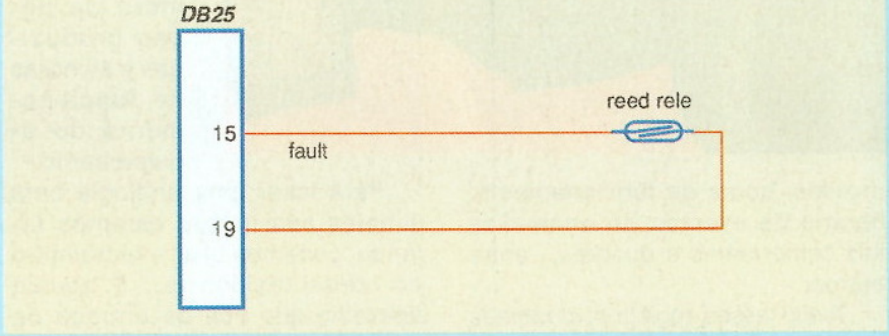
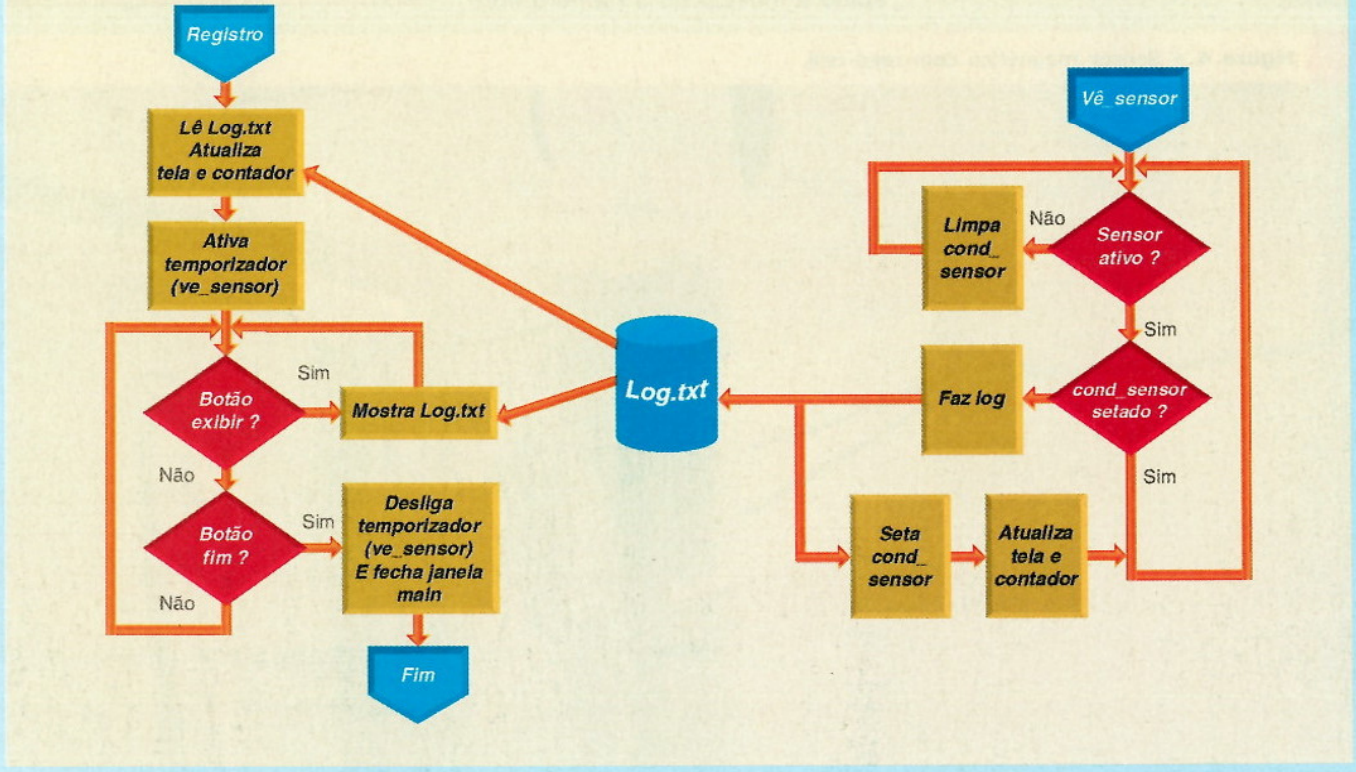


Figura 6 – Fluxograma.



Código-fonte 2

```
aprenda registro
atribua "cond_sensor ~
"nãoativo
atribua "cont 0
criejanela "main "d1 ~
[Registro] 10 10 200 130
crieestático "d1 "e1 ~
[Horario Atual:] ~
10 15 50 20
crieestático "d1 "e2 [] ~
70 15 100 20
crieestático "d1 "e3 ~
[Ultimo Registro: ]~
10 45 70 20
crieestático "d1 "e4 [] ~
70 45 85 20
crieestático "d1 "e5 ~
[Registros: ] 10 70 60 15
crieestático "d1 "e6 [] ~
70 70 50 15
criebotão "d1 "b1 "Fim ~
10 90 40 20 ~
[apaguejanela "d1 ~
desativetemporizador 1 ]
criebotão "d1 "b2 ~
"Exibir 80 90 40 20 ~
[ Exibir ]
atribua "i 0
abraparaler "c:\\log.txt
mudeleitura "c:\\log.txt
enquanto [:i = 0] [ atribua ~
"hora leialista ~
senão évazia :hora [atribua ~
"i 1] [mudeestático "e4 ~
:hora muda_cont ]]
mudeleitura []
fechearq "c:\\log.txt
ativetemporizador 1 200
[ve_sensor]
fim

aprenda Exibir
mostre shell [notepad ~
c:\\log.txt]
fim

aprenda log
abraparaadicionar ~
"c:\\log.txt
mudeescrita "c:\\log.txt
atribua "ultimo_log diahora
escreva :ultimo_log
mudeestático "e4 :ultimo_log
mudeescrita [ ]
escreva [registro adicionado]
fechearq "c:\\log.txt
muda_cont
fim

aprenda muda_cont
atribua "cont :cont+1
mudeestático "e6 :cont
fim

aprenda ve_sensor
mudeestático "e2 diahora
desativetemporizador 1
atribua "sensor ~
portaentradab 889
atribua "sensor bite :chave 8
senão :sensor = 1 ~
[se sãoiguais :cond_sensor ~
"nãoativo ~
[Log atribua "cond_sensor ~
"ativo ] ] ~
[atribua "cond_sensor ~
"nãoativo ]
ativetemporizador 1 200 ~
[ve_sensor]
fim
```

A linha **fechearq "c:\\log.txt** vai fechar o arquivo "log.txt" e, finalmente, a linha **escreva [registro adicionado]** irá escrever no terminal do LOGO a frase " registro adicionado" .

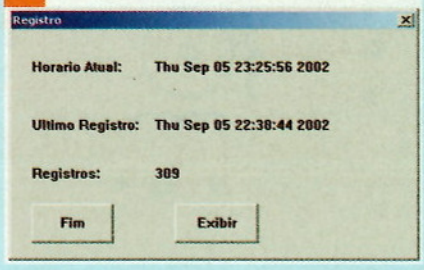
Agora que entendemos todos os novos comandos, vamos analisar o resultado da execução do procedimento "log". Primeiramente, execute o procedimento teclando "log" na janela de comandos do LOGO e pressionando o botão de executa (ou teclando " entra ").

Você verá no terminal do LOGO que ele irá escrever " registro adicionado" .

Depois, abra o programa "Notepad" ou Bloco de Notas do Windows (geralmente ele está em programas, acessórios). A seguir, abra o arquivo "c:\log.txt" e veja um resultado parecido com a da **figura 2**.

Feche o "Notepad" ou Bloco de Notas e execute novamente o procedimento "log" mais três vezes, e veja o resultado abrindo o arquivo "log.txt" com o Notepad ou Bloco de Notas, que deve ser algo parecido com a **figura 3**. Observe que já tínhamos uma linha de dia e hora e foram adicionadas mais 3 linhas.

Figura 7 – Tela do Registro.



Com isso, já temos um procedimento que atualiza o arquivo de registro de eventos. Agora vamos fazer algo prático utilizando os conceitos que aprendemos . Que tal utilizar algo que temos em casa? A proposta é fazer um registro de quantas vezes e em que hora foi aberta a porta de uma geladeira.

Basicamente, vamos colocar um sensor magnético com reed-relé entre o corpo da geladeira e a sua porta (**figura 4**)

Após isso, ligamos o sensor conforme o esquema elétrico da **figura 5**.

Agora, vamos fazer um programa no LOGO que siga a seqüência mostrada no fluxograma da **figura 6**.

Assim, podemos digitar o programa vendo o fluxograma para entender os procedimentos que estamos digitando.

Executando o procedimento "REGISTRO", iremos obter uma tela parecida com a da **figura 7**:

Nela temos o horário atual, o horário do último registro e quantos registros temos no nosso arquivo . A tecla "Exibir" abre o "Notepad" ou Bloco de Notas .

Agora que está tudo funcionando, basta deixar o nosso registrador de eventos armazenar os horários dos ataques noturnos à geladeira !

CONCLUSÃO

Mostramos como manusear arquivos e o modo de utilizá-los para armazenar dados. Fizemos aqui uma sugestão bem simples de registro de eventos. O leitor pode aprimorar e fazer algo mais sofisticado como um controle de uma produção, passagem de veículos , passagem de pessoas, etc. Basta utilizar a imaginação e por mãos a obra ! Bons projetos e até a próxima. ●

Eletrônica básica para Mecatrônica - 7ª parte

Motores de passo

Newton C. Braga

Na parte anterior desta série estudamos os controles lineares e PWMs para motores de corrente contínua comuns. No entanto, em Robótica, Mecatrônica e outras aplicações que envolvem o controle de movimento, encontramos um outro tipo de motor que, pelas suas características, é o ideal para os casos onde a precisão é um requisito importante do projeto. Trata-se do motor de passo ou *stepper motor*, que não só é usado para transmitir movimento, mas também para posicionar de forma precisa as partes de um equipamento. Como funcionam esses motores e como usá-los em projetos de Mecatrônica é o assunto deste capítulo.

O princípio básico de funcionamento de um motor de passo não difere muito de outros tipos de motores: conjuntos de bobinas criam campos magnéticos pela passagem de uma corrente elétrica, e esses campos interagem estabelecendo forças que movimentam as partes móveis do motor.

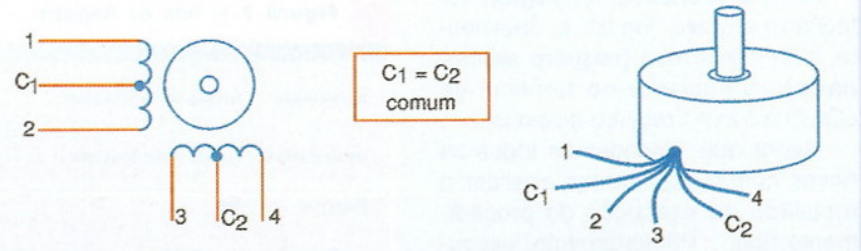
A diferença básica entre um motor de passo e um motor comum está no modo como suas bobinas são energizadas. Isso ocorre porque os motores não devem apenas girar, mas também ter seus eixos posicionados de forma precisa, conforme a aplicação a que se destinam.

Todavia, para que o leitor entenda melhor como esses motores podem ser usados e controlados, será interessante fazermos uma análise de seu princípio de funcionamento.

COMO FUNCIONAM

Os motores de passo convertem informação digital em movimento ou posição. Eles são diferentes dos motores comuns nesse ponto porque, enquanto os motores comuns são controlados pela corrente que atravessa seus enrolamentos, os motores de passo são controlados digitalmente.

Figura 1 - Motor de passo de 4 fases.



Existem três tipos básicos de motores de passo:

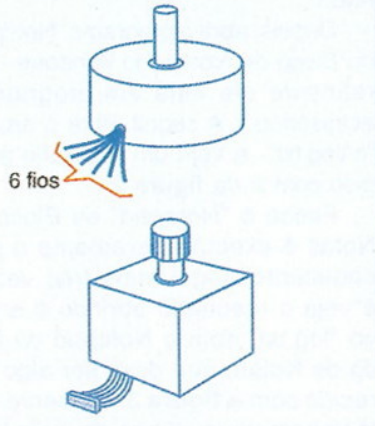
- Ímã permanente
- Relutância variável
- Híbrido.

O modo como as suas bobinas são montadas internamente e sua quantidade, determinam a sua maneira de funcionar.

O tipo mais comum de motor de passo é o de 4 fases, se bem que também possamos encontrar em algumas aplicações tipos de 2 e de 6 fases.

Na **figura 1** damos uma representação simplificada do motor de passo de 4 fases, o qual servirá como

Figura 2 - Tipos de motores de passo de 4 fases.



ponto de partida para nossas explicações.

Esse motor possui 4 bobinas, ligadas duas a duas com um terminal comum. Isso faz com que esse tipo de motor seja facilmente identificado, pois possui 6 fios de conexão, conforme mostra a **figura 2**.

Na operação normal, os fios comuns aos enrolamentos são conectados ao pólo positivo de uma fonte de alimentação. Essas fontes, para a maioria dos casos, são de 12 V e as correntes exigidas pelos enrolamentos variam tipicamente entre 100 e 500 mA conforme o tamanho do motor.

Cada enrolamento é energizado quando a extremidade livre é aterrada por um momento. Cada vez que um enrolamento é energizado, o eixo avança um passo ou uma fração da volta completa.

Se energizarmos os enrolamentos em seqüência, de acordo com a **figura 3**, o eixo irá se movimentar rodando no sentido correspondente.

Uma outra forma de obtermos movimento rotativo do eixo é energizando os enrolamentos pelo sistema ON/OFF, conforme ilustra a **figura 4**.

Nesta forma de energização temos a aplicação combinada de sinais nas bobinas de tal forma a ligar e desligar cada bobina em tempos certos.

Ao usar um motor de passo é importante observar qual tipo de seqüência é a recomendada pelo fabricante para sua operação normal.

CARACTERÍSTICAS

Para usar um motor de passo precisamos conhecer suas características. As principais são:

a) Tensão e Corrente

Embora os motores mais comuns sejam os de 12 V, temos também tipos de 5 V. As correntes indicadas são as que circulam pelas bobinas quando a tensão nominal é aplicada. Podemos, em função da tensão e da corrente, calcular facilmente a resistência do enrolamento aplicando a Lei de Ohm.

Figura 3 - Seqüência de pulsos para um motor de 4 fases.

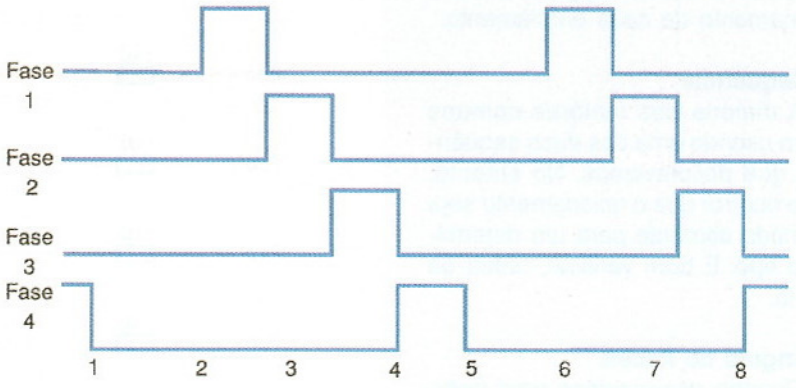


Figura 4 - Acionamento ON-OFF.

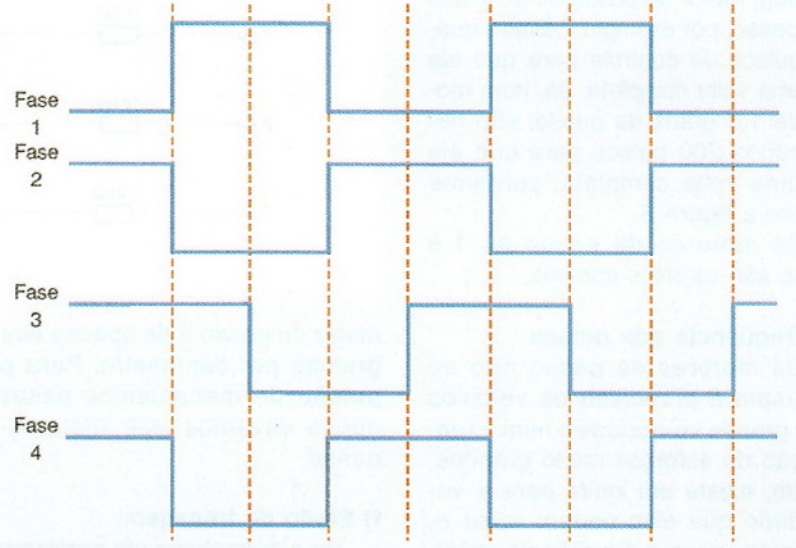
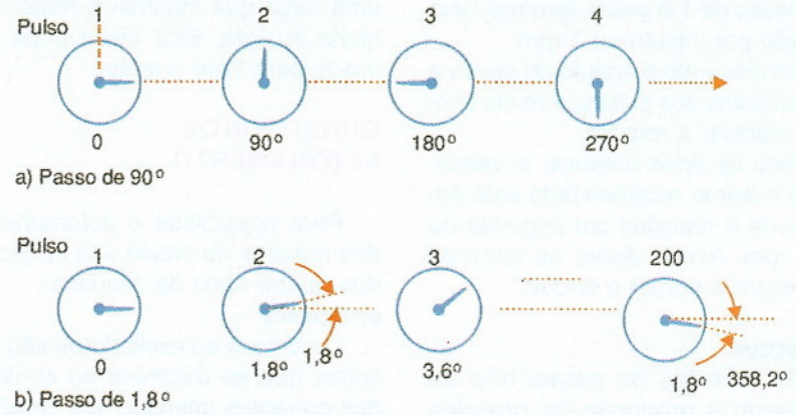


Figura 5 - O ângulo de passo.



Elevador controlado pelo LOGO

Olá! Meu nome é Peter, gostaria de esclarecer algumas dúvidas que tive em relação ao elevador da revista número 6 ("Mecatrônica Fácil"). Eu não encontrei exatamente o reed-relé que está na revista, e gostaria de saber se posso utilizar outro (Ampola REED, Tensão de comutação 200 V, 500 mA, resistência residual do contacto 150:mW). Outra dúvida é sobre o relé mencionado, vocês podem me informar, com mais detalhes, que tipo de relé que foi usado. E, por último, gostaria de saber se existe algum lugar aonde eu posso encontrar com facilidade esses pequenos motores utilizados em robótica. Agradeço desde já pela sua atenção!



Caro Peter,
*Você pode utilizar outro reed-relé no seu projeto. Faça o teste com um ímã e veja se funciona. Depois disso, é só colar na haste do elevador! Quanto ao relé, utilizamos um de 6 volts na bobina para ser alimentado com bateria (4 pilhas). A Saber Marketing (**www.sabermarketing.com.br**) dispõe para venda a caixa de redução utilizada no elevador.*

Novos Microcontroladores

Adquiri a revista "Mecatrônica Fácil" número 6, e gostei bastante dos artigos e da forma como os assuntos são abordados. Estou desenvolvendo algumas aplicações com os microcontroladores PIC, e gostaria de sugerir que nas próximas edições, vocês passassem a abordar os modelos 16f627 e 16f628, pois eles foram desenvolvidos pela Microchip para substituir o 16f84, que não está sendo produzido mais e logo estará em falta no mercado, dificultando aos leitores que não têm essa informação, na aquisição dos microcontroladores, bem como não lhes permitindo que utilizem todos os recursos a mais que a nova linha de PICs está oferecendo.
Atenciosamente,
Marcos

Caro Marcos,
Os microcontroladores 16F62X não são substitutos do 16F84, constituindo sim um aprimoramento. A Microchip garante o fornecimento de seus microcontroladores (o antigo 16C54C é encontrado muito facilmente em lojas e distribuidores) e atualmente nas lojas especializadas o mais comum ainda é o 16F84.
A série 16F8X é igual em comandos à 16F62X. A diferença está no número de comandos e registradores da versão mais nova, que é maior. Portanto, todos os que aprenderem a programar com um PIC16F84 estarão aptos a desenvolverem aplicações com qualquer microcontrolador Microchip. Porém, no final da série o autor escreverá sobre os novos modelos, não somente da série 16F62X como também da série 16F87X maior e mais poderosa, promovendo uma migração mais fácil.

Lista de Material do ADR-1

Caro Senhores da Editora Saber,

Meu nome é Diego Santana, e sou assinante da revista "Mecatrônica Fácil". Gosto muito disso, e por esse e outros motivos, eu e meus amigos resolvemos apresentar o robô ADR-1 na Feira de Ciências 2002.
Estamos construindo-o para a feira, que se realizará em 2 meses, em nosso colégio (Colégio Cenecista Presidente Kennedy-PR), porém encontramos algumas dificuldades sobre quais são os componentes usados para a construção?

Procuramos na revista ("Mecatrônica Fácil" nº 5, Julho - Agosto de 2002), mas percebemos que para esse robô não veio acompanhado impresso a lista de componentes. Solicitamos, se possível, a lista de componentes para o ADR-1 urgentemente! Atenciosamente,
Diego Santana

Caro Diego,
Perdoe-nos a falha. A lista de componentes do ADR-1 está disponível no endereço:
www.mecatronicafacil.com.br\downloads.
Esperamos que o seu projeto seja um sucesso.

Figura 7 - Acionamento com transistores bipolares.

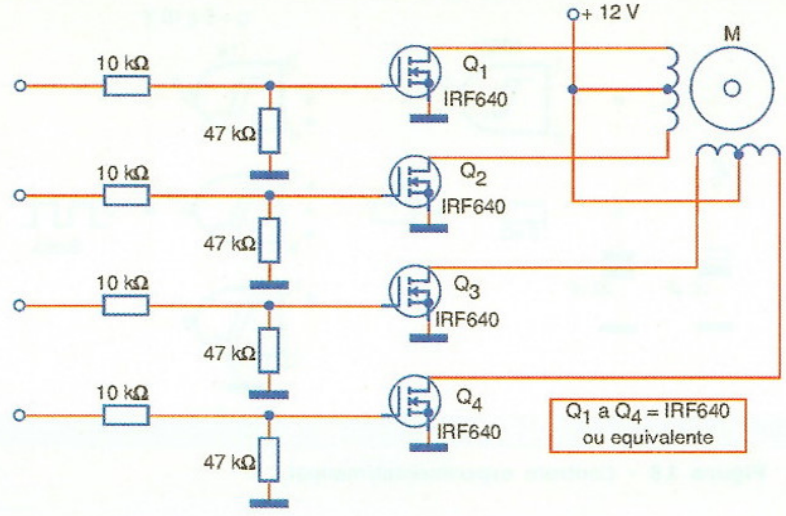
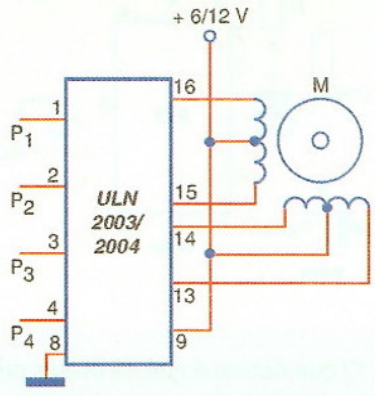


Figura 8 - Controle com CI.

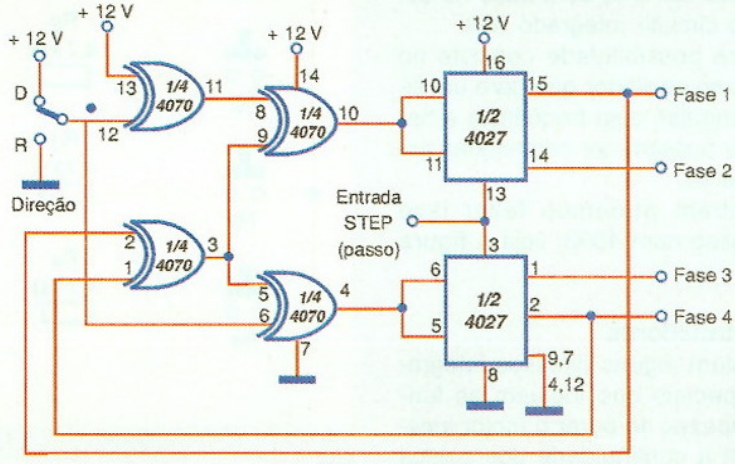


gica digital, principalmente a encontrada nas saídas das portas paralelas dos computadores (PCs).

Isso significa que este circuito pode ser utilizado para controlar o movimento de um motor de passo diretamente a partir de um programa que rode num PC, usando a porta paralela.

O circuito integrado indicado pode ser aplicado para controlar motores de passo de até 500 mA por bobina.

Figura 9 - Seqüenciador CMOS.



b) Seqüenciadores ou *translators*

Esse tipo de circuito tem por finalidade seqüenciar os pulsos na forma exigida pelo motor para que ele rode num sentido ou noutro.

Se o motor for controlado por um PC ou por um microprocessador, esse seqüenciamento pode ser feito pelo próprio programa instalado. No entanto, se assim não ocorrer, precisaremos de um circuito que faça o seqüenciamento, por exemplo, a partir de pulsos gerados por um sensor, uma chave ou um oscilador.

Temos diversas possibilidades para isso. A primeira consiste em um seqüenciador com circuitos digitais comuns, como o mostrado na **figura 9**.

Os pulsos aplicados à entrada desse circuito se convertem na seqüência necessária ao movimento do motor em quatro saídas.

Outra possibilidade é a mostrada na **figura 10**, em que temos um circuito dedicado que reúne todos os elementos para acionar um motor diretamente a partir de pulsos aplicados seqüencialmente em sua entrada.

c) Circuitos de passo

Os circuitos de passo têm por finalidade produzir a seqüência de pulsos de acionamento na velocidade que se deseja fazer o motor girar ou mover.

O tipo mais simples de circuito de passo é o que consiste num simples interruptor, mas como sempre existe o problema do tempo e o repique, alguns refinamentos devem ser acrescentados.

Figura 10 - Controle com CI dedicado.

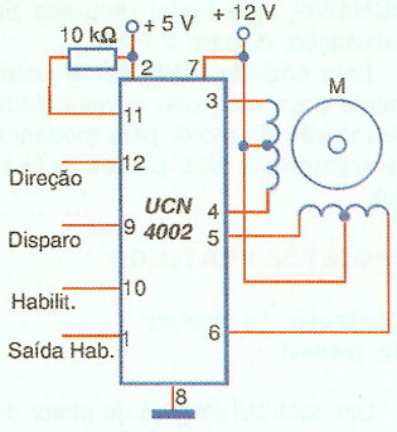


Figura 11 - Gerador de passo.

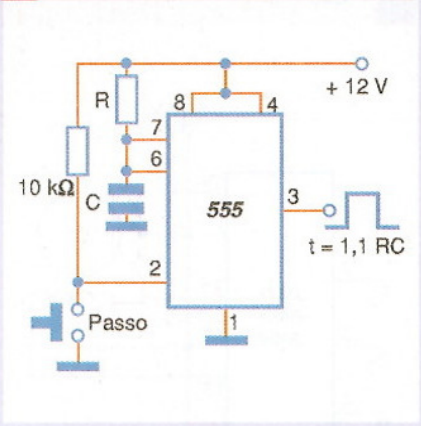
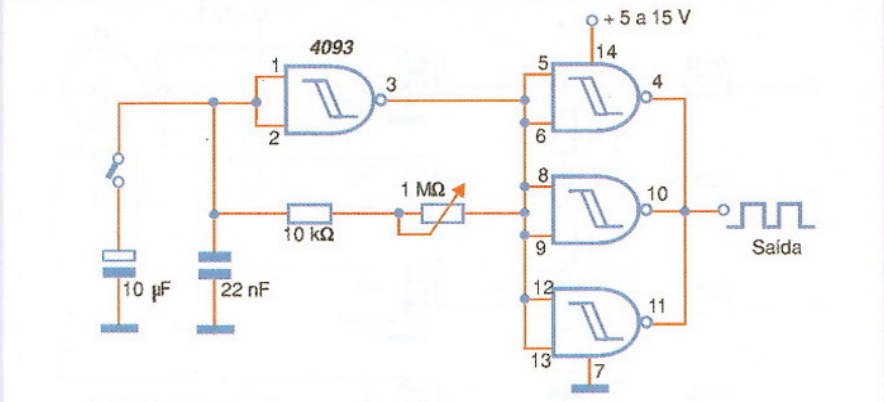


Figura 12 - Outro gerador de passo.



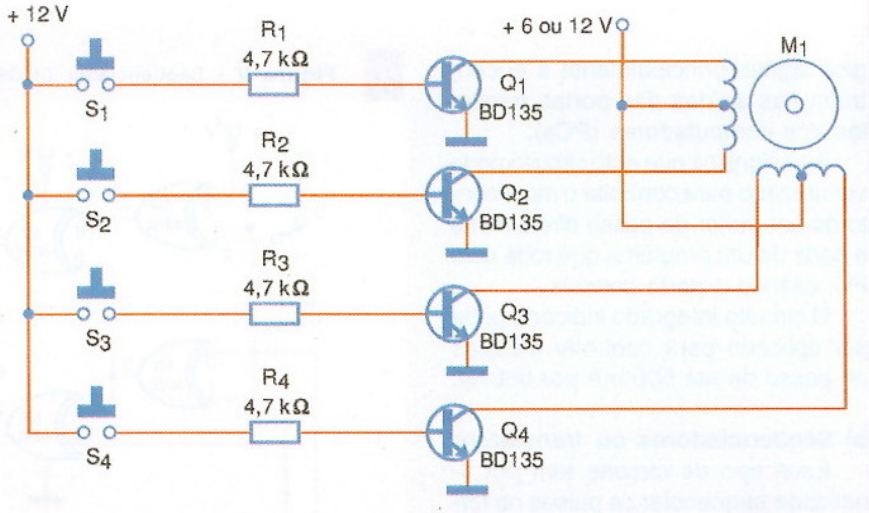
O que ocorre é que os pulsos precisam ter duração constante para um correto acionamento, exigindo-se, por exemplo, um circuito monoestável para essa finalidade.

Na **figura 11** mostramos como isso pode ser feito com base no conhecido circuito integrado 555.

Outra possibilidade consiste no uso de um oscilador que gere um sinal retangular, cuja freqüência e habilitação possam ser controladas externamente.

Também podemos fazer isso como base num 4093; veja a **figura 12**.

Figura 13 - Controle experimental/manual.



d) Controladores

Existem alguns circuitos integrados especiais que incluem as funções capazes de parar o motor mesmo com a continuidade dos pulsos de entrada, inverter sua rotação ou ainda produzir a frenagem forçada com a energização constante de um enrolamento.

Um desses circuitos é o UCN4202, que inclui recursos de habilitação, disparo e PWM.

Esta entrada PWM serve justamente para controlar a velocidade de rotação do motor pela mudança da freqüência dos pulsos aplicados.

PROJETOS PRÁTICOS

Controle de motor de passo

Um controle simples de motor de passo pode ser elaborado com poucos componentes e controlado por chaves de pressão ou pela porta paralela do computador.

Na **figura 13** damos a versão controlada pelos interruptores. Acionando-os em seqüência, o motor vai girar no sentido escolhido, servindo para que o leitor entenda seu princípio de funcionamento.

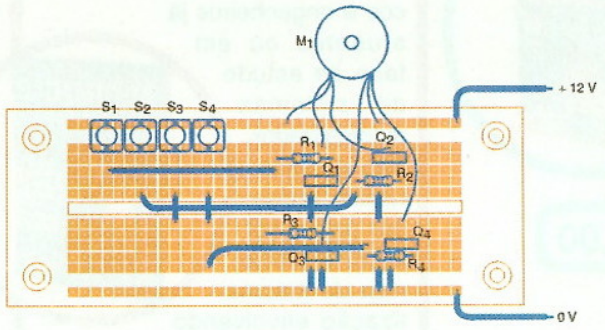
Esta versão pode ser montada numa matriz de contactos, conforme mostra a **figura 14**.

Se o motor não exigir mais do que uns 200 mA por enrolamento, não será necessário dotar os transistores de dissipadores de calor. No entanto, para motores maiores, esses radiadores deverão ser agregados. Eles consistem em pequenas placas

Lista de Material 1

Q_1 a Q_4 - BD135 ou BD137 - transistores NPN de média potência.
R_1 a R_4 - 4,7 k ohms x 1/8 W - resistores - amarelo, violeta, vermelho.
M_1 - Motor de passo de 4 fases
S_1 a S_4 - Interruptores miniatura para placa de circuito impresso, tipo normalmente aberto (NA).
Diversos: matriz de contatos, fios, fonte de alimentação conforme o motor, fios, solda, etc.

Figura 14 - Montagem na matriz de contatos ou placa universal.



de metal parafusadas nos invólucros dos componentes.

A alimentação do circuito deve ser de acordo com o motor, ou seja, normalmente uma fonte de 12 V.

A seqüência de pulsos normais é a que analisamos neste artigo, mas dependendo do motor ela pode ser

Figura 15 - Diagrama da versão controlada pelo PC.

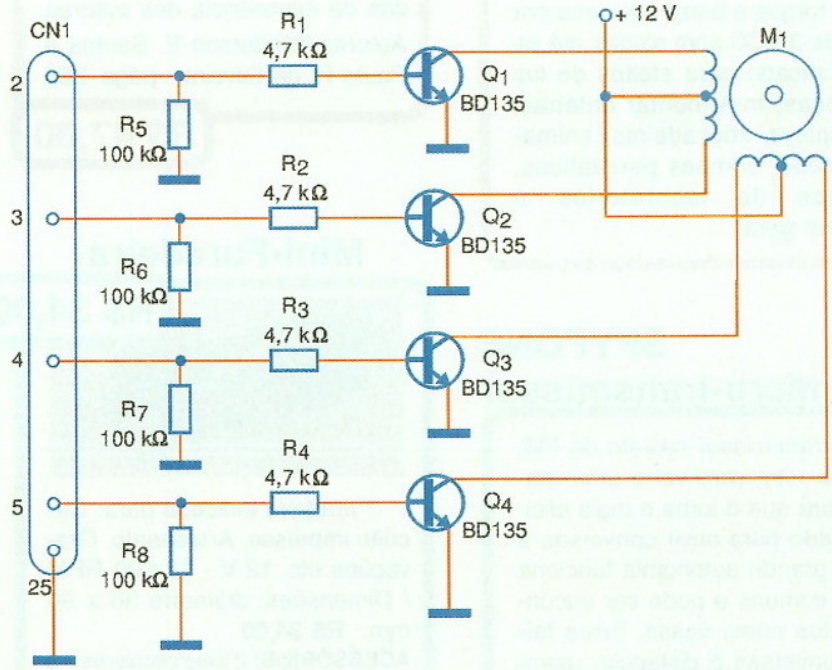
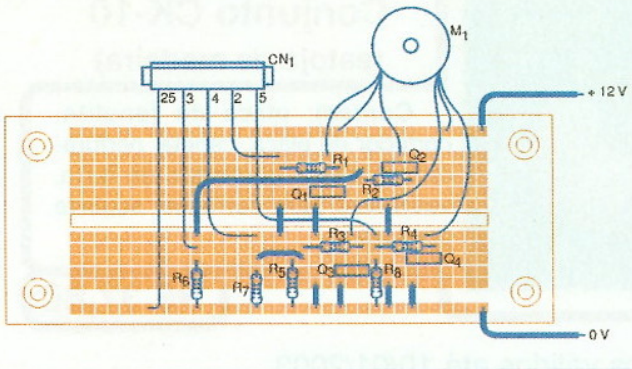


Figura 16 - Montagem na matriz de contatos da versão com chaves.



outra, caso que o próprio leitor pode descobrir experimentalmente acionando as chaves.

Para a versão controlada pelo PC basta agregar quatro resistores ao circuito e um conector DB25 conforme o circuito da **figura 15**.

A montagem desta versão numa matriz de contatos é mostrada na **figura 16**.

Nesta versão, os transistores devem eventualmente ser dotados de radiadores de calor.

Para acionar os diversos passos sugerimos que o leitor consulte o artigo "Controle de um LED Através da Porta Paralela", publicado na revista número 2.

Nesse artigo, Luiz Henrique Corrêa Bernardes ensina como energizar um LED a partir da porta paralela, usando a linguagem LOGO.

Basta ampliar estas explicações tratando as fases do motor como se fossem 4 LEDs acionados pelas diversas saídas da porta. Depois, é só implementar um programa que faça seu acionamento por botões virtuais ou ainda por um programa que faça o seqüenciamento em diversas velocidades.

É claro que outras linguagens podem ser usadas para acionar o motor empregando a porta paralela como, por exemplo, Visual Basic, Delphi, C++, Pascal, etc. ●

Lista de Material 2
(para versão acionada pela porta paralela)

Q_1 a Q_4 - BD135 ou BD137 - transistores NPN de média potência.
R_1 a R_4 - 4,7 k ohms x 1/8 W - resistores - amarelo, violeta, vermelho.
R_5 a R_8 - 100 k ohms x 1/8 W - resistores - marrom, preto, amarelo.
CN_1 - Conector DB25 macho
M_1 - Motor de passo de 4 fases
S_1 a S_4 - Interruptores miniatura para placa de circuito impresso, tipo normalmente aberto (NA).
Diversos: matriz de contatos, fios, fonte de alimentação conforme o motor, fios, solda, etc.

Errata

Na figura 18 do artigo desta série, na edição anterior, os pinos 6 e 7 do circuito integrado 555 estão invertidos.

1. O QUE É A FEBRACE?

A FEBRACE é uma feira anual de Ciências e Engenharia que envolve projetos de alunos das escolas públicas e particulares de todo o Brasil, em diversas categorias estabelecidas a partir das Ciências (Exatas e da Terra, Biológicas, da Saúde, Agrárias, Sociais e Humanas) e Engenharia e suas Aplicações.

A FEBRACE tem como principais objetivos estimular novas vocações em Ciência e Engenharia através do desenvolvimento de projetos criativos e inovadores, e aproximar as escolas públicas e privadas das Universidades, criando oportunidades de interação espontânea entre os alunos e professores das escolas com a comunidade universitária (estudantes universitários, professores, funcionários) para uma melhor compreensão dos papéis das Universidades em Ensino, Pesquisa, Cultura e Extensão.

2. QUEM PODE PARTICIPAR?

Estudantes da 8ª série do Ensino Fundamental e de Ensino Médio e Técnico das escolas públicas e particulares de todo o Brasil, não ultrapassando a idade máxima permitida de 21 anos (até o dia 1° de maio de 2003), que submetem seus projetos no prazo estabelecido e que têm seus projetos selecionados para participação.

Os estudantes podem participar individualmente ou em grupos de no máximo três alunos por projeto. Todos os projetos devem ter a participação de um adulto orientador, com mais de 21 anos de idade. Os projetos submetidos são avaliados por especialistas que selecionam os melhores projetos para participarem da FEBRACE.

3. QUAIS SÃO AS PREMIAÇÕES?

Todos os estudantes e professores orientadores de projetos que participam da FEBRACE recebem certificados de participação. Serão selecionados dentre os participantes, durante a FEBRACE, projetos para representarem o Brasil na feira internacional ISEF (International Science & Engineering Fair), os quais terão suas despesas com viagens aos E.U.A. e estadias pagas pela Intel, patrocinadora oficial da FEBRACE e da ISEF. São também convidados a visitarem a FEBRACE e a oferecerem premiações (computadores, PDA's, equipamentos científicos, bolsas de estudo, estágios, visitas técnicas, literatura, certificados), de acordo com seus respectivos critérios de avaliação, sociedades e associações científicas e profissionais, entidades governamentais e da iniciativa privada.

4. QUAIS SÃO AS CATEGORIAS?

OBSERVAÇÃO:
Todo projeto que envolva pesquisas com seres humanos e/ou animais e/ou substâncias controladas e/ou substâncias perigosas deverá ter aprovação prévia do instituto responsável pelo projeto. Esta aprovação deverá ser emitida por cientista(s) qualificados(s), antes de ser iniciada a pesquisa, e deverá acompanhar os demais documentos da inscrição.

Os projetos submetidos à FEBRACE devem pertencer a uma (ou mais) das seguintes categorias:

CIÊNCIAS EXATAS E DA TERRA: Matemática, Probabilidade e Estatística, Ciência da Computação, Astronomia, Física, Química, Geociências, Oceanografia.

CIÊNCIAS BIOLÓGICAS: Biologia Geral, Genética, Botânica, Zoologia, Ecologia, Morfologia, Fisiologia, Bioquímica, Biofísica, Farmacologia, Imunologia, Microbiologia, Parasitologia.

CIÊNCIAS DA SAÚDE: Medicina, Odontologia, Farmácia, Enfermagem, Nutrição, Saúde Coletiva, Fonoaudiologia, Fisioterapia e Terapia Ocupacional, Educação Física.

CIÊNCIAS AGRÁRIAS: Agronomia, Recursos Florestais e Engenharia Florestal, Engenharia Agrícola, Zootecnia, Medicina Veterinária, Recursos Pesqueiros e Engenharia de Pesca, Ciência e Tecnologia de Alimentos.

CIÊNCIAS SOCIAIS APLICADAS: Direito, Administração, Economia, Arquitetura e Urbanismo, Planejamento Urbano e Regional, Demografia, Ciência da Informação, Museologia, Comunicação, Serviço Social, Economia Doméstica, Desenho Industrial, Turismo.

CIÊNCIAS HUMANAS: Filosofia, Sociologia, Antropologia, Arqueologia, História, Geografia, Psicologia, Educação, Ciência Política, Teologia.

ENGENHARIAS: Eletrônica, Eletrotécnica, Mecânica, Química, Civil, de Minas, de Materiais e Metalúrgica, Sanitária, de Produção, Nuclear, de Transportes, Naval e Oceânica, Aeroespacial, Biomédica.

5. COMO SUBMETER UM PROJETO?

5.1 . TEMA:
Escolher um tópico para o projeto de pesquisa.

5.2 . REQUER APROVAÇÃO?
Se o projeto envolve uma das seguintes áreas:
- Pesquisa com seres humanos
- Animais vertebrados
- Recombinação de DNA
- Substâncias patogênicas ou microorganismos
- Tecido animal ou humano ou seus fluidos
- Substâncias ou equipamentos perigosos

Preencher e enviar um **Pedido de Aprovação** ANTES de iniciar a pesquisa. Aguardar o recebimento do código de aprovação do projeto para iniciar a pesquisa.

5.3. PESQUISA:
Conduzir a pesquisa seguindo o Método Científico ou o Método de Engenharia. Projetos em andamento que serão concluídos até o prazo de submissão ou projetos já completados podem também ser submetidos.

5.4. DIÁRIO DE BORDO:
Todo participante da feira deve elaborar, manter e trazer na feira um Diário de Bordo do Projeto.

No Diário de Bordo, o estudante deve registrar as etapas que realizou para desenvolver o projeto, relatando todos os fatos e as datas respectivas. O Diário de Bordo deve abranger o período relativo a todo o desenvolvimento do projeto e deve conter: o registro detalhado e preciso dos fatos, dos passos, das descobertas e das novas indagações, o registro das datas e locais das investigações; as entrevistas conduzidas, o registro dos testes e resultados alcançados.

5.5. PRÉ-SUBMISSÃO:
Preencher e enviar até 30 de novembro 2002 o formulário de **Planejamento de Projeto**.

5.6. SUBMISSÃO:
Preencher o formulário de **Informações do Projeto** e enviar com o **Relatório de Projeto** até o dia 20 de janeiro 2003.

6. MÉTODO CIENTÍFICO

6.1. ENUNCIAR O PROBLEMA OU A AFIRMAÇÃO
- Qual é seu objetivo?
- Qual é a idéia que você está tentando testar?
- Qual é a pergunta científica que você está tentando responder?

6.2. DESENVOLVA UMA HIPÓTESE
- Pense como seu projeto pode demonstrar seu propósito ou objetivo.
- Faça uma previsão dos resultados do experimento.
- Liste os resultados previstos em termos mensuráveis.

6.3. DESENVOLVA UM PROCEDIMENTO PARA TESTAR A HIPÓTESE
- Explique com detalhe como seu experimento será executado e como ele vai testar sua hipótese.
- Identifique as variáveis (elementos do experimento que mudam para testar a hipótese) e os controles (elementos do experimento que não mudam).
- Especifique como as medidas dos resultados vão provar ou refutar sua hipótese.
- Este procedimento deve ser como uma receita: uma outra pessoa deve poder executar o experimento seguindo o procedimento. Teste com um amigo ou parente para verificar que o procedimento está claro e completo.

6.4. MATERIAIS
- Liste os materiais e os equipamentos que serão utilizados.
- Esta lista deve incluir todos os ingredientes do procedimento.

6.5. OBSERVAÇÕES
- Registre sempre no Diário de Bordo do Projeto todas as observações, os dados e resultados. Estes podem ser medidas ou anotações sobre seu experimento.
- Fotografe os resultados de seu projeto ou as fases do mesmo, se possível e se isto ajudar sua análise ou a apresentação de seu experimento na feira.

6.6. ANÁLISE
- Explique suas observações, dados e resultados.
- Liste os pontos principais que você aprendeu.
- Porque você obteve estes resultados? O que seu experimento provou?
- Sua hipótese estava correta? Seu experimento provou ou refutou sua hipótese? Explique em detalhes.

6.7. CONCLUSÃO
- Responda ao problema ou à afirmação elaborados.
- Qual é o valor de seu projeto?
- Dados os resultados de seu experimento, qual seria a próxima pesquisa a ser desenvolvida? Qual seria a próxima pergunta que deveria ser feita?
- Se você tivesse que refazer a pesquisa, o que você mudaria?

7. MÉTODO DE ENGENHARIA

7.1. RECONHECER NECESSIDADES
- Observa o mundo ao seu redor.
- Quais são os problemas de sua comunidade?
- Quais são as necessidades?
- O que poderia ser melhorado?

7.2. DEFINIR O PROBLEMA
- Defina o problema escolhido.
- Descreva o problema em linhas gerais.
- Reduza o problema focando em um aspecto específico.

7.3. PROPOR ALTERNATIVAS DE SOLUÇÃO
- Pense em algumas alternativas de soluções que você encontrou para resolver o problema.

7.4. AVALIAR AS ALTERNATIVAS DE SOLUÇÃO
- Reflita sobre as soluções que você encontrou.
- Avalie as alternativas. O que elas têm de bom, o que elas têm de ruim?
- Qual seria o impacto delas na comunidade?
- Qual seria a dificuldade de implementar estas alternativas?

7.5. SELECIONAR A ALTERNATIVA PREFERIDA
- Escolha uma solução.
- Justifique sua escolha.

7.6. ESPECIFICAR A SOLUÇÃO E COMUNICAR O PROJETO
- Especifique o seu projeto.
- Faça sua pré-submissão a FEBRACE preenchendo e enviando o Planejamento de Projeto.

7.7. IMPLEMENTAR (FABRICAR E DISPONIBILIZAR) A SOLUÇÃO
- Implemente sua solução.
- Teste seu projeto.
- Submeta a FEBRACE redigindo, preenchendo e enviando o Relatório de Projeto.

OBSERVAÇÕES
- Registre sempre no Diário de Bordo do Projeto todas as observações, os dados e resultados. Estes podem ser medidas, descrições ou anotações.
- Fotografe os resultados de seu projeto ou as fases do mesmo, isto pode ajudar a documentar e a apresentar seu projeto na feira.

Informações mais detalhadas no site: www.lsi.usp.br/febrace



Curso de ELETRÔNICA DIGITAL

CodeWarrior

Automation Studio

ATMEL ARM

Nas Bancas

www.sabereletronica.com.br

SILICOM
INTERNET PROVIDER

- Hospedagem de web-sites
- Acesso discado e dedicado à internet
- Acesso banda larga ADSL
- Registro e manutenção de domínios
- Colocation de equipamentos
- Desenvolvimento e implantação de conectividade a internet
- Além de diversos outros serviços na área de tecnologia da informação

HOSPEDAGEM PROFISSIONAL
19,90*
* A PARTIR DE

Windows 2000

HTTP://WWW.SILICOM.COM.BR

FONE/FAX.: +55 11 6198-2526

RM-1 – Robô Manipulador

Construa um braço mecânico, passo-a-passo, capaz de manipular pequenas "cargas". Todo o controle é feito por um microcomputador, aumentando ainda mais sua possibilidades. Apresentamos sua construção em duas versões: uma com material de fácil acesso e outra com plástico. Assim, o leitor poderá escolher a versão que julgar mais conveniente.

Conversão de energia e transmissão de movimento com LEGO

Neste artigo, o autor mostra como utilizar peças do *kit* da LEGO Dacta associadas a outros materiais eletrônicos para demonstrar a geração de energia elétrica (pela transformação da energia mecânica) a fim de acender pequenas lâmpadas, LEDs ou até mesmo para armazenar esta energia num capacitor.

Medindo com o paquímetro

Ao ver alguém utilizando um paquímetro, é curioso saber como se pode ler uma medida tão pequena em uma escala tão repleta de números. Neste artigo, o autor mostra de maneira clara como se utiliza um paquímetro, instrumento de grande importância para medir comprimentos com grande precisão.

O software que faz os robôs jogarem futebol

Ainda que o aparato eletrônico e mecânico seja essencial, de fato, a criação do futebol de robôs teve muito a ver com a necessidade de fomentar o desenvolvimento de técnicas sofisticadas e avançadas em diversas áreas da Ciência da Computação. Conheça o sistema de software por trás dos robôs-jogadores.

Como gravar um microcontrolador PIC

O uso dos microcontroladores é crescente e o estudante/hobista/técnico não pode ficar de fora. Aprenda como gravar um microcontrolador PIC com o auxílio das ferramentas Microchip e do projeto PROGPIC-1, que é um gravador de baixo custo.

Motores de passo

Estes motores são ideais para casos onde a precisão seja um requisito importante do projeto. Saiba como funcionam esses motores e saiba também como utilizá-los em projetos de Mecatrônica.

Registrando eventos com o LOGO

Em sistemas de Automação e robotizados, é muito importante o registro de eventos para análise ou estatística. Neste artigo, vamos explorar um pouco esse assunto mostrando como fazer registros de eventos utilizando o LOGO.

Automação Simples

O real poder do microcontrolador está em sua habilidade de tomar decisões baseadas em suas "entradas" e, então, manipular o "mundo real" com suas saídas. Neste artigo, o autor explora este conceito propondo uma automação simples utilizando o Basic Step.

Trabalhando com materiais alternativos II

Veja, neste artigo, algumas dicas de como trabalhar com garrafas ou recipientes plásticos descartáveis, aproveitando sua forma previamente moldada nos projetos de mecatrônica.