

Centro Universitário de Brasília- UniCEUB
Faculdade de Ciências Exatas e Tecnologia- FAET
Curso de Engenharia de Computação
Disciplina: Projeto Final
Professor Orientador: Prof°. Aderlon Marcelino Queiroz



**Segurança Residencial:
Monitoramento Visual do Ambiente via
*Internet com Alarme via E-mail***

WINÍCIUS FERRAZ NERES
REGISTRO ACADÊMICO: 2011539/6

Brasília-DF, Dezembro de 2007



**CENTRO UNIVERSITÁRIO DE BRASÍLIA- UNICEUB
FACULDADE DE CIÊNCIAS EXATAS E TECNOLOGIA- FAET
ENGENHARIA DE COMPUTAÇÃO**

**Segurança Residencial:
Monitoramento Visual do Ambiente via
*Internet com Alarme via E-mail***

**WINÍCIUS FERRAZ NERES
REGISTRO ACADÊMICO: 2011539/6**

Monografia apresentada à Banca Examinadora da Faculdade de Ciências Exatas e Tecnologia- UniCEUB, para conclusão do curso de Engenharia de Computação.

Brasília-DF, Dezembro de 2007

Agradecimentos

Agradeço primeiramente a Deus por me prover saúde, o principal bem de qualquer ser humano. Agradeço também aos meus maravilhosos pais Wanderly e Tânia que sempre me deram carinho e proporcionaram condições de acesso à educação em todas as etapas da minha vida. Expresso meu agradecimento também ao professor Aderlon M. Queiroz por me ensinar os caminhos a serem seguidos neste projeto.

Além dos citados, não poderia esquecer minha namorada Nayara, companheira carinhosa e compreensiva em todos os momentos. Agradeço ainda ao meu amigo Bruno Rithele, pessoa de elevado conhecimento que muito me ajudou ao longo deste projeto.

Resumo

Nos últimos anos o acesso à tecnologia por parte da população cresceu muito e tal constatação é ratificada, por exemplo, quando pesquisas confirmam que o número de pessoas que acessam a *Internet* e que possuem telefone celular aumentou muito na última meia década. Na contramão do desenvolvimento do país existem os índices de criminalidade que não diminuíram no mesmo período. Este projeto visa aproveitar as soluções tecnológicas cada vez mais disponíveis e proteger o patrimônio particular através do desenvolvimento de um sistema de alarme que utilize recursos tais como *Internet* e telefone celular. Para implementação deste projeto a porta paralela foi utilizada como canal de comunicação entre o microcomputador e os circuitos responsáveis por controlarem o alarme e o movimento da câmera. Além disto, uma câmera USB foi utilizada para prover a imagem, e um sensor de presença de alarme empregado para monitorar o ambiente. Como complemento, alguns *softwares* de terceiros foram utilizados para prover o acesso remoto via *Internet* à máquina que detém a visualização do ambiente.

Palavras chaves: Alarme, motor de passo, porta paralela, sensor e *Visual Basic*.

Abstract

In recent years the use of technology has grown up dramatically which is confirmed by numbers of people accessing the Internet or people who own mobile phones have increased in the last five years. Countering the sense of evolution, the crime statistics were not reduced in the same period. This project aims to take advantage of technological solutions increasingly available and to protect the assets through the development of an alarm system that uses the Internet and mobile phones. For implementation of this project the parallel port was used as a channel of communication between the PC and the control circuits responsible for the alarm and movement of the camera. Also, a USB camera was used to provide the image and an alarming presence sensor was used to monitor the environment. In addition, some third-party software was used to provide remote access through the Internet to the machine that is responsible for the surveillance of the environment.

Key Words: Alarm, engine step, parallel port, sensor and Visual Basic.

Sumário

Lista de Figuras.....	VII
Lista de Tabelas.....	IX
Lista de Abreviaturas.....	X
1 Introdução.....	1
1.1 Objetivo Geral.....	1
1.1.1 Objetivos Específicos.....	2
1.2 Estrutura da Monografia.....	2
2 Interfaces de Comunicação.....	3
2.1 Tipos de Transmissão.....	3
2.2 Porta Paralela.....	4
2.2.1 Endereçamento de Porta Paralela.....	5
2.2.2 Conectores.....	7
2.3 Porta USB.....	8
2.4 Considerações Finais.....	10
3 Motores Elétricos.....	11
3.1 Motor de Passo.....	13
3.1.1 Motor de Passo de Ímã Permanente.....	14
3.1.2 Motor de Passo de Relutância Variável.....	16
3.2 Operação do Motor de Passo.....	17
3.3 Controle do Motor de Passo.....	19
3.4 Controle via <i>Software</i> do Motor de Passo.....	22
3.5 Considerações Finais.....	26
4 Sensores de Presença.....	27
4.1 Características Específicas dos Sensores de Presença.....	28
4.2 Considerações Finais.....	32
5 Produção Técnica.....	33
5.1 Montagem da Infra-Estrutura do Controle da Câmera.....	34
5.2 Ambiente de Desenvolvimento do Controle da Motor/Câmera.....	38
5.3 Montagem da Infra-Estrutura do Alarme via <i>E-mail</i>	42
5.4 Ambiente de Desenvolvimento do Alarme via <i>E-mail</i>	45
5.5 Resultados Obtidos.....	48
6 Considerações Finais.....	50
Referências Bibliográficas.....	51

Anexos	53
Anexo A: Tabela de pinos da porta paralela.....	53
Anexo B: Código Fonte do Experimento que Aciona <i>Leds</i>	54
Anexo C: Especificação do Motor de Passo.....	55
Anexo D: Especificação do Sensor de Presença	57
Anexo E: Código Fonte do Controle de Motor de Passo e Visualização via Câmera.....	58
Anexo F: Código Fonte do <i>Software</i> de Alarme via <i>E-mail</i>	64

Lista de Figuras

Figura 2.1: Transmissão Serial.....	3
Figura 2.2: Transmissão Paralela.....	3
Figura 2.3: Descrição DB25 Fêmea	6
Figura 2.4: Descrição DB25 Macho.....	6
Figura 2.5: Circuito Simplificado da Porta Paralela	7
Figura 2.6: Intervalo de Endereçamento LPT1	7
Figura 2.7: Conector DB25 Fêmea.....	8
Figura 2.8: Conector DB25 Macho	8
Figura 2.9: Conector Centronics.....	8
Figura 2.10: <i>Plug</i> Padrão USB	9
Figura 2.11: Fios do Cabo USB.....	9
Figura 3.1: Transformação de Energia no Motor.....	11
Figura 3.2: Motores Elétricos.....	11
Figura 3.3: Regra da Mão Direita	12
Figura 3.4: Motor de Passo do Projeto	13
Figura 3.5: Um Pulso Gera um Passo	14
Figura 3.6: Vários Pulsos Geram Vários Passos.....	14
Figura 3.7: Estator.....	15
Figura 3.8: Rotor	15
Figura 3.9: Esquema do Motor de Ímã Permanente.....	15
Figura 3.10: Esquema do Motor de Relutância Variável	16
Figura 3.11: Motor <i>Half Step</i> com Duas Bobinas Energizadas.....	18
Figura 3.12: Motor <i>Half Step</i> com Uma Bobina Energizada	18
Figura 3.13: Circuito Utilizando Porta Paralela para Comunicação	19
Figura 3.14: Esquema Lógico ULN2003	20
Figura 3.15: Esquema Interno do ULN2003.....	21
Figura 3.16: ULN2003 Lógico.....	21
Figura 3.17: Camadas do Windows NT.....	23
Figura 3.18: Funções de <i>Hardware</i> Gerenciadas pela HAL	24
Figura 3.19: Trabalho da <i>Inpout32.dll</i>	25
Figura 4.1: Energias Captadas pelo Sensor.....	27
Figura 4.2: Princípio de Funcionamento do Sensor de Presença	29
Figura 4.3: Sensor por Retroreflexão	29

Figura 4.4: Sensor por Transmissão	30
Figura 4.5: Sensor por Reflexão Difusa	30
Figura 4.6: Sensor Infravermelho Passivo	31
Figura 4.7: Esquema do CI 74LS541	31
Figura 5.1: Ilustração do Sistema.....	33
Figura 5.2: Circuito do Controle do Motor de Passo.....	34
Figura 5.3: Esquema de Conexão do Circuito de Controle do Motor	35
Figura 5.4: Criação de Conta no <i>Site DynDns</i>	36
Figura 5.5: Criação de Nome Válido na <i>Internet</i>	36
Figura 5.6: Instalação do Cliente <i>DynDNS</i>	37
Figura 5.7: Instalação do <i>Radmin Server</i>	37
Figura 5.8: Tela de Configuração do <i>Radmin Server</i>	38
Figura 5.9: <i>Software</i> de Controle do Motor	41
Figura 5.10: Circuito do Alarme via <i>E-mail</i>	42
Figura 5.11: Esquema de Conexão ao CI 74LS541	43
Figura 5.12: CI 7805- Regulador de Tensão	43
Figura 5.13: Esquema de Conexão ao CI 7805	44
Figura 5.14: Esquema de Conexão do Alarme via <i>E-mail</i>	44
Figura 5.15: Instalação do Serviço SMTP	45
Figura 5.16: Primeiro Passo de Configuração do Serviço SMTP	46
Figura 5.17: Segundo Passo de Configuração do Serviço SMTP	46
Figura 5.18: <i>Software</i> Sensor.....	48

Lista de Tabelas

Tabela 2.1: Endereçamento de E/S LPT2	6
Tabela 3.1: Estados do Motor de Passo	17
Tabela 3.2: Modos de Operação do Motor de Passo	18
Tabela 3.3: Acionamento Lógico do Motor (<i>Full Step</i>)	19

Lista de Abreviaturas

A - Ampere

ADSL - Assymmetric Digital Subscriber Line, Linha Digital Assimétrica para Assinante

ANATEL - Agência Nacional de Telecomunicações

BIOS - Basic Input/Output System, Sistema Básico de Entrada e Saída

CI - Circuito Integrado

DMA - Direct Memory Access, Acesso Direto à Memória

DNS - Domain Name System, Sistema de Nomes de Domínios

DOS - Disk Operating System, Sistema Operacional de Disco

DLL - Dynamic Link Library, Biblioteca de Ligação Dinâmica

E/S - Entrada e Saída

HAL - Hardware Abstraction Layer, Camada de Abstração de Hardware

Hz - Hertz

IBOPE - Instituto Brasileiro de Opinião Pública e Estatística

M - Mega

n - nano

SMTP - Simple Mail Transfer Protocol, Protocolo Simples de Transferência de E-mail

USB - Universal Serial Bus, Barramento Serial Universal

V - Volts

1 Introdução

Pesquisa realizada pelo Instituto Brasileiro de Opinião Pública e Estatística (IBOPE) em parceria com a *NetRatings*, constatou que 14,4 milhões de pessoas acessaram a rede mundial de computadores de suas residências no ano de 2006. Os brasileiros navegaram 21 horas e 39 minutos na *Internet* residencial durante o mês de dezembro do mesmo ano, ficando a frente de países como França, Estados Unidos, Austrália e Japão (*site idgnow.uol.com.br*).

Ainda em relação ao acesso à tecnologia, o uso do telefone celular no Brasil nos últimos 5 anos cresceu 288%, segundo a Agência Nacional de Telecomunicações (ANATEL). Os aparelhos móveis tornaram-se mais populares e as operadoras oferecem serviços variados além da telefonia, como por exemplo, envio de *e-mail* e acesso à *Internet* (*site anatel.gov.br*).

Paralelo às facilidades de acesso à *Internet* e telefonia móvel os números da violência urbana aumentam consideravelmente em todo país. Esta violência surge na forma de seqüestros, assaltos, latrocínios e furtos.

Segundo o jornal Correio Braziliense, publicado em 16 de março de 2007, a escala da criminalidade no Distrito Federal sobe 2,5% acima do crescimento populacional. Em conseqüência, o número de seguros residenciais cresceu cerca de 30% em 2006 em relação ao ano anterior (*site correioweb.com.br*).

Indo ao encontro do maior acesso à tecnologia, na forma de *Internet* e telefonia móvel, e ainda a crescente necessidade de segurança motivaram o desenvolvimento de um sistema que possa ser utilizado no monitoramento de residências.

1.1 Objetivo Geral

Tem-se como objetivo principal desenvolver mecanismo de segurança residencial com monitoramento visual do ambiente *via Internet* e alarme através de *e-mail*. Além de ser considerado um sistema de segurança, tal projeto também pode ser usado para monitorar as pessoas que já estão no ambiente fazendo alguma atividade, por exemplo, um empregado que presta serviço em determinada residência.

1.1.1 Objetivos Específicos

Este projeto tem como objetivos específicos:

- Desenvolver interface *hardware/software* capaz de controlar o movimento de uma câmera integrada ao motor de passo;
- Integrar a visualização do ambiente ao controle da câmera para o usuário através da *Internet*, e
- Designar uma condição de alarme que quando ativado origine um *e-mail* para um determinado endereço.

1.2 Estrutura da Monografia

- Capítulo 1 (Introdução): Este capítulo visa explicar os objetivos gerais e específicos deste projeto, proporcionando ao leitor uma visão macro do que será feito na monografia e no desenvolvimento físico.
- Capítulo 2 (Interfaces de Comunicação): Esta parte visa definir as portas paralelas e USB, explicar seu modo de operação, modelos e sua utilidade.
- Capítulo 3 (Motores Elétricos): O capítulo 3 tem como foco esclarecer o funcionamento do motor de passo, seus tipos e características. Além disto, pretende explicar ao leitor como se pode ter o controle do motor, levando em consideração modo de operação e componentes adicionais necessários.
- Capítulo 4 (Sensor de Presença): Este tem como idéia principal explicar o funcionamento de alguns tipos de sensores de presença disponíveis no mercado. Além disto, visa explicitar como se pode trabalhar para tratar o sinal gerado pelo componente.
- Capítulo 5 (Produção Técnica): Esta parte visa explicar a construção física do projeto, levando em consideração as conexões, interfaces e códigos. Além disto, cita os resultados obtidos.
- Capítulo 6 (Considerações Finais): Visa explicitar os problemas encontrados no desenvolvimento do projeto e algumas sugestões para futuras evoluções.

2 Interfaces de Comunicação

2.1 Tipos de Transmissão

Um microcomputador não teria utilidade se não existisse a possibilidade de entrar com dados externos para o processamento ou exteriorizar informações. Genericamente chama-se de interfaces de entrada e saída (E/S) os dispositivos capazes de realizar estas tarefas. Existem duas formas básicas para prover a comunicação entre dois dispositivos digitais: comunicação serial e paralela.

A comunicação serial tem a característica de transmissão *bit a bit*. É utilizado um único canal, e como consequência disto há uma transmissão não tão veloz quanto à paralela, porém a transmissão serial praticamente não sofre com problemas relacionados ao ruído. Este é o método de transmissão é utilizado para dispositivos como teclados, *mouses*, *Universal Serial Bus (USB)* entre outros (TORRES, 2001). A Figura 2.1 ilustra este sinal.

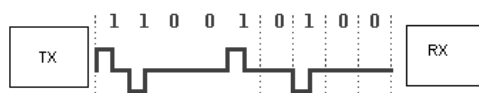


Figura 2.1: Transmissão Serial

(Fonte: TORRES, 2001)

A transmissão paralela tem a capacidade de transmitir todos os *bits* simultaneamente que o transmissor e o receptor são capazes de manipular. Este tipo de transmissão, por exemplo, é utilizado entre o processador e a memória. A Figura 2.2 mostra a transmissão de uma seqüência de *bits*.

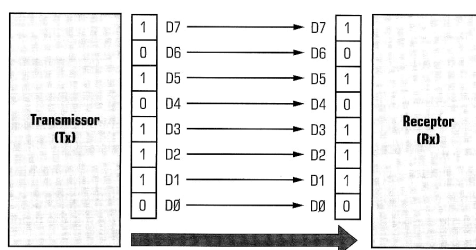


Figura 2.2: Transmissão Paralela

(Fonte: TORRES, 2001)

Devido sua grande disseminação no mercado e vasta quantidade de material de estudo a porta paralela será utilizada neste projeto na comunicação entre os circuitos e o microcomputador. Além desta interface, será utilizada a interface USB para prover a comunicação entre a câmera e o microcomputador.

2.2 Porta Paralela

O funcionamento da porta paralela (também chamada de interface paralela) é relativamente simples. Ela pode ser unidirecional ou bidirecional e está dividida em três modos de operação (TORRES, 2001):

- I. *Standard Parallel Port (SPP)*: Possui máxima taxa de transferência de 150 kB/s (Quilo Bytes por segundo) e as portas paralelas mais antigas só trabalham neste modo. Comunica-se com a Unidade Central de Processamento (CPU) através de um barramento de 8 *bits*. Por padrão este modo trabalha de forma unidirecional, mas habilitando o *modo nibble*¹ a porta trabalhará bidirecionalmente, porém a comunicação será feita a apenas 4 *bits* por vez.
- II. *Enhanced Parallel Port (EPP)*: Este trabalha de forma bidirecional, a comunicação do processador com a porta é feita a 32 *bits* por vez, mas para comunicação com dispositivos externos somente 8 *bits* são utilizados simultaneamente. A taxa de transferência teórica chega a 2 MB/s (Mega Bytes por segundo), entretanto na prática gira em torno de 800 kB/s. Para que a porta paralela trabalhe neste modo é necessário um cabo especial a fim de diminuir o ruído da transmissão.
- III. *Enhanced Capabilities Port (ECP)*: Este é o modo mais avançado e também trabalha de forma bidirecional. A porta neste modo também possui a taxa de transferência de 800 kB/s, mas compacta os dados utilizando do algoritmo *Run Length Encoded* aumentando o desempenho. Este modo ainda utiliza o *Direct Access Memory (DMA)*, assim o processo acontece sem o conhecimento do processador, alcançando melhor desempenho. Da mesma forma

¹ Modo *Nibble* é uma utilização especial da porta paralela no modo SPP que permite realizar leitura de dados de forma bidirecional (TORRES, 2001).

que o EPP, o modo ECP necessita de um cabo com blindagem especial para evitar os ruídos gerados pela alta taxa de transmissão e pela proximidade física dos meios (TORRES, 2001).

2.2.1 Endereçamento de Porta Paralela

O computador nomeia as portas paralelas em LPT1, LPT2, LPT3 e assim por diante. Para a comunicação com essas portas existem alguns endereços de E/S reservados. Para um micro ler ou escrever algum dado utilizando a porta paralela ele deve utilizar o endereço de E/S da porta. Cada endereço está relacionado a uma porta, que por sua vez está ligada ao registrador que a controla. A porta paralela trabalha com três tipos de registradores (*site jobtecltda.com.br*):

- Registrador de dados: Usado para enviar dados através da porta e trabalham com endereços 888 (decimal) ou 378h (hexadecimal);
- Registrador de entradas ou *Status*: Através deste registrador a porta pode receber dados do mundo exterior. Este trabalha utilizando os endereços 889 (decimal) ou 379h (hexadecimal);
- Registrador de controle: A idéia inicial era controlar a impressora, mas também podem ser usados para enviar dados para o mundo externo utilizando os endereços 890 (decimal) ou 37Ah (hexadecimal).

O endereço 378h, no caso da interface LPT1, é considerado o endereço de escrita de dados. Dados escritos neste endereço são enviados através do registrador de dados para os pinos 2, 3, 4, 5, 6, 7, 8 ou 9 da porta paralela (pinos chamados D0 a D7). O endereço seguinte é o 379h que é utilizado para receber dados do mundo exterior. Este endereço recebe dados do Registrador de Entradas que utiliza os pinos 10, 11, 12, 13 e 15 (pinos chamados S3 a S7). Já o endereço 37Ah, na prática, acaba sendo um endereço utilizado para enviar dados para o exterior através do Registrador de Controle utilizando os pinos 1, 14, 16, e 17 (pinos nomeados de C0, C1, C2 e C3 respectivamente). As Figuras 2.3 e 2.4 ilustram os pinos dos conectores DB25.

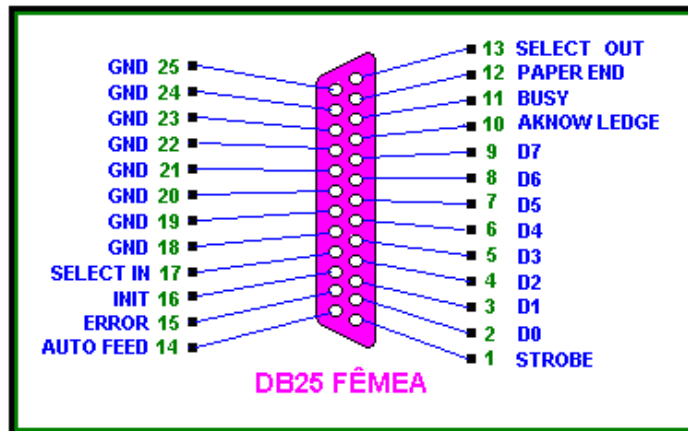


Figura 2.3: Descrição DB25 Fêmea

(Fonte: Site rogercom.com)

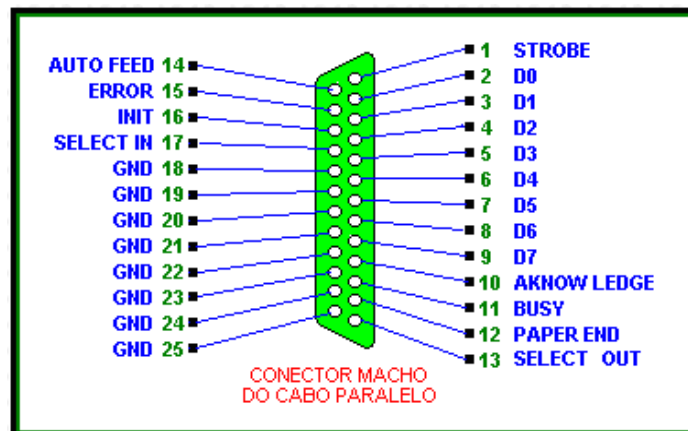


Figura 2.4: Descrição DB25 Macho

(Fonte: Site rogercom.com)

A Tabela 2.1 e a Figura 2.5 ilustram a explicação utilizando o endereço base 278h ([site jobtecltda.com.br](http://site.jobtecltda.com.br)).

Tabela 2.1: Endereçamento de E/S LPT2

(Fonte: VASCONCELOS, 2002)

Porta	Endereço de Dados	Endereço de Entrada	Endereço de Controle
LPT1	278h	279h	27Ah

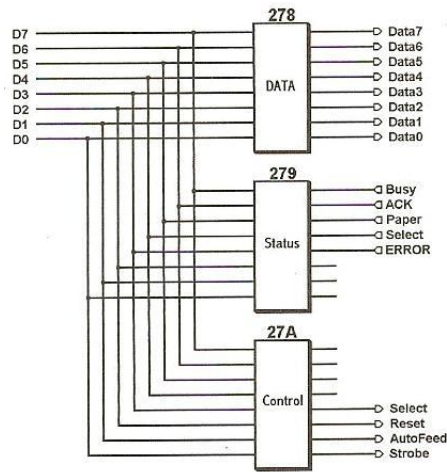


Figura 2.5: Circuito Simplificado da Porta Paralela

(Fonte: VASCONCELOS, 2002)

A Figura 2.6 ilustra o intervalo de endereçamento da porta LPT1 no *Microsoft Windows XP*.

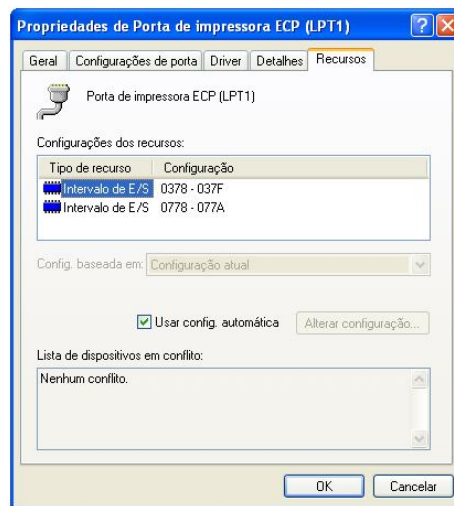


Figura 2.6: Intervalo de Endereçamento LPT1

(Fonte: Autor, 2007)

O item a seguir detalha os tipos de conectores utilizados com a porta paralela.

2.2.2 Conectores

Existem três tipos de conectores que provêm à conexão através da porta paralela, dois destes são os conectores macho e fêmea DB25. O conector DB25

fêmea está localizado geralmente atrás do microcomputador e o macho está conectado a algum dispositivo. As Figuras 2.7 e 2.8 ilustram os conectores DB25 fêmea e macho. Outro conector muito utilizado é conhecido como Centronics, este tem 36 pinos e ainda é utilizado para fazer a conexão com impressoras. A Figura 2.9 exibe o Centronics, além disto, o anexo A lista todos os pinos para o conector DB25 como para o Centronics.



Figura 2.7: Conector DB25 Fêmea

(Fonte: *Site rogercom.com*)



Figura 2.8: Conector DB25 Macho

(Fonte: *Site rogercom.com*)



Figura 2.9: Conector Centronics

(Fonte: *Site rogercom.com*)

Além da porta paralela outra interface utilizada neste projeto é a porta USB. Ela será usada na comunicação entre a câmera e o computador com intuito de prover a imagem do ambiente monitorado ao usuário.

2.3 Porta USB

Cada vez mais os periféricos tais como *mouses*, teclados, impressoras e câmeras são fabricados para o barramento USB. Pode-se conectar até 127 dispositivos diferentes em cada porta USB, basta que seja utilizado um

concentrador de cabos. Todos os dispositivos USB trabalham com comunicação serial e podem ter cabos de conexão com até 5 metros de comprimento. Outra importante característica da porta USB é o fato dela permitir que a alimentação elétrica do dispositivo seja feita pelo cabo de dados (TORRES, 2001).

A primeira versão do barramento USB é conhecida como USB 1.1, versão que utiliza entre 1,5 Mb/s (Mega *bits* por segundo) e 12 Mb/s dependendo do dispositivo conectado. A versão atual do mercado trabalha com até 480 Mb/s e é conhecida como USB 2.0, versão ainda compatível com a primeira versão (TORRES, 2001). A Figura 2.10 ilustra o *plug* padrão USB.



Figura 2.10: Plug Padrão USB

(Fonte: Autor, 2007)

O cabo USB é composto por 4 fios, dos quais 2 fios são utilizados para transportar a energia para o dispositivo, energia que é fornecida e gerenciada pela controladora. O barramento USB pode fornecer no máximo 5V e 500mA de tensão e corrente, respectivamente, para cada porta. Os 2 fios restantes são utilizados para a transferência de dados (*site rogercom.com*). A Figura 2.11 ilustra os fios internos do cabo USB.



Figura 2.11: Fios do Cabo USB

(Fonte: *Site rogercom.com*)

2.4 Considerações Finais

Neste capítulo foram citadas as transmissões paralelas, seriais e suas respectivas definições e funcionalidades. Dentro do âmbito das transmissões paralelas foi citada a porta paralela, interface de grande importância neste projeto. Explicou-se que o endereço de escrita de dados da porta LPT1 é 378h e que os principais pinos de escrita vão de D0 a D7. Além disto, foram citados os modelos existentes no mercado tais como: SPP, EPP e ECP. Outra interface também citada neste segundo capítulo foi a USB, interface que utiliza de comunicação serial que possui alimentação própria e será utilizada na comunicação com a câmera.

Dentro deste projeto outro componente de grande importância é o motor de passo, equipamento que possibilitará o giro da câmera, é o alvo do próximo capítulo.

3 Motores Elétricos

Pode-se chamar motores elétricos as máquinas que transformam energia elétrica em energia mecânica, usualmente disponibilizada num eixo de rotação. A presença da corrente elétrica, seja contínua ou alternada, gera movimento em um eixo, que pode ser utilizado de diversas maneiras dependendo da aplicação do motor (FELIPPO, 2000). As Figuras 3.1 e 3.2 ilustram o motor elétrico.

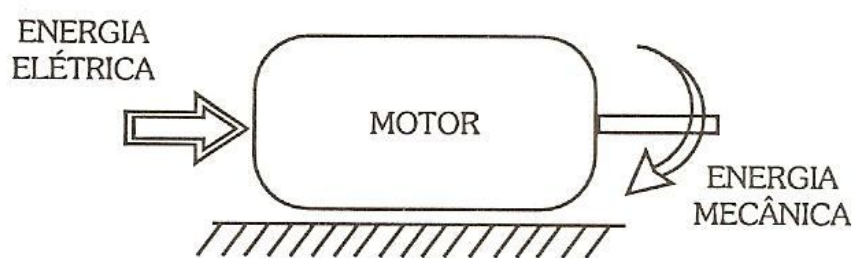


Figura 3.1: Transformação de Energia no Motor

(Fonte: FELIPPO, 2000)



Figura 3.2: Motores Elétricos

(Fonte: Autor, 2007)

Estima-se que no campo dos acionamentos industriais, cerca de 70 a 80% da energia elétrica consumida pelo conjunto de todas as indústrias sejam transformadas em energia mecânica através de motores elétricos. Eles trabalham por fenômenos eletromecânicos ou por eletromagnetismo, como acontece na maioria dos casos (*site eletrônica.org*).

Para entender o funcionamento do motor é importante que alguns conceitos, como do magnetismo seja definido. Pode-se considerar o magnetismo a propriedade que um material possui em atrair metais ferrosos. Esta atração magnética acontece devido ao campo magnético existente ao redor destes materiais. Esta propriedade magnética vem da composição molecular do material. Quando as moléculas do material estão agrupadas compondo um só domínio, esse material possui propriedades magnéticas (CARVALHO, 2006).

Em 1826 o físico francês André Marie Ampère, concluiu que todos os fenômenos elétricos derivam de um princípio único: a ação mútua de correntes elétricas. Este trabalho deu origem à lei de Ampère, sentido convencional e eletrônico para a corrente, e a lei do campo magnético (CARVALHO, 2006).

De acordo com Tipler (2000), experimentalmente é possível observar que: “Quando uma carga elétrica q que tem velocidade \vec{v} num campo magnético \vec{B} , há uma força sobre a carga que é proporcional a q e a velocidade (v), e ao seno do ângulo entre os vetores \vec{v} e \vec{B} . A força é perpendicular a estes dois vetores.” Pode-se resumir este resultado no produto vetorial abaixo:

$$\vec{F} = q \vec{v} \times \vec{B} \quad (3.1)$$

A direção da força vetorial \mathbf{F} é dada pela regra da mão direita conforme a Figura 3.3 ilustra. Quando v gira na direção de B , a direção do vetor \mathbf{F} é a do polegar da mão direita (TIPLER 2000).

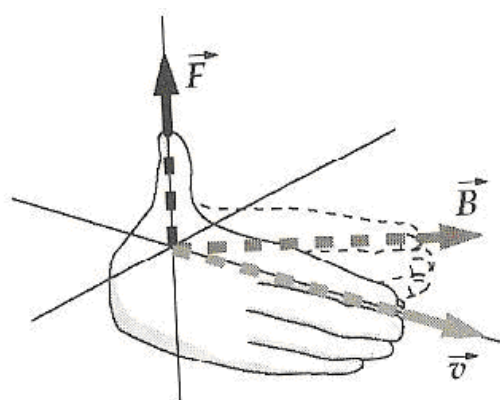


Figura 3.3: Regra da Mão Direita

(Fonte: TIPLER, 2000)

Os motores de corrente contínua e alternada são os tipos mais comuns de motores elétricos, sendo que o motor de corrente alternada ainda é o mais utilizado. Dentre os motores de corrente contínua encontram-se os motores de passos, objeto alvo de estudo do próximo item e de suma importância neste trabalho.

3.1 Motor de Passo

O crescimento da indústria da informática fomentou o aumento da utilização dos motores de passos, pois tais motores fornecem força e precisão que acionam vários dispositivos periféricos de computador. A Figura 3.4 exibe o motor de passo utilizado neste projeto.

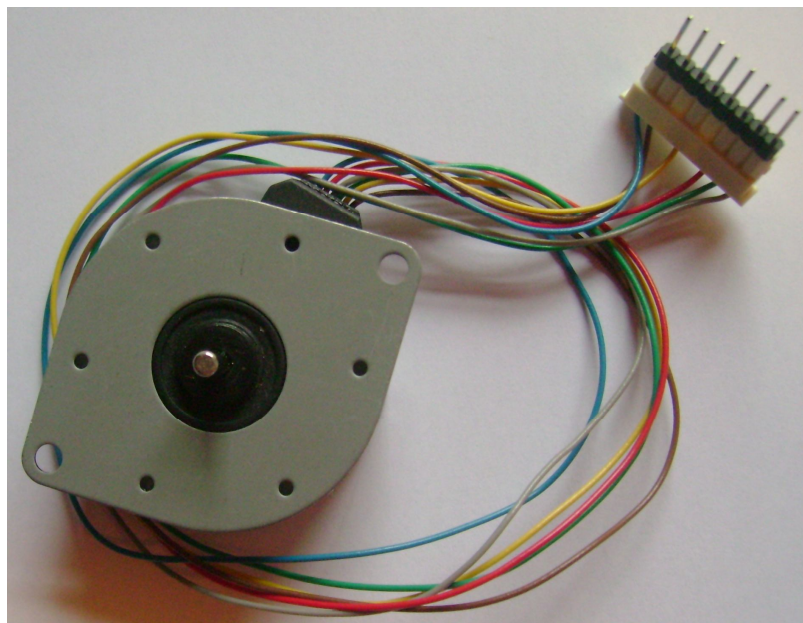


Figura 3.4: Motor de Passo do Projeto

(Fonte: Autor, 2007)

O motor converte energia elétrica em movimento controlado através de pulsos, o que possibilita o deslocamento por passo, onde passo é o menor deslocamento angular possível. Em geral, os motores de passo necessitam entre 2 e 72 passos para completar um giro completo de 360°. As Figuras 3.5 e 3.6 ilustram a relação entre pulsos e passos.

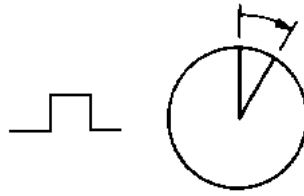


Figura 3.5: Um Pulso Gera um Passo

(Fonte: Autor, 2007)

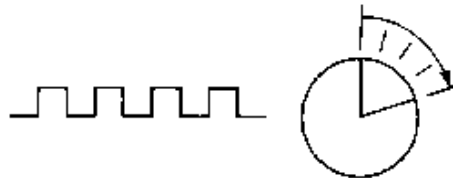


Figura 3.6: Vários Pulsos Geram Vários Passos

(Fonte: Autor, 2007)

Algumas características são comuns a todos os tipos de motores de passo: A velocidade do deslocamento é proporcional à frequência de aplicação dos pulsos lógicos; normalmente o motor tem força resultante igual à zero, sem a necessidade de freio adicional devido ao ímã permanente; e por último, o motor pode girar em velocidade mesmo com a carga, dependendo do torque do motor (CARVALHO, 2006).

Foram citadas algumas propriedades gerais dos motores de passo, porém os motores também apresentam características vinculadas a cada tipo. Os motores de passo estão divididos em motores de passo de ímã permanente ou motores de relutância.

3.1.1 Motor de Passo de Ímã Permanente

O motor de passo de ímã permanente é formado por um rotor, que na verdade é o próprio ímã permanente de alta qualidade (parte móvel do motor), e um estator, este formado de aço silício laminado em que estão instaladas as bobinas que receberão os pulsos. O estator também é caracterizado como a trave fixa onde as bobinas são enroladas. As Figuras 3.7 e 3.8 ilustram o estator e o rotor, respectivamente (CARVALHO, 2006).



Figura 3.7: Estator

(Fonte: *Site eletrônica.org*)

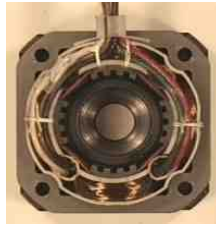


Figura 3.8: Rotor

(Fonte: *Site eletrônica.org*)

De acordo com Carvalho (2006), o motor de ímã permanente, ilustrado na Figura 3.9, possui as seguintes características principais:

- Todas as bobinas podem ser conectadas a um ponto comum;
- Quando o rotor é movimentado manualmente, nota-se que ele tende a agarrar, devido ao ímã permanente; e
- O motor não possui ranhuras.

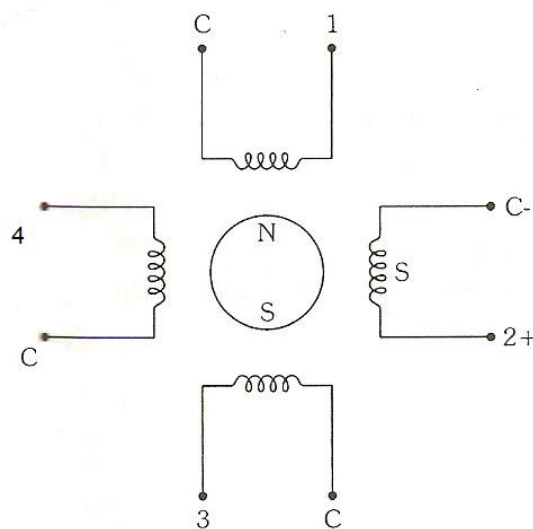


Figura 3.9: Esquema do Motor de Ímã Permanente

(Fonte: CARVALHO, 2006)

O funcionamento do motor de ímã permanente tem o seguinte princípio: Energizar as bobinas uma a uma de modo que elas atraiam o rotor, fazendo que ele gire no mesmo sentido da energização das bobinas (CARVALHO, 2006).

Nota-se na figura que o pólo S da bobina atrairá o pólo N do rotor e repulsará o pólo S, de modo que ocorrerá um passo no sentido horário. Como foi citado também se tem no mercado motores de passo de relutância variável, modelo alvo do próximo item.

3.1.2 Motor de Passo de Relutância Variável

Motores de relutância são formados de um rotor de aço de silício e um estator, este receberá as bobinas que por sua vez formarão o campo magnético ao serem acionadas. O rotor não é um ímã, diferentemente do motor de ímã permanente. A principal característica é que possuem ranhuras com o objetivo de produzir relutâncias² diferentes entre determinadas áreas do componente (CARVALHO, 2006).

A Figura 3.10 ilustra a variação de relutância ao longo do rotor, supondo um motor no qual não foram desenhadas as bobinas entre as bobinas 1 e 2, 2 e 3, 3 e 4, 4 e 1 para simplificar a visualização e entendimento (CARVALHO, 2006).

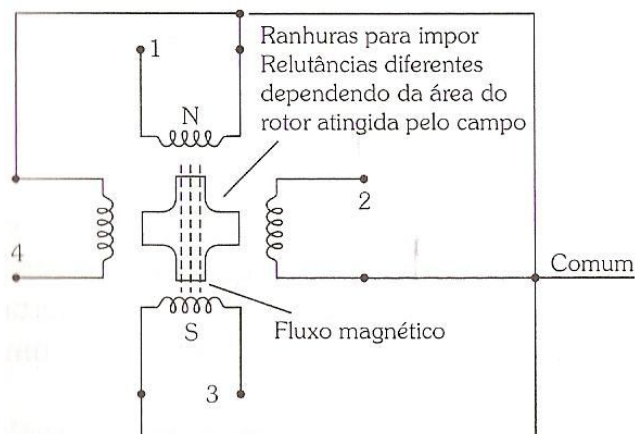


Figura 3.10: Esquema do Motor de Relutância Variável

(Fonte: CARVALHO, 2006)

² De acordo com Carvalho (2006), pode-se considerar relutância como “Oposição oferecida pelo conjunto formador do caminho magnético à passagem do fluxo magnético.” Ela pode ser expressa pela fórmula que divide o comprimento da bobina pelo produto entre área e permeabilidade do meio magnético.

Neste exemplo as bobinas que estão conectadas aos terminais 1 e 3 serão acionadas juntas, assim como as bobinas 2 e 4, formando um fluxo de norte a sul. O fluxo magnético no estator priorizará, como a corrente, a menor relutância. Os dentes no rotor aumentarão a relutância neste setor, pois o ar é o pior condutor magnético. O rotor tende a se posicionar mais adequadamente para a passagem do campo magnético, campo este criado no estator. Os pares de bobinas executam a mesma tarefa (CARVALHO, 2006).

Todos estes motores de passos para terem utilidade devem ser passíveis de controle, e para esta atividade existem componentes que desempenham esta tarefa conforme nos próximos itens.

3.2 Operação do Motor de Passo

É possível caracterizar em três estados e em três modos de operação os motores de passo, conforme as Tabelas 3.1 e 3.2 (*site rogercom.com*).

Tabela 3.1: Estados do Motor de Passo

(Fonte: *Site rogercom.com*)

Estado do motor	Descrição
Desligado	Não há alimentação no motor, não existe consumo de energia e todas as bobinas estão desligadas.
Parado	Pelo menos uma bobina está energizada e o motor permanece parado. Há consumo de energia, mas o motor permanece em uma posição fixa.
Rodando	As bobinas são energizadas em intervalos determinados, proporcionando o giro do motor.

Tabela 3.2: Modos de Operação do Motor de Passo

(Fonte: Site rogercom.com)

Operação	Descrição
Passo Completo 1 (<i>Full Step</i>)	Neste modo somente um bobina por vez é energizada. Isto gera maior velocidade e menor consumo de energia.
Passo Completo 2	Duas bobinas são energizadas simultaneamente a cada passo, gerando maior consumo de energia e maior torque.
Meio Passo (<i>Half Step</i>)	A combinação entre o passo completo 1 e passo completo 2 gera o efeito do motor de operação meio passo. Duas bobinas são energizadas simultaneamente (Figura 3.11) e em seguida somente uma é energizada (Figura 3.12), assim é caracterizado um ciclo. Com isto temos menor velocidade que nos modelos de passo completo, porém com maior precisão. As figuras ilustram o funcionamento do motor <i>half step</i> .



Figura 3.11: Motor *Half Step* com Duas Bobinas Energizadas

(Fonte: Site rogercom.com)



Figura 3.12: Motor *Half Step* com Uma Bobina Energizada

(Fonte: Site rogercom.com)

A forma que o motor irá operar passo depende basicamente do que se deseja controlar e qual o item mais importante: torque, velocidade ou precisão. O tópico seguinte está ligado ao controle do motor de passo.

3.3 Controle do Motor de Passo

Todas as saídas das portas paralelas são lógicas, isto é, quando enviado o *bit* "1" para determinado pino o mesmo terá saída de +5V, exceto os pinos *Select*, *Strobe* e *Autofeed* que possuem lógica invertida, ou seja, colocando *bit* "1" nestes pinos eles apresentarão saída de 0V (*site mrshp.hpg.ig.com.br*).

Levando-se em consideração que os sistemas de acionamento desses motores são digitais têm-se apenas dois níveis lógicos, pode-se montar a Tabela 3.3 para ilustrar a lógica de acionamento (CARVALHO, 2006).

Tabela 3.3: Acionamento Lógico do Motor (Full Step)

(Fonte: CARVALHO, 2006)

Passo	Bobina 1	Bobina 2	Bobina 3	Bobina 4	Valor Decimal
1	1	0	0	0	8
2	0	1	0	0	4
3	0	0	1	0	2
4	0	0	0	1	1

Com intuito de ilustrar a comunicação com dispositivos externos via porta paralela tem-se o exemplo a seguir, ilustrado na Figura 3.13.

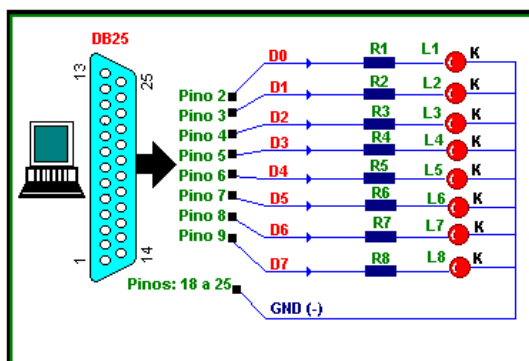


Figura 3.13: Circuito Utilizando Porta Paralela para Comunicação

(Fonte: *Site mrshp.hpg.ig.com.br*)

O objetivo deste experimento é conectar 8 *leds* e acendê-los seqüencialmente conforme comando. Neste exemplo foram utilizados os seguintes componentes: 1 cabo paralelo, 8 resistores de 470 de ohms e 8 *leds* comuns.

Para que o experimento funcione pode-se utilizar um programa compilado que tenha acesso direto ao *hardware*, por exemplo, linguagem C. Este programa definiu a porta LPT1 com o endereço hexadecimal 378h e enviou o valor 10000000 binário para a porta LPT1 para acender o primeiro *led*. Para que o *led* seguinte fosse ativado bastou um deslocamento de *bit*, enviando o valor 01000000, e assim por diante até 00000001 e conseqüentemente o acendimento do último *led*. O anexo B traz na íntegra o código fonte utilizado neste exemplo ([site mrshp.hpg.ig.com.br](http://site.mrshp.hpg.ig.com.br)).

Em circuitos simples, como o citado acima, é possível que dispositivos sejam conectados diretamente à porta paralela do computador, mas em sistemas mais elaborados como temporizadores, controle de relés e motores de passo há a necessidade de um circuito integrado (CI) ou *driver* de potência entre a porta e o dispositivo. Isto é necessário, pois a porta paralela não é capaz de fornecer corrente suficiente para os componentes citados ([site rogercom.com](http://site.rogercom.com)).

Como a porta paralela não fornece a corrente suficiente para o motor (0,4A para o motor deste projeto) usa-se o CI ULN2003 que é composto por transistores *Darlington*. Ele funciona como uma chave, pois assim que detecta o nível de 5V (oriundo da LPT1) o contato entre a fonte de 12V e o motor é fechado. Desta forma a potência da fonte fornece os 0,4A necessários às bobinas do motor. A Figura 3.14 ilustra o conjunto de 7 portas lógicas NOT ligadas aos respectivos diodos e a Figura 3.15 ilustra o esquema interno do CI.

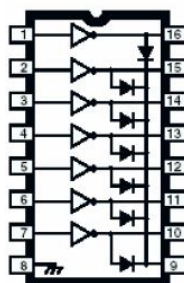


Figura 3.14: Esquema Lógico ULN2003

(Fonte: [Site datasheetcatalog.net](http://Site.datasheetcatalog.net))

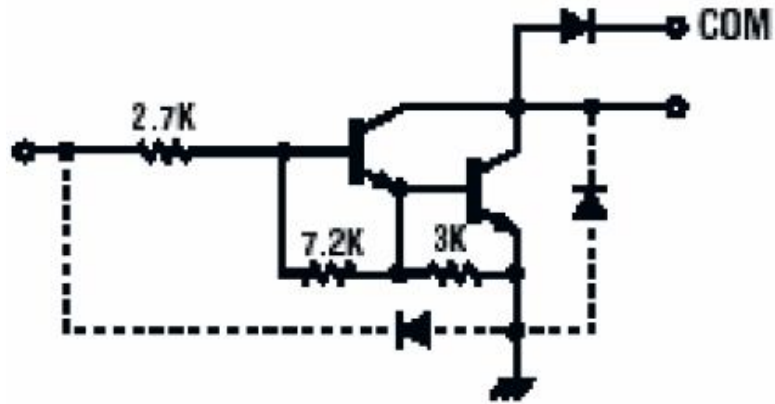


Figura 3.15: Esquema Interno do ULN2003

(Fonte: *Site datasheetcatalog.net*)

O ULN2003 tem 7 entradas que podem controlar até 7 saídas com tensão de saída até 50V e correntes máximas de 500mA, com picos de 600mA em cada canal. Os pinos de 1 a 7 são de entrada, 10 a 16 são pinos de saída, o pino 8 é o terra (0V) e o pino 9 é o positivo (12V) (*site motorpasso.no.sapo.pt*). A Figura 3.16 ilustra o componente.

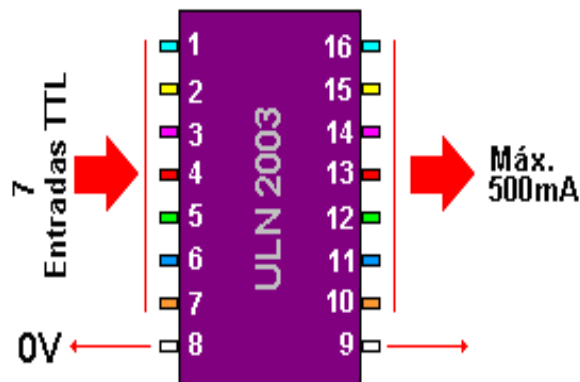


Figura 3.16: ULN2003 Lógico

(Fonte: *Site rogercom.com*)

Neste item vários assuntos relacionados ao controle do motor de passo foram tratados, porém com ênfase a parte física. O próximo tópico abordará pontos relacionados ao controle do motor via *software*.

3.4 Controle via *Software* do Motor de Passo

Desenvolver *softwares* para controlar portas paralelas sobre o antigo *Disk Operating System* (DOS), ou mesmo sobre Windows 95, era uma tarefa um pouco mais simples e direta que produzir os mesmos sistemas para rodarem sobre Windows 2000 ou XP.

Os sistemas operacionais DOS e Windows 95 eram constituídos por um pequeno núcleo, que executavam em *kernel mode* (modo núcleo), acrescido de alguns processos que executavam em *user mode* (modo de usuário). Os processos dos usuários interagiam com os processos do servidor através do modelo cliente-servidor, processo onde o cliente enviava uma requisição para o servidor, que por sua vez realizava o trabalho e retornava o resultado em uma segunda mensagem (TANENBAUM, 1999).

Esta maior simplicidade de acesso ao *hardware*, na prática, acontecia, pois nos sistemas operacionais legados citados existiam a possibilidade de utilizar funções como *Inporb* e *outportb*, ou *_Inp()* e *_Outp()*. Estas funções proporcionam acesso direto ao *hardware* (inclusive à porta paralela), e ainda são muito utilizadas através das linguagens C e C++ (*site logix4u.net*).

Por questões de segurança, desempenho e portabilidade com a chegada dos sistemas operacionais da *Microsoft* baseados na tecnologia NT (*New Technology*), o acesso direto ao *hardware* não foi mais possível para linguagens consideradas alto nível, tais como *Delphi* e *Visual Basic*. Boa parte do sistema operacional, por exemplo, o tratamento de chamadas ao sistema, foi colocado no *kernel mode* (TANENBAUM, 1999).

Um dos objetivos do Windows NT era tornar o sistema operacional portátil, e devido a isto a *Microsoft* ocultou boa parte das dependências de máquina na Camada de Abstração de *Hardware* (HAL). A função dela é oferecer a parte do sistema operacional dispositivos abstratos de *hardware*, desprovidos de especificidades e independentes de máquina, nos quais possam ser utilizados pelo restante do sistema operacional e pelos *drivers*. Dentre os serviços da HAL pode-se citar o endereçamento de dispositivos e acesso aos registradores de dispositivos, inclusive aos da porta paralela. A estrutura do Windows NT é ilustrada na Figura 3.17 (TANENBAUM, 1999).

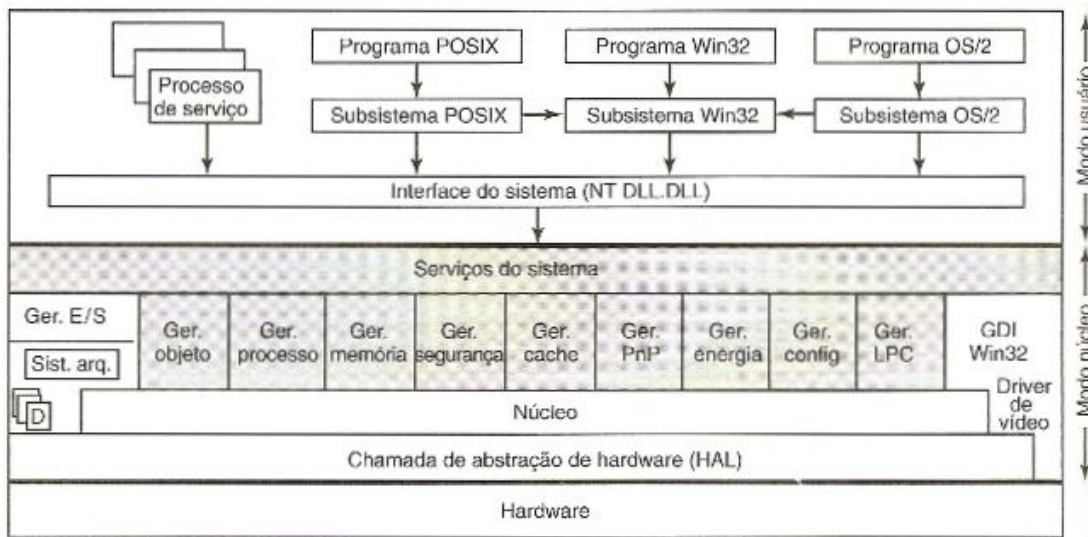


Figura 3.17: Camadas do Windows NT

(Fonte: TANENBAUM, 1999)

A camada de abstração de *hardware* disponibiliza três métodos de leitura e três métodos de escrita para os desenvolvedores de *drivers* usarem no acesso aos registradores dos dispositivos (TANENBAUM, 1999):

- READ_PORT_UCHAR(port) e WRITE_PORT_UCHAR(port,uc);
- READ_PORT_USHORT(port) e WRITE_PORT_USHORT(port,us) e
- READ_PORT_ULONG(port) e WRITE_PORT_ULONG(port,ul).

Além das funções citadas, a camada de abstração de *hardware* é responsável por outros serviços, ilustrados na Figura 3.18, e que são citados abaixo:

- Identificar dispositivos mapeando seus endereços de barramento em um endereço lógico válido no sistema;
- Controlar as interrupções;
- Gerenciar as transferências do DMA;
- Administrar relógios e temporizadores;
- Sincronizar utilização de recursos; e
- Informar e inspecionar a *Basic Input/Output System* (BIOS) sobre configurações dos barramentos e dispositivos de E/S.

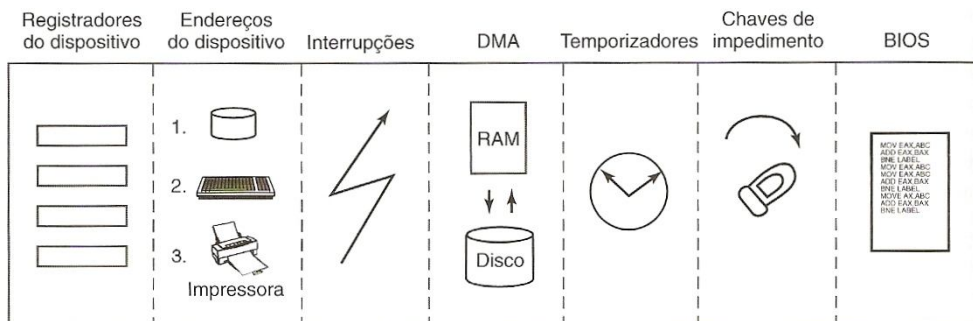


Figura 3.18: Funções de *Hardware* Gerenciadas pela HAL

(Fonte: TANENBAUM, 1999)

De acordo com Figura 3.17, acima da camada de abstração de *hardware* encontram-se o núcleo, ou *kernel*, juntamente como os *drivers* de dispositivos. O objetivo do núcleo é tornar o sistema operacional independente do *hardware*. Para isso ele indica onde o HAL deve permanecer, provê acesso ao *hardware* através da HAL e arquiteta os serviços para implementar abstrações em um nível mais alto (TANENBAUM, 1999).

Acima do núcleo está a parte chamada de executivo, camada está escrita em linguagem C. O executivo é formado por dez componentes que são divididos em gerenciadores de: Objeto, E/S, processos, memória, segurança, cache, *plug and play*, energia, configuração e gerenciador de chamadas de procedimento local. Outra parte muito importante nesta camada da estrutura são os *drivers* de dispositivos. *Drivers* são componentes que atuam no modo núcleo que tratam dos requerimentos das camadas superiores provendo o acesso ao *hardware* (TANENBAUM, 1999).

Ainda analisando a Figura 3.17, percebe-se que o Windows baseado na tecnologia NT é composto de componentes no modo núcleo e componentes no modo usuário. Os componentes do modo usuário são os subsistemas do ambiente, processos de serviço e principalmente as *Dynamic-Link Library* (DLLs). Esses elementos trabalham em conjunto para proporcionar a cada processo de usuário uma interface distinta da interface de chamadas ao sistema do Windows (TANENBAUM, 1999).

Um *software* pode ser desenvolvido para a especificação Win32 e ser executado sem alterações em todas as versões do Windows. Para isto cada DLL agrupa um conjunto de procedimentos de biblioteca, todas em um arquivo com extensão DLL. Quando uma aplicação é acionada, as DLLs necessárias são

localizadas, seja na memória ou em disco, e mapeadas no espaço de endereçamento virtual.

Como já foi explicado, o *software* não possui acesso direto ao *hardware*, nas versões atuais do *Microsoft Windows*. A solução para este projeto foi à utilização da DLL *Inpout32* para prover tal acesso. Dentre as principais características da *Inpout32.dll* pode-se citar (*site logix4u.net*):

- Trabalha sobre os sistemas operacionais Windows XP, NT, 2000 e 2003;
- Não necessita de outros *softwares* adicionais;
- O *driver* de comunicação com o *kernel* é automaticamente instalado quando a DLL é carregada; e
- Apenas as instruções *Inp32* e *Out32* são utilizadas para operação.

O funcionamento da DLL acontece da seguinte forma: O passo inicial da DLL é verificar a versão do sistema operacional quando as funções são chamadas. Caso seja Win9x a DLL usará a função *_inp* e *_outp* para leitura e escrita, respectivamente, da porta paralela. Caso o sistema operacional seja NT, 2000 ou XP a DLL utilizará o *driver hwinterface.sys*. O passo seguinte é receber os dados para e enviá-los à porta paralela (*site logix4u.net*). A Figura 3.19 ilustra o processo citado.

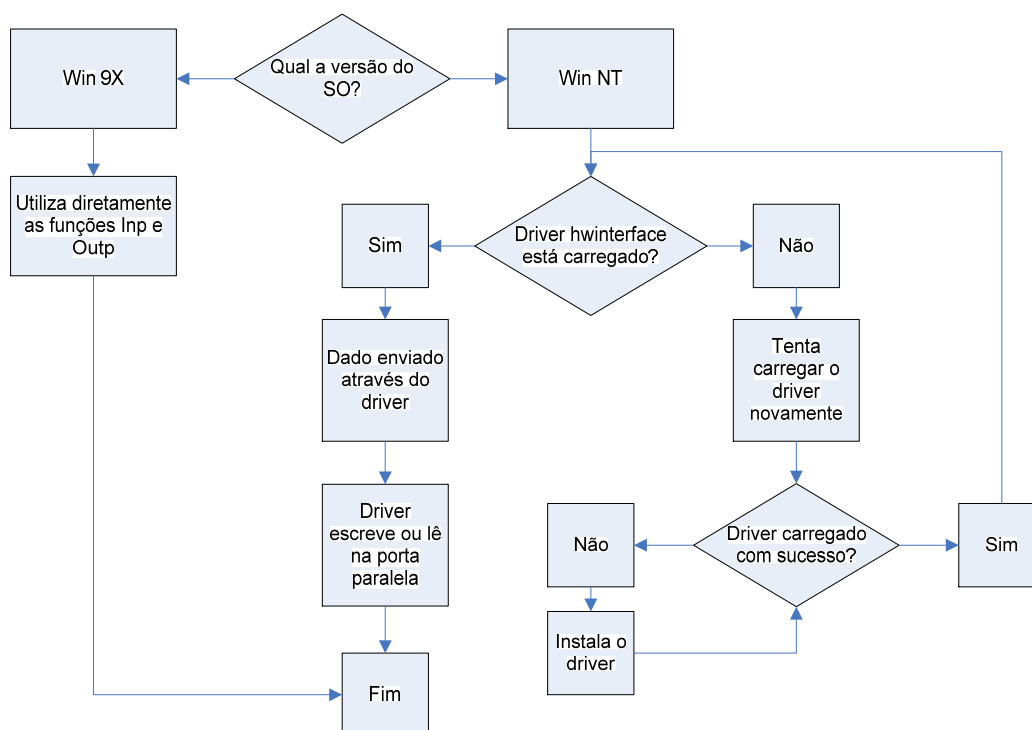


Figura 3.19: Trabalho da *Inpout32.dll*

(Fonte: *Site logix4u.net*)

Além das funções *Inp32()* (utilizada para ler dados na porta especificada) e *Out32()* (utilizada para enviar dados), existem outras funções incorporadas pela DLL utilizada neste projeto (*site logix4u.net*):

- *Dllmain()*: Utilizada quando a DLL é carregada para verificar a versão do sistema operacional e em seguida carregar o *driver hwinterface.sys* se necessário.
- *Closedriver()*: Chamada antes do descarregamento do *driver*.
- *Inst()*: Extrai o arquivo *hwinterface.sys* de binário para a pasta `\systemroot\system32\drivers`.
- *SystemVersion()*: Utilizada para verificar qual sistema operacional é utilizado.

O já citado *driver hwinterface.sys* contém três funções já implementadas (*site logix4u.net*):

- *DriverEntry()*: Ela é chamada quando o *driver* é carregado e cria os objetos *device object* e *symbolic links*.
- *HwinterfaceUnload()*: Chamada quando o *driver* é descarregado para fazer a limpeza da memória.
- *HwinterfaceDeviceControl()*: Estabelece um processo de sincronia com a porta.

3.5 Considerações Finais

O capítulo 3 focou-se em analisar e explicar os conceitos relacionados aos motores de passo. Citou os modelos de relutância variável e imã permanente, e suas respectivas formas de trabalho, além disto, explicou os modos de operação dos mesmos. Este capítulo atentou-se em explanar o CI ULN2003, principal componente no controle do motor de passo através da porta paralela responsável em proporcionar a interface física entre o microcomputador e o motor. Para fazer a interface entre o *kernel mode* e *user mode* foi utilizada a DLL *inpout32*, componente este também detalhado no decorrer deste capítulo. Outro objetivo deste projeto é o monitoramento do ambiente, atividade esta que será feito pelo sensor de presença, foco do próximo capítulo.

4 Sensores de Presença

Um dos principais elementos presentes em processos de automação é o sensor. O termo é designado para definir dispositivos sensíveis a alguma forma de energia no ambiente que pode ser térmica, luminosa, sonora ou cinética. Seguindo esta linha, neste projeto está incluso um sensor que monitorará o ambiente quanto à presença de pessoas (ALBUQUERQUE, 2005).

Geralmente os sensores não possuem características elétricas necessárias para serem utilizados em sistemas de controle, assim nestes casos o sinal de saída deve ser tratado antes da sua leitura no sistema. Este tratamento pode ser feito por circuitos de interface para produção de um sinal que possa ser lido (ALBUQUERQUE, 2005). A Figura 4.1 ilustra as formas de energia possivelmente captadas por um sensor.

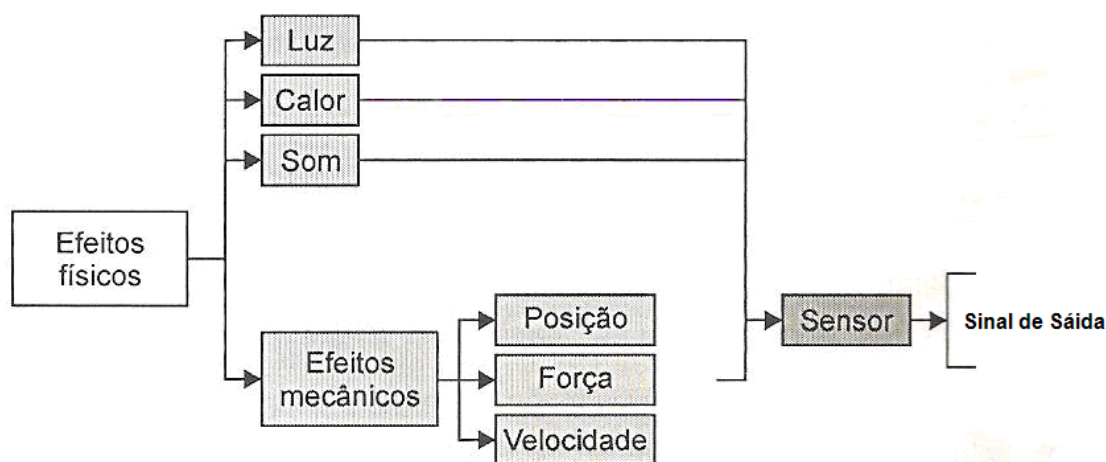


Figura 4.1: Energias Captadas pelo Sensor

(Fonte: ALBUQUERQUE, 2005)

Atualmente no mercado existem os sensores analógicos e digitais. Os analógicos, como o nome sugere, podem assumir qualquer valor no seu sinal de saída ao longo do tempo, desde que dentro do escopo de operação. Este tipo de sensor é útil ao trabalhar com grandezas como pressão, temperatura, umidade e força. Já os sensores digitais podem trabalhar apenas com dois valores e são muito utilizados em detecção de passagem de pessoas e determinação de velocidade (ALBUQUERQUE, 2005).

Um componente importante relacionado aos sistemas sensoriais é o transdutor. Segundo Albuquerque (2005), pode-se definir transdutor como: “Dispositivo completo, que contém um sensor, usado para transformar uma grandeza qualquer em outra que pode ser utilizada nos dispositivos de controle”. Ainda de acordo com Albuquerque (2005), estes componentes transformam os valores de uma grandeza física em sinal de corrente ou tensão que podem ser interpretados por um sistema.

Outro item vinculado aos sistemas sensoriais é o transmissor. Ele funciona como um dispositivo que formata o sinal de saída de um transdutor de acordo com padrões definidos. Os padrões mais utilizados para sinais analógicos são: 3 a 15 PSI; 4 a 20mA; 0 a 20mA e 0 a 10V. Protocolos conhecidos como *HART*, *Asi*, *Fieldbus Foundation*, *PROFIBUS-PA*, *Modbus*, *CAN*, *LonWorks* e *Industrial Ethernet* são os mais utilizados na comunicação entre transmissores e atuadores³ (ALBUQUERQUE, 2005).

Existem várias características que devem ser levadas em consideração no momento da escolha de um sensor, tais como: tipo de saída (pode ser analógica ou digital), sensibilidade, exatidão, alcance, estabilidade e velocidade de resposta.

4.1 Características Específicas dos Sensores de Presença

Os sensores de presença são capazes de detectar a presença de alguém ou algo em determinado ambiente. Este processo pode ser feito utilizando sensores ópticos. O funcionamento dos sensores ópticos está baseado na existência de um emissor e um receptor, que podem, ou não, estar no mesmo objeto. A luz gerada pelo emissor deve alcançar o receptor com intensidade suficiente para fazer com que o sensor comute sua saída (ALBUQUERQUE, 2005).

O sinal de luz gerado pelo emissor é modulador numa frequência, já o receptor do sinal é acoplado a um filtro que somente considera sinais com a mesma frequência do emissor (ALBUQUERQUE, 2005). A Figura 4.2 ilustra a estrutura genérica de um sensor de presença com emissor e receptor em corpos separados.

³ Segundo Albuquerque (2005), pode-se definir atuadores como: “Dispositivos que modificam uma variável controlada. Eles recebem um sinal proveniente do controlador e agem sobre o sistema controlado”.

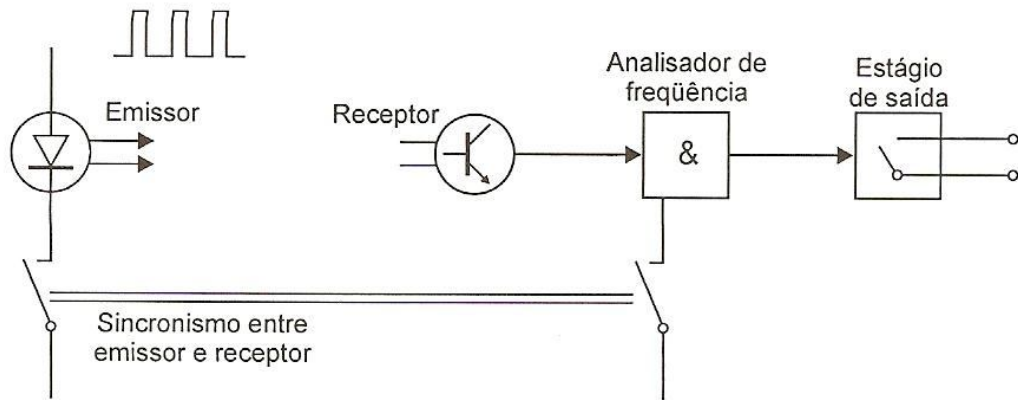


Figura 4.2: Princípio de Funcionamento do Sensor de Presença

(Fonte: ALBUQUERQUE, 2005)

Dentro do grupo dos sensores ópticos existem os sensores ópticos por retroreflexão. Neste modelo de sensor o emissor e receptor se encontram no mesmo corpo. Um feixe de luz é estabelecido entre o emissor e o receptor por intermédio de algum refletor, assim o sensor é ativado quando o objeto interrompe o feixe. A Figura 4.3 ilustra o funcionamento deste tipo de sensor (ALBUQUERQUE, 2005).

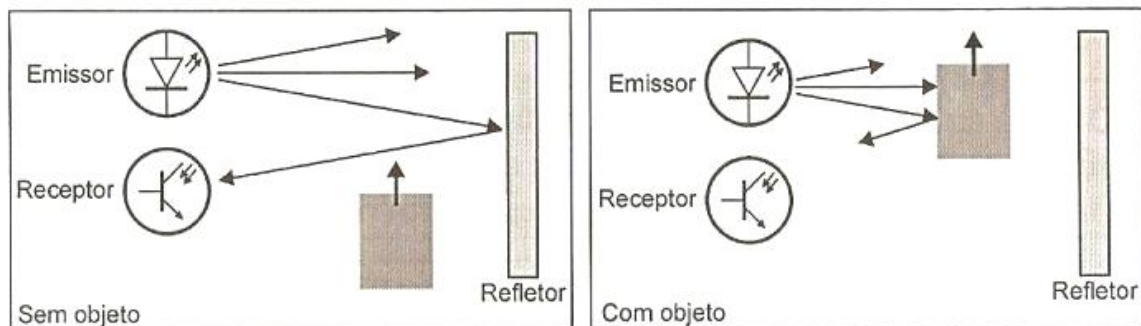


Figura 4.3: Sensor por Retroreflexão

(Fonte: ALBUQUERQUE, 2005)

Outro tipo de sensor óptico conhecido é o sensor por transmissão, ou sensor óptico de detecção por barreira de luz, como também é conhecido. Ele possui emissor e receptor montados em dispositivos separados, diferentemente do sensor óptico por retroreflexão. Ao serem alinhados, o emissor e receptor criam uma barreira de luz entre si, e a presença de um objeto interrompendo essa barreira faz com que o sensor seja ativado. A Figura 4.4 ilustra o sensor não

ativado, e também um objeto interrompendo a barreira de luz entre emissor e receptor (ALBUQUERQUE, 2005).

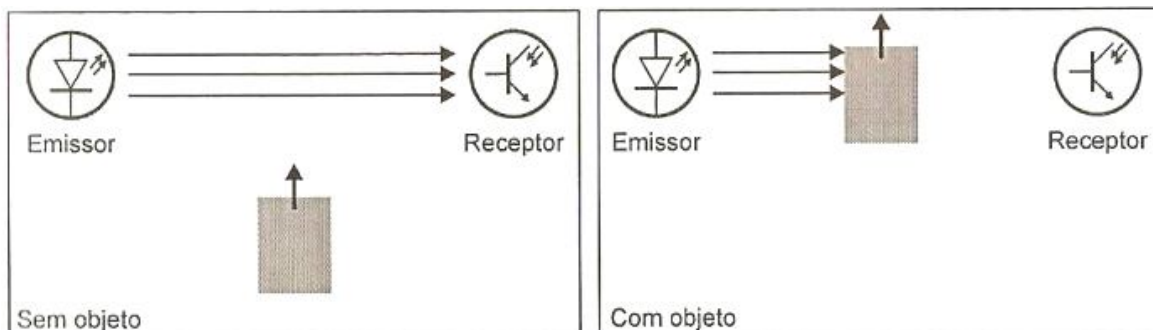


Figura 4.4: Sensor por Transmissão

(Fonte: ALBUQUERQUE, 2005)

Um modelo de sensor óptico muito utilizado, além dos já citados, é o sensor que trabalha por reflexão difusa. Este tipo de equipamento, ilustrado na Figura 4.5, possui o emissor e o receptor montados no mesmo dispositivo. A luz emitida pelo emissor cria uma região ativa cuja presença de algum objeto ou pessoa faz com que a luz refletida de forma difusa para o receptor, ativando assim o sensor (ALBUQUERQUE, 2005).

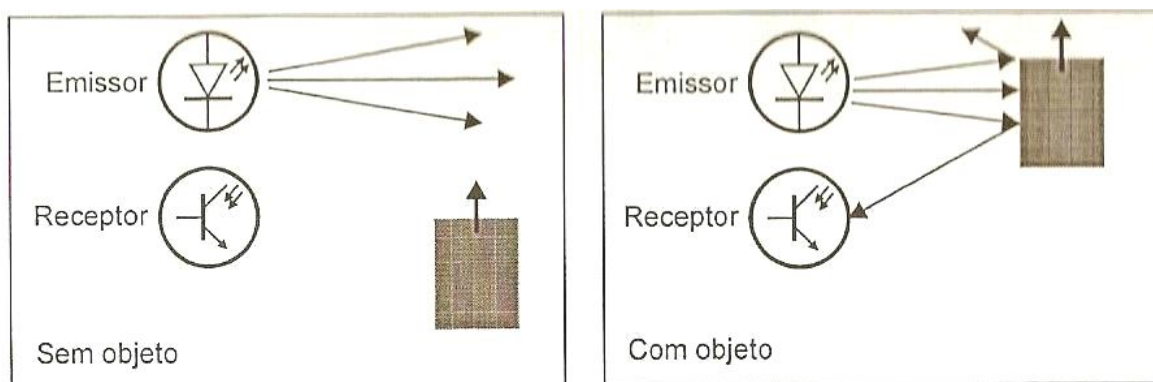


Figura 4.5: Sensor por Reflexão Difusa

(Fonte: ALBUQUERQUE, 2005)

Além de todos os sensores citados, pode-se citar os sensores infravermelhos ativos e passivos. Os ativos possuem o mesmo princípio do sensor de barreira e são utilizados principalmente em aplicações como alarmes e detecção de intrusão. Já o infravermelho passivo trata-se de um receptor de

sinais onde o movimento é detectado através da captação de transmissões de infravermelho que passam pela tela frontal, refletidas diretamente no detector, tendo um circuito eletrônico que conta os pulsos de transmissão e efetua o disparo. Também são muito utilizados em sistemas de alarmes e geralmente possuem ajuste de sensibilidade. O sensor infravermelho passivo normalmente fechado utilizado neste projeto está ilustrado na Figura 4.6 e detalhado no Anexo D.



Figura 4.6: Sensor Infravermelho Passivo

(Fonte: Autor, 2007)

Um dos objetivos deste projeto é designar uma condição de alarme que quando ativado origine um *e-mail* para um determinado endereço. Para que tal objetivo seja alcançado a porta paralela também será utilizada para receber as informações do sensor quanto a presença de pessoas no ambiente monitorado. Isto será possível com ajuda do CI 74LS541. Este componente que necessita de alimentação de 5V, atuará como *buffer* e protegerá os pinos de leitura da porta paralela das correntes excessivas (*site rogercom.com*). A Figura 4.7 ilustra o esquema interno do CI.

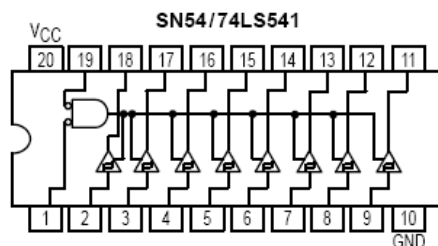


Figura 4.7: Esquema do CI 74LS541

(Fonte: Site datasheetcatalog.net)

4.2 Considerações Finais

Este capítulo descreveu as características dos sensores, dentre elas pode-se destacar o fato que os tais componentes não possuem características para serem utilizados em sistemas de controle de forma direta, logo há a necessidade de uma interface para tratamento do sinal. Na busca deste tratamento do sinal recebido foi citado o CI 74LS541, componente que atuará como um *buffer*, proporcionando a camada física para o tratamento do sinal recebido do sensor.

O próximo capítulo tratará da implementação do projeto, explicará as conexões dos circuitos, códigos utilizados e demais assuntos relacionados.

5 Produção Técnica

O projeto Segurança Residencial: Monitoramento Visual do Ambiente via *Internet* com Alarme via *e-mail*, funcionará da seguinte forma: Dentro do ambiente monitorado haverá um sensor de presença, componente que através da interface paralela enviará um sinal. Este sinal será processado pela interface *hardware/software* que remeterá um alerta caso o ambiente seja invadido. O usuário ao receber este *e-mail*, preferencialmente pelo celular, poderá, via *Internet*, acessar a console que está localizada em sua residência. Esta console proverá controle da câmera instalada no ambiente utilizando um motor de passo que será controlado através da porta paralela. A Figura 5.1 ilustra a montagem do sistema.

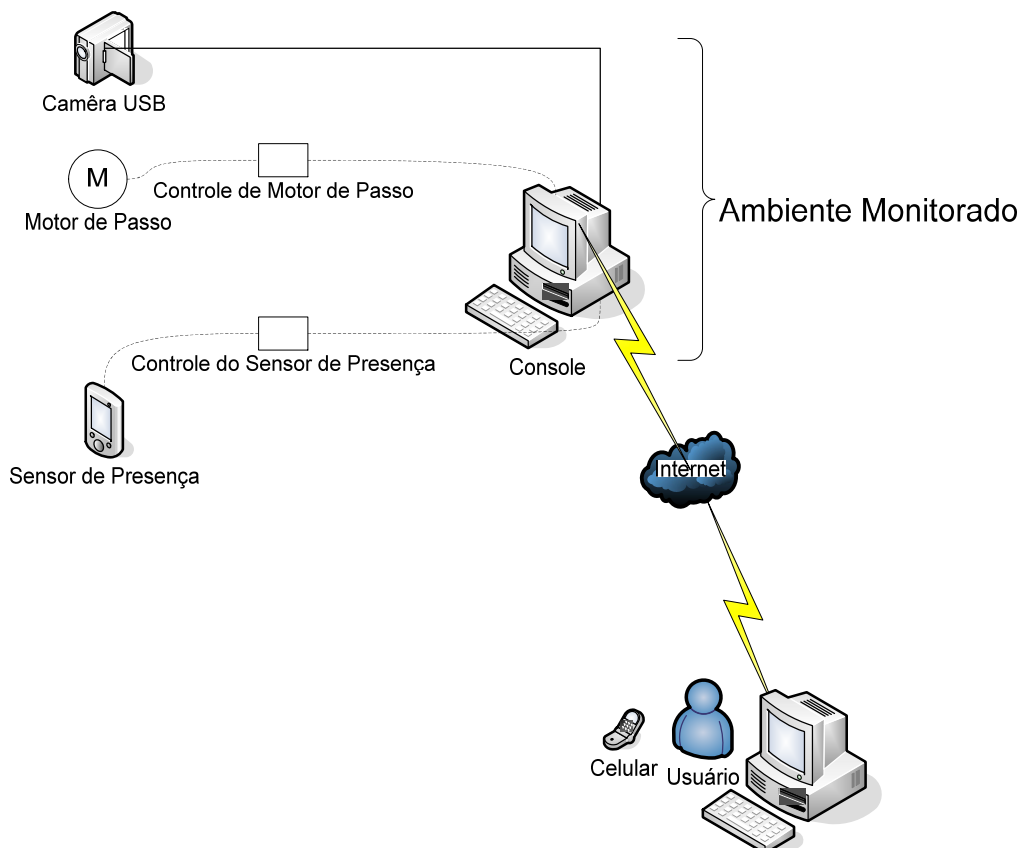


Figura 5.1: Ilustração do Sistema

(Fonte: Autor, 2007)

Para facilitar à compreensão, a descrição dos sistemas será dividida em duas partes: controle da câmera e alarme via *e-mail*.

5.1 Montagem da Infra-Estrutura do Controle da Câmera

O circuito de controle da câmera/motor, ilustrado na Figura 5.2, conta com um *proto-board* com capacidade para 840 pinos, utilizado nos dois circuitos, além dos seguintes componentes:

- Um transistor de potência modelo ULN2003;
- Um diodo Zener 0,5W 12V;
- Um motor de passo unipolar de 0,4A, 12V e passo de 7,5 ° (Informações adicionais no Anexo C); e
- Fonte 12V, 3A e 60Hz de comum uso aos dois circuitos.

Além dos componentes citados, foi utilizada uma câmera USB modelo LG Webpro LIC-200 e para a comunicação do circuito com o microcomputador foi utilizada a porta paralela em modo ECP, além do cabo padrão LPT1.

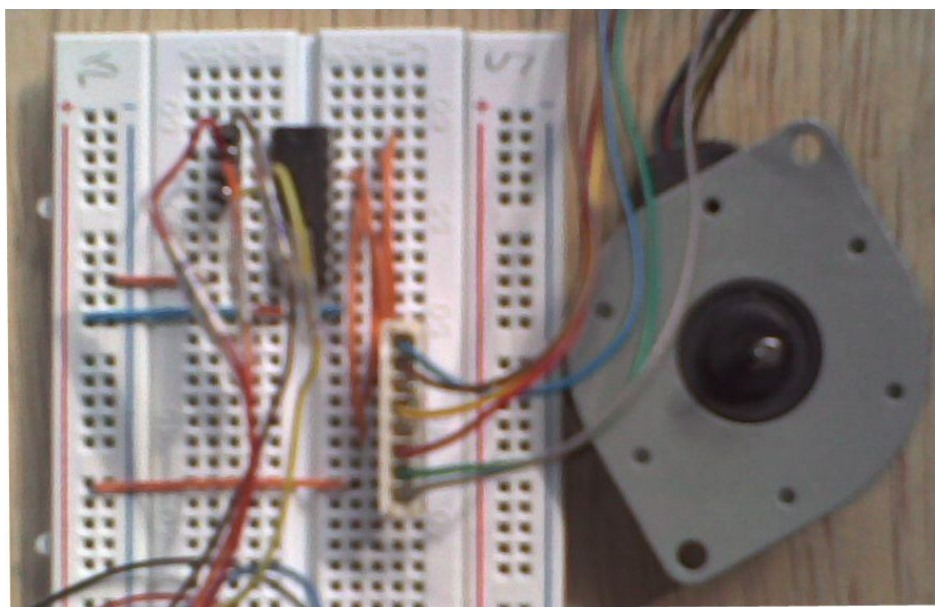


Figura 5.2: Circuito do Controle do Motor de Passo

(Fonte: Autor, 2007)

Neste circuito foram utilizados os pinos 1, 2, 3, e 4 do ULN2003 para prover a conexão com microcomputador, que por sua vez utilizou os pinos 2, 3, 4 e 5 da porta paralela, chamados de D0, D1, D2 e D3, respectivamente. O pino 25 da porta paralela, bem como o pino 8 do ULN2003 foi utilizado para o aterramento.

Para a comunicação com o motor de passo utilizou-se os pinos 16, 15, 14 e 13 do transistor, que por sua vez foram conectados às bobinas do motor.

A fonte de alimentação utilizada possui 12V de saída, tensão necessária pelo motor de passo. Como o ULN2003 também é alimentado com 12V, apenas foi inserido um diodo Zener 12V 0,5W para fins de proteção do CI. A Figura 5.3 ilustra o esquema da montagem deste circuito.

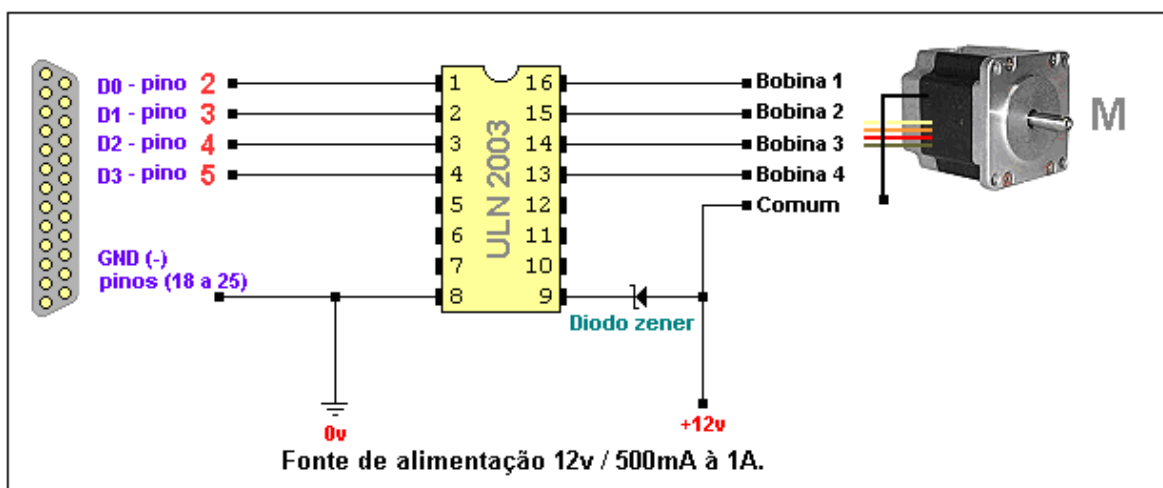


Figura 5.3: Esquema de Conexão do Circuito de Controle do Motor

(Fonte: Site rogercom.com)

Além dos componentes eletrônicos citados foram utilizados dois *softwares* a fim de disponibilizar o acesso ao sistema via *Internet*. A forma escolhida foi a utilização do serviço conhecido como *Dynamic Domain Name System* e emprego do *software Radmin* para prover o acesso remoto.

Domain Name System (DNS) é um serviço de resolução de nomes utilizado para fazer o mapeamento entre o endereço *Internet Protocol* (IP) e o nome de um determinado *host*. Sempre que um *host* é conectado à *Internet* ele recebe um endereço IP dentro de um escopo de operação do provedor, exceto nos planos empresariais que oferecem IP fixo. Com isso, sempre que a conexão for realizada muito provavelmente o IP fornecido será diferente do IP que havia sido entregue na última conexão.

O cenário citado dificulta a conexão remota via *Internet* entre computadores e visando contornar a situação citada surgiu o *DynDNS*. Este sistema gratuito está disponível no site *dyndns.com* e tem como objetivo fazer que sempre que um computador conecte-se à *Internet* possua continuamente o mesmo DNS,

independente do IP que receba, facilitando o acesso remoto, por exemplo. É válido informar que existem outros *softwares* similares ao *DynDns* que poderiam atuar neste projeto.

Para que o *DynDns* fosse empregado neste projeto bastou a criação de uma conta, através do *site*, e cadastramento do DNS *monitoramento.dnsalias.net* para a estação de trabalho que contém a console de gerenciamento da câmera. Além disto, foi necessário a instalação do cliente *DynDnsUpdater*, componente responsável em enviar as atualizações do endereço IP para o *site*, que por sua vez replicará as alterações quando ocorrerem. As Figuras 5.4, 5.5 e 5.6 ilustram o processo de instalação do *DynDNS*.

Create Your DynDNS Account

Please complete the form to create your free DynDNS Account.

- User Information

Username:	<input type="text" value="winiciusf"/>	
E-mail Address:	<input type="text" value="winiciusf@gmail.com"/>	Instructions to activate your account will be sent to the e-mail address provided.
Confirm E-mail Address:	<input type="text" value="winiciusf@gmail.com"/>	
Password:	<input type="password" value="●●●●●●"/>	Your password needs to be more than 5 characters and cannot be the same as your username. choose a password that is a common word, or can otherwise be easily guessed.
Confirm Password:	<input type="password" value="●●●●●●"/>	

Figura 5.4: Criação de Conta no Site DynDns

(Fonte: Site dyndns.com)

Hostname:	<input type="text" value="monitoramento"/> . <input type="text" value="dnsalias.net"/>	
Wildcard:	<input checked="" type="checkbox"/> Yes, alias "*.hostname.domain" to same settings.	
Service Type:	<input checked="" type="radio"/> Host with IP address <input type="radio"/> WebHop Redirect <input type="radio"/> Offline Hostname	
IP Address:	<input type="text" value="200.181.83.64"/> Use auto detected IP address 200.181.83.64. TTL value is 60 seconds. Edit TTL.	

Figura 5.5: Criação de Nome Válido na Internet

(Fonte: Site dyndns.com)

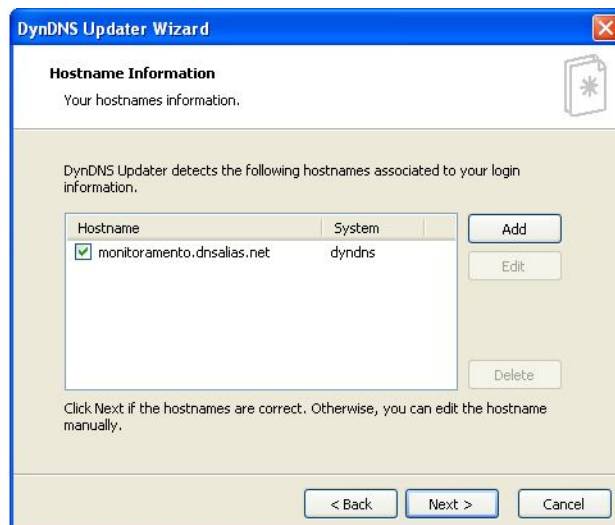


Figura 5.6: Instalação do Cliente *DynDNS*

(Fonte: Autor, 2007)

A partir deste ponto que a estação que possui a console já detém um nome válido na *Internet*, o objetivo passou a ser a instalação e configuração do *software* de acesso remoto *Radmin 3 Remote Control*. O *software* foi desenvolvido pela empresa *Famatech* e está disponível para *download* em versão *trial* através do *site* www.radmin.com. Ressalta-se que outros *softwares* que provêm o acesso remoto poderiam ser utilizado ao invés do *Radmin*.

O *software* possuiu dois arquivos de instalação, sendo um para a máquina a ser acessada e outro para o cliente que fará o acesso remoto. O arquivo versão *server* foi instalado na estação que possui a console e a Figura 5.7 ilustra a instalação.

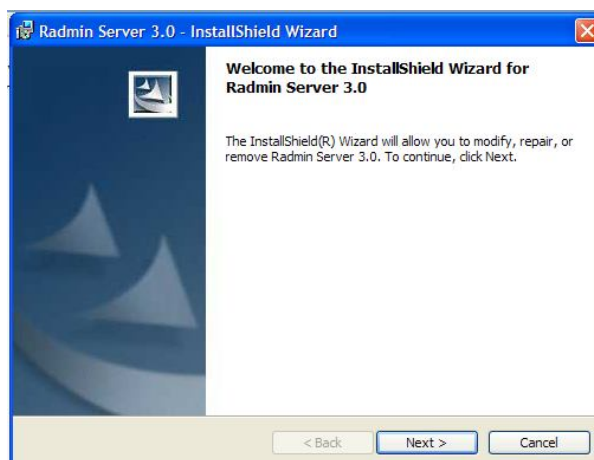


Figura 5.7: Instalação do *Radmin Server*

(Fonte: Autor, 2007)

A Figura 5.8 ilustra a tela de configuração do *software*. A partir dela pode-se configurar o modo inicialização do serviço, porta a ser utilizada (4899 neste projeto) e quais usuários terão acesso de conexão à máquina. Neste projeto a permissão foi concedida ao usuário local chamado *uniceub*. Importante ressaltar a necessidade de instalação do *Radmin* versão *client* na estação que fará o acesso remoto à máquina que possui a console de controle da câmera e visualização do ambiente.



Figura 5.8: Tela de Configuração do Radmin Server

(Fonte: Autor, 2007)

Até este ponto do projeto todas as informações relacionadas à infraestrutura do controle do motor de passo e da câmera foram relatadas. O próximo item tratará do *software* de controle do motor de passo e captação da imagem através da câmera.

5.2 Ambiente de Desenvolvimento do Controle da Motor/Câmera

O ambiente de desenvolvimento e programação utilizado para a elaboração deste projeto é baseado em linguagem *Visual Basic* e compilador *Visual Studio 6 SP6* desenvolvido pela *Microsoft Corporation*. O *Visual Basic* foi a linguagem escolhida para desenvolvimento da interface *hardware/software*, pois possui extensa documentação sobre o controle da porta paralela e boa integração com a DLL *Inpout32*.

A partir deste ponto é possível comentar o código implementado neste projeto na busca do controle do motor de passo/câmera e visualização do ambiente. O projeto PrjLPT1, responsável pelo controle do motor e visualização do ambiente (arquivo .VBP), possui o módulo PrjLPT1.bas e um formulário PrjLPT1.frm. O módulo PrjLPT1.bas tem como objetivo instanciar a DLL *inpout32.dll* e está transcrito abaixo:

‘Módulo PrjLPT1.bas do projeto PrjLPT1.vbp:

```
Public Declare Function Inp Lib "inpout32.dll" _
Alias "Inp32" (ByVal PortAddress As Integer) As Integer
Public Declare Sub Out Lib "inpout32.dll" _
Alias "Out32" (ByVal PortAddress As Integer, ByVal Value As Integer)
```

O código do arquivo PrjLPT1.frm tem objetivo de capturar a imagem oriunda da câmera USB e juntamente com a DLL *Inpout32* controlar o motor de passo. As principais funções utilizadas para movimentação do motor estão listadas abaixo:

‘Trecho do formulário PrjLPT1.frm do projeto PrjLPT1.vbp:

```
Private Sub BtPassoDireita_Click()
    If Envia = 1 Then
        Envia = 8
    Else
        Envia = Envia / 2
    End If
    Out &H378, Envia 'Endereço H378=Endereço da porta paralela LPT1
End Sub ‘Envio de pulsos ao motor de passo conforme tabela 3.3
```

```
Private Sub BtPassoEsquerda_Click()
    If Envia = 8 Then
```

```

        Envia = 1
    Else
        Envia = Envia * 2
    End If
    Out &H378, Envia
End Sub

```

Além do controle do motor de passo, a captura da imagem via câmera USB também está implementada neste formulário. Abaixo as principais linhas do código que foram utilizadas para captura de imagem. O anexo D contempla na íntegra o código fonte.

‘Trecho do formulário PrjLPT1.frm do projeto PrjLPT1.vbp:

```

Private Sub Command2_Click()

    'Inicia a câmera

    mCapHwnd = capCreateCaptureWindow("captura Janela", 0, 0, 0, 320,
    240, Me.hwnd, 0)

    SendMessage mCapHwnd, WM_CAP_DRIVER_CONNECT, 0, 0

    Timer2.Enabled = True

    Timer2.Interval = 50

End Sub

```

```

Private Sub Command3_Click()

    'Desliga a câmera

    SendMessage mCapHwnd, WM_CAP_DRIVER_DISCONNECT, 0, 0

    Timer2.Enabled = False

End Sub

```

```

Private Sub Timer2_Timer()

```

'Exibe imagem continua no pictobox

Clipboard.Clear

SendMessage mCapHwnd, WM_CAP_GRAB_FRAME, 0, 0

SendMessage mCapHwnd, WM_CAP_EDIT_COPY, 0, 0

Picture1.Picture = Clipboard.GetData

DoEvents

Com a infra-estrutura concluída e código implementado tem-se o resultado do *software* de controle de motor de passo ilustrado na Figura 5.9. A interface é simples e auto-explicativa. No campo “Controle contínuo” existe duas setas indicando o sentido do giro do motor. A rotação também pode ser comandada passo a passo através do “Controle manual”. Já a visualização da imagem acontece no campo “Câmera”, basta iniciar no botão *play*. O passo seguinte para a conclusão da etapa de controle do motor foi a fixação da câmera ao motor de passo.

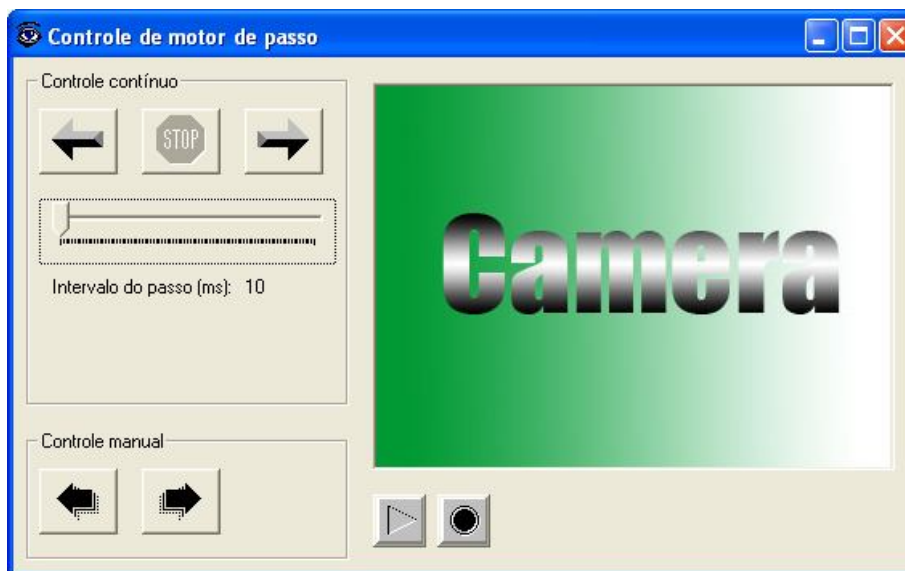


Figura 5.9: Software de Controle do Motor

(Fonte: Autor, 2007)

Como o controle de motor de passo/câmera e a visualização do ambiente estão concluídos pode-se tratar do alarme via *e-mail* no próximo item deste documento.

5.3 Montagem da Infra-Estrutura do Alarme via *E-mail*

Para a montagem do circuito de alarme via *e-mail*, ilustrado na Figura 5.10, foram utilizados os seguintes componentes:

- 1 sensor de presença infravermelho normalmente fechado com alimentação de 12V e 17mA de consumo;
- 1 CI 74LS541;
- 2 capacitores de 100nF e 1 capacitor de 0,33 μ F; e
- 1 regulador de tensão CI 7805.

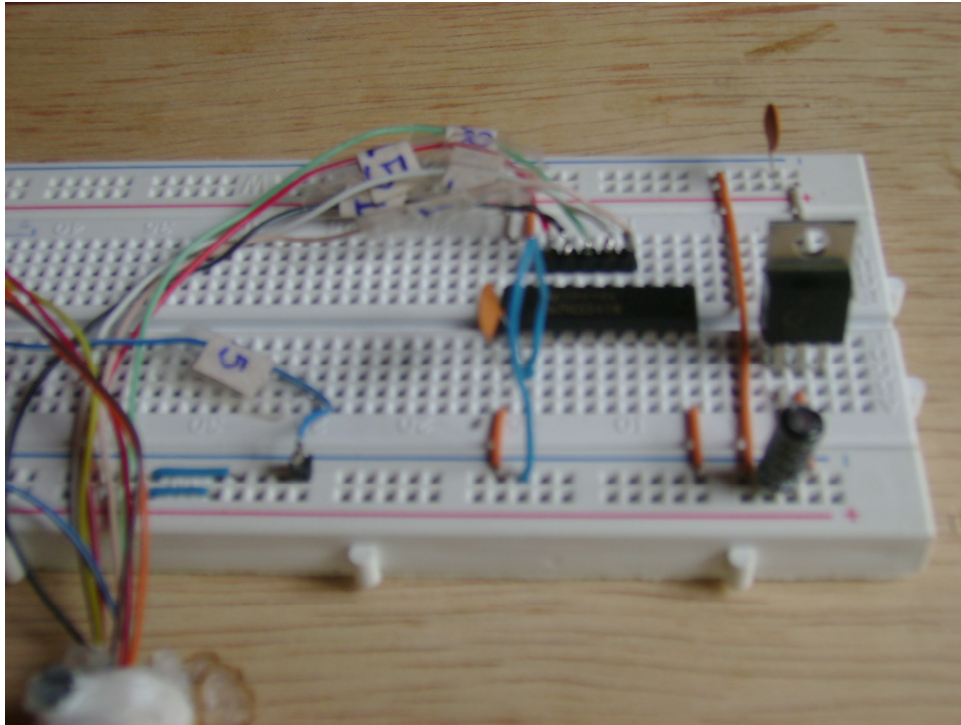


Figura 5.10: Circuito do Alarme via *E-mail*

(Fonte: Autor, 2007)

Neste trabalho foi utilizado apenas um canal do CI 74LS541 para receber os pulsos do sensor de presença. Conforme a Figura 5.11, o sinal do sensor de presença chega ao CI através do pino 2 e através do pino 18 (saída) é conectado ao pino 11 da porta paralela. O CI precisa ser alimentado por uma fonte de 5V com no mínimo 500mA, além disto, é necessário a existência de um capacitor de

100nF entre o canal de entrada e o sensor para o correto funcionamento do CI (site datasheetcatalog.net).

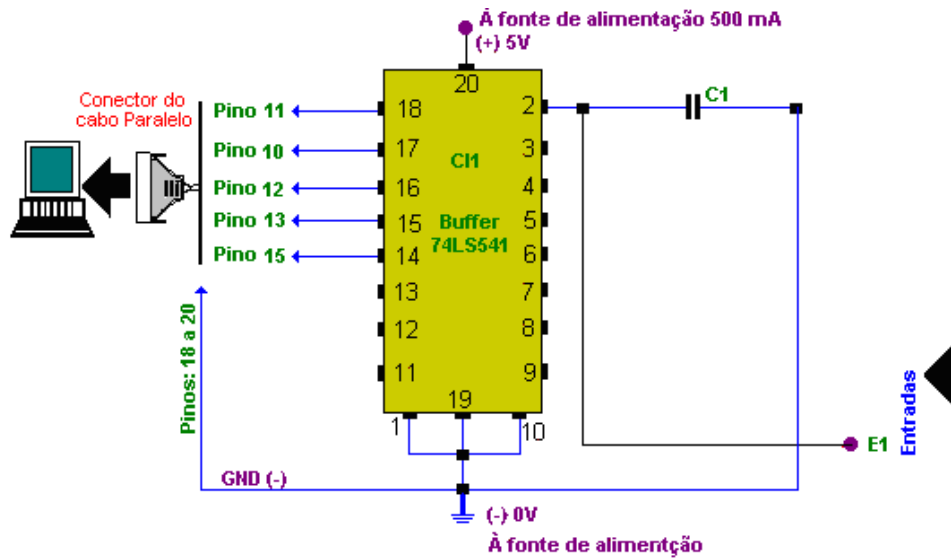


Figura 5.11: Esquema de Conexão ao CI 74LS541

(Fonte: Site rogercom.com)

Como o 74LS541 necessita de alimentação diferente da fornecida pela fonte deste projeto (12V), foi empregado o regulador de tensão CI 7805, ilustrado na Figura 5.12, para prover a alimentação adequada ao *buffer*. Este componente recebe 12V como tensão de entrada e disponibiliza a saída em 5V com corrente de até 1A.

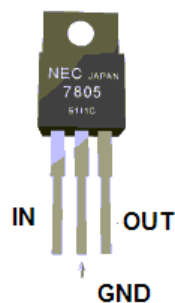


Figura 5.12: CI 7805- Regulador de Tensão

(Fonte: Site ktechno.co.kr)

Para o correto trabalho deste componente foi necessária a inserção de um capacitor de 100nF e outro com capacitância de 0,33µF, conforme o esquema

ilustrado na Figura 5.13. A Figura 5.14 ilustra de forma macro as conexões do circuito do sistema de alarme via *e-mail*.

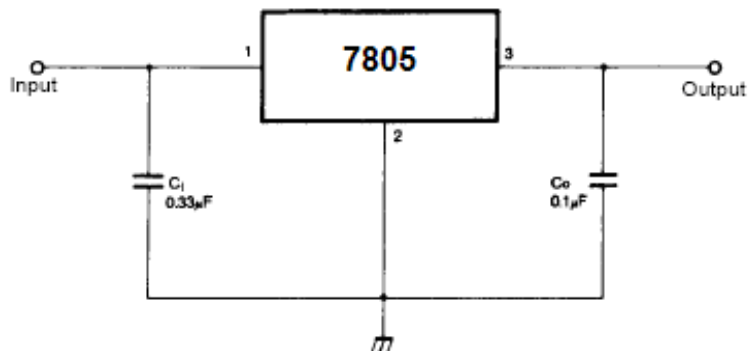


Figura 5.13: Esquema de Conexão ao CI 7805

(Fonte: Site datasheetcatalog.net)

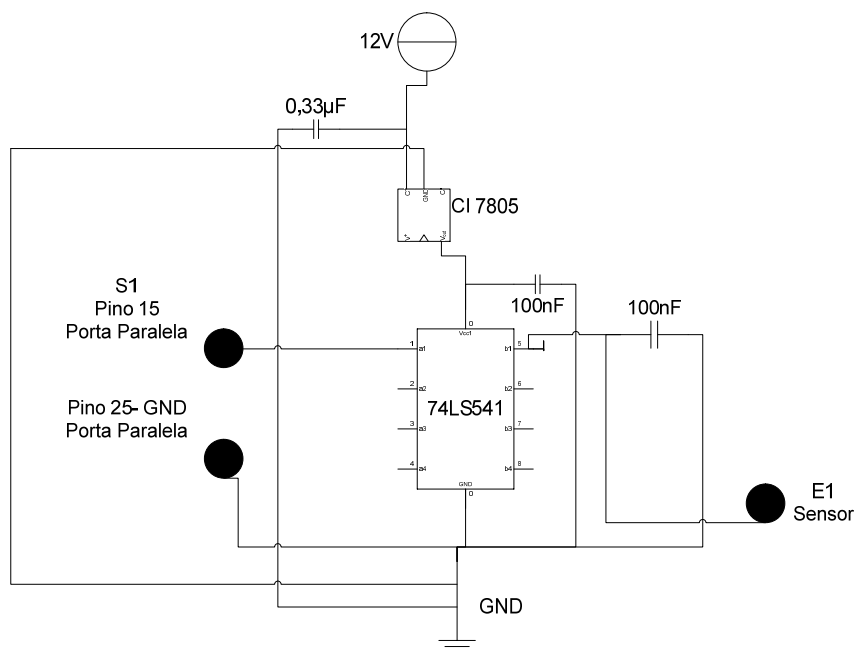


Figura 5.14: Esquema de Conexão do Alarme via *E-mail*

(Fonte: Autor, 2007)

Até este ponto as informações relacionadas à infra-estrutura do sistema de alarme via *e-mail* foram relatadas. O próximo item tratará do *software* de detecção de intrusos e envio do alerta.

5.4 Ambiente de Desenvolvimento do Alarme via *E-mail*

No desenvolvimento do código deste *software* foi utilizada a mesma ferramenta para a produção do sistema anterior, ou seja, *Visual Studio 6*. A partir deste estágio é possível comentar a lógica e o código implementado para desenvolvimento do alarme via *e-mail*.

Na produção do sistema de detecção de intrusos e envio de *e-mail*, foi utilizada para leitura do sensor de presença a porta paralela e o mesmo componente utilizado para o controle do motor de passo, ou seja, a DLL *inpout32*.

A lógica do executável *Sensor* é a seguinte: O sensor de presença utilizado neste projeto trabalha de forma normalmente fechada, ou seja, ele envia pulsos constantemente até que o ambiente monitorado seja invadido. Partindo deste princípio o *software* foi desenvolvido para escutar a cada 1 segundo o sensor, caso não consiga ler o sensor, independente do motivo, um *e-mail* de alarme será enviado.

Como o *e-mail* é enviado a partir da própria máquina, existe o pré-requisito que o serviço *Simple Mail Transfer Protocol* (SMTP) esteja instalado e configurado no microcomputador. A Figura 5.15 ilustra a instalação do SMTP.

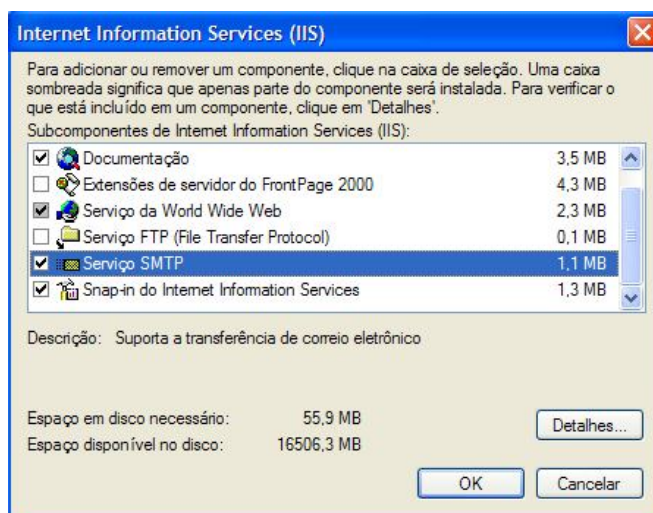


Figura 5.15: Instalação do Serviço SMTP

(Fonte: Autor, 2007)

Tão importante quanto à instalação do serviço SMTP é a configuração do serviço. Todas as configurações podem ser feitas através da console

“Gerenciamento do Computador” do Windows XP. Uma destas configurações é a habilitação do “Acesso anônimo”, conforme a Figura 5.16.

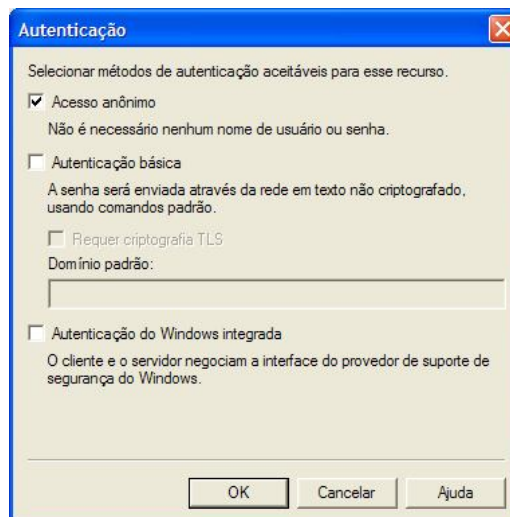


Figura 5.16: Primeiro Passo de Configuração do Serviço SMTP
(Fonte: Autor, 2007)

Além da configuração ilustrada na Figura 5.16, a configuração exibida na Figura 5.17 também deve ser feita. As demais configurações podem ser mantidas.

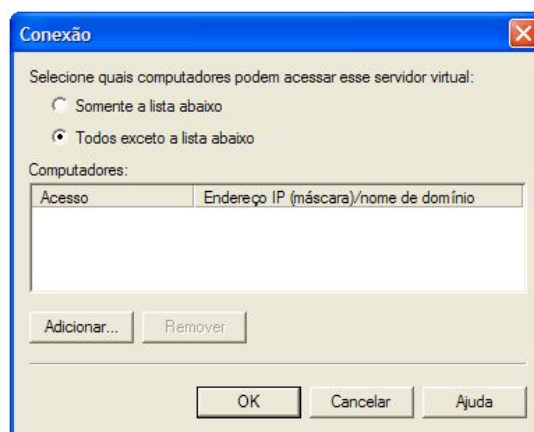


Figura 5.17: Segundo Passo de Configuração do Serviço SMTP
(Fonte: Autor, 2007)

O projeto sensor.vbp também possui dois componentes, um módulo chamado sensor.bas e um formulário chamado sensor.frm. O código no módulo sensor.bas está transcrito abaixo.

‘Sensor.bas do projeto Sensor.vbp:

```
Public Declare Function Inp Lib "inpout32.dll" _  
Alias "Inp32" (ByVal PortAddress As Integer) As Integer  
Public Declare Sub Out Lib "inpout32.dll" _  
Alias "Out32" (ByVal PortAddress As Integer, ByVal Value As Integer)
```

O código do formulário do projeto tem como fim escutar o sensor de presença e enviar um *e-mail* quando o ambiente for invadido. Abaixo a parte relativa à escuta do sensor de presença. O anexo E contempla na totalidade o código do *software* de alarme via *e-mail*.

‘Parte do Sensor.frm:

```
Private Sub Timer1_Timer()  
    Label3.Caption = Now()  
    Valor = Inp(889) ‘Endereço da porta LPT1 em notação decimal  
    Select Case Valor  
        Case 112 ‘Corresponde a 1110000 binário = ambiente vazio  
            Label1.Caption = "Esperando porta"  
            Timer1.Interval = 1000 ‘Intervalo de leitura do sensor em milisegundos  
            Case 120 ‘Corresponde a 1111000 binário = ambiente invadido  
                Label1.Caption = "Alerta!"  
                SendMail "projeto@gmail.com", sEmailDestinatario, "127.0.0.1", "Alerta do  
Sensor", "Ambiente invadido"  
                Timer1.Interval = 60000 ‘Aumento do intervalo de leitura do sensor  
            Case Else  
                Label1.Caption = "Valor não esperado"  
                Timer1.Interval = 100  
            End Select  
    End Sub
```

Com a infra-estrutura concluída e código implementado tem-se o saldo do *software* sensor.exe ilustrado na Figura 5.18. O *software* inicia a escuta do sensor somente após um clique no botão “Atualizar *E-mail*”, que além de iniciar o trabalho do *software* faz a atualização *e-mail* do destinatário no arquivo ConfigEmail.xml. Este arquivo mantém cadastrado o último endereço de *e-mail* do destinatário.



Figura 5.18: Software Sensor

(Fonte: Autor, 2007)

Com toda a parte física implementada e códigos desenvolvidos, pode-se partir para análise dos resultados obtidos neste projeto.

5.5 Resultados Obtidos

Para realização de testes deste projeto foram necessários alguns itens adicionais para completar a infra-estrutura:

- 1 computador conectado à *Internet* via *Asymmetric Digital Subscriber Line* (ADSL) com taxa de 600 kbps (Quilo *bits* por segundo) para *download* e 512 kbps para *upload*. Este computador servirá como console do projeto e possui os *softwares DynDnsUpdater* e *Radmin Server* instalados. Para facilitar a explicação este computador será denominado servidor. Além disto, este microcomputador está conectado aos circuitos via porta paralela;

- 1 computador também conectado à *Internet* ADSL⁴ através de outra linha com a mesma taxa de *download* e *upload*. Este computador teve o *Radmin Client* instalado para fazer o acesso remoto ao computador citado no tópico anterior. Este computador será tratado como cliente; e
- Um celular *Nokia* 6111 habilitado pela empresa TIM para receber *e-mails*.

A estrutura montada dentro laboratório, além dos componentes citados, contemplava o *protoboard* do projeto. O *software* sensor.exe foi executado antes que o ambiente fosse evacuado e logo em seguida o laboratório foi desocupado. Cerca de 5 minutos depois o laboratório foi invadido e em 2 minutos foi recebida uma mensagem texto no celular indicando que havia recebido um *e-mail* na caixa winiciusf@tim.com.br.

Logo em seguida, através do computador cliente que estava em outro local e utilizando o *software Radmin* foi efetuada a conexão remota ao endereço monitoramento.dnsalias.net, endereço correspondente ao computador servidor. A partir deste ponto execução do *software* “Controle do motor de passo” e ativar a captura da imagem. O passo seguinte foi controlar o giro do motor de passo a fim de visualizar todo o ambiente. A velocidade de transição dos *frames* foi apenas razoável, porém a qualidade da imagem foi considerada boa, possibilitando a identificação da pessoa com muita nitidez.

Este capítulo 5 detalhou todos componentes utilizados na montagem dos dois circuitos utilizados, dando ênfase ao CI ULN2003 e ao *buffer* 74LS541, elementos utilizados para prover a interface entre o motor e sensor, respectivamente. Além dos componentes eletrônicos, também foram citados os *softwares* de apoio utilizados durante a implementação deste projeto. Por fim foram citados pontos relacionados ao desenvolvimento do código. Concluindo esta monografia, tem-se na próxima página o capítulo 6 focado nas considerações finais.

4 Foram utilizadas duas conexões à *Internet*, pois não é possível que cliente que fará o acesso remoto tenha o mesmo IP válido na *Internet* do servidor que possui a console. Para demonstração próxima ao ambiente real é possível que exista apenas uma conexão à rede mundial de computadores.

6 Considerações Finais

Mediante todos os expostos considero que foi possível alcançar o resultado proposto, ou seja, criar um sistema de alarme com alerta via *e-mail* e desenvolver *software* de controle do motor/câmera com visualização do ambiente.

A produção técnica de qualquer projeto exige pesquisa, disciplina e dedicação na busca do resultado, contudo é natural encontrar alguns problemas passíveis de solução, dependendo do escopo do projeto.

Dentre as principais dificuldades deste projeto pode-se citar o árduo trabalho para prover as conexões físicas dos circuitos. Boa parte dos fios condutores possuía diâmetro muito pequeno, por exemplo, os fios de conexão dos pinos da porta paralela. O processo de solda destes filetes aos pinos de conexão do *protoboard* exigiu considerável tempo de execução e habilidade manual. Outro ponto que demandou pesquisa e análise estava relacionado ao controle do motor de passo. Implementar a lógica de controle do motor em linhas de código exigiu algumas horas de testes e erros para alcançar o objetivo.

Ainda em relação aos contratemplos, ocorreu o fato que ao ativar a captura de imagem foi percebido que o motor de passo passou a trabalhar em alguns momentos, sem a continuidade no giro, ou seja, com algumas breves interrupções no prosseguimento do movimento. Para análise deste problema foi utilizado o gratuito *software Process Explorer* (disponível para *download* no *site* microsoft.com/technet/sysinternals/Utilities/ProcessExplorer.msp). Quando o executável é ativado percebe-se que apenas uma *thread* é acionada dentro do processo, porém quando a captura da imagem é iniciada duas *threads* são avivadas. Considera-se que esta concorrência entre *threads* seja a causa do problema citado, porém isto não causou qualquer prejuízo para o alcance do objetivo do projeto.

Sugere-se para projetos futuros a implementação de histórico das atividades via registros de *logs*. Além disto, considera-se interessante a adição de controle de giro do motor levando em consideração o ângulo do movimento.

Referências Bibliográficas

1. ALBUQUERQUE, Pedro Urbano Braga de. *Sensores Industriais: Fundamentos e Aplicações*. São Paulo: Érica, 2005.
2. CARVALHO, Geraldo. *Máquinas Elétricas Teoria e Ensaio*s. São Paulo: Érica, 2006.
3. CLUBE DO HARDWARE [Home Page]. Disponível em <<http://clubedohardware.com.br>>. Acesso em: 20 de março de 2007.
4. CORREIOWEB [Home Page]. Disponível em <<http://www.correioweb.com.br/noticias/materias.php?id=2677866&sub=Distrito>>. Acesso em: 16 de março de 2007.
5. DATASHEET CATALOG [Home Page]. Disponível em <<http://www.datasheetcatalog.net>;
<http://www.datasheetcatalog.net/datasheets_pdf/U/L/N/2/ULN2003.shtml;
<http://www.datasheetcatalog.net/datasheets_pdf/7/8/0/5/7805.shtml;
<http://www.datasheetcatalog.net/datasheets_pdf/7/4/L/S/74LS541.shtml>. Acesso em: 20 de agosto de 2007.
6. DYNDNS [Home Page]. Disponível em <<http://www.dyndns.com>>. Acesso em: 1 de novembro de 2007.
7. ELETRÔNICA.ORG [Home Page]. Disponível em <<http://www.eletronica.org>>. Acesso em: 16 de junho de 2007.
8. EMBEDDED SYSTEM [Home Page]. Disponível em <<http://www.logix4u.net>>. Acesso em: 10 de setembro de 2007.
9. FELIPPO, Guilherme Filho. *Motor de Indução*. São Paulo: Érica, 2000.
10. HOME PAGE ROGERCOM [Home Page]. Disponível em <<http://www.rogercom.com>>. Acesso em: 18 de março de 2007.
11. IDG NOW [Home Page]. Disponível em <<http://www.idgnow.uol.com.br>>. Acesso em: 16 de março de 2007.
12. INFOWESTER [Home Page]. Disponível em <<http://www.infowester.com.br>>. Acesso em: 28 de agosto de 2007.
13. INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA [Home Page]. Disponível em <<http://www.ibge.gov>>. Acesso em: 16 de março de 2007.

14. JOBTEC [Home Page]. Disponível em <<http://www.jobtecltda.com.br>>. Acesso em: 10 de setembro de 2007.
15. KTECHNO [Home Page]. Disponível em <http://www.ktechno.co.kr/ls_parts/img_part/7805.gif>. Acesso em: 30 de outubro de 2007.
16. MACORATTI.NET [Home Page]. Disponível em <<http://www.macoratti.net>>. Acesso em: 1 de outubro de 2007.
17. MERCADO BR Shopping [Home Page]. Disponível em <<http://www.mercadobr.com.br>>. Acesso em: 4 de junho de 2007.
18. MICROSOFT TECHNET [Home Page]. Disponível em <<http://www.microsoft.com/technet>>. Acesso em: 12 de novembro de 2007.
19. MIKROELETRONIKA [Home Page]. Disponível em <<http://www.mikroe.com>>. Acesso em: 3 de agosto de 2007.
20. MOTORPASSO [Home Page]. Disponível em <<http://www.motorpasso.no.sapo.pt>>. Acesso em: 23 de outubro de 2007.
21. MRS [Home Page]. Disponível em <<http://www.mrshp.hpg.ig.com.br>>. Acesso em: 20 de março de 2007.
22. PC REMOTE CONTROL SOFTWARE- RADMIN [Home Page]. Disponível em <<http://www.radmin.com>>. Acesso em: 1 de novembro de 2007.
23. PORTAL ANATEL [Home Page]. 2007. Disponível em <<http://www.anatel.gov.br>>. Acesso em: 10 de março de 2007.
24. TANENBAUM, Andrew. *Sistemas Operacionais Modernos*. 2. ed. Rio de Janeiro: LTC, 1999.
25. TIPLER, Paul. *Física Eletricidade e Magnetismo, Ótica*. 4. ed. Rio de Janeiro: LTC, 2000.
26. TORO, Vicent Del. *Fundamentos de Máquinas Elétricas*. Rio de Janeiro: LTC, 1999.
27. TORRES, Gabriel. *Hardware Curso Completo*. 4. ed. Rio de Janeiro: Axcel, 2001.
28. VASCONCELOS, Laércio. *Hardware Total*. São Paulo: Makron, 2002.

Anexos

Anexo A: Tabela de pinos da porta paralela

Sinal	Pino (DB25)	Pino (Centronics)	E/S	Descrição
Strobe	01	01	S	Indica se o dado está pronto ou não para ser transmitido. 0= Dado pronto para ser transmitido, 1= Dado não está pronto para ser transmitido.
D0 a D7	2 a 9	2 a 9	S	<i>Bits</i> de dados.
Ack	10	10	E	Indica se o dispositivo está pronto para receber dados.
Busy	11	11	E	Indica que o dispositivo não está pronto para receber dados.
Paper end	12	12	E	Indica que não há papel na impressora.
Select	13	13	E	Indica que o dispositivo está on-line.
Auto Feed	14	14	S	A impressora move o papel para o início da próxima linha.
Error	15	32	E	Indica algum erro.
Init	16	31	S	Reinicia o dispositivo.
Select Input	17	36	S	Dados podem ser transferidos para dispositivo apenas quando esta linha estiver em "0".
GND	18 a 25	19 até 30	S	Terra

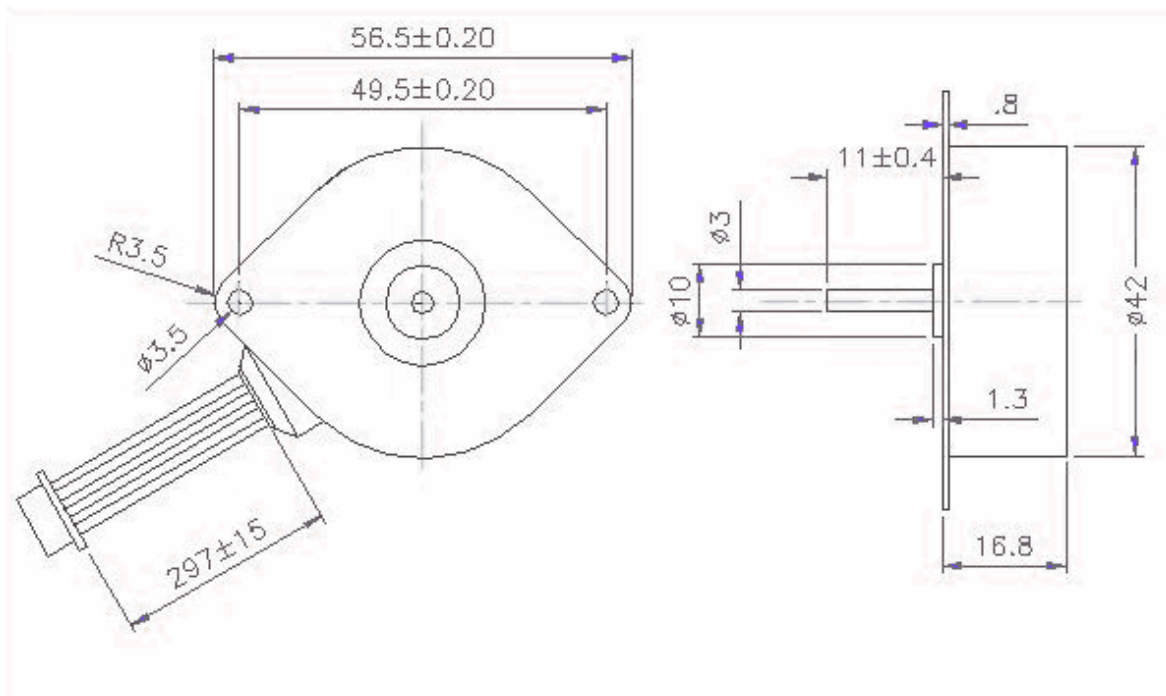
(Fonte: [Site clubedohardware.com.br](http://Siteclubedohardware.com.br))

Anexo B: Código Fonte do Experimento que Aciona *Leds*

```
//Envia sinal para a Porta Paralela LPT1:
#include <stdio.h>
#include <dos.h> //Carrega bibliotecas
#define LPT1 0x378 //Define o endereço de acesso a porta LPT1
int main(void)
{
    unsigned char numero=128; //Em binário = 10000000
    while( numero > 0 )
    {
        outportb(LPT1, numero); // Envia para a Porta LPT1
        printf("\nPressione a tecla para ascender o próximo LED");
        getch( );
        Valor = numero >> 1; //A cada passagem o bit 1 é movido para à direita
    }
}
```

(Fonte: *Site* mrshp.hpg.ig.com.br)

Anexo C: Especificação do Motor de Passo



Especificações:

Item	Unit	Value
Drive Voltage +/- 10%	V DC	12
Current Per Phase	A	0,4
Resistance Per Winding	Ω	16
N. of Phase	4	
Excitation Mode	Unipolar	
Step Angle	Deg.	$7,5^\circ$
Holding Torque	g.cm	500
Dentent Torque	g.cm	150
Pull-Out Torque/12V/300PPS	g.cm	260

Max Starting Frequency	Pps	800
Max Response Frequency	Pps	700
12V 100pps Temperature Rise	Deg	65°C
Insulation Resistance	MΩ	10M (DC 500V)
Ambient Temperatuer	Deg	-5 °C a +50 °C
Weigth	Gf	90q

(Fonte: *Site mercadobr.com.br*)

Anexo D: Especificação do Sensor de Presença

Ligação Elétrica:



Especificações:

- Alimentação 12vcc;
- Possui compensação de temperatura;
- Led ligado ou desligado;
- Consumo 17 mA;
- Estabilização 60 segundos;
- Cobertura horizontal 75°;
- Cobertura máxima 10m;
- Cobertura vertical 10° a 30°;
- Tempo de disparo 2 segundos.

Anexo E: Código Fonte do Controle de Motor de Passo e Visualização via Câmera

‘Módulo PrjLPT1.bas do projeto PrjLPT1.vbp:

‘Declarando funções Inp32 da biblioteca que se encontra na DLL

```
Public Declare Function Inp Lib "inpout32.dll" _
```

```
Alias "Inp32" (ByVal PortAddress As Integer) As Integer
```

```
Public Declare Sub Out Lib "inpout32.dll" _
```

```
Alias "Out32" (ByVal PortAddress As Integer, ByVal Value As Integer)
```

‘Formulário PrjLPT1.frm do projeto PrjLPT1.vbp:

‘Declarando as funções da DLL avicap32.dll

```
Private Declare Function capCreateCaptureWindow Lib "avicap32.dll" Alias  
"capCreateCaptureWindowA" (ByVal lpszWindowName As String, ByVal dwStyle  
As Long, ByVal X As Long, ByVal Y As Long, ByVal nWidth As Long, ByVal  
nHeight As Long, ByVal hwndParent As Long, ByVal nID As Long) As Long
```

```
Private Declare Function SendMessage Lib "USER32" Alias "SendMessageA"  
(ByVal hwnd As Long, ByVal wParam As Long, ByVal lParam As  
Any) As Long
```

‘Declarando as funções da API USER32

```
Private Declare Function ReleaseCapture Lib "USER32" () As Long
```

```
Private Const WM_CAP_DRIVER_CONNECT As Long = 1034
```

```
Private Const WM_CAP_DRIVER_DISCONNECT As Long = 1035
```

```
Private Const WM_CAP_GRAB_FRAME As Long = 1084
```

```
Private Const WM_CAP_EDIT_COPY As Long = 1054
```

```
Private Const WM_CAP_DLG_VIDEOSOURCE As Long = 1065
```

```
Private Const WM_CAP_DLG_VIDEOFORMAT As Long = 1066
```

Private Const WM_CLOSE = &H10

Private mCapHwnd As Long

Dim Envia As Integer

Dim ContProgress As Integer

Dim Direita As Boolean

Private Sub BtDireita_Click()

 Timer1.Enabled = True

 Direita = True

 BtDireita.Enabled = False

 BtStop.Enabled = True

 BtEsquerda.Enabled = True

End Sub

Private Sub BtEsquerda_Click()

 Timer1.Enabled = True

 BtEsquerda.Enabled = False

 BtStop.Enabled = True

 Direita = False

 BtDireita.Enabled = True

End Sub

'Envio de pulsos ao motor para prover o giro para direita

Private Sub BtPassoDireita_Click()

 If Envia = 1 Then

 Envia = 8

 Else

```
        Envia = Envia / 2
    End If
    Out &H378, Envia
End Sub
```

'Envio de pulsos ao motor para prover o giro para esquerda

```
Private Sub BtPassoEsquerda_Click()
    If Envia = 8 Then
        Envia = 1
    Else
        Envia = Envia * 2
    End If
    Out &H378, Envia
End Sub
```

```
Private Sub BtStop_Click()
    Timer1.Enabled = False
    ProgressBar1.Visible = False
    ProgressBar1.Value = 0
    BtDireita.Enabled = True
    BtEsquerda.Enabled = True
End Sub
```

```
Private Sub Command2_Click()
    'Inicia a câmara
    mCapHwnd = capCreateCaptureWindow("captura Janela", 0, 0, 0, 320,
    240, Me.hwnd, 0)
    SendMessage mCapHwnd, WM_CAP_DRIVER_CONNECT, 0, 0
```

Timer2.Enabled = True

Timer2.Interval = 50

End Sub

Private Sub Command3_Click()

'Desliga a câmera

SendMessage mCapHwnd, WM_CAP_DRIVER_DISCONNECT, 0, 0

Timer2.Enabled = False

End Sub

Private Sub Form_Load()

Timer1.Enabled = False

Timer2.Enabled = False

Label1.Caption = Slider1.Value

Envia = 8

Direita = True

ProgressBar1.Visible = False

ProgressBar1.Value = 0

ContProgress = 0

BtStop.Enabled = False

End Sub

Private Sub Form_Terminate()

'Desliga a câmera

SendMessage mCapHwnd, WM_CAP_DRIVER_DISCONNECT, 0, 0

Timer2.Enabled = False

End Sub


```

Private Sub Slider1_Scroll()
    Label1.Caption = Slider1.Value
    Timer1.Interval = Slider1.Value
End Sub

Private Sub Timer1_Timer()
    If Not ProgressBar1.Visible Then
        ProgressBar1.Visible = True
    End If

    If ContProgress = 100 Then
        ContProgress = 0
    Else
        ContProgress = ContProgress + 1
    End If

    ProgressBar1.Value = ContProgress

    If Direita Then
        If Envia = 1 Then
            Envia = 8
        Else
            Envia = Envia / 2
        End If

        BtDireita.Enabled = False
    Else
        If Envia = 8 Then
            Envia = 1
        Else
            Envia = Envia * 2
        End If
    End If
End Sub

```

Out &H378, Envia

End Sub

Private Sub Timer2_Timer()

'Exibe imagem continua no picture box

Clipboard.Clear

SendMessage mCapHwnd, WM_CAP_GRAB_FRAME, 0, 0

SendMessage mCapHwnd, WM_CAP_EDIT_COPY, 0, 0

Picture1.Picture = Clipboard.GetData

DoEvents

End Sub

Anexo F: Código Fonte do *Software* de Alarme via *E-mail*

‘Módulo Sensor.bas do projeto Sensor.vbp:

```
Public Declare Function Inp Lib "inpout32.dll" _
```

```
    Alias "Inp32" (ByVal PortAddress As Integer) As Integer
```

```
Public Declare Sub Out Lib "inpout32.dll" _
```

```
    Alias "Out32" (ByVal PortAddress As Integer, ByVal Value As Integer)
```

‘Formulário Sensor.frm do projeto Sensor.vbp:

```
Dim sEmailDestinatario As String
```

```
Dim objDOM      As New DOMDocument
```

‘Declarando XML que receberá endereço do *e-mail*

```
Private Sub cmdEmail_Click()
```

```
    objDOM.loadXML "<Sensor><Email>" & txtEmail.Text &  
    "</Email></Sensor>"
```

```
    sEmailDestinatario = txtEmail.Text
```

```
    objDOM.Save App.Path & "/ConfigEmail.xml" ‘Salva o e-mail do  
    ‘destinatário
```

```
    Timer1.Enabled = True
```

```
End Sub
```

Private Sub Command1_Click() ‘Passando parâmetros para envio do *e-mail*

```
    SendMail "projeto@gmail.com", sEmailDestinatario, "127.0.0.1", "E-mail de  
    teste do projeto de monitoramento", "E-mail de teste do projeto de monitoramento  
    do Winícius Ferraz"
```

```
End Sub
```

```

Private Sub Form_Load()
    objDOM.Load App.Path & "/ConfigEmail.xml"
    sEmailDestinatario = objDOM.selectSingleNode("/Sensor/Email").Text
    txtEmail.Text = sEmailDestinatario
End Sub

```

```

Private Sub Timer1_Timer()
    Label3.Caption = Now()
    Valor = Inp(889) 'Endereço da porta LPT1 em notação decimal
    Select Case Valor
        Case 112 'Corresponde a 1110000 binário = ambiente vazio
            Label1.Caption = "Esperando porta"
            Timer1.Interval = 1000 'Intervalo de leitura do sensor
        Case 120 'Corresponde a 1111000 binário = ambiente invadido
            Label1.Caption = "Alerta!"
            SendMail "projeto@gmail.com", sEmailDestinatario, "127.0.0.1", "Alerta do
Sensor", "Ambiente invadido" 'Passando parâmetros para envio do e-mail
            Timer1.Interval = 60000 'Aumento do intervalo de leitura do sensor
        Case Else
            Label1.Caption = "Valor não esperado"
            Timer1.Interval = 100
    End Select
End Sub

```

'Função de envio de e-mail

```

Private Sub SendMail(MailFrom As String, MailTo As String, SrvSMTP As String,
TXTAssunto As String, TXTCorpo As String)
    Dim lobj_cdomsg As CDO.Message

```

```
Set lobj_cdomsg = New CDO.Message
'Add the Project Reference Miscrosoft CDO WINDOWS FOR 2000
lobj_cdomsg.Configuration.Fields(cdoSMTPServer) = SrvSMTP
lobj_cdomsg.Configuration.Fields(cdoSMTPConnectionTimeout) = 30
lobj_cdomsg.Configuration.Fields(cdoSendUsingMethod) =
doSendUsingPort
lobj_cdomsg.Configuration.Fields.Update
lobj_cdomsg.To = MailTo
lobj_cdomsg.From = MailFrom
lobj_cdomsg.Subject = TXTAssunto
lobj_cdomsg.TextBody = TXTCorpo
lobj_cdomsg.Send
Set lobj_cdomsg = Nothing

End Sub
```