

O Uso da Robótica como Apoio ao Ensino de Disciplinas do Curso de Ciência da Computação

Mário Yoshihiko de Souza Haratani¹, Emerson Rodrigo Alves Tavares¹

¹Departamento de Ciência da Computação
Universidade Presidente Antônio Carlos (UNIPAC) – Antônio Carlos, MG – Brasil
mharatani@hotmail.com, emersontavares@unipac.br

Resumo. *Este artigo descreve a construção de um robô, controlado pela porta paralela de um microcomputador. É sugerida a utilização da robótica no ensino, onde pertinente, de forma que indique aos alunos os pontos práticos de aplicação dos conhecimentos lecionados em um curso de graduação em ciência da computação.*

Palavras-chave: *robótica, aplicação no ensino.*

1. Introdução

Há tempos, o homem tenta construir robôs que possam realizar trabalhos em seu lugar, pelos mais variados motivos: por serem trabalhos pesados, perigosos, repetitivos ou que exijam precisão. Marcos Gera [GERA 2002], em sua tese de mestrado, diz que a priorização dos estudos voltados para as “aplicações” leva os alunos a preocuparem-se apenas com a seguinte questão: “onde vou aplicar isso?”.

É exatamente esta questão que, muitas vezes, pode ser respondida com a robótica, pois muitos conteúdos tratados ao longo de um curso de Ciência da Computação são utilizados na construção de robôs. Agregando-se a isso o fascínio que os robôs despertam, obtemos um ícone, cuja maior força reside na atenção que é capaz de atrair, que pode ajudar a amenizar a sensação de inaplicabilidade do que é lecionado.

2. Objetivo do trabalho

Desenvolver um robô que possa ser útil no auxílio da aprendizagem de alguns conteúdos presentes na ementa do curso de ciência da computação.

Sugerir a implementação de um recurso alternativo de aplicação prática do conhecimento formal apresentado nas disciplinas de computação nos cursos de graduação em informática, na qual o conhecimento transmitido seja aplicado na construção gradativa de um robô, a ser executada pelos alunos.

3. Considerações preliminares

Tendo como foco a aplicação do projeto em sistemas de ensino, não são abordados neste artigo os cálculos para o dimensionamento dos componentes eletrônicos, sendo que mais detalhes sobre esta parte do trabalho podem ser obtidos em [JONES 2002] e [BIASI 1968].

Em uma outra análise, há que se considerar que um projeto desta natureza envolve custos que nem sempre podem ser arcados pelas instituições de ensino. Por este

motivo, definiu-se que todo material a ser utilizado no projeto deveria ser:

- 1- reaproveitado de equipamentos obsoletos ou parcialmente danificados, ou
- 2- encontrado no mercado a baixo custo.

Ainda na linha do custo, destacam-se mais dois aspectos: a modularização e o uso da porta paralela de um microcomputador.

A modularização é importante quando se observa o módulo de endereçamento/acionamento de cada motor, pois a construção de apenas dois módulos é mais que suficiente para que se entenda o seu funcionamento.

Já o uso da porta paralela de um microcomputador evita a utilização de componentes específicos para controle, conhecidos como PIC's, que embora baratos e de uso mais corrente no mercado, prejudicam a compreensão por serem de implementação um pouco mais complexa, além de exigirem a aquisição de outros equipamentos.

4. Elementos de hardware utilizados no robô

Nesta seção serão apresentados os elementos essenciais de hardware do projeto.

4.1. O motor de passo

O motor de passo é um motor especialmente construído de forma que o movimento de seu eixo possa ser controlado. É ele o responsável por traduzir os *comandos* desejados em *movimento e trabalho*.

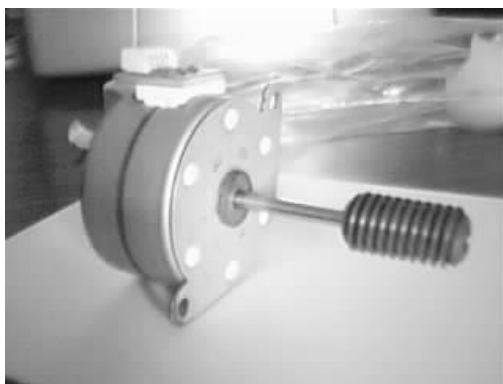


Figura 1. Motor de ímã permanente (com fuso sem-fim)

O projeto prevê a utilização de motores de ímã permanente e quatro enrolamentos, com uma ou duas derivações centrais, uma vez que este tipo é o mais usado pelos fabricantes de periféricos, ver Figura 1.

4.2. A porta paralela

A porta paralela é uma interface bidirecional presente na maioria dos microcomputadores tipo PC. Seu uso mais comum é a comunicação com impressoras. O padrão IEEE 1284 é o que descreve a porta paralela de comunicações e nele se encontra a melhor definição disponível da porta paralela [JONES 2002].

A porta paralela, na saída do microcomputador (Figura 2), utiliza um conector DB-25, cuja pinagem está descrita na Figura 3.



Figura 2. Porta paralela

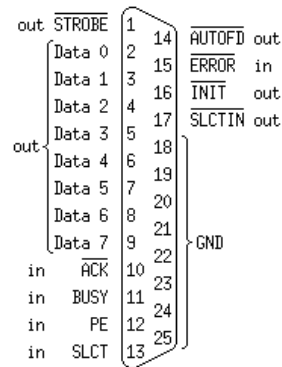


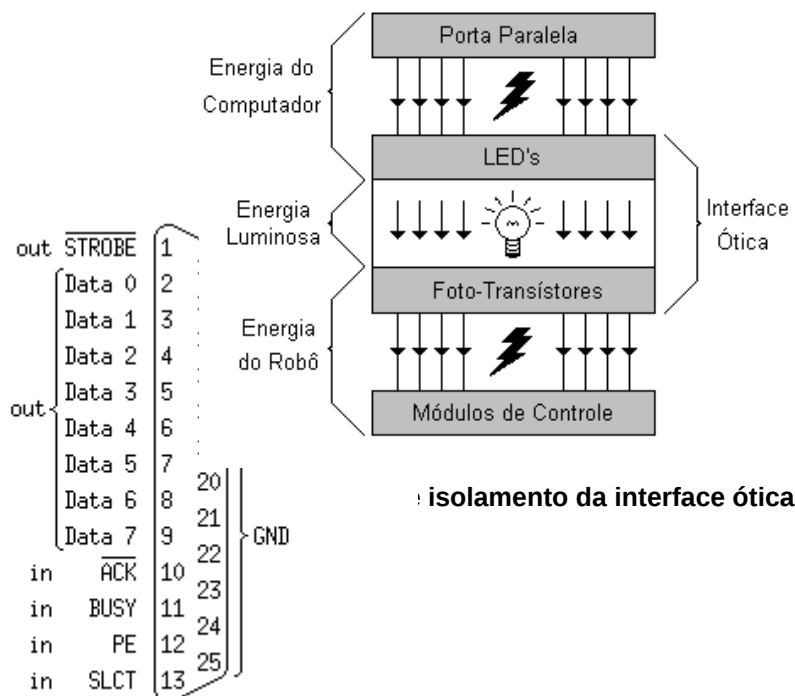
Figura 3. Pinagem da porta paralela [JONES 2002]

Pode-se observar que existem apenas oito saídas de dados (data 0 até data 7) na porta paralela. Considerando-se que são quatro enrolamentos por motor, serão necessários quatro sinais elétricos para acioná-lo. Desta forma, apenas dois motores poderiam ser controlados utilizando-se a porta paralela, a menos que sejam utilizados circuitos adicionais.

4.3. A interface ótica

A porta paralela de qualquer computador é uma interface sensível. Sua proteção original reside em apenas alguns componentes que “drenam” um eventual excesso de corrente. A interface ótica aumenta essa proteção, pois utiliza optoacopladores, que bloqueiam o acesso elétrico direto e isolam a porta paralela dos circuitos do robô. Os optoacopladores são capazes de suportar altas tensões (5300Vac em condições de teste) entre a entrada (LED's) e a saída (Foto-transistores) [4N25 2006][4N33 2006]. Essa excelente margem de proteção torna a interface imprescindível, uma vez que um eventual erro na construção do robô ou mesmo um defeito não danificará o computador.

A interface isola eletricamente o microcomputador do restante dos circuitos, da forma mostrada na Figura 4.



4.3.1. Esquema eletrônico

É importante observar que a linha PE deve ser ligada ao “terra” da porta paralela, o que pode ser feito nos pinos 18 a 25 da mesma. O esquema eletrônico está na Figura 5.

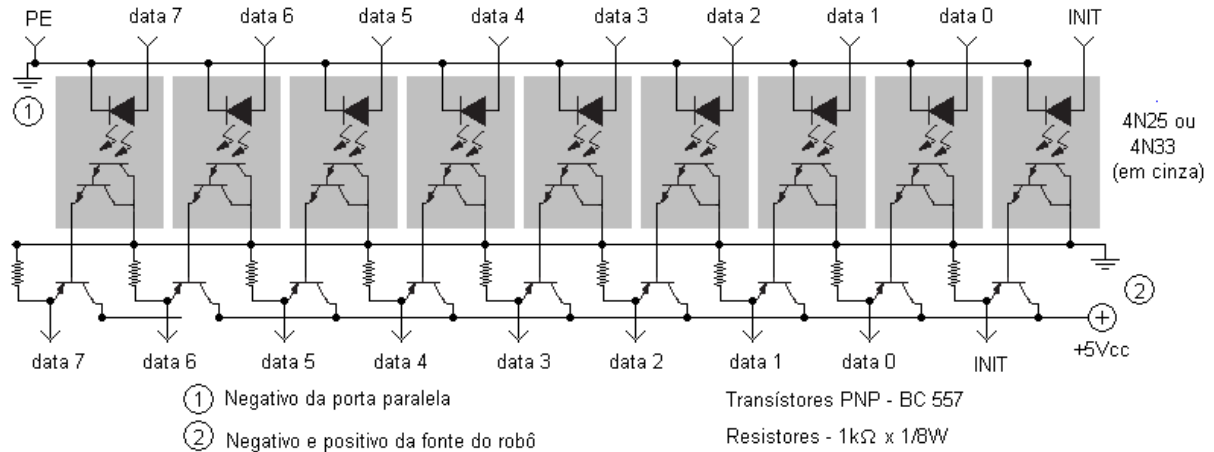


Figura 5. Esquema eletrônico da interface ótica

4.4. Fontes secundárias de energia

Conforme já visto acima, a interface ótica isola o computador dos circuitos de controle do robô. Com isso há a necessidade de utilização de fontes secundárias de energia. No projeto foram utilizadas duas fontes de microcomputadores, ligadas em paralelo, sendo que as linhas de 12 volts são usadas para alimentação dos motores e uma de 5 volts para a alimentação dos circuitos eletrônicos.

4.5. O gerador de pulsos (clock)

Os circuitos de controle necessitam de um meio de estar em sincronia com os dados que são enviados pela porta paralela. Para isso foi utilizado um gerador de onda quadrada, com frequência superior à do envio de dados. Desta forma, pode-se garantir que o enviado pelo computador será gravado na memória do controlador, pois sempre haverá no mínimo um pulso de sincronismo para cada informação disponibilizada na porta paralela, conforme mostrado na Figura 6.

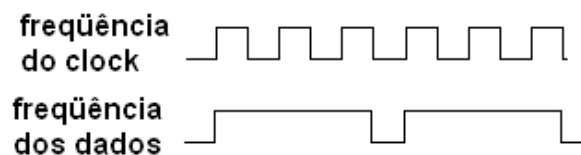


Figura 6. Diferença de frequências

A construção do gerador de pulsos foi feita com base no *Datasheet* da National Semiconductor do circuito integrado LM 555 [LM555 2006]. O esquema eletrônico do gerador está representado na Figura 7.

A frequência dos pulsos é calculada utilizando-se a fórmula:

$$f = \frac{1}{T} = \frac{1,44}{(Ra+2Rb)C}$$

4.5.1 Esquema eletrônico

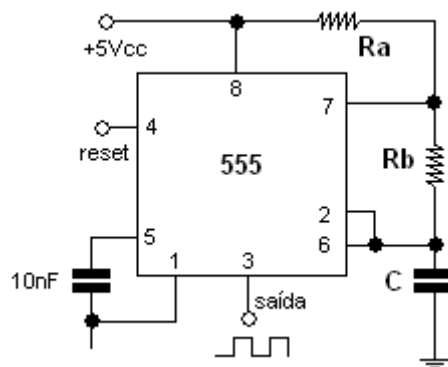


Figura 7. Esquema eletrônico do gerador de pulsos

4.6. Controlando mais de dois motores de passo com a porta paralela

O funcionamento básico de um motor de passo está descrito na Figura 8. Eles não mantêm a posição do eixo sem energia e, por esse motivo, torna-se necessário armazenar o dado destinado a um determinado controlador em uma memória. De outra forma, ao endereçar qualquer outro dado a outro controlador o primeiro dado seria perdido (e o eixo não manteria sua posição, tornando impossível o controle dos motores).

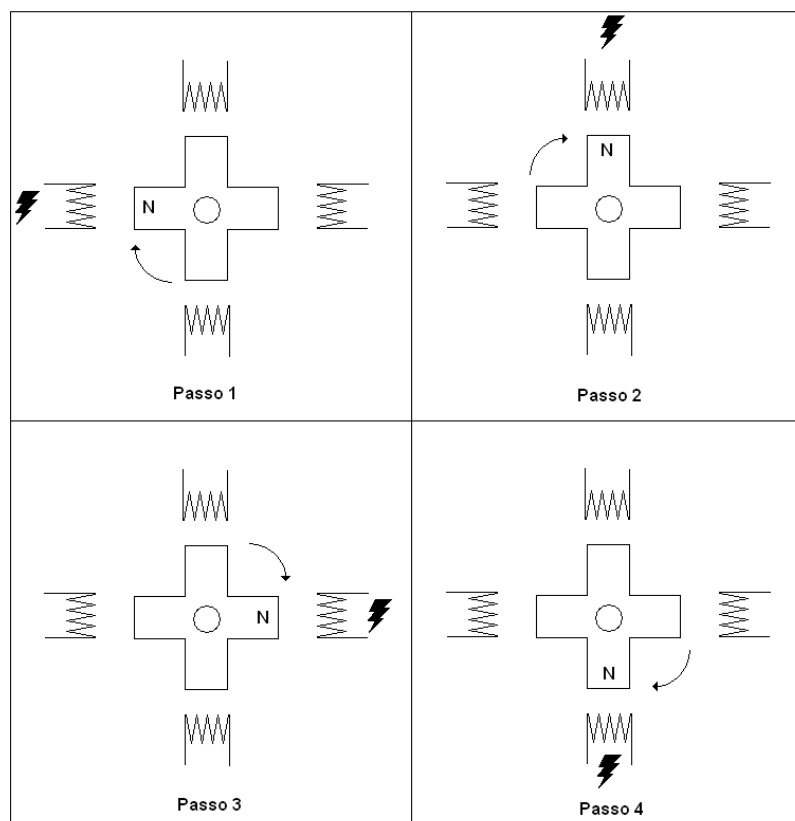


Figura 8. Giro do eixo de um motor de passo (básico)

4.6.1. O módulo controlador de motor de passo

A Figura 9 exibe o diagrama de blocos do controlador. É importante ressaltar que as oito linhas de bits chegam aos módulos em paralelo, ou seja, os mesmos bits chegam de forma idêntica em cada um deles.

Os dados chegam das linhas *data0* a *data7* através da interface ótica. Os quatro primeiros bits (controle) informam quais as bobinas serão ativadas (1-ativada/ligada, 0-desativada/desligada).

Os outros quatro bits formam o endereço do módulo controlador que deve ser ativado. O bloco comparador verifica se estes bits são iguais ao de algum módulo. Em caso positivo a memória deste módulo é habilitada e o pulso de clock externo efetiva a gravação dos quatro primeiros bits nela.

Desta forma, a memória é quem se encarrega de manter a posição do eixo do motor de passo, através do driver, permitindo que outros dados possam ser enviados para outro módulo, sem afetar a memória dos demais módulos e, conseqüentemente, a posição do eixo dos demais motores.

Cada bloco (“comparador”, “memória” e “driver”) é um circuito integrado. “O **circuito integrado** é um circuito eletrônico completo miniaturizado manufaturado sobre um pequeno chip de silício” [MARTINS e PUGET 2000].

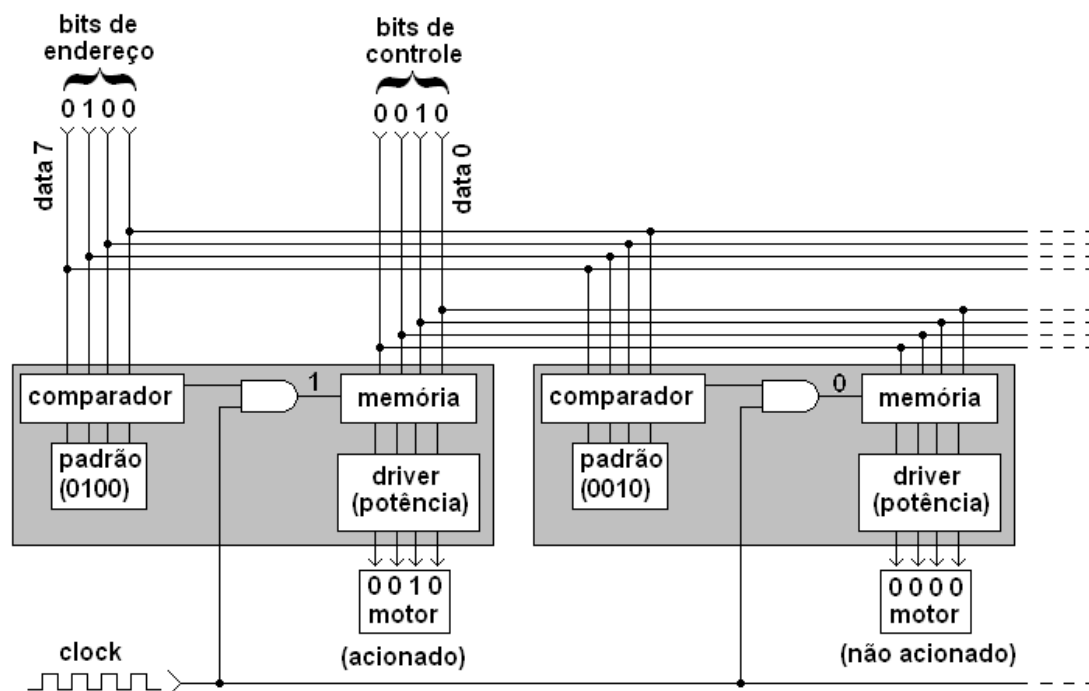


Figura 9. Diagrama de blocos do módulo controlador

4.6.1.1. Esquema eletrônico

O circuito integrado 74LS194[74LS194 2006] é um “*parallel-in, parallel-out shift-register*”, que pode ser utilizado como memória caso os pinos S0 e S1 sejam ligados em paralelo, convertendo o modo de entrada de dados em uma linha “*enable*” (habilita)

[JONES 2002]. O comparador é o 74LS85[74LS85 2006] e o *driver* é o ULN2003[ULN2003 2006], que lida diretamente com os motores. O esquema eletrônico pode ser visto na Figura 10.

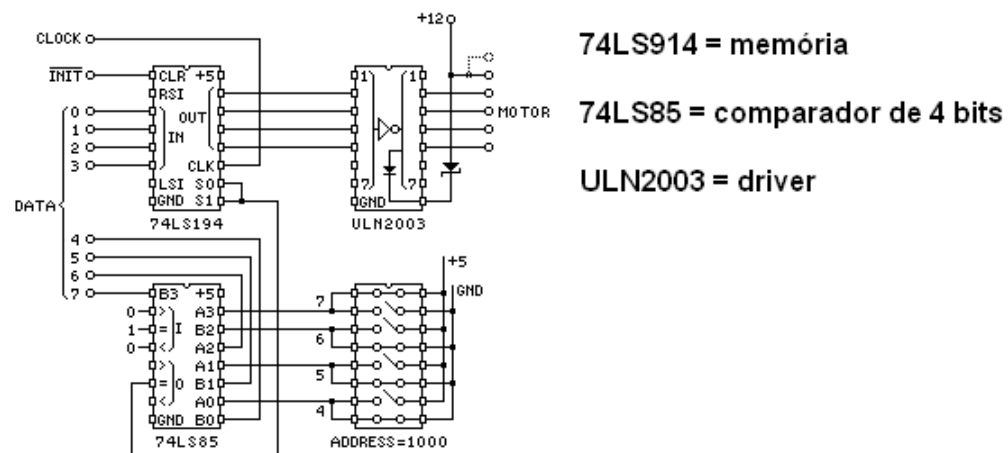


Figura 10. Esquema eletrônico do controlador [JONES 2002]

4.6.1.2 Etapa de potência adicional

No projeto foram utilizadas etapas de potência adicionais que suportassem a corrente exigida por dois dos motores. Estas etapas foram construídas com transistores de potência e foram ligadas entre o *driver* e o motor em questão, cujo esquema eletrônico está na Figura 11.

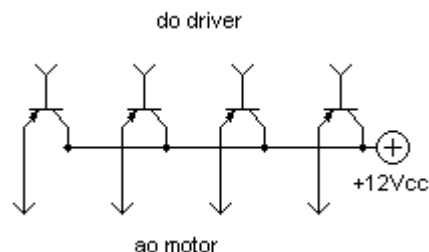


Figura 11. Esquema eletrônico da etapa de potência adicional

5. Software

A necessidade básica do projeto, em termos de software, é enviar dados pela porta paralela, que serão interpretados pelos módulos e convertidos em movimento. Devido a essa simplicidade conceitual, não foi utilizada nenhuma metodologia formal para modelagem.

Para o envio dos bytes é usada a função **outportb()**, presente no pacote da linguagem C/C++ para MS-DOS da Borland® (atualmente Inprise®). O endereço para envio dos bytes na plataforma MS-DOS® ao Windows98® é 378h para a porta paralela 1 (LPT1) e é esse o endereço utilizado no projeto. A sintaxe da função **outportb()** conforme definida no "help" do turbo C é a seguinte:

void outportb(int portid, unsigned char value); onde:

portid é 378h (ou 0x378 na sintaxe da linguagem C) e
value é o byte que se deseja enviar para portid.

Este byte é composto pelo endereço do controlador do motor, (bits de 4 a 7) e pelo(s) bit(s) correspondente(s) à(s) bobina(s) que deve(m) ser ativada(s) (bits 0 a 3), conforme exemplificado na Tabela 1. Quando duas bobinas adjacentes são ativadas ao mesmo tempo e, em seguida somente uma delas é ativada, acontece o que é conhecido como “*half step*” (meio passo). Isso proporciona um movimento mais suave do eixo do motor e maior torque, mas também implica em maior demanda de corrente elétrica [JONES 2002].

Tabela 1. Sequência de bits para um giro completo do eixo (*half-step*)

Passos	Endereço bits 4 a 7				Bobinas bits 0 a 3				Byte Resultante
1	1	0	0	0	0	0	0	1	129
2	1	0	0	0	0	0	1	1	131
3	1	0	0	0	0	0	1	0	130
4	1	0	0	0	0	1	1	0	134
5	1	0	0	0	0	1	0	0	132
6	1	0	0	0	1	1	0	0	140
7	1	0	0	0	1	0	0	0	136
8	1	0	0	0	1	0	0	1	137
1	1	0	0	0	0	0	0	1	129
.									.
.									.
.									.

Um cuidado especial deve ser tomado com relação à velocidade de transmissão dos dados, pois os motores levam uma fração de tempo entre a energização da bobina e o efetivo movimento [JONES 2002]. Como esse tempo varia de motor para motor e a quantidade de bytes enviados por intervalo de tempo será diretamente proporcional à velocidade do computador que estiver executando a aplicação, é mais eficaz experimentar valores, por tentativa e erro, até que sejam encontrados valores adequados.

O programa da listagem 1 (disponível na mídia digital) faz a leitura do teclado e envia os dados correspondentes para a porta paralela, proporcionando um controle interativo do robô.

Uma outra abordagem é fazer a leitura de um arquivo contendo “comandos” que podem ser interpretados e traduzidos em movimento. O programa da listagem 2 (disponível na mídia digital) faz a leitura de um arquivo texto que contém comandos. Estes comandos estão estruturados na forma da Tabela 2 (sem os comentários).

Cada linha do arquivo é um comando, que contém o motor ao qual está direcionado, o comando em si (giro completo do motor ou interpretação direta do dado) e quatro bits, que correspondem às bobinas do motor que devem ser acionadas. Caso o comando seja o giro completo do motor, os quatro bits devem ser 0000 para giro à direita e 1111 para giro à esquerda.

Tabela 2. Estrutura do arquivo de comandos

Motor	Comando	Dado / Direção				Comentário
0	0	0	0	0	1	Ativa a primeira bobina do motor 0
1	0	1	0	0	1	Ativa a primeira e a última bobina do motor 1
3	0	0	0	1	1	Ativa a primeira e a segunda bobina do motor 3
0	1	0	0	0	0	Giro no motor 0 à direita
1	1	0	0	0	0	2 Giros no motor 1 à direita
1	1	0	0	0	0	
2	1	1	1	1	1	Giro no motor 2 à esquerda
0	1	1	1	1	1	2 Giros no motor 0 à esquerda
0	1	1	1	1	1	
0	0	0	0	0	0	Desliga todas as bobinas dos motores
1	0	0	0	0	0	
2	0	0	0	0	0	
3	0	0	0	0	0	
4	0	0	0	0	0	

6. Exemplo do uso de conteúdo das disciplinas do curso de Ciência da Computação

Nesta seção são enumerados alguns dos possíveis usos do projeto em sala de aula.

6.1 Sistemas de Numeração (Álgebra booleana)

Podem ser facilmente abordados, com ênfase no sistema binário, buscando consolidar a sua relação direta com a eletrônica e a computação. Há um monitor no painel frontal do robô que exibe, usando LED's, os bits que compõem o byte enviado pela porta paralela, auxiliando nesta tarefa.

6.2 Portas Lógicas (Arquitetura e Organização de Computadores, Álgebra Booleana)

Portas lógicas estão presentes em cada circuito integrado. O circuito comparador é particularmente interessante, devido à sua concepção interna. Um bom exercício para os alunos pode ser a “construção” de um comparador de 4 bits a partir da combinação de portas lógicas (AND, OR, NOT).

6.3 Barramentos e Endereçamento (Arquitetura e Organização de Computadores)

O conceito de barramento pode ser explorado em sala, tomando-se como exemplo as linhas *data* do projeto, com uma breve alusão ao endereçamento de memória, através do circuito comparador.

6.4 Clock (Arquitetura e Organização de Computadores, Física)

Para o clock do projeto foi utilizado um *timer* de precisão. O cálculo da frequência é baseado em resistores e capacitores, cujo estudo é abordado na disciplina de Física.

Além disso, o clock é indissociável dos processadores, sendo que a sua aplicação no projeto é similar à do clock dos computadores.

6.5 Software (Programação, Estruturas de Dados)

Os softwares do projeto, apesar de simples, utilizam diversas estruturas de codificação que podem ser tomadas como exemplo para as fases iniciais do curso.

7. Parte mecânica

O projeto está estruturado de forma que a parte mecânica pode ser direcionada para diversos tipos de aparelhos que utilizem motores de passo (plotters, posicionadores, braços, etc).

Neste projeto foi desenvolvido um braço mecânico, dotado de uma “garra”, uma base giratória e três articulações. Todo o conjunto foi montado em um gabinete de microcomputador, conforme mostrado na Figura 12.



Figura 12. O protótipo montado

8. Conclusão e sugestões para trabalhos futuros

Esta seção finaliza este artigo. É necessário destacar que todas as sugestões apresentadas visam ampliar as possibilidades de aplicação do robô no ensino e não torná-lo mais eficiente.

8.1 Conclusão

Durante o desenvolvimento do projeto foi observada a aplicabilidade direta de conhecimentos adquiridos durante o curso de ciência da computação. Foi verificado que o protótipo reflete de forma direta a implementação dos objetivos propostos e que este pode ser usado como ferramenta de apoio ao ensino, sendo esta a principal contribuição do projeto para o meio acadêmico.

Como em qualquer projeto, algumas dificuldades foram encontradas, sendo que as maiores estavam relacionadas com a dificuldade de encontrar os componentes dentro das premissas estabelecidas (reaproveitados ou encontrados a baixo custo).

Uma outra dificuldade foi tentar fazer o controle do robô através da movimentação de um mouse – possibilidade vislumbrada durante o desenvolvimento do projeto – que não pôde ser levado a termo devido ao elevado tempo de resposta do protótipo em relação ao movimento do mouse. Dois fatores foram decisivos: a lentidão da parte mecânica, tal como implementada (engrenagem acionada por fuso sem-fim), e o tempo de resposta dos componentes utilizados na interface ótica.

8.2 Sugestões para trabalhos futuros

- Criar dispositivo programável, para tornar o robô independente do microcomputador, utilizando PIC's ou componentes semelhantes;
- Dotar o robô de sensores de posição e controlá-lo utilizando um sistema de coordenadas em 3D via software;
- Desenvolver um software que funcione em ambiente gráfico para controle interativo do robô;
- Desenvolver um software tradutor que converta comandos do tipo “Gire 30° à direita” nos bytes correspondentes;
- Montar um novo protótipo, cujo movimento esteja ligado diretamente aos eixos dos motores, de forma a possibilitar a implementação do mouse como dispositivo de movimento.
- Montar uma nova interface ótica, utilizando componentes similares que tenham tempo de resposta mais curto.

Referências Bibliográficas

- [4N25 2006] 4N25 – Datasheet – Fairchild Semiconductor – disponível em: <http://www.ortodoxism.ro/datasheets/fairchild/4N25.pdf>. Acesso em 10.06.2006.
- [4N33 2006] 4N33 – Datasheet – Motorola Semiconductor – disponível em: <http://www.ortodoxism.ro/datasheets/motorola/4N30.pdf>. Acesso em 10.06.2006.
- [74LS85 2006] 74LS85 – Datasheet – Fairchild Semiconductor – disponível em: <http://www.ortodoxism.ro/datasheets/fairchild/DM74LS85.pdf>. Acesso em 10.06.2006.
- [74LS194 2006] 74LS194 – Datasheet – Fairchild Semiconductor – disponível em: http://www.tranzistoare.ro/datasheets/70/375495_DS.pdf. Acesso em 10.06.2006.
- [BD330 2006] BD330 – Datasheet – Philips Semiconductors – disponível em: <http://www.ortodoxism.ro/datasheets/philips/BD330.pdf>. Acesso em 10.06.2003.

- [BIASI 1968] BIASI, R. S. (1968) – *Enciclopédia Record de Eletricidade e Eletrônica* 7 ed. – Rio de Janeiro: Record.
- [GERA 2002] GERA, Marcos Silvestre. (2002) – Globalização e educação – <http://libdigi.unicamp.br/document/?down=vtls000265182>. Biblioteca Digital da Unicamp – tese de mestrado. Acesso em 03.05.2007.
- [JONES 2002] JONES, Douglas W. (2002) <http://www.cs.uiowa.edu/~jones/step/> - Site of the Department of Computer Science – University of Iowa. Acesso em 04.10.2002.
- [LM555 2006] LM555 – Datasheet – National Semiconductor – disponível em: <http://cache.national.com/ds/LM/LM555.pdf>. Acesso em 20.02.2006
- [MARTINS e PUGET 2000] MARTINS, Antônio Cesar Germano, PUGET, Maria Augusta Constante (2000) – Evolução dos Computadores – Introdução à Ciência da Computação, disponível em: <http://www.sorocaba.unesp.br/professor/amartins/aulas/icc/histcomp.PPT>. Acesso em 08.06.2007.
- [ULN2003 2006] ULN2003 – Datasheet – Allegro Microsystems, Inc – disponível em: http://www.allegromicro.com/en/Products/Part_Numbers/2001/2001.pdf. Acesso em 10.06.2006.