



CENTRO UNIVERSITÁRIO DE BRASÍLIA - UNICEUB
FACULDADE DE CIÊNCIAS EXATAS E TECNOLOGIA – FAET
CURSO: ENGENHARIA DE COMPUTAÇÃO
DISCIPLINA: MONOGRAFIA ACADÊMICA
PROFESSOR ORIENTADOR: ADERLON MARCELINO QUEIROZ

CONTROLE REMOTO INFRAVERMELHO PARA AUTOMAÇÃO

RAFAEL RIBEIRO REZENDE LOMBARDI

REGISTRO ACADÊMICO: 2012652/1

Brasília/DF, Julho de 2006



CENTRO UNIVERSITÁRIO DE BRASÍLIA - UNICEUB
FACULDADE DE CIÊNCIAS EXATAS E TECNOLOGIA – FAET
CURSO: ENGENHARIA DE COMPUTAÇÃO
DISCIPLINA: MONOGRAFIA ACADÊMICA
PROFESSOR ORIENTADOR: ADERLON MARCELINO QUEIROZ

CONTROLE REMOTO INFRAVERMELHO PARA AUTOMAÇÃO

RAFAEL RIBEIRO REZENDE LOMBARDI

REGISTRO ACADÊMICO: 2012652/1

Monografia apresentada como requisito para conclusão do curso de Engenharia de Computação no UniCEUB – Centro Universitário de Brasília.

Brasília/DF, Julho de 2006



CENTRO UNIVERSITÁRIO DE BRASÍLIA - UNICEUB
FACULDADE DE CIÊNCIAS EXATAS E TECNOLOGIA – FAET
CURSO: ENGENHARIA DE COMPUTAÇÃO
SUPERVISÃO DE MONOGRAFIA ACADÊMICA

MEMBROS DA BANCA EXAMINADORA

3.2.1. MEMBROS DA BANCA	<i>ASSINATURA</i>
1. PROFESSOR^(a) ORIENTADOR^(a) Prof. (a): MC Aderlon Marcelino Queiroz	
2. PROFESSOR^(a) CONVIDADO^(a) Prof ^(a) .: MC Maria Marony Souza Farias Nascimento	
3. PROFESSOR^(a) CONVIDADO^(a) Prof ^(a) .: MC José Julimá Bezerra Júnior	
MENÇÃO FINAL::	

Brasília/DF, 07 de Julho de 2006.

DEDICATÓRIA

À Jesus,

Por sempre estar ao meu lado, me guiando em cada passo na minha vida.

À Minha família,

Que me deu a base necessária para a formação da minha personalidade, independência e educação, me incentivando sempre a lutar pelos meus objetivos e ter um ideal de vida.

AGRADECIMENTOS

À Deus,

Agradeço primeiramente a Jesus por tudo o que tem feito na minha vida, me guiando e iluminando em todos os momentos para que eu possa tomar decisões corretas.

À Família,

Meus pais, Hélio e Eliane e meu irmão Thiago por sempre estarem ao meu lado durante todo o tempo e por me permitirem o direito de completar meus estudos, sempre desejando o melhor tanto para mim como para meu irmão.

À Mariana,

Minha namorada, que esteve ao meu lado em momentos muito importantes e marcantes da minha vida, me trazendo sempre grande alegria com sua presença.

Ao Aderlon,

Meu professor orientador que me ajudou muito na elaboração da monografia, sempre corrigindo e propondo melhorias para o projeto.

Aos meus amigos do peito, que caminharam juntos nessa jornada maravilhosa de aprendizagem, dos quais sempre me lembrarei dos grupos de

estudo, futebol, churrascos, festas e demais momentos que passamos.

"O único homem que está isento de erros, é aquele que não arrisca acertar."

(Albert Einstein)

"Viver é a coisa mais rara do mundo. A maioria das pessoas apenas existe." (Oscar

Wilde)

"Diante da vastidão do tempo e da imensidão do espaço é uma alegria para mim compartilhar uma época e um planeta com você." (Carl Sagan)

RESUMO

Controle Remoto Infravermelho para Automação é um projeto que tem como objetivo automatizar algumas atividades cotidianas com o uso do controle remoto. O principal objetivo é montar um circuito infravermelho de baixo custo que possa ser manipulado pelos mais variados tipos e marcas de controle remoto.

Este trabalho consiste em um estudo sobre processos e conceitos necessários para a implementação do circuito infravermelho, conectado através da porta serial ao computador. A execução de tarefas através do processamento desse computador foi feita na medida em que os sinais do controle remoto são decodificados pelo software livre LIRC (Linux Infrared Remote Control, Controle Remoto Infravermelho Linux). Tais aplicações incluem o controle do cursor do mouse remotamente; Envio de combinações de teclas a qualquer programa e a execução desses a partir de um botão pré-programado; Acender e apagar uma lâmpada, usando a porta paralela do computador.

O projeto descreve a montagem de um segundo circuito conectado à porta paralela do computador que irá receber sinais dos pinos de dados para ativar ou desativar o circuito conectado à lâmpada. Dessa forma, através de um simples botão do controle remoto será possível acender e apagar lâmpadas.

Palavras-chave: Infravermelho, Automação, Controle remoto.

ABSTRACT

The objective of Infrared Remote Control for Automation project is to automate some common tasks of day with a remote control. The main purpose is to build a cheap infrared circuit that can be manipulated by the most common kinds and brands of remote controls.

It consists of a study on processes and concepts for the implementation of the infrared circuit connected to a personal computer by the serial port. The execution of tasks will be made by the computer after the signals of remote control be decoded by the free software LIRC (Linux Infrared Remote Control). Some of the applications are the control of the mouse movements remotely; Executing programs just by one button press; Turn on and off a lamp connected to a parallel port circuit.

The project foresees the build of a second circuit that will be connected to computer parallel port. This circuit will receive signals from data pins to activate or deactivate a circuit connect to the relay coil that is directly connected to the lamp. In this way, with just a button press it will be possible to turn on and turn off lamps.

Keywords: Infrared, Automation, Remote control.

SUMÁRIO

ÍNDICE DE FIGURAS	XII
ÍNDICE DE TABELAS.....	XIII
ÍNDICE DE TABELAS.....	XIII
ÍNDICE DE EQUAÇÕES	XIV
LISTA DE SIGLAS E NOMENCLATURAS.....	XV
LISTA DE SIGLAS E NOMENCLATURAS.....	XV
1. INTRODUÇÃO	1
1.1. MOTIVAÇÃO	3
1.1. OBJETIVOS.....	3
2. INFRAVERMELHO	5
2.1. CONCEITOS INICIAIS	5
2.2. INFRAVERMELHO WIRELESS.....	9
2.3. CARACTERÍSTICAS.....	9
2.4. DETECÇÃO E NEP.....	10
3. CIRCUITO INFRAVERMELHO	13
3.1. PADRÃO RS-232.....	13
3.2. CIRCUITO/HARDWARE.....	19
3.2.1. <i>Características TSOP 1738</i>	23
3.2.2. <i>Funcionamento</i>	24
4. AUTOMAÇÃO.....	27
4.1. APLICAÇÕES.....	27
4.2. CURSOR DO MOUSE	27
4.3. INTERFACE PARALELA	28
4.4. CIRCUITO LED	32
4.5. CIRCUITO LÂMPADA.....	34
4.6. FLUXOGRAMA	37
.....	37
5. LIRC – LINUX INFRARED REMOTE CONTROL	38
5.1. INSTALAÇÃO DO PACOTE LIRC	39
5.1.1. <i>Dependências</i>	39
5.1.2. <i>Instalação</i>	41
5.1.3. <i>Teste</i>	49
5.1.4. <i>Programas LIRC</i>	50
5.1.5. <i>Diagrama de blocos</i>	52
6. CONSIDERAÇÕES FINAIS.....	54
6.1. CONCLUSÃO	54
6.2. DIFICULDADES	54
6.3. SUGESTÕES PARA TRABALHOS FUTUROS	56
REFERÊNCIAS BIBLIOGRÁFICAS	57
7. APÊNDICES.....	59

7.1. FORMATO	59
APÊNDICE A	60
7.1.1. <i>LIRCD.CONF</i>	60
APÊNDICE B	64
7.1.2. <i>LIRCMD</i>	64
APÊNDICE C	68
7.1.3. <i>LIRCRC</i>	68
APÊNDICE D	72
7.1.4. <i>LPT_COMPLETO</i>	72
APÊNDICE E	73
7.1.5. <i>ESQUEMA ELÉTRICO CIRCUITO INFRAVERMELHO</i>	73
APÊNDICE F	74
7.1.6. <i>ESQUEMA ELÉTRICO CIRCUITO LÂMPADA</i>	74

Índice de Figuras

FIGURA 2.1: ESPECTRO ELETROMAGNÉTICO	6
FIGURA 2.2: FORMAS DE TRANSMISSÃO DO INFRAVERMELHO	8
FIGURA 3.1: PINOS DA INTERFACE SERIAL RS-232	15
FIGURA 3.2: PINOS CONECTOR RS-232	15
FIGURA 3.3: PINOS USADOS NO PROJETO	17
FIGURA 3.4: CONEXÕES FÍSICAS DO CIRCUITO RECEPTOR INFRAVERMELHO	20
FIGURA 3.5: ILUSTRAÇÃO DAS CONEXÕES ENTRE OS COMPONENTES	20
FIGURA 3.6: PINOS DO REGULADOR DE TENSÃO 78L05	21
FIGURA 3.7: PINOS DO RECEPTOR TSOP 1738	22
FIGURA 3.8: REPRESENTAÇÃO DAS CONEXÕES EM UMA MATRIZ 5X5	25
FIGURA 3.9: FOTO DO CIRCUITO MONTADO NO PROTOBOARD	26
FIGURA 4.1: PINOS DA INTERFACE PARALELA RS-232	29
FIGURA 4.2: TRANSISTORES EM SÉRIE ORIGINANDO CORRENTE	31
FIGURA 4.3: TRANSISTORES EM SÉRIE DISSIPANDO CORRENTE	31
FIGURA 4.4: POLARIZAÇÃO DO LED	32
FIGURA 4.5: LED	32
FIGURA 4.6: ESQUEMA DE CONEXÃO DA PORTA PARALELA AOS LEDS	33
FIGURA 4.7: ESQUEMA DE MONTAGEM DO CIRCUITO PARA AUTOMAÇÃO LÂMPADA	35
FIGURA 4.8: ESQUEMA COM A MESMA FUNCIONALIDADE DO CIRCUITO ABORDADO ANTERIORMENTE	36
FIGURA 4.9: FLUXOGRAMA	37
FIGURA 5.1: CONFIGURAÇÃO DO KERNEL	40
FIGURA 5.2: CONFIGURAÇÃO DO KERNEL	41
FIGURA 5.3: SCRIPT DE CONFIGURAÇÃO LIRC	42
FIGURA 5.4: CONFIGURAÇÃO DO DRIVER	43
FIGURA 5.5: CONFIGURAÇÃO DO DRIVER	43
FIGURA 5.6: CONFIGURAÇÃO DE ENDEREÇO ENTRADA/SAÍDA E INTERRUPÇÃO	44
FIGURA 5.7: OPÇÃO PARA TRABALHAR APENAS COM O MÓDULO RECEPTOR	44
FIGURA 5.8: CONFIGURAÇÃO DO LIRC	45
FIGURA 5.9: COMPILAÇÃO E INSTALAÇÃO DO LIRC	46
FIGURA 5.10: ALTERAÇÃO DE PERMISSÃO	47
FIGURA 5.11: ESPECIFICAÇÃO DO DRIVER E ENDEREÇAMENTO	48
FIGURA 5.12: ESPECIFICAÇÃO DO DIRETÓRIO DAS BIBLIOTECAS	49
FIGURA 5.13: DIAGRAMA DE BLOCOS LIRC	52
FIGURA 7.1: ESQUEMA ELÉTRICO CIRCUITO INFRAVERMELHO	73
FIGURA 7.2: ESQUEMA ELÉTRICO CIRCUITO LÂMPADA	74

Índice de Tabelas

TABELA 2.1: SUBDIVISÕES DO INFRAVERMELHO	7
TABELA 3.1: RELAÇÃO DE PINOS E FUNÇÕES DO PADRÃO RS-232	18
TABELA 4.1: RELAÇÃO DOS PINOS E BITS REFERENTES	33

Índice de Equações

EQUAÇÃO 2.1: NEP	11
EQUAÇÃO 2.2: DETECTIVIDADE ESPECTRAL	11
EQUAÇÃO 2.3: DETECTIVIDADE ESPECÍFICA	12
EQUAÇÃO 2.4: RESPONSABILIDADE	12

Lista de Siglas e Nomenclaturas

C	- Common, Comum
CMOS	- Complementary Metal-Oxide Semiconductor, Semicondutor de Metal-Óxido Complementar
CTS	- Clear to Send, Livre para enviar
D	- Detectividade Espectral
D*	- Detectividade Específica
DCD	- Data Carrier Detect, Detector de Portadora de Dados
DCE	- Data Communication Equipment, Equipamento de Comunicação de Dados
DSR	- Data Set Ready, Dados Prontos
DTE	- Data Terminal Equipment, Equipamento Terminal de Dados
DTR	- Data Terminal Ready, Terminal de dados Pronto
FIR	- Far Infrared, Infravermelho Distante
GND	- Ground, Terra
GNU	- General Public License, Licença Pública Geral
I/O	- Input/Output, Entrada/Saída
IRDA	- The Infrared Data Association, Associação de Dados Infravermelho
IRQ	- Interrupt Request, Requisição de Interrupção
ISA	- Industry Standard Architecture, Arquitetura Padrão de Indústria
LIRC	- Linux Infrared Remote Control, Controle Remoto Infravermelho Linux
MIR	- Medium Infrared, Infravermelho Médio

NC	- Normally Closed, Normalmente Fechado
NEP	- Noise Equivalent Power, Potência Equivalente ao Ruído
NIR	- Near Infrared, Infravermelho Próximo
NO	- Normally Open, Normalmente Aberto
PCI	- Peripheral Component Interconnect, Componente Periférico de Interconexão
PCM	- Pulse Codification Modulation, Modulação por Codificação de Pulso
PDA	- Personal Digital Assistants, Assitência Digital Pessoal
R	- Responsitividade
RTS	- Request to send, Pedido para enviar
RxD	- Received Data, Receptor de dados
TTL	- Transistor-Transistor Logic, Lógica Transistor-Transistor
TxD	- Transmitted Data, Transmissor de dados
UART	- Universal Asynchronous Receiver Transmitter, Receptor e Transmissor Assíncrono Universal
XIR	- Extreme Infrared, Infravermelho Extremo

1. INTRODUÇÃO

Este trabalho descreve como montar um circuito infravermelho de baixo custo para facilitar e automatizar tarefas cotidianas.

Foram abordados conceitos sobre o funcionamento da tecnologia infravermelha bem como todo o processo de montagem do circuito, programação e aprendizagem do controle remoto além da customização do *kernel* do sistema operacional para a acoplagem do circuito.

A aplicação prática de diversas matérias cursadas durante a faculdade foi de fundamental importância durante a implementação do projeto. Certas habilidades como a montagem de circuitos, soldagem em placas integradas foram aprimoradas durante esse período.

O estudo durante todo o projeto, buscou sempre melhorar os resultados obtidos durante a execução de testes e implementar novidades. Além disso, várias dificuldades foram encontradas durante a implementação. Essas dificuldades são descritas sucintamente nas considerações finais da monografia. Essas dificuldades serviram para enriquecer o trabalho e descobrir realmente a causa dos problemas.

O projeto tem um forte embasamento teórico e busca fornecer os passos necessários para a implementação do circuito receptor, mostrando algumas das possíveis aplicações que podem ser desenvolvidas.

A monografia começa apresentando a teoria do infravermelho e seus conceitos fundamentais. Logo em seguida foram descritos o funcionamento das portas seriais e

paralelas, necessárias para o funcionamento dos circuitos receptor infravermelho e da aplicação que tem como função acender e apagar uma lâmpada através do controle remoto.

No capítulo três é enfatizado a parte de montagem do circuito receptor propriamente dito. Nessa fase são descritos os componentes utilizados bem como o funcionamento desse hardware através do software livre LIRC.

No capítulo quatro iniciou-se a fase de testes e as aplicações abordadas nessa monografia. A primeira a ser descrita é a possibilidade de controlar o cursor e ações do *mouse* pelo controle remoto, seguida pela execução de programas e o agendamento de scripts. Para finalizar, são apresentadas as soluções para a parte de automação com a montagem de um circuito capaz de acender e apagar uma lâmpada através de um botão do controle remoto.

Já no capítulo cinco vem a descrição e o modo de instalação do software livre LIRC que faz a decodificação dos sinais recebidos pelo controle remoto. Neste capítulo ainda são descritos alguns dos principais programas do pacote LIRC.

Por fim, são descritas algumas propostas para a continuação desse projeto. Além disso, na seção apêndices existem informações técnicas adicionais como códigos fontes de programas e arquivos de configuração do *software* LIRC.

1.1. MOTIVAÇÃO

Neste projeto é apresentada uma proposta para controle e automação utilizando para isso um controle remoto infravermelho universal. Os principais fatores que motivaram a realização deste trabalho são os seguintes:

- O controle de diversas aplicações sem a utilização de uma conexão por fios, permitindo uma maior mobilidade.
- Utilização de plataforma com código livre licenciado pela GNU (*General Public License*, Licença Pública Geral).
- O custo-benefício do projeto, já que os componentes utilizados na montagem do circuito são baratos em relação a gama de aplicações que podem ser desenvolvidas.
- Consolidação dos conceitos adquiridos durante o curso de engenharia, como a montagem de circuitos.

1.1. OBJETIVOS

Este trabalho tem como principais objetivos:

1. O funcionamento do circuito infravermelho e sua integração com o *software* livre LIRC. O circuito infravermelho captura os pulsos enviados pelo controle remoto enviando-os para o computador através da interface serial. O *software* LIRC decodifica esses pulsos, transformando-os em sinais válidos para que o computador possa processá-los e apresentá-los em tempo real. Para que isso fosse possível foi necessário o teste de *hardware* e *software* isoladamente.

2. Um segundo objetivo a ser alcançado é o funcionamento de algumas aplicações isoladas do mundo externo, ou seja, que dependa somente do circuito receptor montado e de aplicações existentes no servidor em questão. Tais aplicações são, por exemplo, o controle do cursor do mouse e conseqüentemente a possibilidade da apresentação de slides remotamente. A execução e manipulação de programas existentes, a execução de *scripts* e agendar atividades.

3. O último objetivo é com relação a aplicações conectadas ao mundo externo. Poderíamos citar inúmeras aplicações, mas o foco está voltado para o controle de uma lâmpada. Dessa forma, foi possível ao final da implantação do projeto controlar esse dispositivo através do controle remoto. Esse mesmo tipo de circuito pode ser usado como exemplo para diversos tipos de aplicações envolvendo eletroeletrônica.

2. INFRAVERMELHO

2.1. CONCEITOS INICIAIS

O astrônomo William Herschel descobriu o infravermelho em 1800. William, sabendo que a luz solar continha todas as cores do espectro eletromagnético, queria saber qual cor era responsável pelo aquecimento dos objetos. Dessa forma, Herschel idealizou um experimento usando um prisma, papelão e termômetros com bulbos pretos, medindo as temperaturas de diferentes cores. Assim, concluiu que a temperatura mais elevada ocorria além da luz vermelha e que não era visível ao olho humano. Na época, chamou essa radiação invisível de “raios caloríficos” o que hoje é chamado de radiação infravermelha.

“A radiação infravermelha, está localizada abaixo (por isso o termo “infra”) da faixa do vermelho na luz visível e acima de ondas de rádio no espectro eletromagnético”.¹

Energia infravermelha é uma luz invisível, pois seu comprimento de onda é muito longo para ser detectado pelo olho. Assim sendo, é a parte do espectro eletromagnético percebido como calor. Todos os objetos com temperatura acima de zero absoluta emitem calor, portanto podem ser detectados através da tecnologia infravermelha.

A radiação infravermelha provoca vibração de átomos ou grupos de átomos em um composto. Estas vibrações podem ter amplitudes e velocidades diferentes, ocorrendo em torno das ligações covalentes que une os átomos, ou grupos de átomos.

A radiação visível corresponde a uma estreita faixa do espectro eletromagnético. Para cada cor do espectro, está associado um comprimento de onda. Os comprimentos de

¹ Vanzetti Riccardo. *Practical Applications of Infrared Techniques*. New York: John Wiley & Sons, Inc., 1972. p.7

onda variam desde alguns poucos décimos de bilionésimos de metro, como é o caso das ondas emitidas pelo raio-x, até várias dezenas de quilômetros como é o caso de ondas de radio.



Figura 2.1: Espectro eletromagnético
Fonte: <http://www.las.inpe.br/~cesar/Infrared/conceitos>

Os limites do infravermelho são o visível no lado de menor comprimento de onda e as microondas no lado de maior comprimento de onda.

“Nós podemos notar que a região do infravermelho é dividida em três áreas: o infravermelho próximo, o intermediário, e o infravermelho distante. Essas divisões são arbitrárias em seus limites superiores e inferiores.”²

Na tabela 2.1 são apresentadas as subdivisões do infravermelho de acordo com a faixa de comprimento de onda:

² Vanzetti Riccardo. *Practical Applications of Infrared Techniques*. New York: John Wiley & Sons, Inc., 1972. p.9

Tabela 2.1: Subdivisões do Infravermelho

Fonte: <http://www.las.inpe.br/~cesar/Infrared/conceitos>

Designação	Abreviação	Comprimento de onda (μm)
Infravermelho próximo	NIR	0,75 – 3
Infravermelho médio	MIR	3 – 6
Infravermelho distante	FIR	6 – 15
Infravermelho extremo	XIR	15 – 1000

“As origens da radiação infravermelha podem ser divididas em três grupos, baseados na sua massa: Partículas subatômicas cujas vibrações se enquadram no infravermelho próximo; Partículas atômicas cujas vibrações produzem radiação no infravermelho intermediário; Moléculas, nas quais as vibrações e rotações geram radiação no infravermelho distante.”³

³ Vanzetti Riccardo. *Practical Applications of Infrared Techniques*. New York: John Wiley & Sons, Inc., 1972. p.9

Para transmitir dados os sistemas infravermelhos utilizam frequências muito altas, um pouco abaixo da luz visível no espectro eletromagnético. Igualmente à luz, o sinal infravermelho não pode penetrar em objetos opacos. Assim as transmissões por infravermelho são diretas ou difusas.

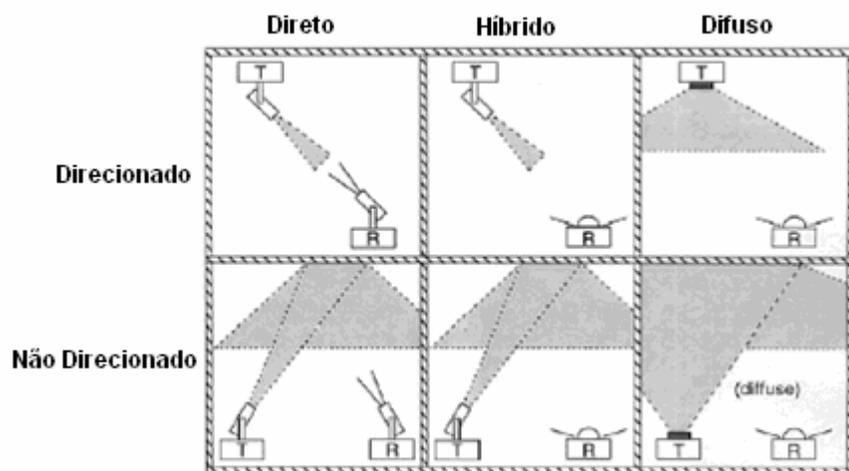


Figura 2.2: Formas de transmissão do infravermelho

Fonte: www.dc.ufscar.br/lsredes/material/Apresent3.pdf

Este trabalho é focado no modo difuso e não direcionado, pois é assim que funciona para controles remotos e aparelhos de DVD, televisão, som e etc. O campo de visão do receptor não precisa ser direcionado e restrito a uma pequena linha imaginária reta com o emissor. Ambos precisam estar no mesmo ambiente.

2.2. INFRAVERMELHO WIRELESS

O infravermelho Wireless é o uso da tecnologia sem fio, através da qual, ondas eletromagnéticas são responsáveis por transmitir sinais (dados) através do meio físico ar.

A tecnologia Wireless infravermelho é usada para pequenas e médias distâncias de comunicação, podendo atingir até vinte metros dependendo das condições do ambiente, do transmissor e receptor utilizados. A largura de banda utilizada é a mesma tanto no emissor quanto no receptor. A comunicação é intermitente, diminuindo as possibilidades de interferência.

2.3. CARACTERÍSTICAS

“Quando a energia infravermelha colide com algum material físico, a porção que é absorvida pelo corpo produz um efeito primário: Aumenta a energia de partículas atômicas e subatômicas da qual a matéria física é formada, o que podemos detectar como o fenômeno de aumento de temperatura. Essa mudança pode ser mensurada por um dos efeitos secundários, como a variação de características físicas (volume, pressão, etc.), ou características químicas (como em fotografia infravermelha), ou em propriedades elétricas (condutividade, coeficiente dielétrico, emissão secundária, etc.). A velocidade com que esses efeitos secundários se manifestam pode ser de algumas horas, necessárias para uma reação química ou de frações de nano segundos para uma segunda emissão”.⁴

⁴ Vanzetti Riccardo. *Practical Applications of Infrared Techniques*. New York: John Wiley & Sons, Inc., 1972. p.30

2.4. DETECÇÃO E NEP

Todos os parâmetros de detectores infravermelhos são medidos em cima de termos de ruído e sinal. Dessa forma, podemos dizer que os detectores podem ser sensíveis ou não a certos ruídos. Os sensores são sensíveis quando técnicas de cancelamento de ruído devem ser aplicadas para uma correta interpretação do sinal válido. As técnicas são as seguintes:

- O sinal deve ser de uma natureza repetitiva, ou seja, pulsos de mesma característica, repetidos em uma determinada frequência.
- A existência de uma técnica para cancelar ruídos. Isso é feito através de um equipamento chamado de Analisador de Junção Semicondutor.

Pode-se então considerar que “um detector pode detectar somente sinais cuja magnitude é maior que o ruído gerado por si próprio”⁵. Conseqüentemente, o nível de ruído se torna o fator limitante, ou seja, quanto menor o ruído gerado pelo receptor, menor será o nível de sinal que ele pode detectar.

NEP (*Noise Equivalent Power*, Potência Equivalente ao Ruído) é a potência do sinal quando esta se iguala à potência do ruído. Assim, quanto menor o valor de NEP, maior será a sensibilidade do detector.

⁵ Vanzetti Riccardo. *Practical Applications of Infrared Techniques*. New York: John Wiley & Sons, Inc., 1972. p.32

$$\text{NEP} = \frac{\Phi_{\text{inc}}}{\left(\frac{V_s}{V_n} \right)} \quad (\text{W}) \quad \text{(Eq. 2.1)}$$

Equação 2.1: NEP

Onde,

Φ_{inc} é a potência radiante incidente (W);

V_s é a tensão *rms* do sinal de saída (V);

V_n é a tensão *rms* produzida pelo ruído (V);

Detectividade espectral (D) é o inverso de NEP. Essa fórmula foi criada com a finalidade de aumentar-se a magnitude com a melhoria da performance, ou seja, quanto maior o valor da detectividade espectral, maior será a sensibilidade do detector.

$$D = \frac{1}{\text{NEP}} \quad (\text{W}^{-1}) \quad \text{(Eq. 2.2)}$$

Equação 2.2: Detectividade espectral

Onde,

D é a detectividade espectral;

NEP é a Potência Equivalente ao Ruído dada na fórmula anterior;

Considerando a dependência do desempenho de um detector em relação a sua área (A_D) e a faixa de ruído, normaliza-se a detectividade espectral e utiliza-se a detectividade específica (D^*).

$$D^* = D\sqrt{A_D} \quad (\text{cmHz}^{1/2}\text{W}^{-1}) \quad \text{(Eq. 2.3)}$$

Equação 2.3: Detectividade específica

Onde,

D^* é a detectividade específica;

D é a detectividade espectral;

A_D é a área do detector;

“Responsividade de um detector é definido como a habilidade de converter radiação em um sinal de tensão de saída. A medida é indicada como R . É expressa em relação à tensão de saída (*rms*) por watt de radiação incidente.”⁶

$$R = \frac{D^*V_n}{\sqrt{A_D\Delta_f}} \quad \text{(Eq. 2.4)}$$

Equação 2.4: Responsividade

Onde,

R é a responsividade;

D^* é a detectividade específica;

V_n é a tensão *rms* produzida pelo ruído;

A_D é a área do detector;

Δ_f é a largura de banda;

⁶ Vanzetti Riccardo. *Practical Applications of Infrared Techniques*. New York: John Wiley & Sons, Inc., 1972. p.33

3. CIRCUITO INFRAVERMELHO

Nesse capítulo foi enfatizada a montagem do circuito receptor infravermelho e o modo de funcionamento. Antes de seguir para a montagem do circuito, se faz necessário o aprendizado de alguns conceitos importantes sobre a comunicação entre o circuito e o microcomputador através do padrão RS-232.

3.1. PADRÃO RS-232

RS-232, também chamado de conexão serial, é um padrão para a troca de dados binários entre um **DTE** (*Data Terminal Equipment*, Equipamento Terminal de Dados) e um **DCE** (*Data Communication Equipment*, Equipamento de Comunicação de Dados). Pode utilizar tanto conectores DB9 (nove pinos) como conectores DB25 (vinte e cinco pinos). O padrão de velocidade para cabos de até 15 metros é de até 115200 bits por segundo.

No protocolo de comunicação RS-232, caracteres são enviados como conjunto de bits. A codificação mais usada é o “start-stop assíncrono” que usa um bit de início, seguido por sete ou oito bits de dados, um bit de paridade e mais um ou dois bits de parada, precisando de no mínimo dez bits para enviar um caractere. Por isso é necessário dividir por dez a taxa de transmissão para obter a velocidade de transmissão.

Esse tipo de comunicação é definido na categoria “single-ended” e foi introduzida no mercado em 1962. Canais independentes são estabelecidos em dois caminhos (Full-Duplex). Os sinais RS232 são representados por níveis de voltagem através

da presença de tensão ou ausência dela. O estado “pronto” tem o nível de sinal negativo e o estado “ativo” tem o sinal positivo.

A interface RS-232 pressupõe um aterramento comum entre o dispositivo terminal de dados (DTE) e o dispositivo de comunicação (DCE). Isso é válido para conexões em pequenas distâncias.

Os dados desse tipo de comunicação são bipolares, o que significa que na faixa entre +3 Volts e + 12 Volts corresponde ao estado ligado (zero lógico) enquanto que na faixa entre -3 Volts e -12 Volts corresponde ao estado desligado (um lógico). Os computadores atualmente, têm a capacidade de ignorar níveis de tensão negativos e aceitam 0 Volts como desligado. Efetivamente, esses computadores precisam de tensões positivas menores do que 3 Volts para entrar em estado lógico igual a zero. Dessa forma, qualquer circuito de 5 Volts com corrente direta (DC) pode alimentar diretamente circuitos RS-232.

O sinal de saída da porta serial nos computadores geralmente varia de +12 Volts a -12 Volts, contudo esse valor não é válido para a maioria dos laptops que possuem uma saída entre +6 Volts e -6Volts. Nesse intervalo existe a chamada área morta (geralmente entre -2 Volts e +2 Volts), designada para absorver os ruídos da linha. Nessa área não existe presença nem ausência de pulsos.

Os *chips* típicos da interface RS-232, também chamados de UARTs (*Universal Asynchronous Receiver Transmitter*, Receptor e Transmissor Assíncrono Universal) permitem até 50 miliamperes (mA) de corrente por pino de saída (output), portanto se o dispositivo a ser conectado precisar de mais de 50 miliamperes (mA) de corrente será necessário usar no mínimo 2 pinos como fonte. Esse não é o caso do circuito receptor infravermelho que precisa de apenas 5 Volts para operar e uma corrente de um pouco mais de 1 miliampere (mA).

O conector DB9, utilizado pelo padrão RS-232, é demonstrado na Figura 3.1 com as funções de cada pino:

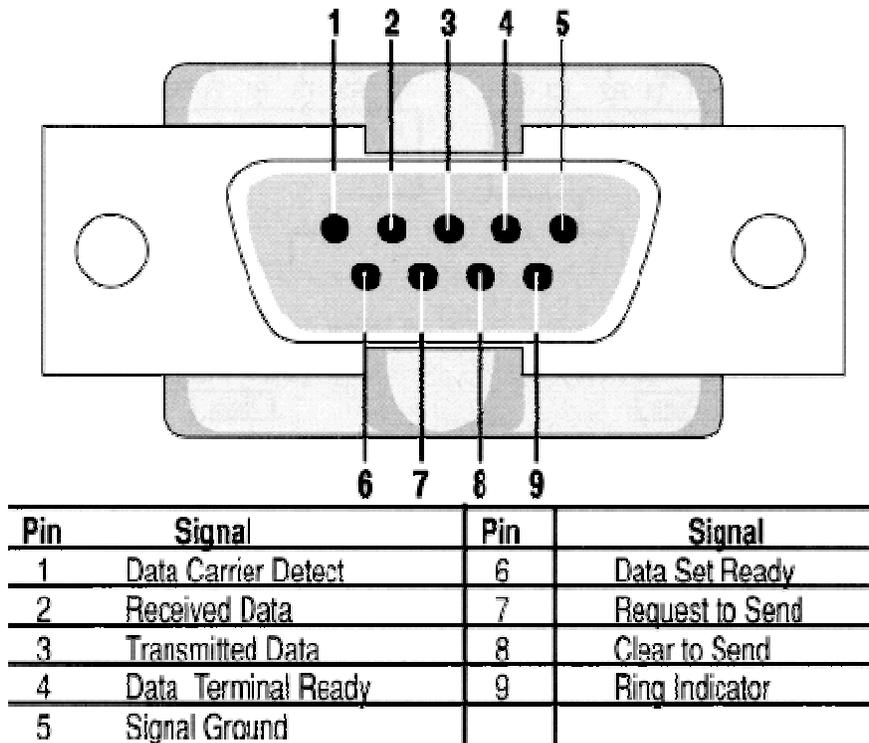


Figura 3.1: Pinos da interface serial RS-232
 Fonte: <http://www.arcelect.com/rs232.htm>



Figura 3.2: Pinos conector RS-232
 Fonte: <http://www.arcelect.com/rs232.htm>

Pino 1 – *Data Carrier Detect* – Detector de portadora de dados. Esse pino é responsável por detectar os sinais transmitidos pelo controle remoto. Indica se uma boa

portadora está sendo recebida pelo **DCE** (*Data Communication Equipment*, Equipamento de Comunicação de Dados).

Pino 2 – *Received Data* – Receptor de dados. Esse pino é responsável por receber dados de um dispositivo **DCE** (*Data Communication Equipment*, Equipamento de Comunicação de Dados). Não será usado no projeto, pois não existe a necessidade de se receber dados do controle, mas apenas pulsos.

Pino 3 – *Transmitted Data* – Transmissor de dados. A função desse pino é transmitir dados para um dispositivo **DCE** (*Data Communication Equipment*, Equipamento de Comunicação de Dados).

Pino 4 – *Data Terminal Ready* – Terminal de dados Pronto – Esse pino indica quando o **DTE** (*Data Terminal Equipment*, Equipamento Terminal de Dados) está pronto para transmitir dados.

Pino 5 – *Signal Ground* – Sinal de Terra. Responsável pelo aterramento do circuito.

Pino 6 – *Data Set Ready* – Pino responsável por indicar se o **DCE** (*Data Communication Equipment*, Equipamento de Comunicação de Dados) está pronto para enviar e receber dados.

Pino 7 – *Request to Send* – Pedido para enviar – Esse pino é responsável por alimentar o circuito conectado na porta serial. Geralmente a alimentação é entorno de 10 a 12 Volts.

Pino 8 – *Clear to Send* – É o pino responsável por indicar se a transmissão pode proceder.

Pino 9 – *Ring Indicator* – Pino responsável por detectar uma chamada telefônica. Este sinal é utilizado por modems que trabalham com linha discada.

Para o projeto, não foi necessário a utilização de todos os pinos. Os pinos usados pelo conector foram os seguintes:

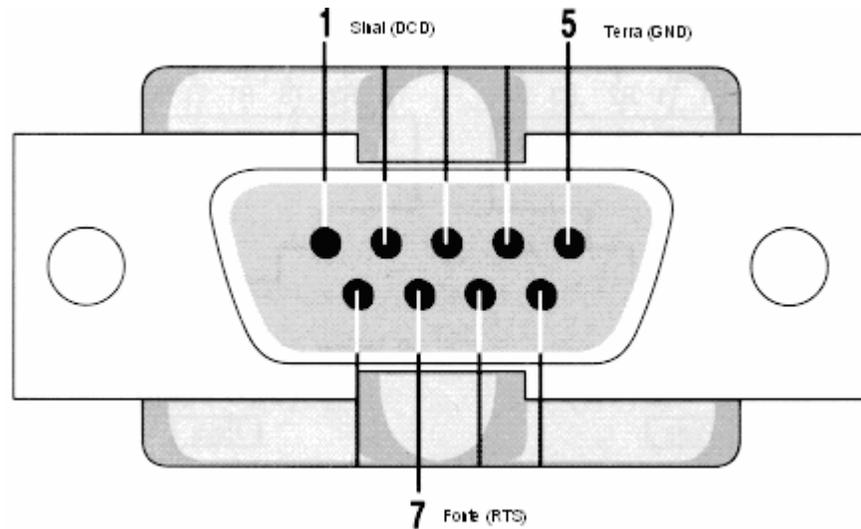


Figura 3.3: Pinos usados no projeto
Fonte: <http://www.arcelect.com/rs232.htm>

- Pino 5 (**GND**) – Terra
- Pino 7 (**RTS**) – Fonte de tensão positiva
- Pino 1 (**DCD**) – Sinal

O pino **DCD** é responsável pela presença ou ausência de sinal, que é detectado pelo dispositivo receptor. A alimentação do circuito fica por conta do pino **RTS**. Essa alimentação é de 12 Volts (no computador usado durante o projeto) e precisa ser estabilizada em 5 Volts para alimentar o circuito. Por último o pino Terra (**GND**) é o aterramento comum tanto para o computador (**DTE**) como para o circuito receptor (**DCE**) e por essa razão se os equipamentos estiverem muito longe, com diferentes fontes de energia, o aterramento se degradará entre os dois dispositivos e a comunicação irá falhar.

Na tabela 3.1 são demonstrados os principais pinos dos conectores DB9 e DB25, suas nomenclaturas e siglas referentes:

Tabela 3.1: Relação de pinos e funções do padrão RS-232

Fonte: <http://www.lirc.org>

Sigla	25 Pinos	9 Pinos	Função
TxD	2	3	Transmite dados
RxD	3	2	Recebe dados
RTS	4	7	Fonte
CTS	5	8	Transmissão procede
DSR	6	6	DCE pronto para enviar e receber
GND	7	5	Terra
DCD	8	1	Portadora
DTR	20	4	DTE pronto para transmitir

3.2. CIRCUITO/HARDWARE

O circuito eletrônico pode ser montado facilmente sem grandes custos, possibilitando que qualquer usuário caseiro que tenha curiosidade ou pensa em facilitar as tarefas cotidianas tome a iniciativa para a montagem do circuito.

Antes de descrever a montagem do circuito deve-se ressaltar que o *driver* da porta serial do software LIRC suporta apenas *chips* do tipo UART (*Universal Asynchronous Receiver Transmitter*, Receptor e Transmissor Assíncrono Universal) compatíveis com o padrão 8250. UART é o *chip* responsável por tratar comunicações seriais assíncronas. Os registradores compatíveis com o padrão 8250 geram interrupções para cada caractere enviado ou recebido, que basicamente informa ao processador que existe um dado a ser processado ou para alimentá-lo.

Existem no mercado diversos tipos de receptores infravermelhos, sendo que o utilizado no projeto foi o *Vishay* TSOP 1738. Esse receptor que trabalha com a frequência de 38 kHz está na faixa de operação do controle remoto usado. Na maioria dos casos esse valor fica por volta de 36 kHz a 38 kHz.

Além do receptor são necessários os seguintes componentes: Um capacitor de 4,7 Microfarad (μF); um diodo 1N4148 que protege o circuito de tensão negativa; um resistor de 4,7 quiloohms (Kohm) que tem como objetivo manter a linha **DCD** em uma tensão válida se o receptor não receber nenhum sinal; um regulador de tensão 78L05 que irá converter a tensão de entrada em exatamente 5 Volts para que o circuito possa operar.

As ligações do circuito são ilustradas na Figura 3.4.

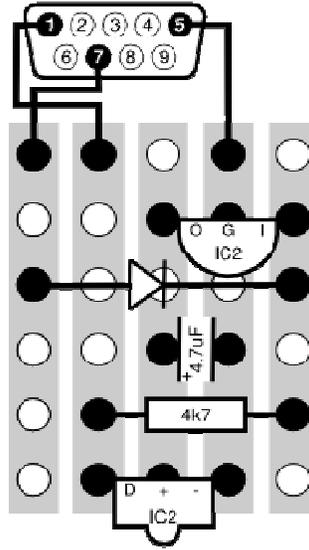


Figura 3.4: Conexões físicas do circuito receptor infravermelho
 Fonte: <http://www.lirc.org>

Para melhor exemplificar os componentes utilizados, a Figura 3.5 demonstra como as conexões foram feitas:

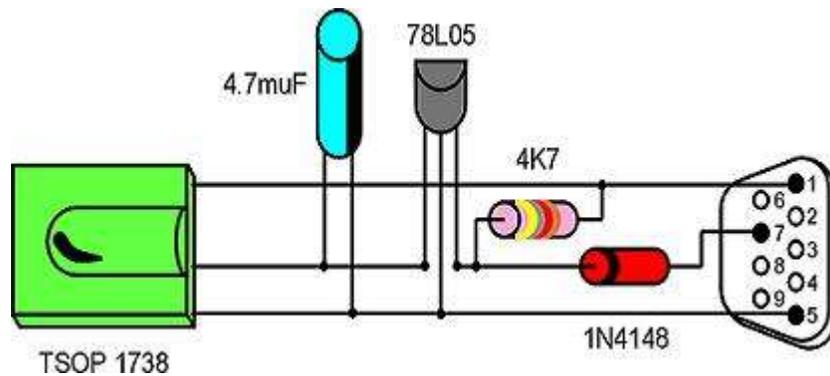


Figura 3.5: Ilustração das conexões entre os componentes
 Fonte: <http://www.lirc.org>

É importante prestar atenção nos pinos do regulador de tensão para que o circuito seja montado corretamente. O pino terra é o intermediário (2) e deve ser ligado em

série ao pino terra do receptor e ao pino terra (**GND**) do conector serial. O pino de entrada (3) deve ser ligado em série ao diodo. Já o de saída (1) será conectado ao pino correspondente à alimentação (**Vcc+**) do receptor. A Figura 3.6 mostra os pinos referentes:

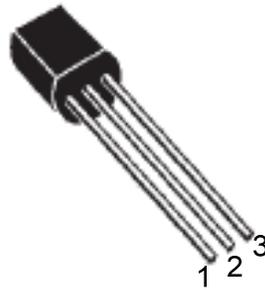


Figura 3.6: Pinos do regulador de tensão 78L05

Fonte: http://www.tranzistoare.ro/datasheets/150/9358_DS.pdf

Pino 1 – Tensão de saída – Conectado em série ao pino de alimentação do receptor TSOP 1738 (**Vcc+**).

Pino 2 – Terra – Conectado em série ao terra do receptor (**GND**) e ao pino terra do conector serial (Pino 5 – **GND**).

Pino 3 – Tensão de entrada – Conectado em série ao diodo, que protege o circuito de tensões negativas.

Os pinos relativos ao receptor TSOP 1738 são: Terra que é ligado ao pino terra da porta serial do microcomputador. O pino de alimentação (**Vs**) do receptor é o intermediário e deve ser conectado à saída do regulador de tensão. O último pino é relativo aos sinais (pulsos) que serão recebidos pelo controle remoto. Esse pino deve ser ligado em série com o resistor e o pino **DCD** da porta serial.

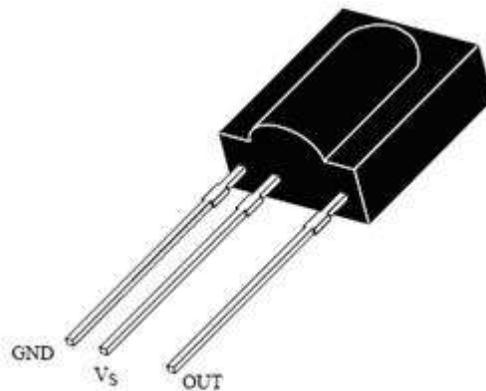


Figura 3.7: Pinos do receptor TSOP 1738
Fonte: <http://irdeo.de/82030.pdf>

3.2.1. Características TSOP 1738

Existem vários tipos de detectores específicos para os mais variados tipos de aplicações infravermelhas. Os detectores quânticos ou fóton detectores são importantes pois serão tratados na implementação física do circuito. “Esses detectores são dispositivos semicondutores cujas características elétricas são uma função do número de cargas que se tornaram disponíveis pela divisão de pares de elétrons causada pela colisão de fótons no material semicondutor.”⁷

Os receptores infravermelhos *Vishay* TSOP 1738 são próprios para sistemas de controle remoto infravermelho. O pino diodo é um semicondutor que opera como um resistor entre a rádio frequência e frequências de microondas. O sinal de saída do receptor pode ser diretamente decodificado por um microprocessador. Esse tipo de receptor suporta a maioria dos códigos de transmissão.

Algumas de suas características principais são: A capacidade de ser um foto detector e um pré-amplificador ao mesmo tempo, possuir um filtro interno para frequências. PCM (*Pulse Codification Modulation*, Modulação por Codificação de Pulso), proteção contra distúrbios elétricos, compatibilidade com tecnologias TTL (*transistor-transistor logic*, lógica transistor-transistor) e CMOS (*complementary metal-oxide semiconductor*, semicondutor de metal-óxido complementar), baixo consumo de energia, alta imunidade em ambientes com muita iluminação, transmissão de dados contínua (acima de 2400 bits/segundo).

⁷ Vanzetti Riccardo. *Practical Applications of Infrared Techniques*. New York: John Wiley & Sons, Inc., 1972. p.45

3.2.2. Funcionamento

A primeira característica a ser levada em conta é que a largura de banda utilizada pelo sistema infravermelho é a mesma tanto no emissor quanto no receptor. A comunicação é intermitente, diminuindo as possibilidades de interferência. Além disso, existe o fator de limitação por parte dos receptores, sendo que os mais sensíveis podem atingir o alcance de até 20 metros dependendo das condições do ambiente.

O circuito é alimentado através da porta serial do microcomputador no pino **RTS** (*Request to Send*, Requisição para enviar). O diodo protege o circuito de tensões negativas. O *driver* instalado na porta serial através do software LIRC muda as características da porta serial durante sua inicialização para que o circuito consiga a tensão positiva necessária para seu funcionamento. Na maioria dos microcomputadores a tensão na porta serial é de aproximadamente 10 Volts. O papel do regulador de tensão é fazer com que essa tensão de entrada se transforme em exatamente 5 Volts. Como a tensão de entrada no regulador 78L05 deve ser no mínimo 2 Volts maior que a tensão de saída, em alguns casos quando a tensão de entrada for menor ou igual a 7 Volts esse circuito não irá funcionar, como é o caso de alguns laptops que possuem tensão de entrada igual ou menor a 6 Volts. Caso a tensão de entrada seja baixa, a solução é trocar o regulador de tensão citado por um regulador de baixa perda, ou seja, esses reguladores conseguem fornecer a tensão de 5 Volts necessária para alimentação do circuito através de uma tensão de entrada no valor de 6 ou 7 Volts.

Quando um pulso é transmitido do controle remoto para o circuito infravermelho, o receptor detecta o sinal e puxa o sinal **DCD** para terra. Essa ausência de tensão é interpretada como um sinal lógico 1 (**DCD=0**) pela porta serial do microcomputador. Na ausência de pulso (sinal), o resistor mantém o sinal do pino **DCD** em um nível de tensão válido, o que representa o sinal lógico 0. Cada botão de um controle remoto possui pulsos diferentes, sendo assim a seqüência de “0”s e “1”s será diferente para cada pulso permitindo com que o software possa aprender o que cada botão corresponde em termos de seqüência binária. A diferença de tensão no circuito varia de 0 Volts a 5 Volts.

Para uma matriz de contatos ou protoboard 5x5 temos o seguinte esquema de ligações:

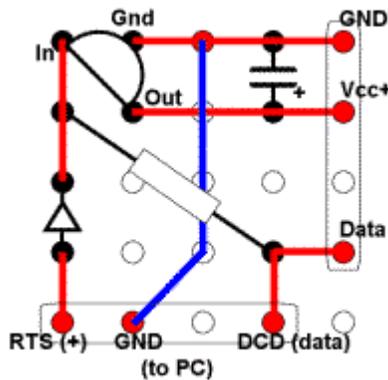


Figura 3.8: Representação das conexões em uma matriz 5x5

Fonte: <http://www.lirc.org>

Após o término da montagem, o circuito receptor infravermelho teve sua configuração conforme a foto ilustrada na Figura 3.9.

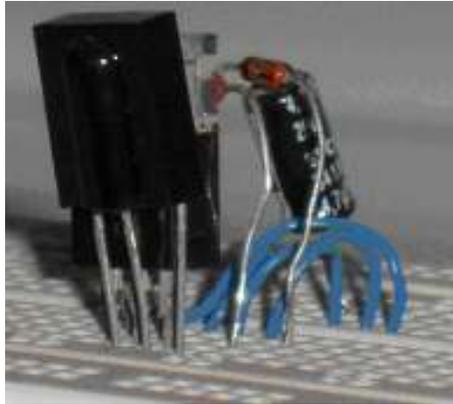


Figura 3.9: Foto do circuito montado no protoboard
Fonte: Autor

4. AUTOMAÇÃO

Neste capítulo são abordadas algumas aplicações para o circuito receptor infravermelho. A intenção é fazer com que algumas tarefas cotidianas sejam automatizadas com o uso do controle remoto.

4.1. Aplicações

A primeira aplicação para o circuito receptor infravermelho será o controle do cursor do *mouse* através do controle remoto. Através dessa funcionalidade será possível realizar apresentações de slides remotamente. Outra aplicação é executar programas, scripts, agendar tarefas e até mesmo desligar o computador através de um botão pré-programado. Já na parte de controle de equipamentos externos será apresentado primeiro o controle de leds conectados à porta paralela do computador. Em automação, ampliaremos esse circuito para controlarmos equipamentos como motores, eletrodomésticos e eletroeletrônicos. A automação a ser demonstrada será o controle de uma lâmpada através do circuito proposto.

4.2. Cursor do mouse

A primeira aplicação é controlar o cursor do *mouse* e seus comandos através do controle remoto. Uma das vantagens é poder passar apresentações de slides sem precisar ficar ao lado do computador. As apresentações se tornam mais dinâmicas e interessantes, podendo ser apresentadas com a interação da platéia.

O software *lircmd* que acompanha o pacote LIRC é responsável por traduzir os comandos do controle remoto em movimentos do *mouse*. Ele faz isso através da simulação da porta do *mouse*. Todos os programas que acompanham o software livre LIRC, necessários para o funcionamento da interface do circuito com o computador serão descritos no próximo capítulo.

4.3. Interface Paralela

Quando se fala em interface paralela (padrão RS-232) logo se pensa em um único hardware: a impressora.

Além da impressora é possível controlar vários tipos de circuitos através da porta paralela. É preciso ter cuidado ao montar circuitos conectados à interface paralela do computador, pois com pequenos erros essa pode vir a ser danificada. Se a porta paralela for integrada com a placa mãe e por algum motivo essa interface vir a ser danificada, o custo para repará-la pode ser muito alto, compensando às vezes até mesmo substituir toda a placa mãe do computador. Por tudo isso, o recomendado é que se compre uma placa **ISA** (*Industry Standard Architecture*, Arquitetura Padrão de Indústria) ou **PCI** (*Peripheral Component Interconnect*, Componente Periférico de Interconexão) com suporte à interface paralela. Caso algum erro ocorra e a placa seja danificada, o custo será muito menor para adquirir outra placa.

O conector da interface paralela possui 25 pinos que na maioria das vezes não serão usados em todos os tipos de hardwares. Muitas vezes, bastam alguns pinos de saída (dados) e pinos referentes à terra para montar variados tipos de circuitos com diversas funcionalidades.

Na Figura 4.1 pode-se visualizar a funcionalidade dos pinos da porta paralela.

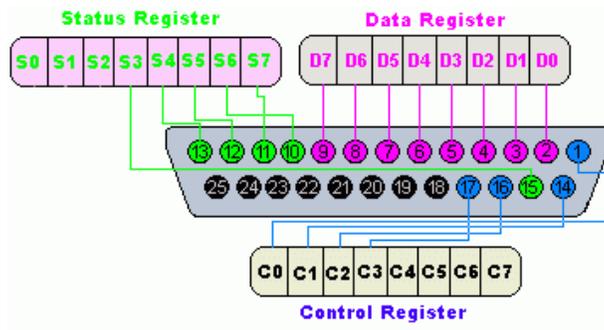


Figura 4.1: Pinos da interface paralela RS-232

Fonte: http://www.linhadecodigo.com.br/artigos/img_artigos/eric_cavalcanti/paralela1.gif

Registrador de dados (*Data Lines*) – Os pinos 2 a 9 são controlados pelo registrador de dados (saída de dados). Através desses pinos pode-se controlar lâmpadas, leds, equipamentos eletrodomésticos, motores e as mais variadas aplicações eletrônicas. Através desse registrador é possível enviar dados do computador para o mundo externo (hardware ligado à porta paralela). O endereço correspondente a esse registrador é o seguinte: 888 (Decimal) ou 378h (Hexadecimal).

Registrador de Status (*Status Lines*) – Esse registrador permite que seja possível receber dados de equipamentos conectados à interface paralela. Seu endereço é o seguinte: 889 (Decimal) ou 379h (Hexadecimal).

Registrador de controle (*Control Lines*) – Tem como função controlar a impressora, embora também seja possível enviar dados para outros tipos de hardware. Tem como endereço: 890 (Decimal) ou 37AH (Hexadecimal).

Os pinos 18 a 25 são usados como terra.

Os pinos controlados pelo registrador de dados operam com tensão de 5 Volts. Dessa forma podemos acionar um desses pinos passando de nível lógico 0 (0 Volts) para

nível lógico 1 (5 Volts). A capacidade de corrente na porta paralela é limitada para poucos miliamperes. Os pinos em questão podem tanto dissipar quanto originar corrente. Esses pinos podem originar até 2,6 miliamperes e dissipar uma corrente de até 24 miliamperes. Eles são formados basicamente por dois transistores empilhados em série entre +5 Volts e terra, sendo que a saída vem da conexão entre eles. A qualquer momento um desses transistores está conduzindo enquanto o outro não conduz.

Para colocar a saída em nível lógico elevado (nível 1), o transistor localizado entre a tensão +5 Volts e a saída conduz, o que origina corrente positiva da saída (output) para terra. Sendo assim, o dispositivo externo conectado entre um dos pinos de saída e um pino terra será energizado.

Originando corrente – Nível lógico igual a 1:

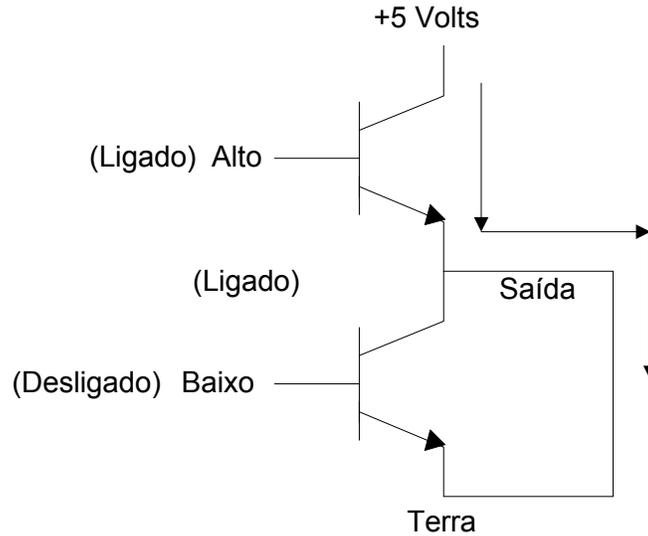


Figura 4.2: Transistores em série originando corrente

Fonte: Autor

Já para colocar a saída em nível lógico baixo (nível 0), somente o transistor localizado entre o terra e a saída conduz, dissipando a corrente para terra.

Dissipando corrente – Nível lógico igual a 0:

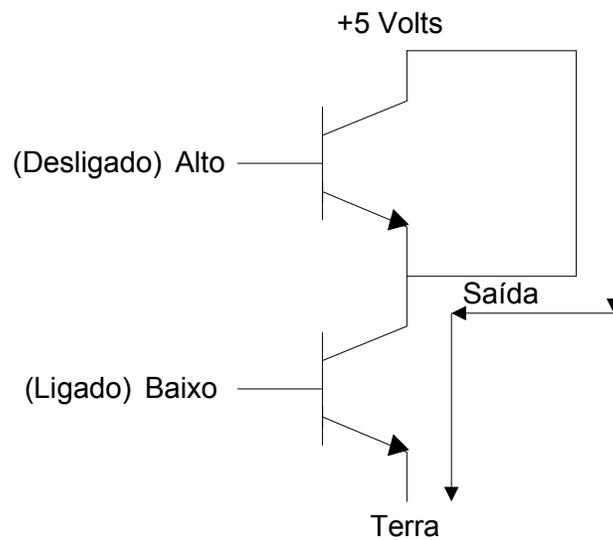


Figura 4.3: Transistores em série dissipando corrente

Fonte: Autor

4.4. Circuito Led

Esse circuito é muito simples de ser montado e os únicos componentes necessários são um resistor de 470 ohms e um led que serão conectados em série. O resistor serve apenas para limitar a corrente retirada da porta paralela para um valor aceitável, fazendo com que o led acenda sem sobrecarregar o chip da porta paralela. Um dos pinos de saída é conectado em série com o resistor que por sua vez é ligado ao ânodo do led. O cátodo é ligado diretamente a algum pino terra.

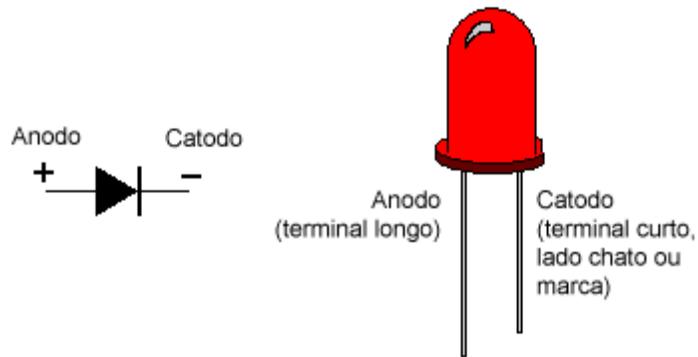


Figura 4.4: Polarização do led

Fonte: <http://www.clubedohardware.com.br/artigos/1147/3>



Figura 4.5: Led

Fonte: <http://en.wikipedia.org/wiki/LED>

Dessa forma o esquema a ser montado é o seguinte:

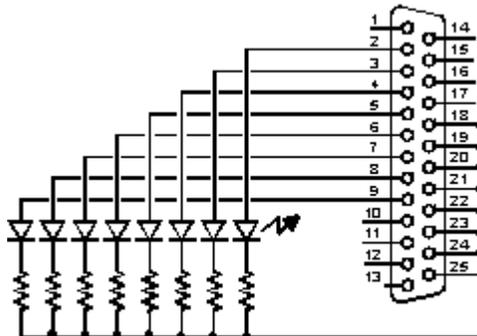


Figura 4.6: Esquema de conexão da porta paralela aos leds
 Fonte: http://www.epanorama.net/circuits/parallel_output.html

O controle dos leds é bastante simples. Ao ativar algum dos pinos correspondente ao registrador de dados a tensão entre este pino e qualquer pino aterrado será igual a 5 Volts, fazendo com que o led fique energizado.

Para manipular, por exemplo, o pino 2 (D0), o comando deve enviar uma saída igual a 1 (Decimal) para habilitá-lo. Dessa forma, o sinal binário enviado será o byte 00000001. Esse sinal habilita o sinal do primeiro pino (D0) e desabilita todos os pinos restantes. Da mesma forma, para habilitar apenas o segundo pino, a saída deve ser acompanhada do número 2 (Decimal), e o byte fica da seguinte forma 00000010. Para desabilitarmos todos os pinos a saída deve ser igual a zero.

O programa para que esta estrutura funcione é exemplificado no apêndice D.

Assim temos a seguinte relação:

Tabela 4.1: Relação dos pinos e bits referentes
 Fonte: Autor

Byte	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
Pino	D7	D6	D5	D4	D3	D2	D1	D0
Decimal	128	64	32	16	8	4	2	1

4.5. Circuito Lâmpada

O circuito com uma lâmpada já é mais complexo e necessita de maiores precauções, já que este é ligado direto à rede elétrica de 110 ou 220 Volts. Essa aplicação já pode ser considerada como uma automação. As pessoas podem ter a comodidade de acender e apagar as luzes de qualquer ambiente da casa sem precisar se deslocar e podem programar através do controle remoto um horário para que todas desliguem, ou ainda, essas tarefas podem ser realizadas remotamente através da internet, bastando que o servidor tenha conexão com a internet.

O circuito a ser demonstrado pode servir de exemplo para outras aplicações. É possível controlar outros tipos de aparelhos como eletrodomésticos, robôs e eletroeletrônicos. O princípio será o mesmo, sempre necessitando de um relé para ativar (chavear) o circuito em questão. Como a porta paralela trabalha com uma tensão de 5 Volts e correntes muito baixas são necessário outras fontes de alimentação para essas automações.

O relé age como uma chave. Os três pinos que descrevem seu funcionamento são descritos a seguir.

- Normally Open (NO).
- Normally Closed (NC).
- Common (C).

- Um circuito integrado 74LS541 para proteger a porta paralela de altas correntes.
- Um led verde para sinalizar que o circuito do relé está ativado.
- Um diodo 1N4148 em paralelo com o relé para prevenir picos de tensão.
- Um resistor de 2,2 Kohms para proteger o transistor de correntes elevadas.
- Um resistor de 470 Ohms para limitar a corrente que passa pelo led.
- Um transistor BD 137 para realizar o chaveamento entre os circuitos.
- Um relé de 12 Volts na bobina que suporte até cargas de 220 Volts/10A.
- Uma fonte de alimentação estabilizada de 12 Volts para alimentar o relé.

Outra forma de montar um circuito semelhante é a seguinte:

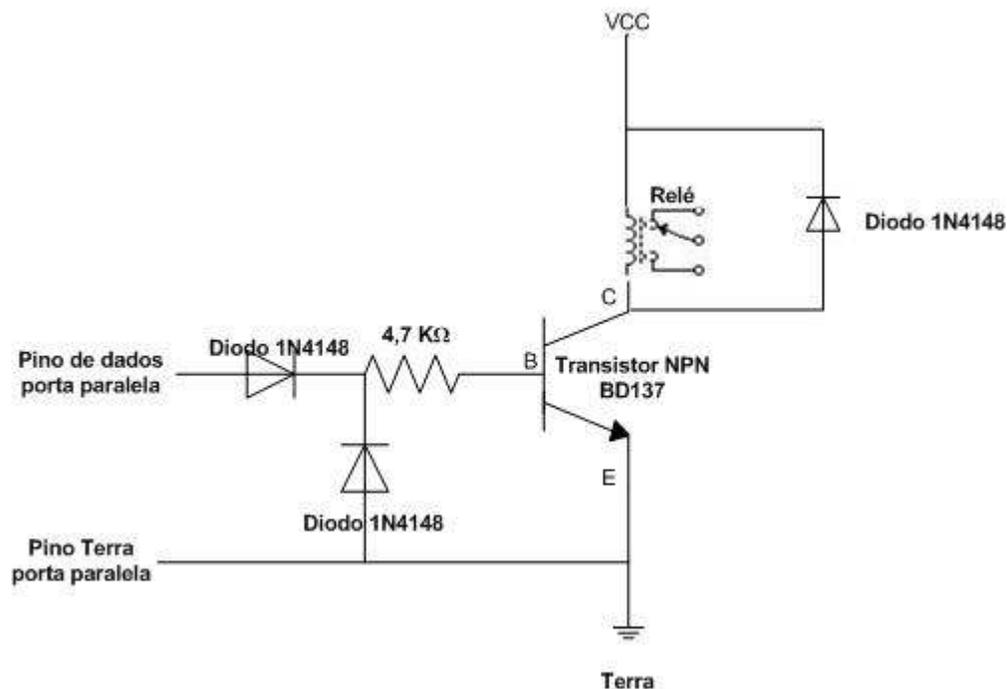


Figura 4.8: Esquema com a mesma funcionalidade do circuito abordado anteriormente
 Fonte: Autor

No circuito apresentado na Figura 4.8, ao invés de usarmos o circuito integrado para proteger a porta paralela de altas correntes, usamos dois diodos 1N4148 entre o pino de saída e terra. Dessa forma, ao receber o sinal do pino de dados, o transistor fecha o circuito do relé (coletor) com o terra (emissor).

Quando a corrente que passa pelo relé é retirada, esse gera um grande pico de tensão já que sua indutância é altíssima. O diodo em paralelo com o relé previne picos de tensão (gerados pelo relé) de danificar o computador.

4.6. Fluxograma

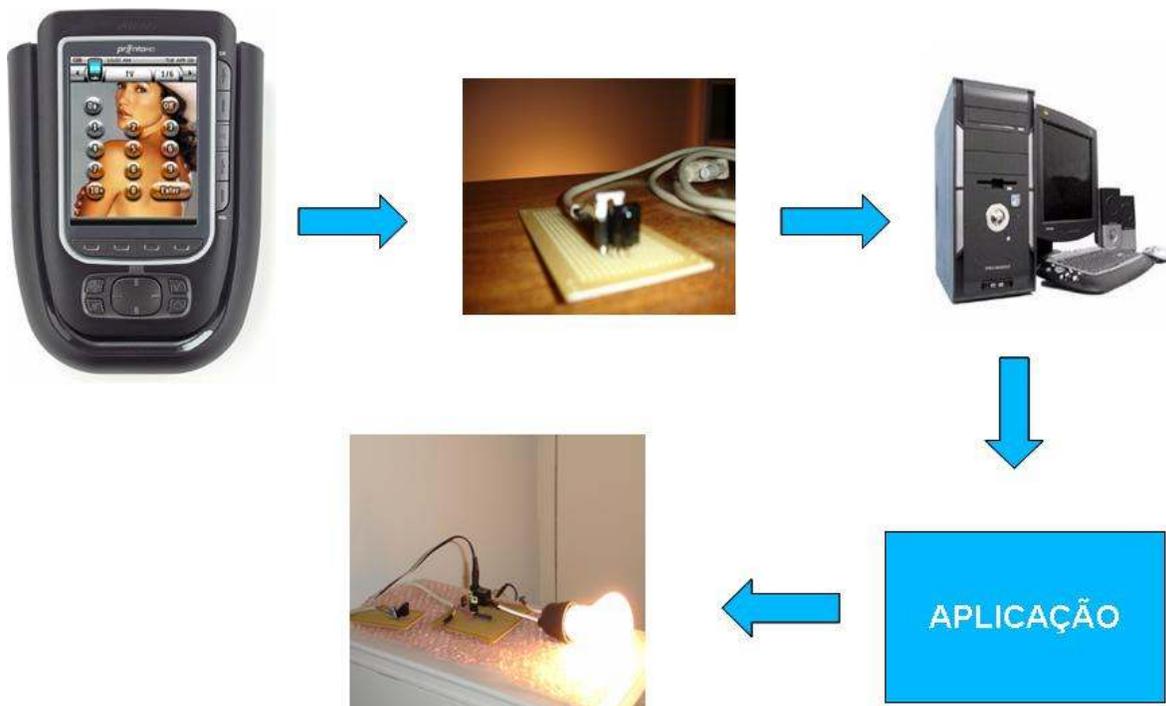


Figura 4.9: Fluxograma
Fonte: Autor

5. LIRC – LINUX INFRARED REMOTE CONTROL

O *software* livre LIRC (*Linux Infrared Remote Control*, Controle Remoto Infravermelho Linux) inicialmente escrito por Ralph Metzler e atualmente mantido por Christoph Bartelmus, tem como principal objetivo decodificar e enviar sinais infravermelhos dos mais variados tipos de controles remotos.

O processo principal do *software* LIRC é o *daemon lircd* que irá decodificar os sinais infravermelhos recebidos pelo *driver* da porta serial e prover esta informação de forma que o processador possa interpretá-la. Além disso, o *software* aceita comandos para que sinais infravermelhos sejam enviados.

Através do LIRC é possível controlar o computador pelo controle remoto, tornando possíveis as mais diversas aplicações como as citadas em capítulos anteriores.

5.1. Instalação do pacote LIRC

Para o correto funcionamento do *software* é necessária certa atenção e compreensão do processo de instalação do pacote. Essas instruções são citadas a seguir.

5.1.1. Dependências

Alguns pacotes são necessários que já estejam instalados para que o programa LIRC obtenha sucesso na sua instalação. Para executar o script de instalação é necessário que o sistema operacional possua o pacote *dialog*. Com esse script de instalação o processo se torna mais fácil e compreensível.

Para compilar e usar todas as ferramentas disponíveis é necessário que se tenha instalado o pacote *svgalib*. Caso no momento da instalação esteja faltando algum pacote de dependência, o script de instalação notificará o usuário quais arquivos estão faltando e consequentemente as devidas ferramentas não serão compiladas.

Antes de iniciar a instalação é necessário que o *kernel* do sistema operacional seja recompilado. O arquivo fonte deve ser independente de qualquer distribuição Linux, pois durante a compilação e instalação do software, o arquivo *Makefile* e a configuração do *kernel* irão corrigir qualquer problema que possa vir a ocorrer. O link www.kernel.org permite adquirir os arquivos fontes das mais variadas versões de *kernel*. No projeto, a versão de *kernel* utilizada é a 2.4.22, considerada estável.

É recomendado que no momento da recompilação do *kernel*, o *driver* da porta serial seja compilado como um módulo do *kernel*, pois normalmente o *driver* da porta serial é carregado no momento que é identificado, fazendo com que os módulos do LIRC não

possam utilizá-lo. A opção *Standard/generic (8250/16550 and compatible UARTs) serial support* deve ser configurada como módulo, para que os módulos do LIRC sejam carregados antes do *driver* da porta serial. Outra solução para isso é usar o comando `setserial /dev/ttySx uart none` para liberar a porta para uso do LIRC.

Antes de compilar o *software* LIRC é necessário escolher qual *driver* utilizar. O script `setup.sh` facilita essa tarefa. Após escolher o *driver* é só salvar a configuração e compilar o *software*. Caso o pacote *dialog* não esteja instalado será necessário configurar os parâmetros manualmente através do comando `./configure (parâmetros)`.

Na Figura 5.1 é demonstrado a opção de *menu* que deve ser configurada para habilitar o suporte ao módulo *Standard/generic (8250/16550 and compatible UARTs) serial support*.

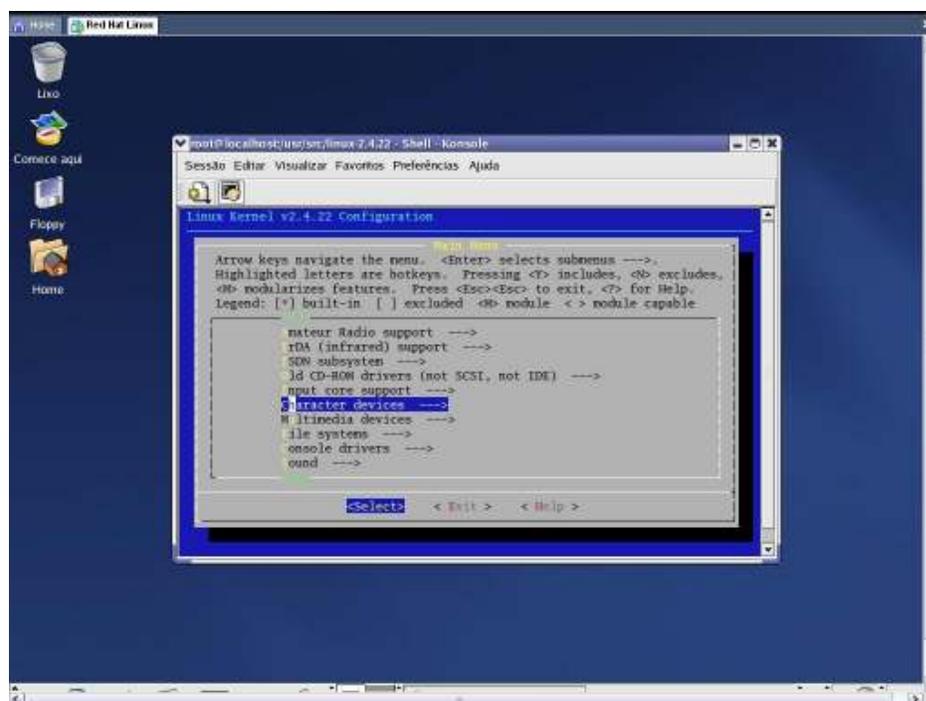


Figura 5.1: Configuração do Kernel
Fonte: Autor

Na Figura 5.2 é mostrado o referido *driver* da porta serial configurado como módulo (M).

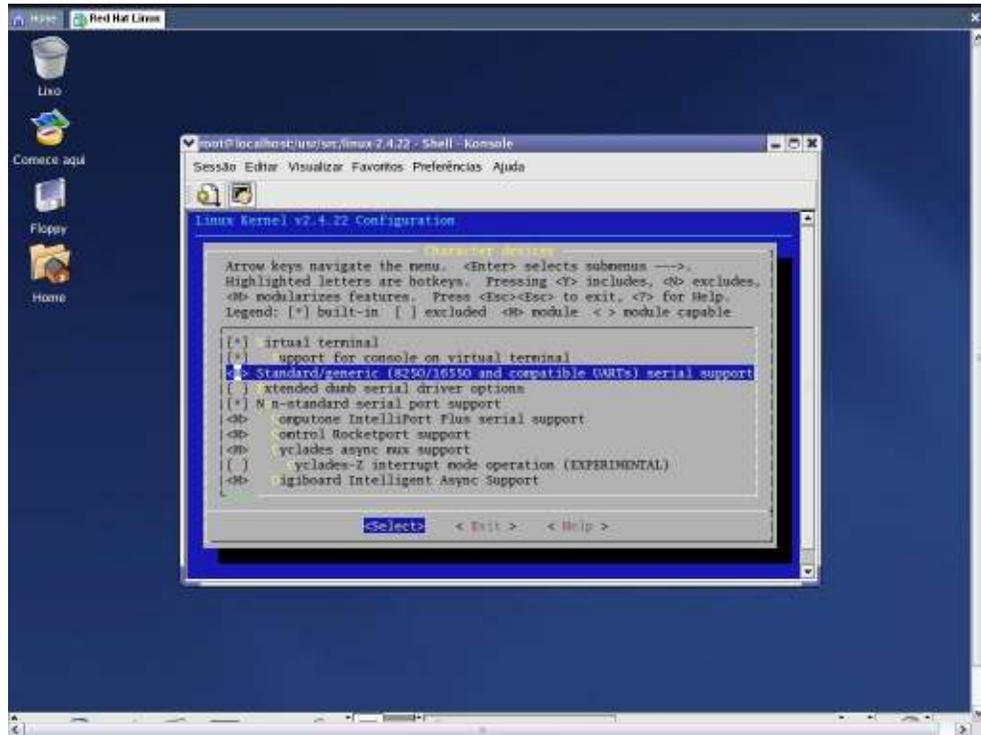


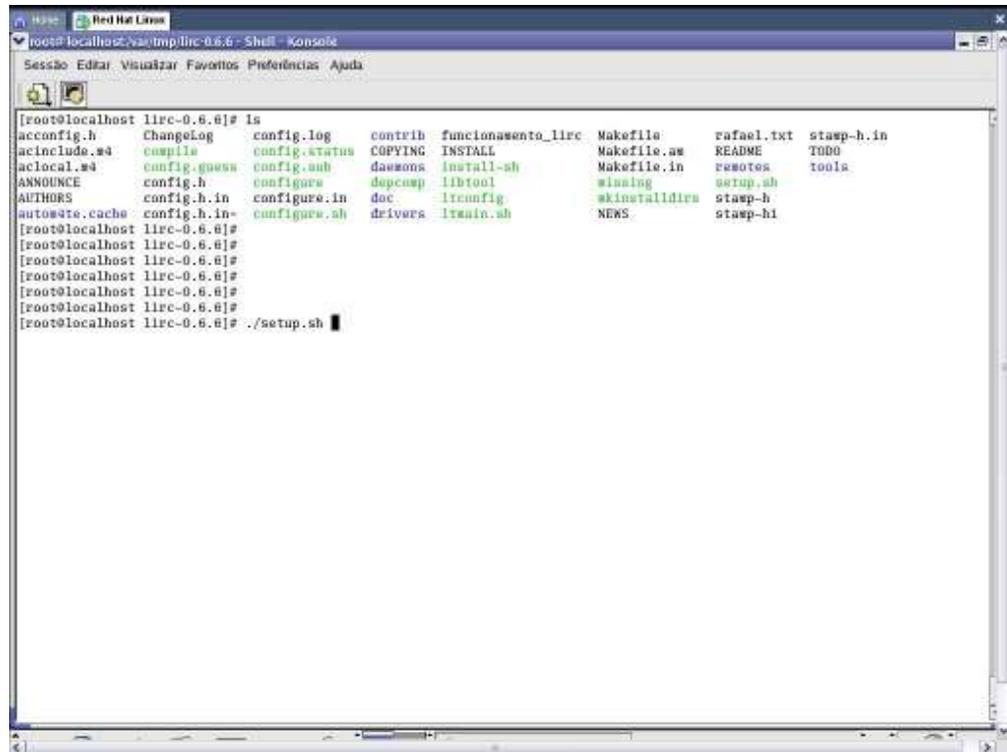
Figura 5.2: Configuração do Kernel

Fonte: Autor

5.1.2. Instalação

Depois de compilado, para instalar o programa com as configurações realizadas basta digitar o comando *make install* Todos os arquivos binários e módulos são instalados nos locais padrões. Os dispositivos são gerados no diretório */dev*.

As próximas figuras demonstram as telas de configuração, compilação e instalação do software LIRC a partir do script *setup.sh*.



```
[root@localhost lirc-0.8.6]# ls
acconfig.h      ChangeLog      config.log      contrib        funcionamento_lirc  Makefile       rafael.txt     stasp-h.in
acinclude.m4    compile        config.status  COPYING       INSTALL           Makefile.am    README        TODO
aclocal.m4      config.guess   config.sub     dswoms        install-sh        Makefile.in    remotes       tools
ANNOUNCE       config.h       configure      depcomp       libtool           missing        setup.sh
AUTHORS        config.h.in   configure.in   doc           lircnfig         skinstalldirs stasp-h
autosetsrc.cache  config.h.in-configure.sh  drivers        ltmain.sh       NEWS
[root@localhost lirc-0.8.6]#
[root@localhost lirc-0.8.6]#
[root@localhost lirc-0.8.6]#
[root@localhost lirc-0.8.6]#
[root@localhost lirc-0.8.6]#
[root@localhost lirc-0.8.6]#
[root@localhost lirc-0.8.6]# ./setup.sh
```

Figura 5.3: Script de configuração LIRC

Fonte: Autor

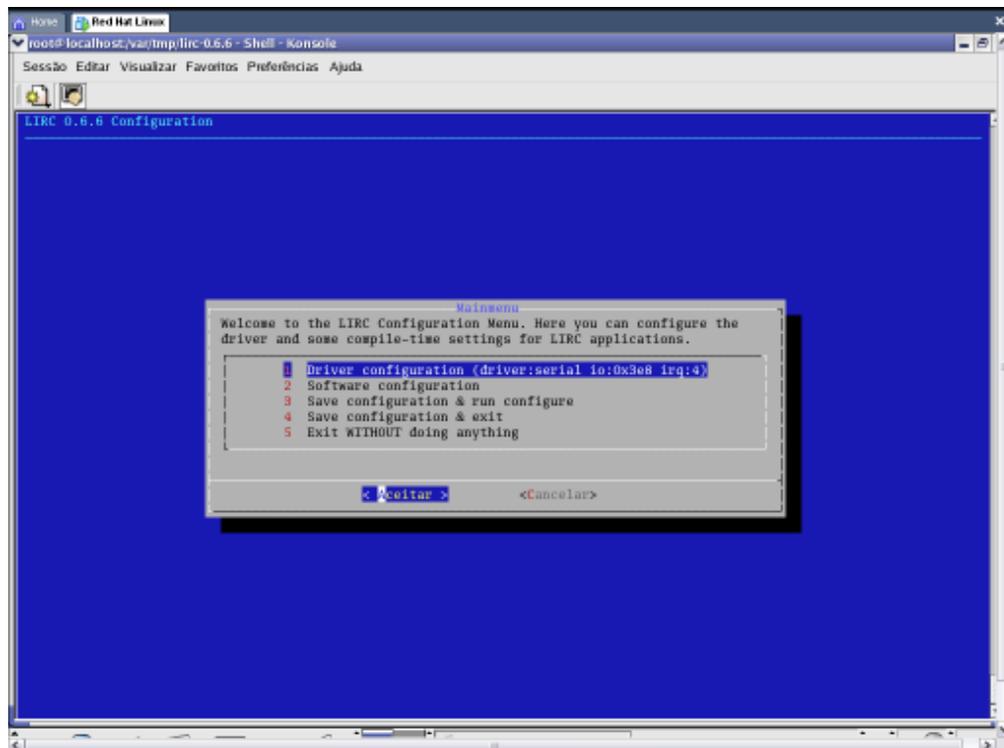


Figura 5.4: Configuração do driver

Fonte: Autor

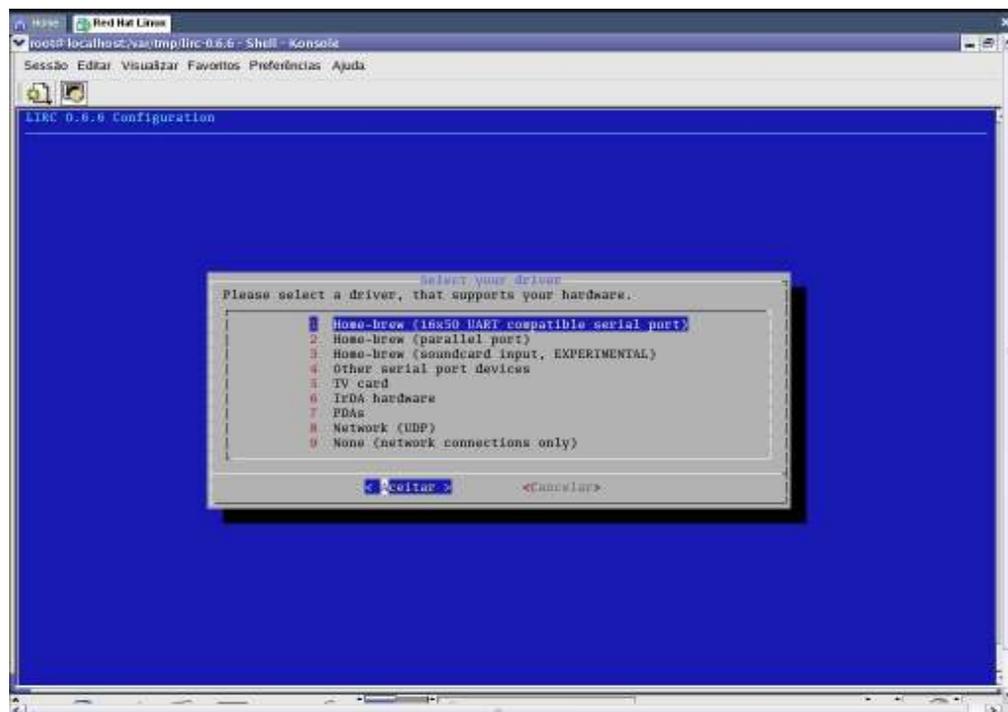


Figura 5.5: Configuração do driver

Fonte: Autor

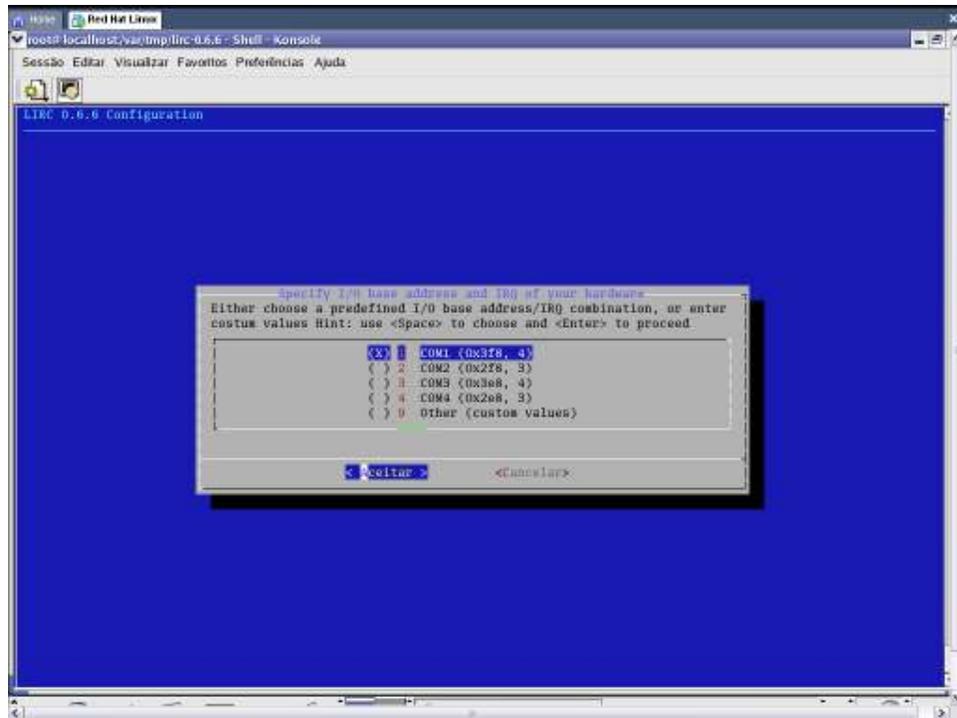


Figura 5.6: Configuração de endereço Entrada/Saída e Interrupção
Fonte: Autor

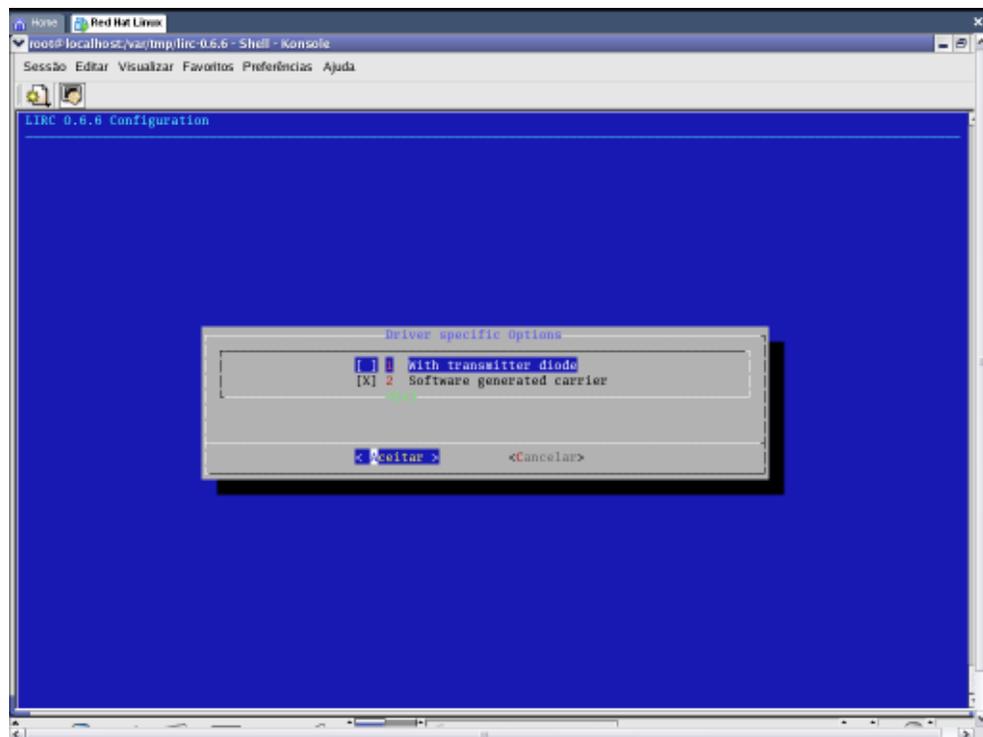


Figura 5.7: Opção para trabalhar apenas com o módulo receptor
Fonte: Autor

Depois de tudo configurado o próximo passo é salvar a configuração e compilar o *software* através da opção *save configuration & run configure*.

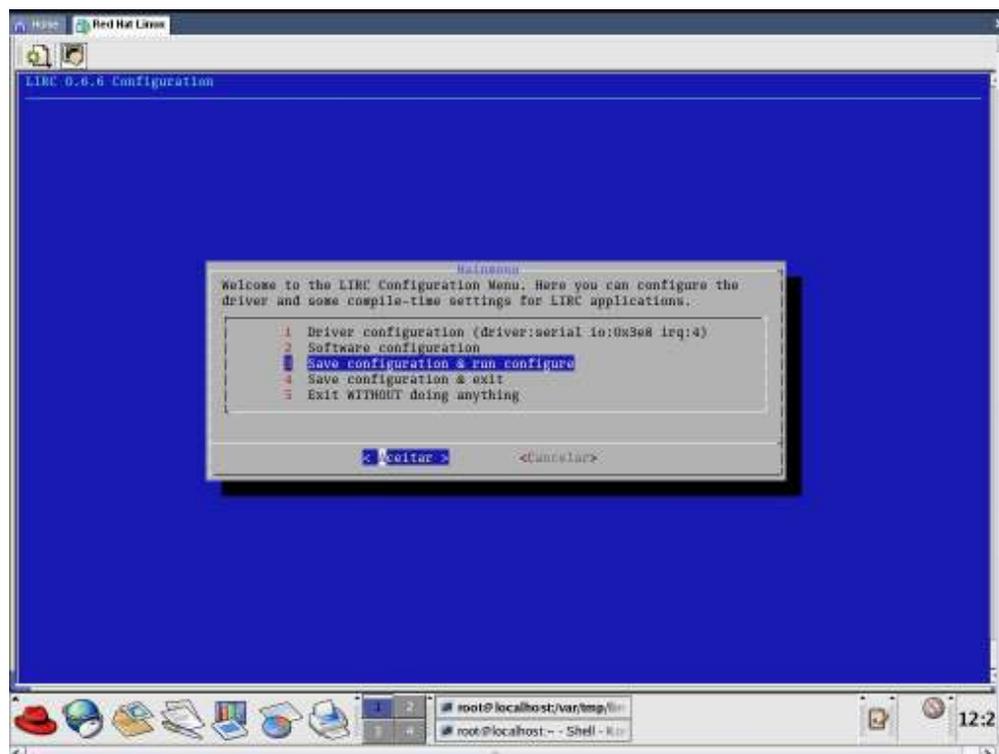
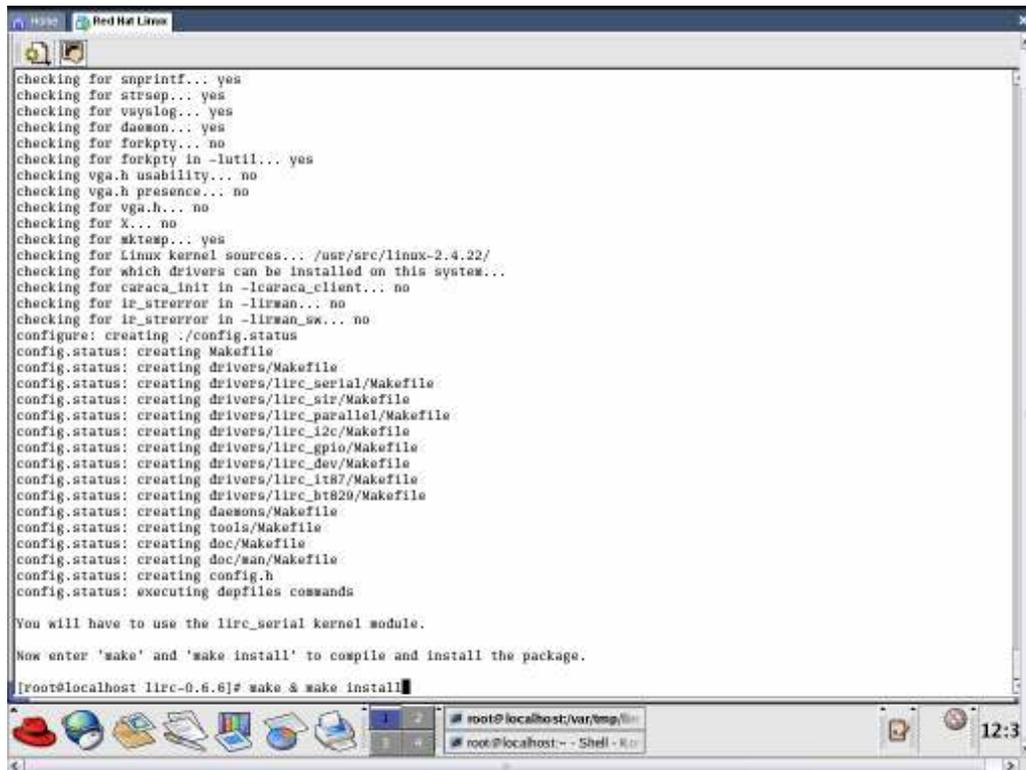


Figura 5.8: Configuração do Lirc
Fonte: Autor

Para compilar e instalar o *software*, após sua configuração deve-se digitar os comandos *make* e *make install* respectivamente.



```
checking for snprintf... yes
checking for strtsep... yes
checking for vsyslog... yes
checking for daemon... yes
checking for forkpty... no
checking for forkpty in -lutil... yes
checking vga.h usability... no
checking vga.h presence... no
checking for vga.h... no
checking for X... no
checking for mktemp... yes
checking for Linux kernel sources... /usr/src/linux-2.4.22/
checking for which drivers can be installed on this system...
checking for caraca_init in -lcaraca_client... no
checking for ir_strerror in -lirwan... no
checking for ir_strerror in -lirwan_sw... no
configure: creating ./config.status
config.status: creating Makefile
config.status: creating drivers/Makefile
config.status: creating drivers/lirc_serial/Makefile
config.status: creating drivers/lirc_sir/Makefile
config.status: creating drivers/lirc_parallel/Makefile
config.status: creating drivers/lirc_i2c/Makefile
config.status: creating drivers/lirc_gpio/Makefile
config.status: creating drivers/lirc_dev/Makefile
config.status: creating drivers/lirc_it87/Makefile
config.status: creating drivers/lirc_bt820/Makefile
config.status: creating daemons/Makefile
config.status: creating tools/Makefile
config.status: creating doc/Makefile
config.status: creating doc/man/Makefile
config.status: creating config.h
config.status: executing depfiles commands

You will have to use the lirc_serial kernel module.

Now enter 'make' and 'make install' to compile and install the package.

[root@localhost lirc-0.6.6]# make & make install
```

Figura 5.9: Compilação e instalação do Lirc

Fonte: Autor

O *lircd* (*LIRC System Daemon*, Daemon do Sistema LIRC), principal programa do *software* LIRC executa todo o trabalho de decodificar sinais infravermelhos. Além disso, o pacote possui um segundo *daemon* chamado *lircmd* que decodifica os comandos do controle remoto para movimentos do *mouse*. Por padrão, esses *daemons* não são principiados quando o sistema operacional é iniciado. Portanto, para não ter que executar toda vez que o computador seja iniciado, o ideal é que seja criado um script de iniciação para os *daemons*.

O próximo passo é configurar o dispositivo `/dev/lircd` (*Socket* para comunicação com o *lircd*) com as permissões para que outros usuários possam se conectar ao *lircd*. As permissões são alteradas utilizando-se o comando `chmod 666 /dev/lircd`. Isso faz com que qualquer usuário do sistema operacional possa utilizar o *socket*.

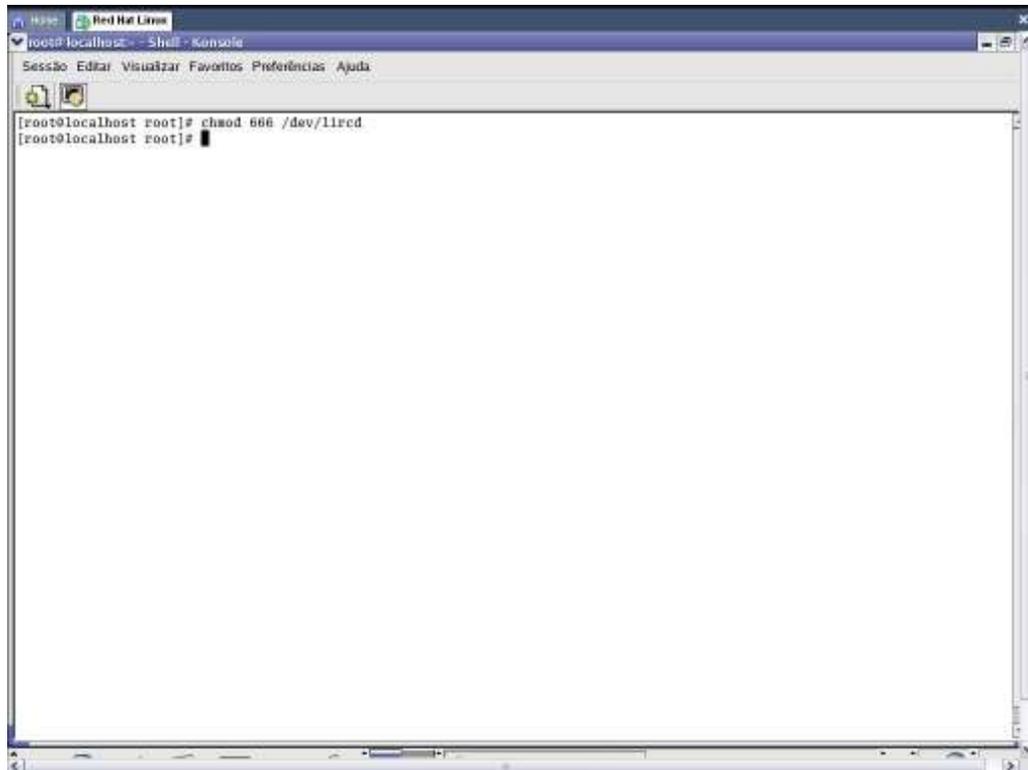
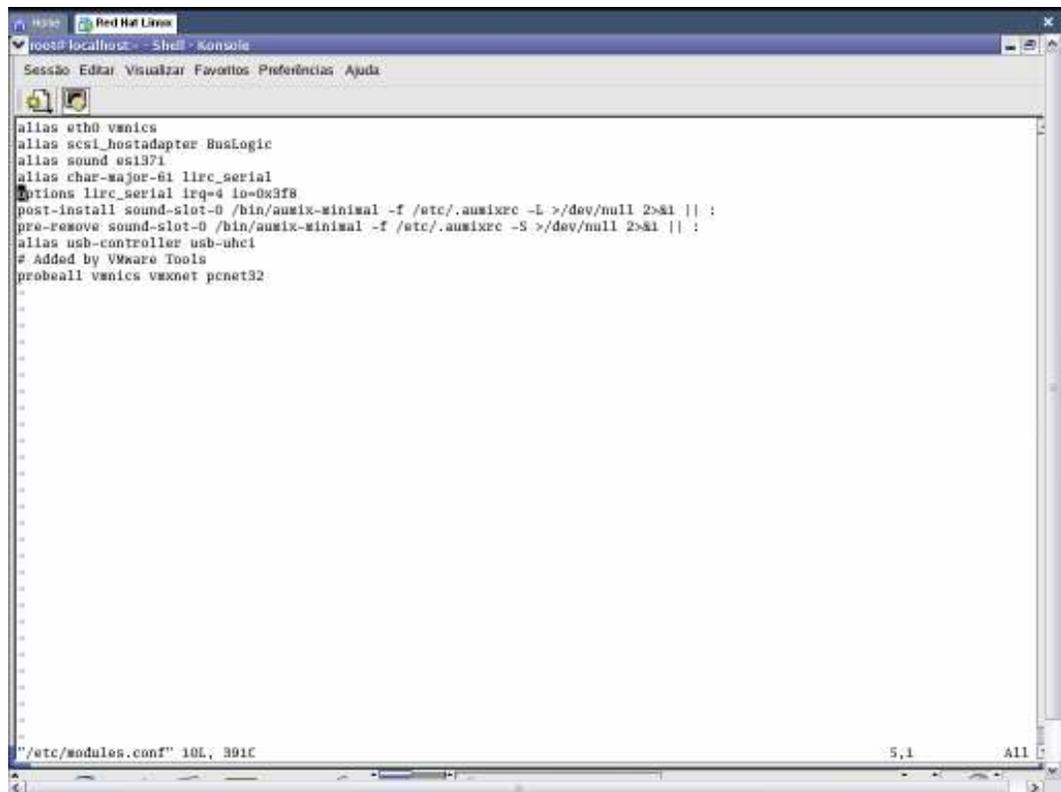


Figura 5.10: Alteração de permissão

Fonte: Autor

O *driver* deve ser especificado no arquivo `/etc/modules.conf`, adicionando o seguinte parâmetro: `alias char-major-61 lirc_serial`

Para especificar endereço IRQ (*Interrupt Request*, Requisição de Interrupção) e I/O (*Input/Output*, Entrada/Saída) para o *driver* da porta serial deve ser adicionada a seguinte linha no arquivo `/etc/modules.conf`: `options lirc_serial irq=4 io=0x3f8`



```
alias eth0 vmnic
alias scsi_hostadapter BusLogic
alias sound es1371
alias char-major-61 lirc_serial
options lirc_serial irq=4 io=0x3f8
post-install sound-slot-0 /bin/auxix-minimal -f /etc/.auxixrc -L >/dev/null 2>&1 || :
pre-remove sound-slot-0 /bin/auxix-minimal -f /etc/.auxixrc -S >/dev/null 2>&1 || :
alias usb-controller usb-uhci
# Added by VMware Tools
probeall vmnic vmxnet pcnet32
```

Figura 5.11: Especificação do driver e endereçamento
Fonte: Autor

Os valores IRQ (*Interrupt Request*, Requisição de Interrupção) e I/O (*Input/Output*, Entrada/Saída) mudam de acordo com as portas seriais que estão em uso. Esses valores podem ser encontrados no **CMOS** (*complementary metal oxide semiconductor*, Semicondutor complementar de óxido metal) ou no diretório */proc* do sistema de arquivo.

Para complementar a configuração, é necessário adicionar a linha */usr/local/lib* em */etc/ld.so.conf* e executar o comando *ldconfig*. Esse arquivo especifica diretórios padrões adicionais, que são usados como *links* para localizar bibliotecas compartilhadas durante a execução do programa.

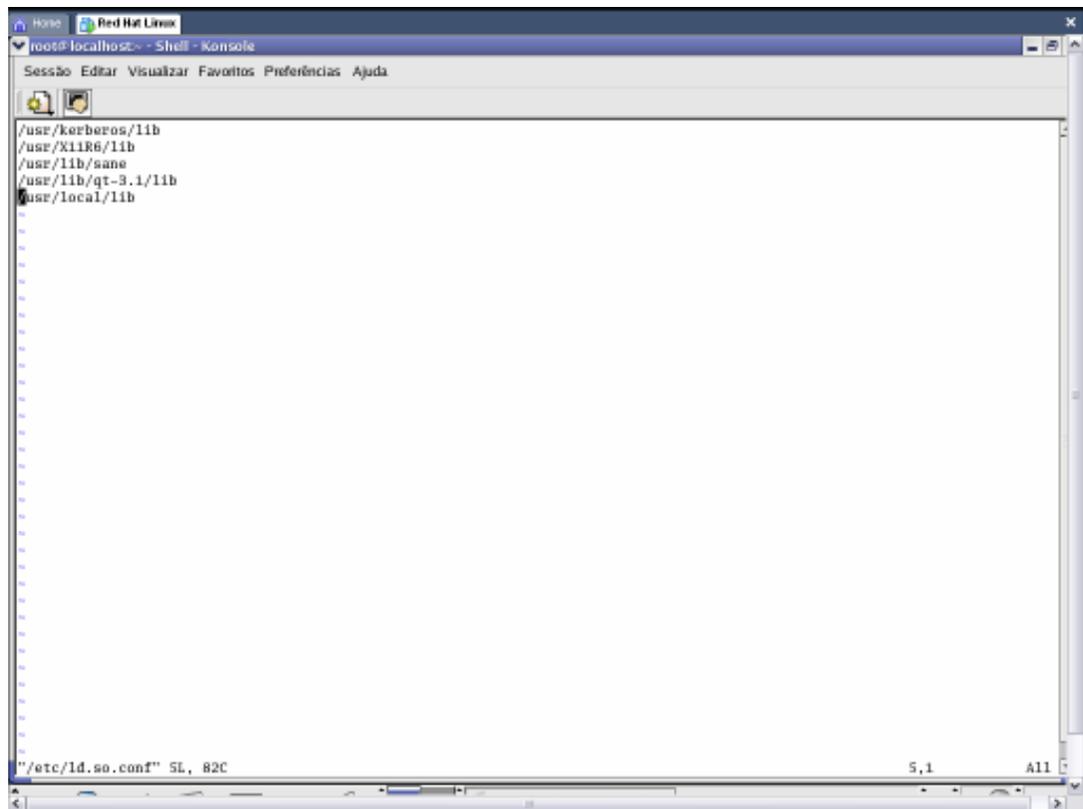


Figura 5.12: Especificação do diretório das bibliotecas

Fonte: Autor

5.1.3. Teste

Para testar o *hardware*, o programa *mode2* deve ser executado. Esse programa carrega o *driver* da porta serial e mostra os sinais infravermelhos na tela em forma de pulsos e espaços. A partir do momento em que o programa está em execução é necessário que alguns botões do controle remoto sejam pressionados para verificar se o circuito foi montado de forma correta.

Caso não apareça nenhum sinal de saída na tela, os procedimentos são: Verificar primeiramente se o *driver* escolhido foi o correto com os parâmetros certos, em seguida verificar se controle remoto está funcionando corretamente e finalmente verificar se o circuito infravermelho está funcionando.

Para executar o *daemon lircd*, basta rodar o comando *lircd /etc/lircd.conf*, onde se encontra o arquivo de configuração do controle remoto. Se o controle remoto a ser usado já possuir um arquivo de configuração, basta copiá-lo para o diretório */etc* com o nome *lircd.conf*. Caso não exista o arquivo para o controle em questão, o programa *irrecord* deve ser usado para “ensinar” o LIRC a entender o controle. Este programa deve ser executado sem que o *daemon* principal *lircd* esteja em execução. O programa é auto-explicativo e pedirá para apertar cada botão do controle informando os nomes correspondentes.

5.1.4. Programas LIRC

5.1.4.1. LIRCD

Daemon que decodifica sinais infravermelhos provendo uma interface uniforme para diversas aplicações. A conexão com o *daemon* é feita através do *socket /dev/lircd*.

5.1.4.2. LIRCMD

Traduz sinais infravermelhos em eventos do *mouse*. Esse *daemon* pode simular os seguintes protocolos usados por diversos tipos de *mouse*: MouseSystems, IntelliMouse e IMPS/2. Seu trabalho é o de capturar os botões recebidos pelo *lircd* e convertê-los em movimentos e ações do *mouse*. Para que isso se torne possível é necessário criar um arquivo de configuração somente para os eventos do *mouse*. Esse arquivo é o *lircmd.conf*.

5.1.4.3. IREXEC

Tem a finalidade de executar programas ao pressionar um botão. Ao escolher esse programa para ser executado alguns outros parâmetros de configuração devem ser definidos de acordo com a necessidade.

5.1.4.4. IRCAT

Imprime caracteres de acordo com a configuração.

5.1.4.5. IRRECORD

Esse programa é fundamental para aprendizagem do controle remoto. O programa é auto-explicativo e pedirá para apertar cada botão do controle informando os nomes correspondentes. Dessa forma, é possível criar o arquivo de configuração do controle remoto caso não exista no diretório *contrib*.

5.1.4.6. MODE2

Esse programa mostra o tamanho dos sinais infravermelhos através de pulsos e espaços. Através desse programa é possível certificar-se de que o circuito foi montado corretamente ou não. Além dele, existem ainda o *smode2* e o *xmode2* que podem ser executados através da interface gráfica mostrando a forma de onda captada pelo circuito.

5.1.5. Diagrama de blocos

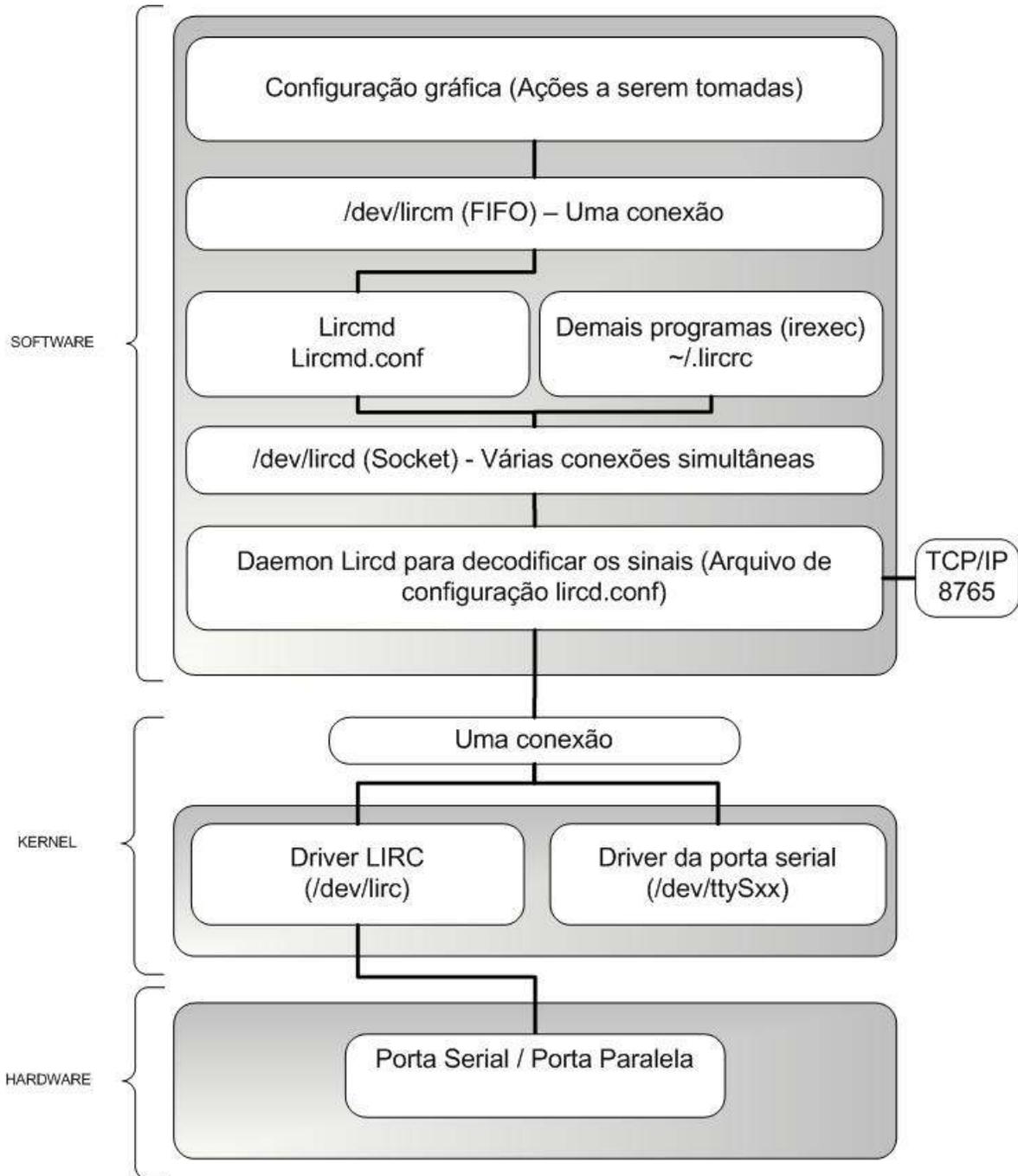


Figura 5.13: Diagrama de blocos LIRC
Fonte: Autor

O funcionamento é descrito baseando-se no diagrama apresentado na Figura 5.13. Como visto, o *driver* da porta serial usado pelo sistema operacional concorre com o *driver* usado pelo sistema LIRC (*Linux Infrared Remote Control*, Controle Remoto Infravermelho Linux). Assim o dispositivo `/dev/lirc` deve estar carregado como *driver* enquanto que o dispositivo `/dev/ttySxx` não deve ser carregado durante a iniciação do sistema operacional. Por isso, que ao configurar os parâmetros do *kernel* devemos compilá-lo com a opção *Standard/generic (8250/16550 and compatible UARTs) serial support* configurada como módulo.

Após o sinal ser recebido pelo *hardware* e transmitido para o *kernel* do sistema operacional através da conexão serial, este é processado pelo *Daemon LIRCD* (processo principal em execução pelo software LIRC) que aceita as conexões na porta TCP 8765 e decodifica os sinais recebidos. Após serem processados, os sinais serão tratados pelo *socket /dev/lircd* que aceita várias conexões simultâneas. A partir daí serão divididos os processos responsáveis por controlar os movimentos e ações do *mouse* e os demais processos que utilizam programas secundários do pacote LIRC para executar diversos programas e tomar certas ações ou agendar determinados *scripts*.

Caso a ação seja destinada ao dispositivo *mouse*, o processo *lircmd* irá tratar esses sinais comparando ao arquivo de configuração *lircmd.conf* fazendo com que os sinais sejam passados para o dispositivo `/dev/lircm` que se encarregará de realizar as ações necessárias. Dessa forma, os movimentos e ações do *mouse* devem necessariamente ser tratados pelo processo *lircmd* (*lircm Daemon*).

Caso os sinais recebidos pelo *hardware* sejam tratados por programas secundários do pacote LIRC estes serão processados conforme as configurações do arquivo *.lircrc* que se encontra no diretório *home* do usuário.

6. CONSIDERAÇÕES FINAIS

6.1. Conclusão

Neste trabalho foi apresentada uma proposta de automação utilizando o controle remoto infravermelho com o objetivo de facilitar as tarefas cotidianas sem fins comerciais. Ao longo do trabalho apresentado pode-se observar que existem uma gama de aplicações a serem desenvolvidas.

Apenas algumas dessas aplicações foram abordadas no projeto. Outros tipos de aplicações como o controle de motores, automatização predial, controle e programação de outros aparelhos eletrônicos podem ser desenvolvidos.

6.2. Dificuldades

Várias dificuldades foram encontradas durante a implementação do projeto.

A primeira foi o fato de que o receptor infravermelho *Vishay* TSOP 1738 não ter sido facilmente encontrado. Este receptor foi adquirido pela *internet* após longo tempo de procura.

Outra dificuldade no início foi a falta de experiência na montagem dos circuitos e posteriormente sua soldagem em placas de circuito integrado.

Além disso, problemas na compilação do *kernel* foram bastante freqüentes até que fosse possível compilá-lo com os devidos módulos.

A implementação do circuito não foi possível no *notebook*. Foi criada uma máquina virtual através do *software vmware*, na qual foi instalado o programa LIRC. O

problema esta relacionado justamente ao fato de que o *notebook* não segue o padrão RS-232 referente à tensão provida pela porta serial. O valor dessa tensão é de 6 Volts no *notebook* utilizado e por isso o regulador de tensão utilizado no projeto não conseguiu prover a tensão de 5 Volts, necessário para o funcionamento do circuito.

Outra grande dificuldade foi encontrar uma boa bibliografia sobre o assunto. A solução foi recorrer para bibliografias internacionais que abrangem melhor o tema.

Apesar de todas as dificuldades, o resultado final correspondeu às expectativas, sendo que foi possível implementar a automatização da lâmpada através do controle remoto.

O custo para montar o circuito receptor é pequeno tendo em vista o grande número de aplicações que podem ser elaboradas, o que deve servir de incentivo para que outros projetos com base nesse possam implementar novas aplicações e até expandir o uso do circuito para que o mesmo tenha outras finalidades.

6.3. Sugestões para Trabalhos Futuros

Ampliação e desenvolvimento de outras aplicações, como por exemplo, a automatização de uma casa com vários circuitos distribuídos pelos mais diversos ambientes e conectados a um servidor central. Dessa forma, é possível realizar diversas tarefas com apenas um ou mais controles remotos distribuídos pela casa. Com isso, seria possível abrir uma porta ao acionar uma trava elétrica ou programar o microondas para ligar certo horário. Outra possibilidade é acionar pequenos motores para fechar ou abrir persianas. Poderia ser elaborado ainda um *display* acoplado ao controle remoto para receber informações de status dos mais diversos objetos da casa.

Ainda nessa linha, é possível adicionar a esse circuito um led transmissor de infravermelho para que se possa controlar outros dispositivos como o aparelho de TV, DVD, som e demais aparelhos eletrônicos comandados por infravermelho.

Outra sugestão é modificar o circuito para implementar funções de transferência de dados através do protocolo IRDA (*The Infrared Data Association*, Associação de Dados Infravermelho), assim seria possível transferir dados de **PDAS** (*Personal Digital Assistants*, Assitência Digital Pessoal) ou celulares para o computador e vice-versa.

REFERÊNCIAS BIBLIOGRÁFICAS

ARC Electronics. RS232 Data Interface a tutorial on Data Interface and cables. Disponível em <<http://www.arcelect.com/rs232.htm>>. Acesso em: out 2005

BARTELMUS, CRISTOPH. LIRC Linux Infrared Remote Control. Disponível em <<http://www.lirc.org>>. Acesso em: jun 2006

BOSCHETTI, CESAR. Conceitos Fundamentais: Radiação, Energia e Ondas. Disponível em <<http://www.las.inpe.br/~cesar/Infrared/conceitos.htm>>. Acesso em: set 2005

ENGDAHL, TOMI. Parallel port interfacing made easy: Simple circuits and programs to show how to use PC parallel port output capabilities. Disponível em <http://www.epanorama.net/circuits/parallel_output.html>. Acesso em: fev 2006

FAGUNDES, EDUARDO. Fundamentos de Wireless LAN. Disponível em <http://www.efagundes.com/Artigos/Wireless_LAN.htm>. Acesso em: jul 2005

FLIR SYSTEMS. História do Infravermelho. Disponível em <http://www.flirthermography.com/brazil/about/ir_history.asp>. Acesso em: ago 2005

MEDEIROS, MIGUEL. Infravermelho (IV). Disponível em <<http://www.quiprocura.net/infra.htm>>. Acesso em: set 2005

Mobilecomms-technology.com. IRDA (INFRARED DATA ASSOCIATION), INTERNATIONAL. Disponível em <<http://www.mobilecomms-technology.com/projects/irda/>>. Acesso em: mar 2006

NOBILI, PEDRO. INFRAVERMELHO. Disponível em <www.dc.ufscar.br/lsredes/material/Apresent3.pdf>. Acesso em: abr 2006

PIROPO, BENITO. Jornal o Estado de Minas. Artigo Conexões sem fio. Disponível em <<http://www.bpiropo.com.br/em20031215.htm>>. Acesso em: set 2005

ROGÉRIO, ANTONIO. RORGERCOM Pesquisa e Desenvolvimento. Disponível em <<http://www.rogercom.com/>>. Acesso em: mai 2006

SearchNetworking.com. IR Wireless. Disponível em: <http://searchnetworking.techtarget.com/sDefinition/0,,sid7_gci293474,00.html?offer=wlanq>. Acesso em: ago 2005

SILVA, ADAILTON. As tecnologias de Redes Wireless. Disponível em <<http://www.rnp.br/newsgen/9805/wireless.html#ng-ieee>>. Acesso em: ago 2005

STRANGIO, CHRISTOPHER E. The RS232 Standard. Disponível em <http://www.camiresearch.com/Data_Com_Basics/RS232_standard.html>. Acesso em: out 2005.

SUMMER, W. **Ultra-violet and infra-red engineering**. London: Isaac Pitman, 1962.

TORRES, GABIEL, LIMA, CÁSSIO. Construindo Protótipos Usando a Porta Paralela. Disponível em <<http://www.clubedohardware.com.br/printpage/1147>>. Acesso em: fev 2006

VANZETTI, Riccardo. **Practical applications of infrared techniques**: A new tool in a new di mension for problem solving. New york: J Wiley, 1972.

VISHAY TELEFUNKEN. Photo Modules for PCM Remote Control Systems. Disponível em <<http://jap.hu/electronic/tsop17.pdf>>. Acesso em: ago 2005

Wikipédia A enciclopédia livre. Artigo RS-232. Disponível em <<http://pt.wikipedia.org/wiki/RS-232>>. Acesso em: out 2005

7. APÊNDICES

7.1. Formato

A seguir são descritos os formatos de arquivos de configuração e do código fonte dos programas tratados no projeto, além dos esquemas elétricos de cada circuito.

APÊNDICE A

7.1.1. LIRCD.CONF

O arquivo *lircd.conf* que foi aprendido através do programa *irrecord*, possui o seguinte formato:

```
# Please make this file available to others
# by sending it to <lirc@bartelmus.de>
# this config file was automatically generated
# using lirc-0.6.6(serial) on Thu Aug 12 16:53:23 2004
# contributed by brand: lircd.conf.panasonic.conf
# model no. of remote control:
# devices being controlled by this remote:

begin remote

name lircd.conf.panasonic.conf

bits      32

flags SPACE_ENC

eps       30

aeps     100

header    3533 1704

one       470 1268

zero     470 399
```

ptrail 474
pre_data_bits 16
pre_data 0x4004
gap 74353
min_repeat 1
toggle_bit 0

begin codes

1	0x000000009000801
2	0x000000009008881
3	0x000000009004841
4	0x00000000900C8C1
5	0x000000009002821
6	0x00000000900A8A1
7	0x000000009006861
8	0x00000000900E8E1
9	0x000000009001811
0	0x000000009009891
r-tune	0x000000009801C95
add/dlt	0x000000009808C05
input	0x00000000900030A
book_mark	0x000000009000900
memory	0x00000000900CAC3
reset	0x000000009002A23

display	0x000000009009C95
track-1	0x000000009004D44
track-2	0x000000009008D84
speed	0x0000000098050D9
vol_up	0x000000001000405
vol_down	0x000000001008485
ch_up	0x000000009002C25
ch_down	0x00000000900ACA5
mute	0x000000001004C4D
rotate	0x0000000098020A9
prog	0x000000009808009
clear	0x00000000980A029
menu	0x000000009806AE3
rec/time	0x000000009001019
stop	0x000000009000009
play	0x000000009005059
ff	0x00000000900C0C9
rew	0x000000009004049
pause/still	0x000000009006069
slow	0x00000000900F0F9
skip	0x000000009001F16
sleep	0x00000000100F0F1
on_timer	0x00000000100D0D1
power	0x00000000900BCB5

end codes

end remote

Cada botão é representado por um código em hexadecimal. “Os bits 0-23 contêm o tamanho pulso/espaco em microssegundos. O bit 24 é representado por zero quando significa espaco ou um quando significa pulso. Todos os outros bits não são usados”⁸

⁸ Conforme BARTELMUS, Christoph. <http://www.lirc.org> Acesso em 22 maio/ 2006 às 23h37.

APÊNDICE B

7.1.2. LIRCMD

Esse *daemon* é usado para emular um *mouse* com o controle remoto. Dependendo do arquivo de configuração o programa converte sinais infravermelhos em movimentos e ações do *mouse*. Atualmente suporta três tipos de protocolos usados por *mouse*: MouseSystems, IntelliMouse e IMPS/2.

O *lircmd* pode ser usado com dois tipos de aplicações gráficas: X11 e gpm. O projeto utilizou a interface gráfica X11 versão 4. Os seguintes parâmetros devem ser adicionados na configuração do arquivo */etc/X11/XF86Config*.

```
Section "InputDevice"
    Identifier "LIRC-Mouse"
    Driver     "mouse"
    Option    "Device" "/dev/lircm"
    Option    "Protocol" "IMPS/2"
    Option    "SendCoreEvents"
    Option    "Buttons" "5"
    Option    "ZAxisMapping" "4 5"
EndSection
```

Além disso, a linha destacada em negrito deve ser adicionada na seção

ServerLayout:

```
Section "ServerLayout"
    Identifier "Default Layout"
    Screen 0 "Screen0" 0 0
    InputDevice "Mouse0" "CorePointer"
    InputDevice "Keyboard0" "CoreKeyboard"
    InputDevice "DevInputMice" "AlwaysCore"
    InputDevice "LIRC-Mouse"
EndSection
```

O arquivo de configuração *lircmd.conf* utilizado para emular os movimentos do

mouse possui o seguinte formato:

```
#
# lircmd config file
#

PROTOCOL IMPS/2
#PROTOCOL MouseSystems

# ACCELERATOR start max multiplier

ACCELERATOR 2 30 5

ACTIVATE lircd.conf.panasonic.conf menu

MOVE_N lircd.conf.panasonic.conf 2
MOVE_NE lircd.conf.panasonic.conf 3
MOVE_E lircd.conf.panasonic.conf 6
MOVE_SE lircd.conf.panasonic.conf 9
MOVE_S lircd.conf.panasonic.conf 8
MOVE_SW lircd.conf.panasonic.conf 7
MOVE_W lircd.conf.panasonic.conf 4
MOVE_NW lircd.conf.panasonic.conf 1
#MOVE_N lircd.conf.panasonic.conf ARROW_UP
#MOVE_E lircd.conf.panasonic.conf ARROW_RIGHT
#MOVE_S lircd.conf.panasonic.conf ARROW_DOWN
#MOVE_W lircd.conf.panasonic.conf ARROW_LEFT
MOVE_IN lircd.conf.panasonic.conf ch_up
MOVE_OUT lircd.conf.panasonic.conf ch_down

BUTTON1_TOGGLE lircd.conf.panasonic.conf 5
BUTTON2_TOGGLE lircd.conf.panasonic.conf 0
BUTTON3_TOGGLE lircd.conf.panasonic.conf play

BUTTON1_CLICK lircd.conf.panasonic.conf rotate
BUTTON2_CLICK lircd.conf.panasonic.conf mute
BUTTON3_CLICK lircd.conf.panasonic.conf prog

# BUTTONx_CLICK, BUTTONx_UP, BUTTONx_DOWN are also possible
```

Onde:

PROTOCOL – Este campo refere-se ao protocolo usado pelo mouse. As três opções são: MouseSystems, IntelliMouse and IMPS/2

ACCELERATOR – Neste campo é possível alterar a velocidade do cursor de acordo com o tempo em que o botão fica pressionado. O primeiro valor (start) corresponde ao ponto inicial de aceleração. O segundo campo (max) especifica o número máximo de pixels o cursor pode mover com um único comando. Já o terceiro campo (multiplier) é o número que multiplicado pela quantidade de repetições de um mesmo sinal dá como resultado o número de pixels que o cursor deve mover-se.

ACTIVATE – Tem como parâmetros o nome do controle remoto e o referido botão para ativar o uso do controle para emular os movimentos do mouse.

TOGGLE_ACTIVATE – Tem a mesma função de ACTIVATE. A diferença é que o comando TOGGLE_ACTIVATE se especifica o botão do controle remoto que serve tanto para iniciar o modo de emulação do mouse quanto para parar esse modo. Já no comando ACTIVATE, basta apenas pressionar um botão que não é usado pelo driver lircmd para sair do modo de emulação do mouse.

MOVE_[N [E | W] | E | S [E | W] | W] – Especifica o botão que faz com que o cursor do mouse se mova para um determinado ponto cardeal. O exemplo abaixo deixa claro que o botão “2” do controle remoto panasonic faz com que o mouse se mova na direção norte, ou seja, para cima.

MOVE_[IN|OUT] – Esse comando funciona apenas com os protocolos IntelliMouse e IMPS/2, emulando as ações da “rodinha” do mouse.

BUTTONx_[CLICK | DOWN | UP | TOGGLE] – Simula eventos do mouse de acordo com a esquerda (x=1), meio (x=2) ou direita (x=3).

APÊNDICE C

7.1.3. .LIRCRC

O arquivo *.lircrc* deve existir para que seja possível executar diversos tipos de comandos através do controle remoto. A sintaxe desse arquivo é descrita abaixo:

```
begin
  prog      = ...
  remote    = ...
  button    = ...
  repeat    = ...
  delay     = ...
  config    = ...
  mode      = ...
  flags     = ...
end
```

“Essa sintaxe diz o seguinte: Qual o programa (prog) deve fazer o que (config, mode, flags) se você pressionar um certo botão (remote, button) por um tempo específico (repeat, delay).”⁹

PROG – Indica qual o programa a ser executado.

REMOTE – Indica o controle remoto a ser usado para o determinado comando.

BUTTON – Especifica qual o botão deve ser pressionado para a execução do programa. Sequências de botões podem ser especificadas para a execução de um programa, basta para isso repetir as opções remote/button quantas vezes forem necessárias.

REPEAT – Especifica o que acontece se houver repetição ao pressionar um botão. O valor padrão é zero, fazendo com que o driver ignore os botões repetidos.

⁹ Conforme BARTELMUS, Christoph. <http://www.lirc.org> Acesso em 12 maio/ 2006 às 17h54.

DELAY – Indica ao programa para ignorar um número específico de repetição de um botão antes de usar a configuração de repetição acima. É usado para prevenir que um evento seja executado mais de uma vez em um curto período. O valor padrão desse campo é zero, o que desabilita o uso dessa função.

CONFIG – A execução da aplicação depende dessa função. Portanto, as ações que devem ser tomadas dependem diretamente desse campo, como por exemplo a ação de imprimir alguns caracteres na tela, ou abrir e fechar o driver do CD, ou ainda executar o programa de acender e apagar a lâmpada.

MODE – Indica ao programa para iniciar em um modo específico.

FLAGS – Parâmetros adicionais ao MODE.

O arquivo *.lircrc* usado pelo projeto ficou da seguinte forma:

```
# defaults:
#
# remote = *
# repeat = 0
#

begin
    remote = *
    button = power
    prog = irexec
    repeat = 0
    config = echo "Bem Vindo!"
end

begin
    remote = *
    button = on_timer
    prog = irexec
    repeat = 0
    config = /usr/bin/ooimpress&
end

begin
    remote = *
    button = sleep
    prog = irexec
    repeat = 0
    config = /usr/bin/eject
end

begin
    remote = *
    button = stop
    prog = irexec
    repeat = 0
    config = shutdown -h now
end

begin
    remote = *
    button = 0
    prog = irexec
    repeat = 0
    config = /home/rafael/lpt_completo 0
```

end

begin

remote = *

button = 1

prog = irexec

repeat = 0

config = /home/rafael/lpt_completo 4

end

APÊNDICE D

7.1.4. LPT_COMPLETO

O programa *lpt_completo* envia um determinado valor para a porta paralela do computador. Os valores são dados como parâmetro para o programa, sendo que podem ser tanto decimal (0-255) como em formato hexadecimal (0x00-0xFF). Esse programa é utilizado para acender e apagar a lâmpada.

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <asm/io.h>

#define base 0x378      /* Endereço da porta paralela*/

main(int argc, char **argv)
{
    int value;

    if (argc!=2)
        fprintf(stderr, "Error: Esse programa precisa de um valor entre 0 e 255.\n"),
        exit(1);

    if (sscanf(argv[1], "%i", &value)!=1)
        fprintf(stderr, "Error: Parâmetro não é um número.\n"),
        exit(1);

    if ((value<0) || (value>255))
        fprintf(stderr, "Valor numérico inválido. O número deve estar entre 0 e 255\n"),
        exit(1);

    if (ioperm(base,1,1))
        fprintf(stderr, "Error: Não consegue acessar a porta %x\n", base),
        exit(1);

    outb((unsigned char)value, base);
}
```

APÊNDICE E

7.1.5. ESQUEMA ELÉTRICO CIRCUITO INFRAVERMELHO

A Figura 7.1 demonstra o esquema elétrico do circuito infravermelho montado com a relação dos componentes utilizados.

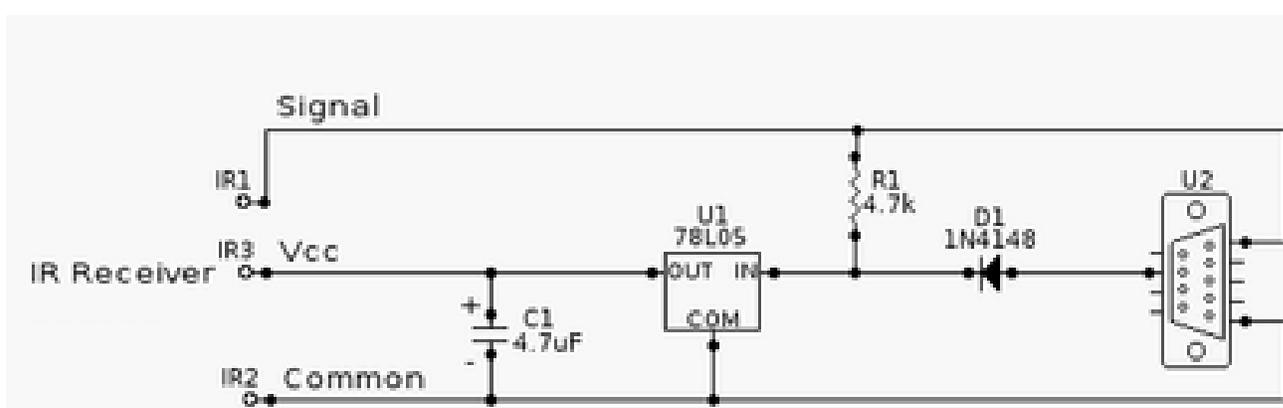


Figura 7.1: Esquema elétrico circuito infravermelho

Fonte: http://www.vivaolinux.com.br/imagens/artigos/comunidade/thumb_LIRC_serial_rx1.png

Os componentes utilizados para a montagem do circuito foram os seguintes:

- 1 Receptor infravermelho *Vishay TSOP 1738*;
- 1 Capacitor de 4,7 μF ;
- 1 Regulador de tensão 78L05;
- 1 Resistor de 4,7 Kohms;
- 1 Diodo 1N4148.

APÊNDICE F

7.1.6. ESQUEMA ELÉTRICO CIRCUITO LÂMPADA

A Figura 7.2 demonstra o esquema elétrico do circuito que tem como função ligar e desligar a lâmpada.

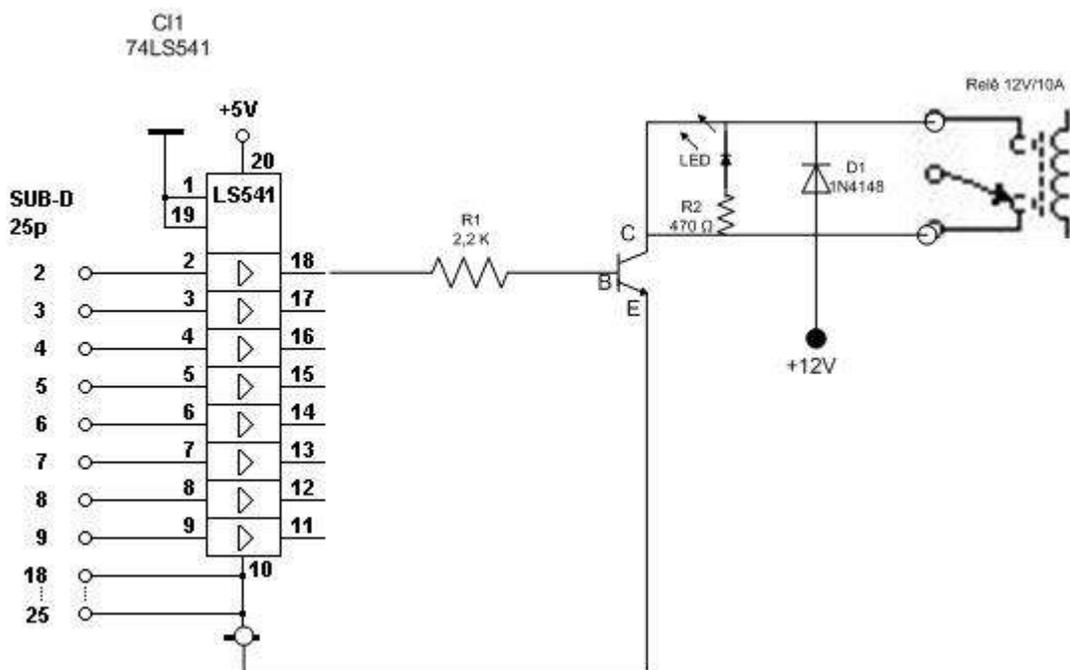


Figura 7.2: Esquema elétrico circuito lâmpada

Fonte: Autor

Os componentes utilizados para a montagem do circuito foram os seguintes:

- 1 LED verde ;
- 1 Diodo 1N4148;
- 1 Resistor R1: 2,2 Kohm;
- 1 Resistor R2: 470 ohm;
- 1 Transistor BD 137;

- 1 Circuito Integrado 74LS541;
- 1 Relé de 12 Volts na bobina – 220 V/ 10 A;
- 1 Fonte de alimentação estabilizada de 12 Volts.