

UNIVERSIDADE ESTADUAL DE CAMPINAS - UNICAMP

**FACULDADE DE ENGENHARIA ELÉTRICA E DA
COMPUTAÇÃO - FEEC**

TESE DE DOUTORADO

Título:

Arquitetura para Laboratório de Acesso Remoto com
Aplicações Educacionais

Autor: Marco Túlio Chella

Orientador: Prof^o. Dr. Elnatan Chagas Ferreira

Banca Examinadora:

Fábio Ferrentini Sampaio – UFRJ

Francisco Javier Ramirez Fernandez – USP

José Antonio Siqueira Dias - DEMIC/FEEC/UNICAMP

Heloisa Vieira da Rocha – IC/UNICAMP

Oséas Valente de Avilez Filho - DEMIC/FEEC/UNICAMP

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE -
UNICAMP

C418a Chella, Marco Túlio
Arquitetura para laboratório de acesso remoto com
aplicações educacionais / Marco Túlio Chella. --Campinas,
SP: [s.n.], 2006.

Orientador: Elnatan Chagas Ferreira.
Tese (doutorado) - Universidade Estadual de Campinas,
Faculdade de Engenharia Elétrica e de Computação.

1. Internet na educação. 2. Ensino à distância. 3.
Educação à distância.. 4. Laboratórios eletrônicos. 5.
Eletrônica – Instrumentos digitais. 6. Recursos de redes de
computadores. I. Ferreira, Elnatan Chagas. II.
Universidade Estadual de Campinas. Faculdade de
Engenharia Elétrica e de Computação. III. Título.

Titulo em Inglês: Architecture for laboratory of remote access with
educational applications.

Palavras-chave em Inglês: Distance learning, Electronics laboratories,
Remote access laboratory.

Área de concentração: Eletrônica, Microeletrônica e Optoeletrônica.

Titulação: Doutor em Engenharia Elétrica

Banca examinadora: Fáblio Ferrentini Sampaio, Francisco Javier Ramirez
Fernandez, José Antonio Siqueira Dias, Heloisa Vieira
da Rocha, Oséas Valente de Avilez Filho.

Data da defesa: 31/07/2006

Resumo

Este trabalho apresenta uma arquitetura para o desenvolvimento de experimentos controlados remotamente via rede Internet com enfoque em aplicações educacionais. Utiliza-se a rede Internet como infra-estrutura de comunicação, aplicativos e hardware específico para instrumentação baseada em computador, um sistema composto de placa eletrônica e aplicativos para controle e monitoramento de experimentos reais. Os experimentos criados com a ferramenta apresentada possibilitam ao usuário com um computador conectado à rede Internet controlar e obter informações relacionadas ao experimento no qual ele está atuando.

Abstract

This work presents an architecture for the development of remote controlled experiments over the Internet with application in educational environments. Internet is used as infrastructure of communication, software and the specific hardware for instrumentation based on computer, electronic circuits and software to control and monitoring of real experiments. The experiment created with the presented tool makes possible to the user with computer plugged on Internet control and get information on the experiment over control.

“Nós somos o que fazemos constantemente. Então excelência não é um ato, mas um hábito.”

Aristóteles

Dedicatória

Aos meus familiares pela compreensão e incentivo.

Agradecimentos

Ao meu orientador professor Elnatan, pela oportunidade e apoio.

A CAPES pelo suporte financeiro.

Sumário

Introdução.....	13
Motivação.....	14
Objetivo.....	15
Metodologia.....	16
Organização do Documento.....	16
Capítulo 1 - Tecnologias de Hardware e Software.....	19
Laboratórios de acesso remoto com propósito educacional.....	19
Telelab.....	19
Ilab.....	21
Laboratório Virtual da Escola Politécnica – USP.....	22
Rexlab.....	24
Hardware.....	25
Microcontroladores.....	25
Barramento USB.....	27
Descrição dos aspectos técnicos do padrão USB.....	29
Controladores USB.....	32
Cypress Semiconductor.....	33
Netchip.....	33
Microchip.....	34
FTDI.....	34
National Semicondutores.....	35
Barramento GPIB.....	35
Barramento SPI.....	37
Software.....	40
Comunicação por Socket.....	40
XML.....	41
COM e DCOM.....	44
Java.....	45
Tecnologias empregadas no desenvolvimento.....	46
Capítulo 2 - Arquitetura do Laboratório Remoto.....	49
Caso de uso para o desenvolvimento de um sistema de experimento.....	49
Interfaces gráficas.....	52
Arquitetura do laboratório de acesso remoto.....	53
Capítulo 3 - Projeto e implementação do Hardware.....	57
PIC16f877.....	58
FT245.....	60
MAX232.....	64
Registro de deslocamento CD4094.....	65
Potenciômetro Digital AD7376.....	66
AD7501.....	67
Arquitetura do hardware.....	68
Placa de Controle.....	70
Caso de uso da Placa de Controle.....	72
Inicializar o hardware.....	73
Ler ft245.....	73

Decodificar comando.....	74
Escrever no ft245.....	75
Escrever no 4094	76
Selecionar experimento	78
Escrever no potenciômetro digital.....	79
Escrever na chave	82
Escrever no banco de resistor	83
Escrever no ponto de medição.....	86
Ler Usart e Escrever na Usart.....	88
Placa de Experimento	91
Bloco Eletrônico Chave Eletrônica	94
Bloco Eletrônico Chave de Relé.....	95
Bloco Eletrônico Potenciômetro Digital.....	97
Testes do sistema de hardware	98
Capítulo 4 - Projeto e implementação do software	101
Editor de Experimentos	101
Barra de ferramentas.....	102
Área de trabalho.....	102
Janela de propriedades.....	103
Caso de uso do Editor de Experimentos.....	103
Criar objeto	104
Potenciômetro Digital.....	105
Chave.....	107
Ponto de Medição	108
Hyperlink	110
Rótulo	111
Posicionar objeto	111
Redimensionar objeto.....	112
Atribuir propriedades ao objeto.....	113
Excluir objeto	113
Salvar Projeto	114
Abrir projeto	114
Criar XML com definição do projeto	115
Cliente do Experimento Remoto	117
Iniciar Conexão com a rede.....	119
Ler arquivo de definição do experimento.....	119
Criar objetos	119
Enviar comandos de controle	120
Encerrar conexão com a rede.....	120
Servidor Placa de Controle.....	120
Conectar a rede	122
Desconectar da rede.....	123
Carregar servidor	123
Receber dados.....	123
Tratar dados	124
Apresentar dados	124
Iniciar comunicação USB.....	124

Escrever no dispositivo USB.....	126
Exibir dados e mensagens de estado	126
Encerrar comunicação USB e o Datasocket	126
Servidor GPIB	127
Iniciar comunicação Internet	128
Carregar servidor Socket	129
Configurar Hardware GPIB.....	129
Inicializar aquisição GPIB.....	129
Ler Dados GPIB	130
Executar Comandos GPIB.....	131
Tratar dados para exibição.....	131
Aguardar Comandos do Cliente GPIB	133
Enviar dados pela Internet	133
Exibir dados.....	133
Exibir mensagens de estado.....	134
Encerrar comunicação Internet.....	134
Cliente Gpib.....	135
Iniciar comunicação Internet	136
Sincronizar com Servidor GPIB.....	137
Enviar comandos	137
Receber dados.....	138
Exibir dados.....	138
Exibir mensagens.....	139
Encerrar comunicação Internet.....	140
Capítulo 5 - Avaliação do sistema.....	141
Experimento 1	142
Experimento 4	153
Conclusão	159
Trabalhos futuros.....	161
Referências bibliográficas	163
Anexo 1	167
Anexo 2	171
Anexo 3	175

Índice de Figuras

Figura 1.1 - Tela inicial do Telelab	20
Figura 1.2 - Tela do cliente do Telelab.....	21
Figura 1.3 - Cliente para experimento no ILab	22
Figura 1.4 - Cliente Laboratório Virtual USP	23
Figura 1.5 - Site do Laboratório Virtual da USP.....	24
Figura 1.6 - Interface gráfica Cliente Rexlab	25
Figura 1.7 - Camada de comunicação do protocolo USB	30
Figura 1.8 - Pacote Token.....	32
Figura 1.9 - Diagrama de tempo da comunicação GPIB.....	36
Figura 1.10 - Diagrama de tempo da comunicação SPI.....	39

Figura 1.11 - Conexão de dispositivos SPI.....	39
Figura 2.12 - Diagrama da arquitetura do Laboratório Remoto.....	54
Figura 3.13 - Microcontrolador PIC16F877.....	58
Figura 3.14 - Configuração mínima do PIC16F877.....	60
Figura 3.15 - Componente FT245 em blocos.....	61
Figura 3.16 - Diagrama de tempo e sinais para operação de escrita do FT245.....	62
Figura 3.17 - Diagrama de tempo e sinais para operação de leitura do FT245.....	63
Figura 3.18 - Configuração típica do componente MAX232.....	65
Figura 3.19 - Componente CD 4094.....	65
Figura 3.20 - Tabela verdade do componente CD4094.....	66
Figura 3.21 - Diagrama interno do AD7376.....	66
Figura 3.22 - Diagrama de tempo do componente AD7376.....	67
Figura 3.23 - Diagrama em blocos do componente AD7501.....	67
Figura 3.24 - Tabela verdade do AD7501.....	68
Figura 3.25 - Arquitetura do sistema de hardware.....	69
Figura 3.26 - Sistema de hardware.....	69
Figura 3.27 - Diagrama em blocos da Placa de controle.....	70
Figura 3.28 - Dados do CD4094 entre a Placa de controle e Experimentos.....	71
Figura 3.29 - Diagrama de arquitetura do programa do microcontrolador PIC16F877.....	72
Figura 3.30 - Estrutura do protocolo de comunicação.....	75
Figura 3.31 - Operação de escrita no CD4094.....	77
Figura 3.32 - Diagrama para seleção da Placa de Controle.....	78
Figura 3.33 - Diagrama de tempo da comunicação SPI.....	80
Figura 3.34 - Diagrama do hardware para controle do potenciômetro digital.....	81
Figura 3.35 - Diagrama de controle para Bloco Chave.....	82
Figura 3.36 - Palavra de dados para Bloco Eletrônico chave.....	83
Figura 3.37 - Palavra de dados para Bloco Eletrônico Banco de Componentes.....	84
Figura 3.38 - Diagrama do circuito para controle de Banco de Componentes.....	85
Figura 3.39 - Componente AD7501.....	86
Figura 3.40 - Diagrama em blocos do Ponto de Medição.....	87
Figura 3.41 - palavra de dados para ponto de medição.....	88
Figura 3.42 - Comunicação serial RS232 da placa de controle.....	89
Figura 3.43 - Placa de Experimento.....	91
Figura 3.44 - Diagrama da Placa de Experimento.....	93
Figura 3.45 - Detalhe do ponto de medição com os cabos conectados.....	94
Figura 3.46 - Detalhe do circuito do ponto de medição.....	95
Figura 3.47 - Circuito do Bloco Eletrônico Chave de Relé.....	96
Figura 3.48 - Bloco Eletrônico Chave de Relé.....	96
Figura 3.49 - Bloco Eletrônico Potenciômetro Digital.....	97
Figura 3.50 - Aplicativo inicial para diagnóstico do hardware.....	98
Figura 3.51 - Aplicativo para avaliar funções da Placa de Controle.....	99
Figura 4.52 - Interface gráfica do editor de experimentos.....	102
Figura 4.53 - Barra de ferramentas do Editor de Experimentos.....	102
Figura 4.54 - funcionalidades do editor de experimentos.....	104
Figura 4.55 - Objeto potenciômetro digital representando dois valores.....	105
Figura 4.56 - Janela de propriedades do potenciômetro digital.....	106
Figura 4.57 - Representação gráfica do objeto chave.....	107

Figura 4.58 - Janela para atribuição das propriedades ao objeto chave	108
Figura 4.59 - Detalhe da representação gráfica do ponto de medição.....	109
Figura 4.60 - Janela para atribuição das propriedades ao objeto ponto de medição	109
Figura 4.61 - Janela para atribuição de propriedades ao objeto Banco de Resistores	110
Figura 4.62 - Janela para atribuição de propriedades do objeto hyperlink.....	111
Figura 4.63 - Janela para atribuição de propriedades ao objeto rótulo.....	111
Figura 4.64 - Redimensionamento de objeto.....	112
Figura 4.65 - Janela de propriedades do objeto rótulo	113
Figura 4.66 - Definição XML do projeto	116
Figura 4.67 - Diagrama de funcionalidades do Cliente do Experimento Remoto.....	118
Figura 4.68 - Diagrama de funcionalidades para Servidor Placa de Controle	121
Figura 4.69 - Interface Gráfica do Servidor Placa de Controle	122
Figura 4.70 - Comunicação com o hardware USB.....	125
Figura 4.71 - Arquitetura do Servidor GPIB	128
Figura 4.72 - Interface gráfica do Cliente GPIB	135
Figura 4.73 - Diagrama das funcionalidades do Cliente GPIB	136
Figura 4.74- Botão de osciloscópio (a) e botão na interface gráfica (b)	137
Figura 4.75 - Objeto CWGraph com forma de onda captura pelo Osciloscópio.....	139
Figura 4.76 - Mensagens de estado e configuração.....	140
Figura 5.77 - Transistor com polarização fixa	143
Figura 5.78 - Transistor polarizado com resistor no emissor	143
Figura 5.79 - Transistor polarizado com divisor de tensão	144
Figura 5.80 - Circuito para Experimento 1	145
Figura 5.81 - Placa de Experimento para Experimento 1	146
Figura 5.82 - Inserir circuito no editor de experimentos	147
Figura 5.83 - Inserir Bloco Eletrônico Chave no Editor de Experimentos.....	148
Figura 5.84 - Configurar propriedades do Bloco Eletrônico Chave.....	149
Figura 5.85 - Circuito com dois transistores e chaves para troca.....	150
Figura 5.86 - Configuração do rótulo no editor de experimentos	150
Figura 5.87 - Configuração das propriedades do Objeto Hyperlink.....	151
Figura 5.88 - Configuração das propriedades do Bloco Ponto de Medição	152
Figura 5.89 - Experimento de polarização no navegador web	152
Figura 5.90 - Diagrama esquemático do experimento quatro	153
Figura 5.91 - Interface Gráfica do Experimento 4.....	154
Figura 5.92 - Detalhe da interface gráfica com chave aberta	155
Figura 5.93 - Detalhe da interface gráfica com chave fechada	155
Figura 5.94 - Detalhe da seleção do ponto de medição	156
Figura 5.95 - Detalhe interface gráfica com potenciômetro digital.....	156

Índice de Tabelas

Tabela 1.1 Exemplo de um documento XML	43
Tabela 3.2 -Procedimento Lerft245.....	74
Tabela 3.3 - Exemplo de utilização de ler_ft245.....	74
Tabela 3.4 - Decodificação do protocolo de comunicação.....	75

Tabela 3.5 - Pseudocódigo da função escreve_ft245	76
Tabela 3.6 - Declaração da função escreve_4094	77
Tabela 3.7 - Pseudocódigo da função escreve_4094.....	77
Tabela 3.8 – Função escreve_4094 usada para selecionar experimento	78
Tabela 3.9 - Pseudocódigo da função escreve_pot_digital.....	80
Tabela 3.10 - Pseudocódigo para função de escrita no potenciômetro digital	81
Tabela 3.11 - Pseudocódigo para função escrever na chave	83
Tabela 3.12 - Pseudocódigo para função escrever_banco_resistor	86
Tabela 3.13 - Pseudocódigo para função ponto de medição	88
Tabela 3.14 - Pseudocódigo para função de escrita da usart.....	90
Tabela 3.15 - Pseudocódigo para função de leitura da usart	90
Tabela 3.16 - Pseudocódigo para escrita de cadeia na usart.....	91
Tabela 4.17 - Código para a criação do objeto chave.....	105
Tabela 4.18 - Procedimento para excluir objeto no editor de experimentos	114
Tabela 4.19 - Procedimento para salvar projeto no editor de experimentos	114
Tabela 4.20 - Procedimento para leitura do arquivo XML.....	115
Tabela 4.21 - Procedimento para instanciar um objeto DOM.....	116
Tabela 4.22 - Exemplo de arquivo de projeto em formato XML.....	117
Tabela 4.23 - Procedimento para conectar o cliente do experimento remoto ao servidor .	119
Tabela 4.24 - Procedimento para carga do arquivo de definição do projeto	119
Tabela 4.25 - Procedimento para envio de comandos pela rede	120
Tabela 4.26 - Procedimento para encerrar a conexão de rede	120
Tabela 4.27 - Procedimento para desconectar da rede	123
Tabela 4.28 - Procedimento para carregar servidor.....	123
Tabela 4.29 - Procedimento para recepção de dados.....	123
Tabela 4.30 - Procedimento para iniciar comunicação USB.....	125
Tabela 4.31 - Procedimento para escrever no hardware USB	126
Tabela 4.32 - Procedimento para encerrar a comunicação com o hardware USB	127
Tabela 4.33 - Procedimento para conexão à rede.....	128
Tabela 4.34 - Procedimento para iniciar servidor de datasocket.....	129
Tabela 4.35 - Procedimento para configurar comunicação com hardware GPIB	129
Tabela 4.36 - Configuração da aquisição de dados do hardware GPIB	130
Tabela 4.37 - Procedimento para aquisição de dados GPIB.....	131
Tabela 4.38 - Procedimento para seleção da base de tempo	131
Tabela 4.39 - Obtenção do dados para ajuste da exibição.....	132
Tabela 4.40 - Fórmula para ajustes dos dados exibidos	132
Tabela 4.41 - Conversão dos dados para exibição.....	133
Tabela 4.42 - Envio dos dados via rede Internet	133
Tabela 4.43 - Exibição dos dados.....	134
Tabela 4.44 - Procedimento para encerrar comunicação com a internet.....	135
Tabela 4.45 - Procedimento para iniciar a comunicação com a internet.....	137
Tabela 4.46 - Pacote de dados do comando GPIB	138
Tabela 4.47 - Envio de comando GPIB.....	138
Tabela 4.48 - Fechar comunicação com a rede Internet.....	140
Tabela 5.49 - Blocos eletrônicos do experimento 1	146
Tabela 5.50 - Propriedades das chaves para a configuração polarização fixa.....	149

Introdução

Durante a última década pode-se observar o rápido desenvolvimento das redes de computadores e a Internet. A WWW (World Wide Web), um dos serviços da Internet mais populares, propiciou meios para a publicação e acesso a grandes volumes de informações em formatos textuais e multimídia. Estas características da WWW se mostraram promissoras para desenvolvimento de ambientes para educação à distância ao oferecer recursos que permitem:

- Identificação, avaliação e integração de uma grande variedade de informação;
- Colaboração, discussão, troca e comunicação de idéias;
- Participação em experiências simuladas, aprendizagem e parcerias cognitivas;
- Expressão e construção coletiva de conceitos, significados artísticos e cognitivos.

(Lucena e Fucks, 2000).

Com o propósito de explorar estas potencialidades em aplicações educacionais, ambientes e ferramentas têm sido desenvolvidos como o WEBCT (Webct, 2002) , Teleduc (Teleduc, 2003) e Aulanet (Aulanet, 2003). Estas ferramentas oferecem recursos para gerenciamento de conteúdos e alunos, disponibilização de materiais e comunicação entre usuários.

Outra tecnologia que tem apresentado desenvolvimento crescente nos últimos anos são os sistemas de instrumentação e medida baseados em computador. Esses sistemas têm como elementos básicos os equipamentos de medida, o computador e o aplicativo responsável pelo controle do fluxo de informações que ocorre por meio de interfaces padronizadas para comunicação homem-máquina e máquina-máquina .

Empresas como National Instruments (National, 2001), fabricante do aplicativo LabView e hardware, tem oferecido uma grande gama de produtos que atendem os mais diversos propósitos na área de instrumentação.

Ao incorporar nos sistemas de instrumentação baseados em computador recursos para comunicação via rede Internet abre-se a possibilidade para que a atuação e monitoramento desses sistemas possam ocorrer remotamente, inclusive por múltiplos usuários, dispensando a presença física no laboratório. Estas possibilidades têm sido

exploradas pela indústria, pela medicina e também em aplicações educacionais caracterizando os laboratórios remotos.

Diversas universidades e centros de pesquisa no mundo têm desenvolvido trabalhos relacionados ao controle de instrumentos remotamente via rede Internet. Alguns como Telelab (Casini, Prattichizzo e Vicino 2002), AIM (Shen et al, 1999) e PEARL (Schafer, Seigneur e Donnelly, 2002) foram projetados com propósito educacional.

Os vários sistemas de laboratórios remotos estudados apresentam uma característica comum: foram desenvolvidos por pesquisadores com habilidade nas várias tecnologias relacionadas à programação de aplicativos para Internet, conhecimento aprofundado do hardware, protocolo de rede e comunicação dos instrumentos que fazem parte do ambiente.

Essa abordagem para o desenvolvimento de sistemas para laboratório de acesso remoto resulta em sistemas que atendem de forma específica a determinados experimentos. A alteração ou elaboração de novos experimentos envolverá a implementação de novos aplicativos e componentes de hardware o que normalmente consome tempo e recursos financeiros.

O trabalho desenvolvido nessa pesquisa propõe e implementa um sistema no qual o professor, que deseja aplicar os experimentos remotos com seus alunos, pode elaborar, alterar e disponibilizar experimentos para acesso remoto de forma rápida já que ele não precisa se aprofundar nos aspectos técnicos relacionados às atividades com experimentação remota

Motivação

Além da formação teórica baseada em aulas tradicionais, a experiência de fazer na prática proporcionada pelos laboratórios são elementos vitais na formação do profissional de engenharia (Hua e Ganz, 2003). Contudo, configurar e manter um laboratório disponível por longos períodos pode representar um custo financeiro alto. Com um laboratório remoto pode-se otimizar a utilização de recursos, já que o mesmo pode estar disponível por maior tempo e acessível a mais de uma instituição de ensino, e conseqüentemente atender a um número maior de alunos que podem atuar nos experimentos simultaneamente em locais e horários distintos.

Outro aspecto a considerar é que não havendo a necessidade de presença física do aluno no laboratório elimina-se a necessidade de instalações com maior espaço físico, o que pode representar diminuição nos custos de manutenção.

Provendo acesso a experimentos remotos pode-se atender a demanda existente de ensinar ao aluno o uso de equipamentos técnicos complexos, introduzindo-o ao estado da arte na prática de sua área de estudo e preenchendo as expectativas do que é esperado em sua formação. Ao prover acesso a experimentos envolvendo equipamentos de custo elevado e algumas vezes sensíveis pode-se ampliar a faixa de utilização desses equipamentos o que de certo modo proporciona uma economia de escala (Schafer, Seigneur e Donnelly, 2002).

Do ponto de vista pedagógico acredita-se que todo o aprendizado possível de ser realizado no ambiente presencial, poderá ser propiciado também à distância, numa abordagem em que diferentemente da sala de aula tradicional, engloba comunicação e conhecimento baseado na liberdade, na pluralidade e na cooperação de forma mais ampla possível (Silva, 2000).

Objetivo

O propósito de desenvolver uma ferramenta para criação de experimentos remotos é oferecer recursos para que o professor possa elaborar seu experimento que envolva o controle de instrumentos e acionamento de relês, chaves eletrônicas, potenciômetros digitais, entre outros, sem se preocupar com os aspectos relacionados à programação da comunicação do hardware, dos instrumentos de medição e integração com a rede Internet, podendo se concentrar nos aspectos relacionados ao conteúdo que deseja explorar.

A contribuição deste trabalho é o desenvolvimento de uma plataforma de suporte a experimentos na área de eletrônica, na qual seja possível realizar, à distância e localmente vários experimentos distintos concebidos e implementados em laboratórios reais atendendo as seguintes características:

- Custo baixo e decrescente por estudante. Depois dos investimentos iniciais, o laboratório remoto tende a ficar mais econômico em relação ao convencional, pois pode atender a um número maior de alunos.
- Modularidade, ao propiciar meios para que sejam agregados novos experimentos, equipamentos e aplicativos.
- Execução dos experimentos em tempo real e com interação do aluno.

- Permita a utilização dos procedimentos experimentais já existentes nos laboratórios convencionais podendo ser facilmente integrado aos recursos didáticos empregados como livros texto, ilustrações, entre outros.
- Aplicação no modo remoto e presencial.
- Acesso compartilhado a equipamentos de custo alto, dispensando a aquisição por cada instituição ou laboratório.
- Flexibilização de horários para execução dos experimentos, otimizando o tempo de estudo dos alunos e a utilização dos equipamentos.
- Recursos para integração a sistemas para criação e gerenciamento de cursos em EAD como o Aulanet (2003) e Teleduc (2003).

Metodologia

Para o desenvolvimento do trabalho de pesquisa foi adotada a seguinte metodologia:

- Definição dos objetivos e justificativas para a pesquisa
- Estudo dos sistemas para laboratórios de acesso remoto e tecnologias associadas
- Levantamento dos requisitos
- Definição da arquitetura de software e hardware
- Implementação do hardware
- Desenvolvimento do software
- Avaliação do sistema e conclusões.

Organização do Documento

A redação deste trabalho foi dividida em cinco capítulos.

No capítulo I é realizado o estudo de alguns laboratórios para acesso remoto desenvolvidos com propósitos educacionais. Também são estudadas as várias tecnologias de hardware e software disponíveis para o desenvolvimento de sistemas para controle, automação, visualização e comunicação de dados.

Com base nas informações estudadas no capítulo I, o capítulo II discute os requisitos para o sistema a ser desenvolvido e detalha a arquitetura de hardware e software desenvolvido.

O capítulo III aborda o desenvolvimento dos diversos sistemas de hardware.

No capítulo IV é apresentada a implementação dos sistemas de software e a integração dos sistemas para testes e diagnósticos do hardware.

A avaliação dos sistemas, conclusão e sugestão para trabalhos futuros está no capítulo V.

Após os capítulos está a bibliografia e três anexos com informações adicionais sobre os sistemas de hardware e software desenvolvidos.

Capítulo 1 - Tecnologias de Hardware e Software

Um sistema para o desenvolvimento de experimentos remotos é normalmente constituído por um conjunto de tecnologias de software e hardware que pode assumir as mais diversas configurações em função da escolha dos recursos tecnológicos empregados em sua implementação.

Este capítulo discute as diferentes opções de hardware e software disponíveis para o desenvolvimento de uma arquitetura para realização de experimentos remotamente via rede internet.

Inicialmente são apresentados alguns laboratórios de acesso remoto desenvolvidos por instituições de pesquisa e suas características. Em seguida são apresentados componentes de hardware e software avaliados para utilização nas implementações realizadas nesse trabalho.

No final do capítulo são descritas as opções tecnológicas escolhidas para o desenvolvimento da arquitetura e os critérios que justificaram as escolhas.

Laboratórios de acesso remoto com propósito educacional

Algumas instituições de ensino têm pesquisado e desenvolvido laboratórios de acesso remoto para aplicações educacionais. Esses laboratórios têm como propósito comum oferecer aos alunos a oportunidade de acessar e controlar experimentos reais, compartilhar equipamentos de custo elevado e trabalhar de forma cooperativa com outros alunos e professores.

Dos vários laboratórios avaliados foram selecionados quatro para descrição. A razão para tal é que os mesmos representam o que tem sido realizado atualmente no desenvolvimento e aplicação desses sistemas.

Telelab

Em desenvolvimento por uma equipe de pesquisadores da Universidade Siena na Itália. O laboratório tem aplicação nos curso controle de sistemas. Utilizando uma arquitetura cliente/servidor permite que o aluno envie controladores desenvolvidos com o Matlab/Simulink para um servidor no qual ele é processado e executado em um sistema real. O aluno pode visualizar o experimento por meio de uma câmera que transmite

imagens e capturar dados relacionados à execução do experimento. A partir de uma página na internet (figura 1.1) o aluno tem acesso a experimentos como controle de velocidade de um motor, controle do fluxo de água em um tanque, controle de um robô construído com blocos de montagem, entre outros experimentos.

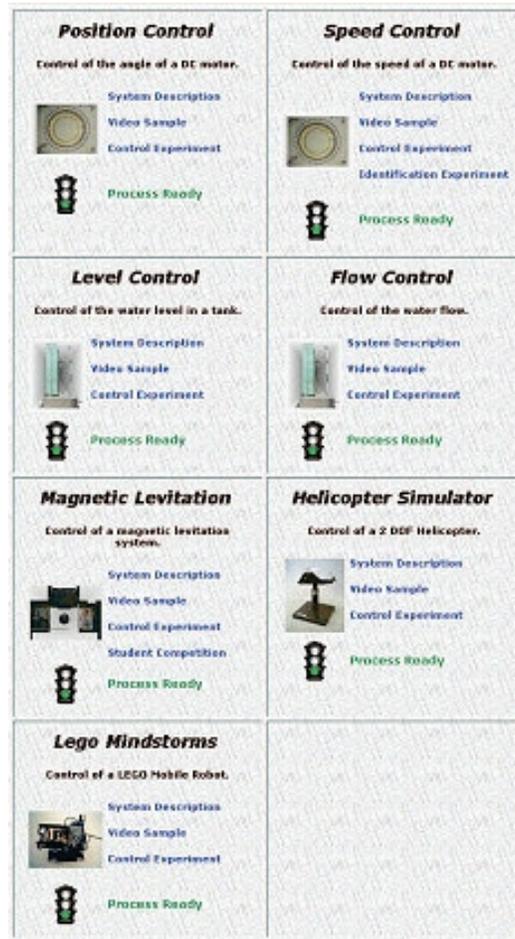


Figura 1.1 - Tela inicial do Telelab

No modo cliente um aplicativo Java exibe os dados do experimento em execução e o vídeo que permite a visualização dos movimentos realizados pelo dispositivo no qual o aluno está atuando(figura 1.2).

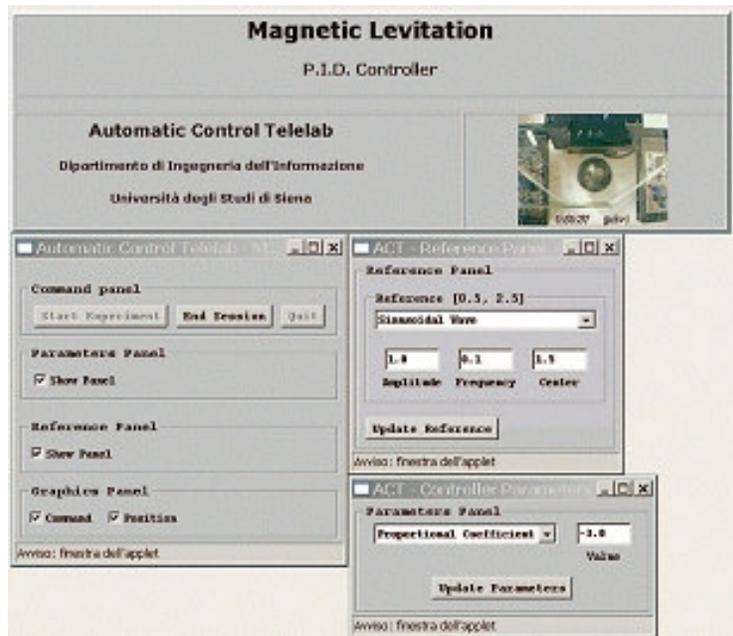


Figura 1.2 - Tela do cliente do Telelab

Publicações dos pesquisadores envolvidos no projeto Telelab (Casini, Prattichizzo e Vicino 2002), (Casini, Prattichizzo e Vicino 2003), (Casini, Prattichizzo e Vicino 2004) indicam que o sistema tem sido utilizado de forma efetiva em atividades com alunos e pesquisas têm sido realizadas com o propósito de avaliar a utilização de experimentação remota como ferramenta educacional.

Ilab

Este laboratório de acesso remoto foi desenvolvido por equipe de pesquisadores da universidade MIT (Massachusetts Institute of Technology) com patrocínio da empresa Microsoft (Ilab,2005). Foram desenvolvidos recursos para que o laboratório opere nos modos batch, interativo e experimentos com sensores. No modo batch os experimentos são configurados previamente, enviados para o laboratório remoto onde são executados e os dados gerados são enviados posteriormente para o aluno. Com o modo interativo os alunos podem controlar em tempo real o experimento alterando parâmetros e observando o resultado em tempo real. O modo experimento com sensores permite aos alunos visualizar dados de sensores, mas sem poder atuar sobre o sistema.

A arquitetura do sistema é baseada no modelo de Web Services e constituído pelo cliente acessível por um navegador web, o service broker que fornece os sistemas comuns

como controle de acesso e agendamento, e o lab server responsável pela execução dos experimentos.

Os experimentos desenvolvidos incluem caracterização de semicondutores (figura 1.3) realizado no modo batch, troca de calor no qual os alunos mudam os elementos térmicos e passagem de fluidos em um sistema e registram os valores obtidos pelos sensores. Este experimento é executado no modo interativo.

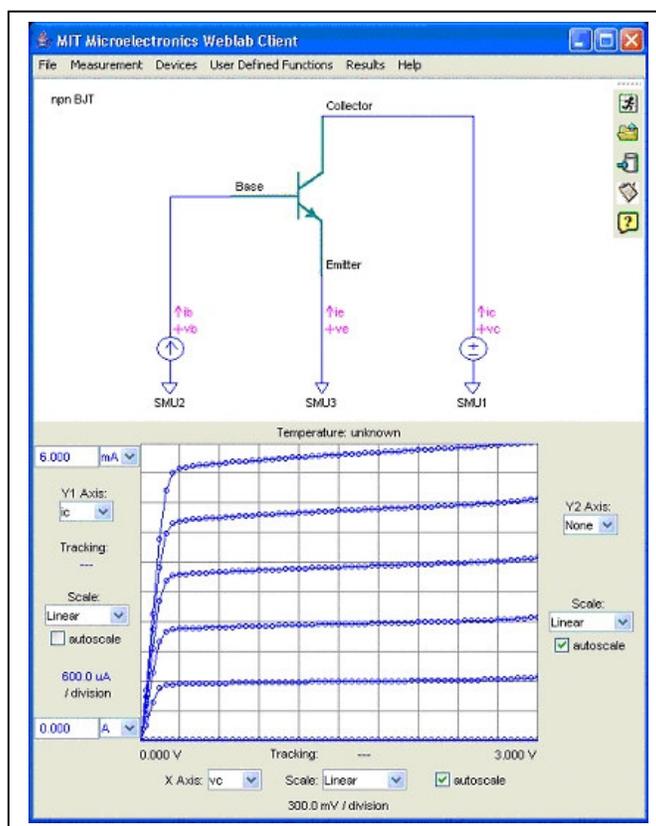


Figura 1.3 - Cliente para experimento no ILab

O experimento estação fotovoltaica é um exemplo de experimento executado no modo experimentos com sensores.

O MIT disponibiliza o sistema de Ilab, incluindo código fonte sem custo para que instituições possam realizar seus experimentos e avaliações. Informações sobre a utilização efetiva não estão disponíveis.

Laboratório Virtual da Escola Politécnica – USP

Trabalho de pesquisa que tem como objetivo a implementação de um sistema físico que utiliza o acervo instrumental disponível no grupo de sensores integráveis da

Universidade de São Paulo para o controle de parâmetros relacionados ao funcionamento de dispositivos eletrônicos em um experimento genérico. O controle e aquisição de dados são realizados via rede Internet por meio de uma interface amigável e intuitiva (Borges, 2002).

A versão final desse trabalho resultou em um sistema constituído por um aplicativo servidor que se comunica com o hardware para controle dos equipamentos e a rede Internet para receber comandos de controle e enviar dados com informações sobre o experimento.

Para o desenvolvimento dos aplicativos foi utilizado a ferramenta Labview no módulo servidor e a linguagem Java no módulo cliente.

O hardware para medição é uma placa de aquisição produzida pela empresa National Instruments. Essa placa é conectada ao circuito eletrônico experimental para estudo do efeito da temperatura sobre o comportamento de um diodo e um termistor possibilitando a variação de parâmetros e obtenção dos sinais resultantes.

O aplicativo cliente executado remotamente a partir de um applet embutido em um navegador WEB permite o envio de parâmetros de controle e a visualização dos dados obtidos (Figura 1.4).

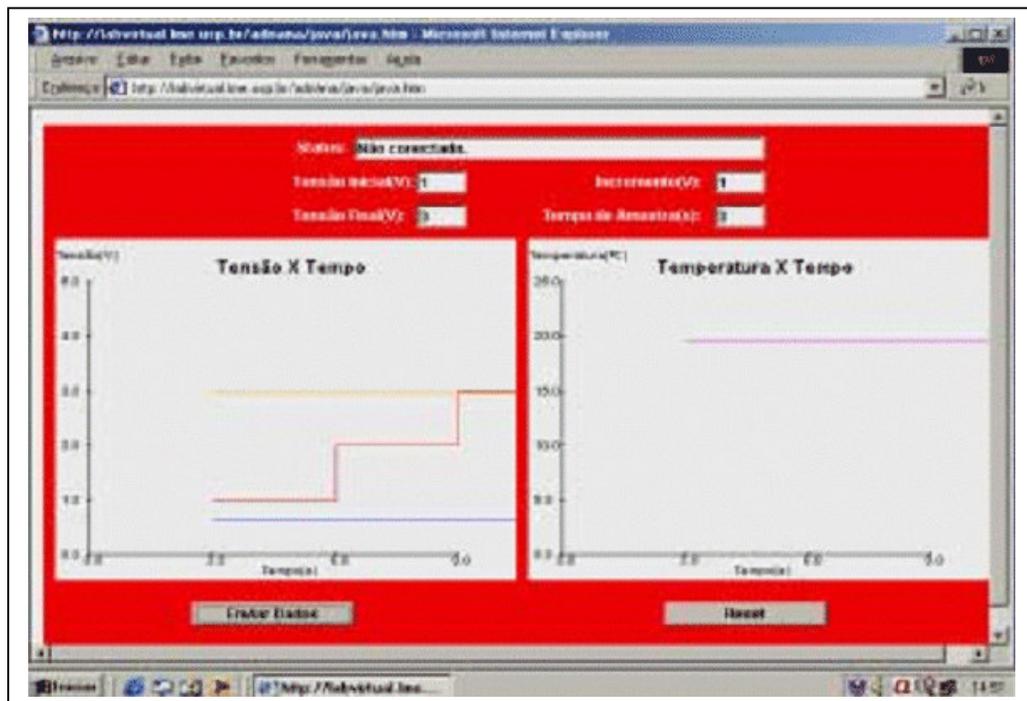


Figura 1.4 - Cliente Laboratório Virtual USP

Para divulgação de informações sobre o projeto e acesso ao experimento remoto foi desenvolvido um site (Figura 1.5).



Figura 1.5 - Site do Laboratório Virtual da USP

Rexlab

Projeto de pesquisa da Universidade Federal de Santa Catarina desenvolve um laboratório de acesso remoto com foco em experimentos para microcontroladores (Rexlab,2004). O sistema foi desenvolvido com base no modelo cliente/servidor. O servidor contém o hardware constituído por uma placa para desenvolvimento com microcontroladores e aplicativos para comunicação com o hardware e a rede internet. O cliente é constituído por um aplicativo que deve ser obtido no site do laboratório e instalado no computador cliente. Por meio desse aplicativo cliente o aluno pode enviar o programa já compilado para o laboratório remoto para execução na placa de desenvolvimento. O resultado da execução pode ser visualizado em uma janela de vídeo e dados como estado de registradores do microcontrolador podem ser obtidos (Figura 1.6).

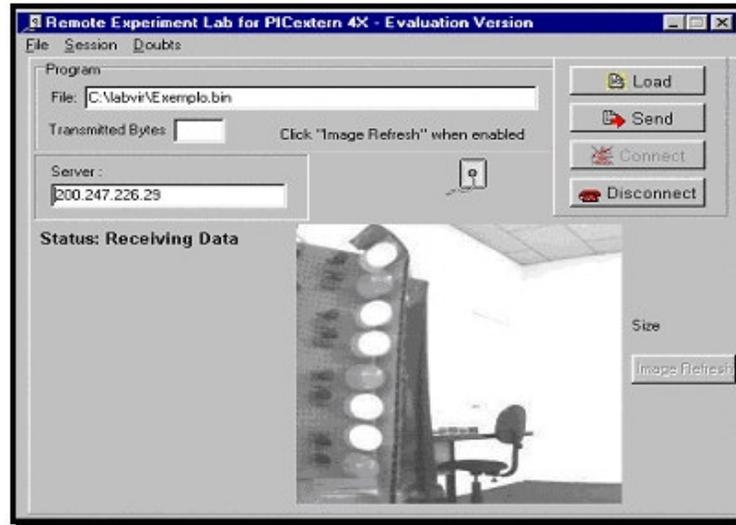


Figura 1.6 - Interface gráfica Cliente Rexlab

Os sistemas para laboratório de acesso remoto descritos empregam em sua implementação tecnologias diversas, mas basicamente seguem o modelo cliente/servidor. Nesse modelo o servidor é constituído pelos aplicativos que controlam o hardware no qual está montado o experimento e a comunicação com a rede Internet. O módulo cliente é um aplicativo que se comunica com o servidor e possibilita o envio de dados que controlam o hardware do experimento existente no servidor. Os dados gerados no experimento e obtidos por meio de instrumentos de medição são enviados e apresentados de forma gráfica ou textual no módulo cliente.

Hardware

Nesta seção são discutidas e apresentadas as várias opções tecnológicas disponíveis para o interfaceamento do computador PC e desenvolvimento de dispositivos com aplicação em instrumentação. As tecnologias com maior aplicabilidade no desenvolvimento dos sistemas do Laboratório Remoto são abordadas com maior detalhe.

Microcontroladores

O microcontrolador é um computador programável, normalmente encapsulado em um invólucro, otimizado para controlar dispositivos eletrônicos por meio de pinos lógicos e interfaces padrão de comunicação como USB, serial RS232 e outras mais específicas. Composto por uma central de processamento, com memória e interfaces de entrada e saída, enfatiza a auto-suficiência, em contraste com um microprocessador de propósito geral,

como os utilizados nos computadores pessoais, que requerem componentes adicionais para prover as funções necessárias.

Atualmente existe um grande número de fabricantes de microcontroladores, sendo os principais apresentados a seguir:

- Analog Devices - Fabrica microcontroladores de 16 bits e DSPs(Processador Digital de Sinais) de 32 bits, além de uma ampla variedade de componentes analógicos digitais que podem se comunicar com microcontroladores como potenciômetros digitais, conversores analógicos digitais, chaves eletrônicas entre outros.
- Atmel - Oferece três famílias de microcontroladores: a linha 8051, a AT91 baseada na tecnologia ARM e AVR componentes RISC de 8 bits.
- Freescale - Divisão da empresa Motorola produz uma ampla linha de microcontroladores na faixa de 8 a 32 bits sendo mais populares as séries 68H*, 68***, Coldfire, MCore e DSP56800.
- Microchip - Produz os PIC* uma família de microcontroladores que processam dados de 8 bits (recentemente foi lançada uma família de 16 bits com prefixo 24F) com extensa variedade de modelos e periféricos internos, com arquitetura Harvard e conjunto de instruções RISC (sets de 35 instruções e de 76 instruções). Possui recursos para programação por memória FLASH, EEPROM e OTP. Os microcontroladores PIC têm famílias com núcleo de processamento de 12 bits, 14 bits e 16 bits operando em velocidade de 0 (DC) a 40MHz com ciclo de instrução mínimo de 4 períodos de clock, o que permite uma velocidade de até 10 MIPS.
- Texas Instruments - Concentra sua produção em microcontroladores para DSP, mas também oferece a linha MSP430 de microcontroladores de 16 bits com tecnologia de memória flash.
- Zilog - Pioneira ao lançar a linha Z80 atualmente oferece uma linha de microcontroladores baseados no núcleo do Z80 com destaque para recursos de acesso a rede ethernet.

A linha de microcontroladores 8051 lançada pela empresa Intel em 1997 e muitas vezes considerada obsoleta é produzida atualmente por vários fabricantes como Analog Devices, Atmel, Dallas, Infineon, Silicon Labs, ST Microelectronics, entre outros. Estes fabricantes a partir do conjunto original de instruções de 8 bits do 8051 desenvolveram versões que

incorporaram os mais diversos recursos de hardware e software, capacidade de processamento, memória, interfaces de entrada e saída e ferramentas de desenvolvimento. Desse modo o 8051 pode ser considerado uma boa opção no desenvolvimento de projetos baseados em microcontroladores.

Barramento USB

O barramento USB tem se mostrado uma opção viável para o desenvolvimento de hardware com o propósito de instrumentação e controle de dispositivos externos ao computador PC.

Recentemente vários fabricantes têm oferecido componentes chamados controladores USB que operam geralmente conectados a microcontroladores e facilitam o desenvolvimento de sistemas para interfaceamento.

A seguir serão apresentadas de forma resumida algumas características do protocolo de comunicação USB e depois alguns controladores disponíveis no mercado atualmente.

O barramento USB (Universal Serial Bus) é uma solução para comunicações que envolvam o computador PC e dispositivos externos conectados a ele. Possibilita aplicações que vão desde projetos experimentais até produtos fabricados em escala industrial. O padrão USB foi desenvolvido com características que atendem a demanda de dois públicos: os usuários que desejam usar periféricos em seus computadores de forma simplificada e desenvolvedores de hardware e software que implementam os aplicativos que se comunicam com os dispositivos (Axelson , 2001) .

Do ponto de vista do usuário, a simplificação na instalação e utilização de dispositivos USB pode ser considerada a partir de características como: configuração automática, uma interface para muitos dispositivos, liberação de recursos para outros dispositivos, conexão simplificada, conexão sem necessidade de desligar os equipamentos (hot pluggable), em alguns casos dispensa fonte de alimentação.

Alguns outros benefícios para o usuário são:

- Velocidade, o padrão USB suporta três velocidades de barramento: alta velocidade a 480 Megabits por segundo (versão 2.0), velocidade total a 12 Megabits por segundo, e baixa velocidade a 1,5 Megabits por segundo (Anderson, 2001).

- Confiabilidade, decorrente dos projetos de hardware e dos protocolos de transferência de dados. As especificações de hardware para drivers USB e cabos eliminam em grande parte os ruídos que podem causar erros. O protocolo tem recursos para detecção de erros e notificação para que o host ou periférico providencie uma nova transmissão. A detecção, notificação, e retransmissão são executadas normalmente pelo hardware, dispensando tratamento por meio de programação ou alguma intervenção do usuário.
- Baixo Custo, ainda que seja mais complexo que as interfaces mais antigas, seus componentes e cabos tem se tornado mais baratos ao longo do tempo.
- Baixo consumo de energia ao dispor de recursos no software e hardware que automaticamente desativam os dispositivos, quando os mesmos não estão em uso, mantendo-os ativos apenas para que respondam quando solicitados.

Do ponto de vista do desenvolvedor destacam-se alguns pontos favoráveis a projeto de sistemas baseados no padrão USB:

- Flexibilidade, dispondo de quatro tipos de transferência e três velocidades viabiliza a aplicação em uma extensa faixa de dispositivos e periféricos. Existem modos de transferência adequados para troca de dados em blocos pequenos ou extensos, com ou sem restrições de tempo. Para dados que não podem tolerar atrasos, existe a garantia da taxa de transferência ou o mínimo tempo entre as transferências. Essas características são particularmente importantes em sistemas operacionais como o Microsoft Windows, no qual o acesso a periféricos em tempo real frequentemente é um desafio.
- Suporte pela maioria dos sistemas operacionais modernos, para troca de dados entre os dispositivos e computador, os sistemas operacionais e fabricantes de controladores USB oferecem drivers que suportam as operações mais comuns como: detectar quando um dispositivo é conectado ou removido do sistema, comunicação com o novo dispositivo conectado e a forma de se estabelecer a troca de dados como o mesmo.
- Disponibilidade de controladores USB, todo periférico USB deve incluir um controlador capaz de manipular os detalhes da comunicação USB. Alguns controladores são computadores completos que incluem a CPU e memória para armazenar códigos específicos do dispositivo que são executados no periférico. Outros manipulam somente as tarefas específicas do padrão USB, com um barramento de dados e controle

que deve ser conectado a um microcontrolador. Muitos dos controladores são baseados em arquiteturas já bastante disseminadas acrescidos de circuitos e código (firmware) para suporte USB. A sessão controladores USB, neste capítulo, descreve vários controladores pesquisados e considerados no desenvolvimento deste trabalho.

- Documentação, o USB Implementers Forum , Inc. (USB-IF) é uma associação sem fins lucrativos fundada pelas empresas que desenvolveram o padrão USB. Por meio do website (www.usb.org) mantido por essa associação é possível obter informações, ferramentas e certificações relacionadas ao desenvolvimento de sistemas USB.

Descrição dos aspectos técnicos do padrão USB

A especificação técnica do padrão USB é fornecida pelo documento Universal Serial Bus Revision 2.0 (USB.ORG,2000). Esse documento com aproximadamente seiscentas páginas descreve os componentes eletrônicos/elétricos e as especificações de software para o desenvolvimento de sistemas que atendam o padrão USB. A seguir será feita uma breve descrição desse padrão.

A interface USB deve ser considerada um barramento e não uma porta, haja vista que se forem utilizados hubs podem ser conectados diversos dispositivos a uma única porta.

Todos os dispositivos devem estar conectados diretamente ou por meio de um hub ao computador host que atua como um controlador para toda a comunicação USB. Restrições a esta topologia incluem o máximo de sete hubs, o máximo de cinco metros entre cada conexão e um número máximo de 127 dispositivos.

Camadas da comunicação USB

A comunicação no padrão USB utiliza uma série de camadas conforme mostrado na figura 1.7. Cada camada executa uma determinada função e opera com as camadas subjacentes a fim de que a comunicação com o usuário ocorra de forma transparente (Baker e Lozano, 2000).

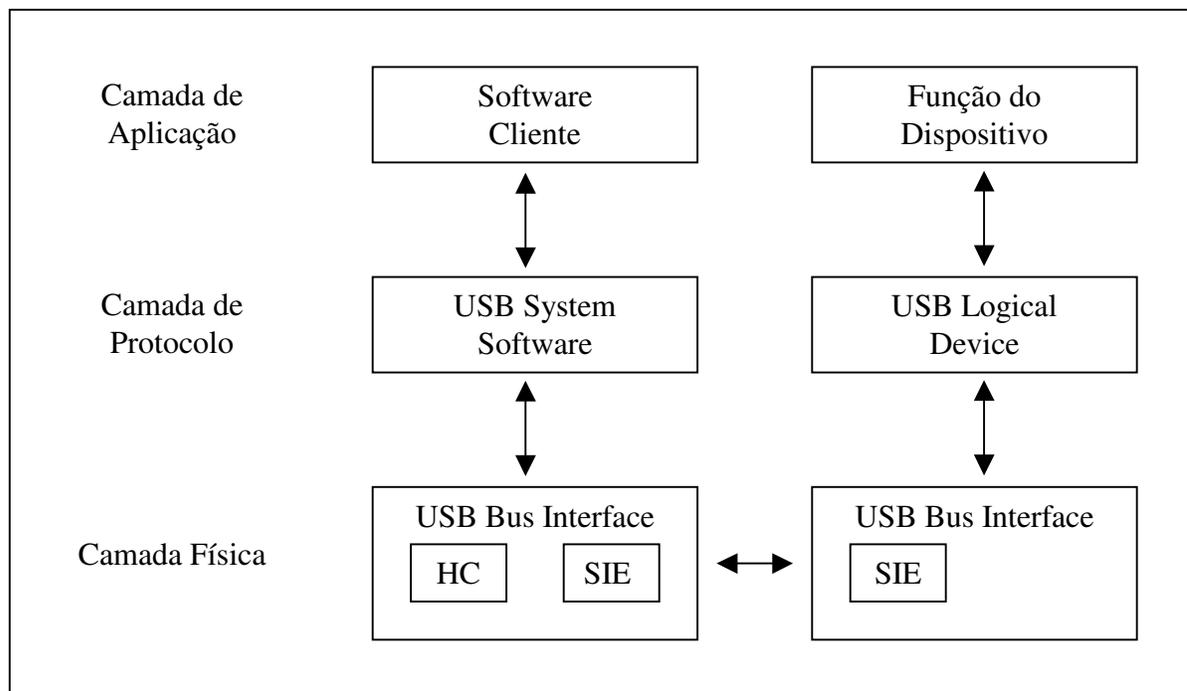


Figura 1.7 - Camada de comunicação do protocolo USB

A Camada Física

A camada física é a base da comunicação USB consistindo do hardware que transmite os sinais digitais "1" e "0". Pelo fato de ser uma comunicação serial toda comunicação ocorre por meio de um par de condutores elétricos. Outras duas linhas fornecem a alimentação de 5 Volts aos dispositivos e a referência de terra.

A camada física no computador (host) é constituída de dois elementos: O SIE (Serial Interface Engine) e o HC (Host Controller). O SIE, que também faz existe no dispositivo periférico, realiza funções que incluem serialização das transmissões, codificação e decodificação dos sinais, geração e verificação de CRC (Cyclic Redundancy Check), e detecção dos pacotes de identificação (PID). O HC é responsável por iniciar as transações e controle de acesso a USB. Ele divide o tempo em quadros e define um pacote de quadro inicial SOF (Start-Of-Frame) a cada intervalo de quadro. Outra função executada é o processamento das requisições de dados do host e manipulação dos erros.

A Camada de Protocolo

A camada intermediária é denominada "USB System Software" no host e "USB Logical Device" no dispositivo. O "USB System Software" aloca largura de banda e gerencia a alimentação do barramento. Também fornece a interface com o HC, identifica, enumera, e provê os serviços de dados requeridos pelos dispositivos no barramento. O "USB Logical Device" é composto por um ou mais endpoints unidirecionais que transferem dados.

Os endpoints são blocos de memória para dados ou registradores no controlador do dispositivo. Um dado armazenado em um endpoint pode ser dado recebido, ou dado aguardando para ser transmitido. O controlador do host também contém memória para dados recebidos e para dados que aguardam transmissão, porém no host essa área de memória ou registrador não recebe o nome de endpoint .

Os dispositivos USB podem conter vários endpoints, divididos em endpoints para envio de dados, para recebimento de dados e para configuração.

Camada de Aplicação

Operando no lado Host, a camada de aplicação é o software cliente pelo qual o usuário ou determinado aplicativo interage com o dispositivo USB.

Protocolo USB

As transações no protocolo USB ocorrem por meio de pacotes. Três tipos de pacotes são necessários para completar uma transação - Token, Data, e Handshake. Cada transação faz parte de uma transferência maior que pode assumir três modos: Por interrupção, bulk e isócrona.

Tipos de Pacotes

Se o host deseja enviar ou receber dados a um dispositivo, ele envia um pacote token endereçado ao dispositivo. Este token é um pacote especial com a composição mostrada na figura 1.8. O ADDR (Device Address) e ENDP (endpoint) somente definem qual endpoint em qual device deve receber ou enviar dados. O campo CRC detecta erros e EOP indica o fim do pacote.

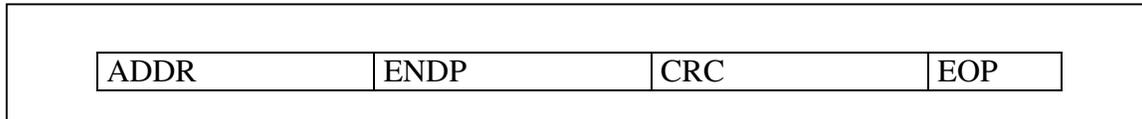


Figura 1. 8 - Pacote Token

Desde que o host envie um token para o dispositivo, o mesmo inicia o envio ou a recepção de pacotes de dados. Pacotes de Handshake são opcionais e contém apenas um bit que indica sucesso ou falha na transação do seguinte modo:

- ACK - a transmissão do pacote foi bem sucedida
- NAK - o dispositivo não está habilitado para enviar ou receber dados em decorrência de um erro. O host ou dispositivo deve reenviar os dados.
- STALL - o endpoint ou o comando de configuração não é suportado.
- NYET - dispositivo ainda não respondeu

Tipos de transferência

O primeiro tipo de transferência, transferência isócrona, garante largura de banda, mas não se preocupa com a precisão nos dados. Neste tipo de transferência não ocorre a fase de Handshake. Dispositivos para áudio e vídeo normalmente usam este tipo de transferência pelo fato da velocidade de transmissão ser essencial, mas sujeita a erros o que não compromete o resultado em razão da visão e audição humana normalmente não serem sensíveis para detectar pequenos erros. Transferências Bulk, geralmente utilizadas em equipamentos como impressoras e digitalizadores variam a largura de banda alocada, mas garantem a precisão dos dados. Teclados, dispositivos apontadores e outros tipos de equipamentos de entrada de dados que regularmente necessitam reportar pequenos blocos de dados ao Host utilizam transferência por interrupção. Um dispositivo USB que utiliza transferência por interrupção define na sua configuração inicial o tempo de intervalo no qual ele deseja enviar ou receber dados. Nesse modo o Host deve consultar o dispositivo no intervalo de tempo especificado para verificar se o mesmo gerou alguma interrupção.

Controladores USB

A seguir são descritos alguns controladores USB disponíveis no mercado e que foram considerados no desenvolvimento do sistema da placa de controle.

Cypress Semiconductor

Produz uma extensa linha de microcontroladores específicos para comunicação USB nas velocidades de 1.5 Mbps, 12 Mps e 480 Mps.

A linha inicial é a Cypress M8 em que predominam controladores de baixa velocidade (1.5 Mbps) com aplicação em dispositivos apontadores e teclados para computadores.

Com a aquisição das empresas AnchorChips, EzUSB e Scanlogic expandiu sua oferta de controladores USB com soluções para as faixas de alta velocidade com produtos das linhas EZ-USB, ISD-300 e USB CY16. A maioria dos produtos dessas séries são baseadas em microcontroladores que empregam o conjunto de instruções do Intel 8051.

Para a utilização dos controladores Cypress é necessário a implementação de um firmware com funções para interpretar e responder adequadamente às requisições de dados, coordenar os modos Suspend e Resume, e se comunicar com o hardware USB do controlador por meios dos Endpoints, registradores USB e Endpoints de interrupção.

As ferramentas de desenvolvimento da Cypress são pagas, comercializadas na forma de kits de desenvolvimento que incluem compiladores, ferramentas de depuração e simulação, placa de desenvolvimento, documentação. Os preços desses kits variam de 200 a 500 dólares.

Netchip

Oferece os controladores de periféricos NET2270, NET2280, NET2890 compatíveis com a especificação USB 2.0. Esses controladores apresentam as seguintes características:

- Otimizado para comunicação bidirecional entre dispositivos como câmeras digitais, impressoras e produtos de áudio entre outros.
- Tecnologia Dynamic Virtual Endpoint permite até 30 fluxos de dados independentes possibilitando a garantia de altas taxas de transferência.
- Controle por software da desconexão, enumeração, modo Suspend, e reset
- Detecção de erros e reenvio de pacotes com falha.
- Documentação e exemplos que facilitam a integração e implementação de firmware com microcontroladores comuns do mercado.

- Oferece kits de desenvolvimento, mas que não são imprescindíveis para o desenvolvimento de aplicações.

Microchip

Os microcontroladores PIC16C745 e PIC16C765 possuem módulos para comunicação USB na camada SIE (Serial Interface Engine) de baixa velocidade. A Microchip fornece um firmware que provê uma interface na camada intermediária entre a SIE e a aplicação do usuário. Esta aplicação também denominada CH9 implementa as funcionalidades do capítulo nove da especificação USB versão 1.1 que descreve as requisições que o host faz ao dispositivo USB e como esse dispositivo responde a essas requisições. As funções Get (receber dados do host) e Put (enviar dados para o host) são disponibilizadas para possibilitar a transferência de dados entre o barramento USB e a aplicação do usuário.

Com o propósito de atender a especificação USB 2.0 a Microchip disponibilizou a partir de 2005 os microcontroladores PIC18F2550, PIC18F2455, PIC18F4550 e PIC18F4455. Este microcontroladores possuem memória de programa que varia de 24 kb a 32 kb, 2048 bytes de RAM, EEPROM para dados com 256 bytes, porta USB com velocidade de 12 Mbits com 16 Endpoints. O MPLAB, um conjunto integrado de ferramentas para desenvolvimento e depuração de aplicativos para os microcontroladores é fornecido gratuitamente pela Microchip.

FTDI

Esta empresa especializada na conversão de periféricos padrões em computadores como o serial RS232 e Centronics para o padrão USB oferece os seguintes dispositivos:

- FT8U232, é um conversor USB para comunicação assíncrona RS232 com taxa de transferência de 300 bps a 920 Kbps, 384 bytes no buffer para recepção e 128 bytes para o buffer de saída, regulador integrado de 3.3 V, não utiliza firmware, pois toda as operações do protocolo USB são realizadas pelo hardware, drivers para o sistemas operacionais Windows, Linux e Mac distribuídos gratuitamente.
- FT8U245AM converte os dados recebidos no protocolo USB em um barramento paralelo. Algumas características desse componente são:
 - o Barramento paralelo de 8 bits
 - o Taxa de transferência de 1 Mbytes/segundo

- o Pinos para controle RD/WR, RXF (buffer de entrada cheio) e TXE (buffer de saída vazio).
- o FIFO para transmissão de 384 bytes e FIFO para recepção de 128 bytes.
- o Regulador integrado de voltagem para 3.3 volts
- o Interface serial para programação de EEPROM que pode armazenar a identificação do dispositivo USB e cadeias de caracteres.
- o Não necessita de firmware
- o Drivers e ferramentas de programação para Windows 98/ME, Windows 2000/XP, LINUX e Apple MAC

National Semicondutores

Esta empresa comercializa os controladores USB USBN9603 e USBN9604 que disponibilizam no mesmo componente para a comunicação dos dados um barramento paralelo e o padrão serial Microwire. Possibilita que os sistemas baseados em microcontroladores sejam desenvolvidos sem a necessidade do aprendizado de uma nova arquitetura. Algumas características desses componentes são:

- Utilização da interface serial ou paralela
- Sinais de controle permitem a operação multiplexada do barramento paralelo.
- Regulador embutido de 3.3 Volts
- Um endpoint de controle com 8 bytes, três endpoints para transmissão com 64 bytes e três endpoints para recepção.

Barramento GPIB

Com o propósito de padronizar a comunicação entre instrumentos de laboratório, a empresa Hewlet-Packard desenvolveu o protocolo de comunicação chamado de HPIB (Hewlet-Packard Interface Bus). Este padrão foi adotado por outros fabricantes de instrumentos o que levou o IEEE (Institute of Electrical e Eletronic Engineers) a desenvolver um documento para padronizar este protocolo. Esta especificação recebeu o nome de GPIB (General Purpose Interface Bus) ou IEEE Standard 488-1978(IEEE, 1978) (Webster e Tompkins, 1987). A adoção do padrão GPIB permite a um computador o controle de até 15 dispositivos a partir de uma única interface. Um dispositivo conectado ao barramento pode enviar dados ou comandos a outros 14 dispositivos presentes no barramento. Por utilizar um handshake assíncrono a troca de dados ocorre com taxas de

transferência que variam conforme o dispositivo. As especificações do hardware limitam a taxa de transmissão máxima a 2 Mbits/s.

O barramento GPIB é constituído por um conjunto de 16 linhas. Oito linhas são utilizadas nas transferências de dados e comandos, e oito linhas como função de controle.

Qualquer dispositivo conectado ao barramento GPIB pode enviar ou receber dados, e ao menos um deve operar como controlador. O dispositivo que envia dados é denominado *talker*, o que recebe dados é chamado de *listener* e o que controla é referenciado como *controller*. As oito linhas de dados enviam bytes que podem ser dados ou comandos originados em um dispositivo *talker* para todos os dispositivos no barramento que estejam atuando como *listener*. Os dispositivos normalmente dispõem de buffer de dados com o propósito de evitar a sobrecarga de tráfego no barramento.

Os dados no GPIB são transmitidos em seqüências de bytes. As linhas de controle DAV (Data Valid), NRFD (Not Ready For Data) e NDAC (Not Data Accepted) gerenciam a transferência dos bytes ao realizar o handshake entre os dispositivos que estão enviando e os que estão recebendo. O handshake assegura que um byte seja colocado no barramento somente quando todos os dispositivos que operam como *listener* estejam habilitados, que os dispositivos *listener* somente realizem uma operação de leitura do barramento se houver um byte válido, e os dispositivos *talker* mantenham os dados no barramento até que o byte possa ser lido por todos os dispositivos *listener*. O diagrama de tempo da figura 1.9 ilustra como as linhas DAV, NRFD e NDAC são utilizadas na troca de dados.

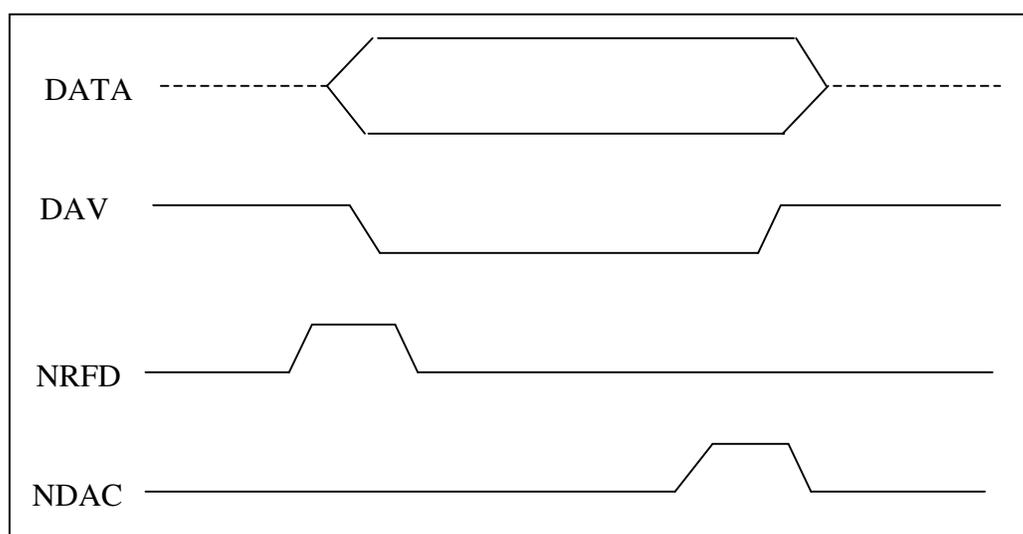


Figura 1.9 - Diagrama de tempo da comunicação GPIB

Após disponibilizar o dado no barramento, o dispositivo transmissor deve aguardar até que o sinal NRFD vá ao nível lógico alto. Com o sinal NRFD em nível alto o dispositivo talker fica habilitado a colocar o dado no barramento, sinalizando ao colocar o sinal DAV em nível lógico baixo quando existe um dado válido no barramento.

Quando os dispositivos *listener* detectam o nível lógico baixo na linha DAV, é realizada a leitura do barramento. Cada dispositivo que termina a operação desabilita a linha NDAC. Após a execução da leitura pelo último dispositivo a linha NDAC é levada ao nível lógico alto indicando ao *talker* que os dados foram recebidos. Após o recebimento dos dados pelos dispositivos listener o dispositivo talker remove os dados do barramento e desabilita o sinal DAV.

Os dispositivos conectados ao barramento GPIB podem enviar ou receber dados, controlar o barramento, ou executar operações que envolvam a combinação dessas funções. Antes de qualquer transmissão o controlador de barramento, entre outras funções, deve especificar qual dispositivo é *talker* e qual é *listener*. As funções de barramento são iniciadas pelo envio de comandos de barramento aos outros dispositivos presentes ao barramento.

Os comandos de barramento são enviados da mesma forma que os dados. A indicação de que o dado presente no barramento é um comando é determinado pelo sinal ATN (Attention) em nível lógico alto.

No sistema do laboratório remoto é utilizado o osciloscópio digital Hewlett-Packard modelo 54503A dotado de interface GPIB e um computador PC com uma placa da empresa National Instruments modelo AT-GPIB/TNT.

Com esse conjunto de hardware foi desenvolvido um sistema de software que possibilita o controle e visualização remota via rede internet dos sinais presentes no osciloscópio. O capítulo IV descreve em detalhes o desenvolvimento desse sistema e sua integração ao laboratório remoto.

Barramento SPI

O barramento SPI (Serial Peripheral Interface) é um padrão de interface serial para aplicações que envolvam a comunicação entre microcontroladores e periféricos que adotam essa especificação. Definido inicialmente pela empresa Motorola para a linha de microcontroladores MC68HCxx atualmente é implementado em muitos microcontroladores

produzidos por empresas como Microchip, Atmel, Analog Devices e periféricos como potenciômetros digitais, conversores analógicos digitais, memórias entre outros.

Constituído fisicamente por quatro linhas de sinal é basicamente uma interface de comunicação simples indicada para a conexão e comunicação de dispositivos externos com microcontroladores a baixa e média velocidade, suportando taxas de transferência de até 1 Mbaud. Um clock síncrono desloca dados que podem ser recebidos e enviados pelo microcontrolador em blocos de 8 bits (Catsoulis,2002).

O barramento SPI é uma interface no modo mestre/escravo. Quando dois dispositivos estão conectados, um é o mestre responsável por gerar o clock e o outro é o escravo. A comunicação ocorre no modo full-duplex na qual os dados são transmitidos e recebidos simultaneamente.

Os sinais do barramento SPI são:

- SCLK é sinal de clock serial, sempre gerado pelo dispositivo mestre. Este sinal controla quando e em qual velocidade os dados serão trocados entre dois dispositivos.
- SS (Slave Select) seleciona o dispositivo escravo para o recebimento dos sinais de dado e clock permitindo que vários dispositivos escravos possam ser conectados em paralelo. Quando o sinal SS é levado ao nível lógico 0 o dispositivo escravo obtém o controle do barramento.
- SDO (Serial Data Out) saída de dados do dispositivo.
- SDI (Serial Data Input) entrada de dados para o dispositivo.

No barramento SPI os dados normalmente mudam durante a borda de subida ou descida do clock, ocorrendo deste modo o sincronismo dos dados. O ponto em que o sinal é lido corresponde ao oposto do qual ele muda. Na figura 1.3 pode ser visto o diagrama da comunicação do SPI para a situação em que o dado muda na borda de descida do clock e está disponível para leitura na borda de subida.

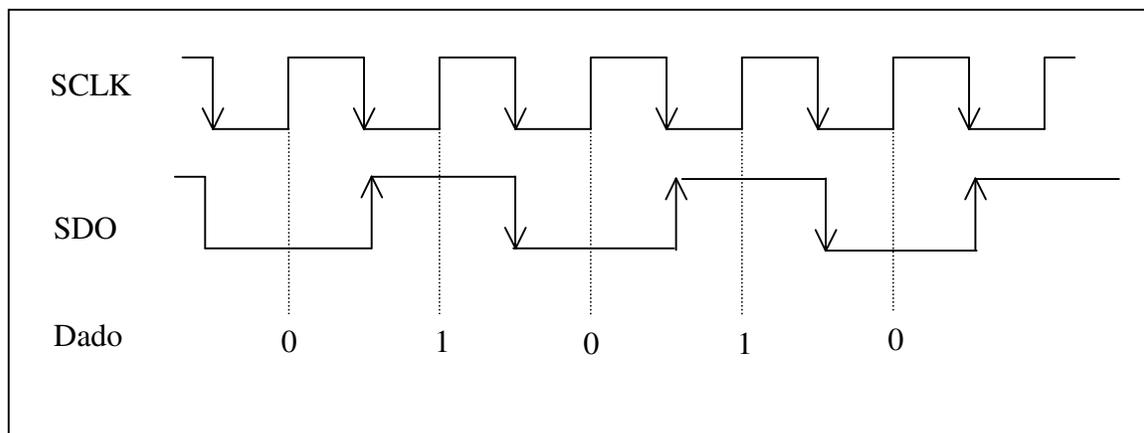


Figura 1.10 - Diagrama de tempo da comunicação SPI

Em uma configuração típica o barramento SPI é conectado com a topologia de laço. Os dados partem do dispositivo mestre pelo pino SDO e os dados originados no dispositivo escravo são recebidos pelo pino SDI do dispositivo escravo. A figura 1.11 ilustra essa configuração.

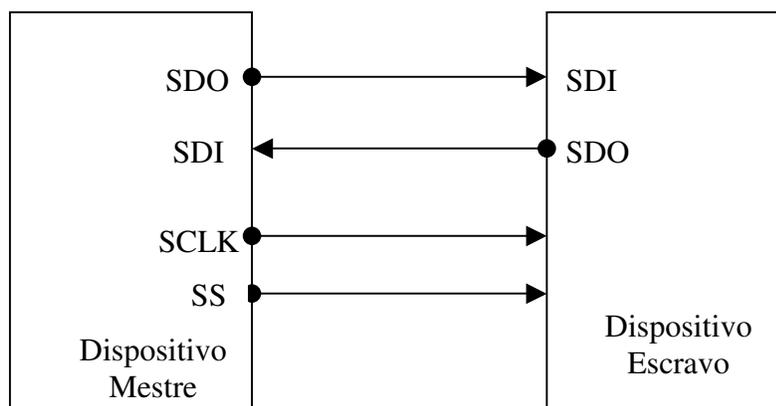


Figura 1.11 - Conexão de dispositivos SPI

Neste trabalho foram utilizados potenciômetros digitais da empresa MAXIM que emprega o barramento SPI como forma de recebimento dos dados que alteram os parâmetros resisitivos do mesmo. No capítulo III é discutida a implementação das rotinas no microcontrolador para a comunicação SPI e o hardware para o microcontrolador e o potenciômetro digital.

Software

A operação do laboratório de acesso remoto compreende a comunicação em rede, com instrumentos comerciais, e hardware específico implementado no projeto. Todos esses sistemas devem estar integrados e acessíveis ao usuário por meio de uma interface gráfica adequada. Para desenvolver estes sistemas de software, foram estudadas as várias opções disponíveis atualmente, tendo como referência tecnologias padrão aceitas amplamente favorecendo a operação nas mais diversas plataformas e pode facilitar a evolução dos sistemas.

Comunicação por Socket

Em meados de 1960 um dos departamentos de defesa dos Estados Unidos (ARPA - Advanced Research Projects Agency), incumbiu a Universidade de Berkeley a tarefa de desenvolver uma plataforma padrão de software para suportar uma rede denominada ARPAnet, embrião da Internet no formato como é conhecida atualmente. A universidade de Berkeley que já era reconhecida pelo desenvolvimento do sistema operacional UNIX adicionou uma nova interface a esse sistema operacional para suporte a comunicação em rede. Esta interface é conhecida como BSD (Berkeley Sockets Interface) e se tornou a base para a maioria das interfaces para o protocolo de rede TCP/IP, incluindo o Windows Socket, mais conhecido como Winsock (Comer, 2000).

O socket pode ser considerado como um sistema para comunicação que ocorre em um canal de duas vias. Conectando-se dois sockets é possível trocar dados entre processos, independente desses processos estarem em diferentes computadores. A comunicação por socket adota o modelo cliente-servidor no qual os elementos básicos são os aplicativos servidor e cliente. O servidor é um aplicativo que aguarda a solicitação de dados e usualmente provê algum serviço ou dado a outros aplicativos. O cliente é um programa que se conecta ao servidor normalmente para solicitar ou enviar algum dado.

Em uma rede TCP/IP a identificação das máquinas ocorre pelo endereço IP e é por ele que os sockets são endereçados. Além do endereço IP também é designado um número de porta com o propósito de permitir que mais de um processo possa ser executado no mesmo computador.

Outro parâmetro a ser configurado na implementação da comunicação por socket é o tipo do socket. Os dois tipos mais comuns são o SOCK_STREAM e SOCK_DGRAM .

No tipo SOCK_STREAM os dados trafegam no socket como um fluxo de caracteres dispondo de recursos para estabelecer uma fila. Assim se o aplicativo servidor estiver ocupado atendendo uma conexão e for recebido outro pedido de conexão, a mesma entra em uma fila aguardando até que o servidor possa atende-la. No tipo SOCK_DGRAM os dados trafegam em pacotes denominados datagramas.

No sistema do laboratório remoto existe a comunicação via rede internet entre vários processos e computadores. A comunicação via socket foi utilizada na implementação dos servidores e clientes que realizam a troca de dados entre os instrumentos, interfaces de hardware e interação com o usuário. O capítulo IV descreve a implementação desses aplicativos.

XML

Muitos documentos na rede Internet são armazenados e transmitidos no formato HTML. Baseada no SGML (Standard Generalized Markup Language, ISSO 879) o HTML é uma linguagem simples apropriada para conteúdos baseados em hipertextos, multimídia (vídeos digitais, arquivos de imagens) e exibição de textos.

O SGML é um padrão para definição de formatos e uso de documentos eletrônicos, permitindo que os documentos descrevam sua própria gramática, o que significa, especificar o conjunto de marcas (tag) usadas no documento e a representação do relacionamento da estrutura de marcas.

Uma aplicação HTML é um pequeno conjunto de marcas em conformidade com a especificação SGML. A utilização de um pequeno conjunto de marcas possibilita ao usuário separar a especificação da linguagem do documento e seu conteúdo, o que facilita o desenvolvimento de aplicações. Porém essa facilidade limita o HTML em aspectos importantes relacionados à extensibilidade, estrutura e validação descritos a seguir.

- Extensibilidade, o HTML não permite ao usuário especificar suas próprias marcas de modo a parametrizar ou qualificar semanticamente os dados.
- Estrutura, não oferece suporte a especificação de estruturas necessárias à organização para representar base de dados ou hierarquias baseadas em esquemas orientados a objetos.
- Validação, não dispõe de recursos para que a aplicação consumidora de dados faça validação da estrutura de dados no momento da importação.

Uma aplicação genérica que atenda o padrão SGML torna possível a elaboração de documentos complexos com recursos de extensibilidade, estrutura e validação. Essas características são possíveis pelo fato do SGML definir seus próprios formatos para o documento, pela facilidade de manipular documentos complexos e extensos, e gerenciar grandes repositórios de informação. Porém o SGML contém um conjunto de recursos que não são necessários para a maioria de aplicações baseadas na WEB o que ocasiona uma relação custo/benefício pouco atrativa para os desenvolvedores de soluções para a WEB.

Com o propósito de buscar uma solução que pudesse se beneficiar das vantagens do SGML o W3C (World Wide Consortium) criou o SGML Working Group que se ocupou de desenvolver um conjunto de especificações que possibilitem a utilização das principais características do SGML com facilidade. O objetivo das ações do W3C SGML (W3C, 1996) foi criar meios para que a distribuição de dados que contenham a descrição de sua estrutura com flexibilidade suficiente e na profundidade e complexidade necessária para as aplicações que venham se utilizar dessa estrutura (Moultis e Kirk, 2000).

O resultado do trabalho do grupo W3C SGML foi a definição de um conjunto simplificado do SGML especialmente projetado para aplicações WEB. Este conjunto foi chamado de XML (Extensible Markup Language) (W3C,2006), que conserva as características mais relevantes do SGML como extensibilidade, estrutura e validação em uma linguagem fácil de ser assimilada, utilizada e implementada.

Em relação ao HTML, o XML difere em três aspectos fundamentais:

- O desenvolvedor pode definir novas marcas atribuindo-lhe denominações de acordo com a sua conveniência.
- A estrutura dos documentos pode ser aninhada a qualquer nível de complexidade.
- O documento XML pode conter uma descrição opcional da sua gramática para uso de aplicações que necessitem realizar validação da estrutura de dados.

Os documentos XML tem um formato semelhante aos documentos SGML e HTML. A tabela 1.1 contém o exemplo de um documento XML.

```
<?XML version = "1.0"?>
<ALUNO>
  <NOME>
    Pedro Silva
  </NOME>
  <RA>
    003399
  </RA>
</ALUNO>
```

Tabela 1.1 Exemplo de um documento XML

Neste exemplo o documento inicia com a instrução de processamento `<?XML version = "1.0"?>`. Essa linha de código é chamada declaração de marcação XML e serve para informar ao processador que esse é um documento XML utilizando a versão 1 do XML para estruturar o documento. Os elementos desse documento são Aluno, Nome e Ra. Para que possam ser processados como documentos XML devem ser demarcados. No exemplo `<ALUNO>` indica a abertura do elemento, o que vem a seguir o dado `</ALUNO>` é a marcação de fechamento do elemento. A estruturação do documento se dá por meio do aninhamento dos elementos, no exemplo o elemento `<ALUNO>` aninha os elementos `<NOME>` e `<RA>`.

A seguir são descritas algumas aplicações que podem se beneficiar da utilização do XML:

- Aplicações que necessitem a mediação entre duas ou mais bases de dados heterogêneas.
- Aplicações que transferem para o cliente uma significativa carga de processamento.
- Aplicações que necessitam apresentar os dados com diferentes representações para diferentes usuários.
- Aplicações na quais agentes de software obtêm as informações de acordo com o definido pelo usuário (Erl, 2004).

Na arquitetura do Laboratório remoto o XML é utilizado para armazenar as diversas propriedades dos objetos de hardware do experimento como chaves eletrônicas, potenciômetros digitais, relês, as informações do projeto como nome, os elementos e sua apresentação no cliente, entre outros.

A utilização do XML possibilita um modelo de representação dos dados que garante a independência de plataforma e facilidade para a inserção de novos objetos quer sejam de hardware ou software, alteração dos existentes e a implementação de clientes com as mais diversas tecnologias que oferecem suporte ao XML. No capítulo IV é discutido em detalhes a implementação e uso do XML na arquitetura do Laboratório Remoto.

COM e DCOM

A arquitetura de software COM (Component Object Model) foi desenvolvida pela empresa Microsoft com o objetivo de oferecer uma ferramenta para o desenvolvimento de aplicações baseadas em componentes (Rubin e Brain, 1999). Objetos COM são componentes discretos de identidades únicas que adotam um padrão de interface que permite a aplicações e outros componentes o acesso a suas funcionalidades. Em um ambiente com sistema operacional Windows os objetos COM são mais versáteis que as bibliotecas dinâmicas (Win32 DLL) em razão de serem independente de linguagem e oferecer recursos para comunicação entre processos e se enquadrarem facilmente no modelo de desenvolvimento orientado a objetos. A primeira versão da COM foi lançada em 1993.

Com o propósito de permitir aos objetos COM que a comunicação de processos ocorresse em rede de computadores, no ano de 1995 foi lançada a arquitetura DCOM (Distributed Component Object Model). A DCOM usa o mecanismo de RPC¹ para de forma transparente ao programador realizar a troca de informações entre objetos COM em uma rede.

Com base na tecnologia COM foram desenvolvidos os controles ActiveX, que encapsulam objetos COM e oferecem recursos como auto-registro e definição do ciclo de vida.

Os controles ActiveX podem ser descarregados de um servidor e executados em um navegador web padrão, o que possibilita o desenvolvimento de aplicações distribuídas. Eles tem operação similares a applets Java, mas oferecem acesso completo aos recursos do sistema operacional Windows. Isto propicia mais poder ao controle ActiveX aumentando contudo o risco de danos a aplicativos ou dados no computador no qual ele está sendo

¹ Protocolo que permite a um aplicativo executado em um computador invocar uma subrotina em outro computador sem que o programador precise explicitar código que detalhe essa interação.

executado. Para evitar esses riscos a empresa Microsoft desenvolveu um sistema de registro para que os navegadores web possam identificar e autenticar o controle ActiveX antes que ele seja descarregado e executado.

Ferramentas de desenvolvimento como C++, Microsoft Visual Studio, Delphi da empresa Borland, Labview, entre outras, possibilitam a implementação de aplicativos que incorporam controles Activex.

O estabelecimento do padrão para controles ActiveX fez florescer uma indústria de componentes que atendem os mais diversos tipos de aplicação incluindo instrumentação e automação. Um exemplo são os controles produzidos pela empresa National Instruments para comunicação com o hardware de instrumentos de medição, interfaces gráficas e comunicação em rede.

Java

A linguagem de programação Java foi desenvolvida pela empresa Sun Microsystems no ano de 1991 como parte de um projeto de pesquisa que planejava o desenvolvimento de software para equipamentos eletrônicos do mercado de consumo como televisores, aparelhos de vídeo, áudio e eletrodomésticos em geral. O objetivo principal desse projeto é que a linguagem Java fosse pequena, rápida, eficiente e facilmente portátil para a maioria dos dispositivos de hardware.

Com o crescimento da rede Internet na mesma época do desenvolvimento da linguagem Java, houve uma demanda por aplicações que pudessem ser distribuídas e executadas nas mais diversas plataformas computacionais e também por uma linguagem de propósito geral para o desenvolvimento de aplicativos portáteis para serem usados em plataformas heterogêneas (Potts e Friedel, 1996).

Um sistema mínimo Java é composto pelo compilador e um interpretador ou máquina virtual que é específica para cada plataforma. Ao compilar o código fonte Java é gerado um bytecode que normalmente pode ser executado sem mudanças em qualquer plataforma que disponha da máquina virtual.

Para o desenvolvimento de aplicativos que precisam acessar recursos nativos do hardware da plataforma, como no caso de automação e controle onde é preciso acessar instrumentos, barramentos, portas foi desenvolvido o JNI (Java Native Interface) .

O JNI é um padrão de interfaces para a implementação de métodos nativos da plataforma que são embutidos na máquina virtual Java (Sun, 2003).

Com esse padrão de interface é possível obter compatibilidade binária de métodos nativos nas diversas implementações de máquinas virtuais para uma determinada plataforma de hardware.

Com JNI é possível invocar métodos de classes Java e acessar propriedades como strings e arrays de módulos como DLLs ou executáveis, viabilizando a utilização de Java em processos de automação e controle.

Tecnologias empregadas no desenvolvimento

Após o estudo das várias tecnologias consideradas chave para o desenvolvimento da arquitetura do laboratório foi realizada a seleção levando em conta critérios como aceitação pelo mercado e desenvolvedores, portabilidade, custo, disponibilidade de documentação e recursos de software e hardware.

Do ponto de vista de software o XML foi escolhido para a troca de dados entre os vários aplicativos o que garante interoperabilidade entre aplicativos distintos e facilidade para evoluções nos sistemas. Esses dados são transportados remotamente pela rede Internet por meio de sockets.

O acesso do usuário ao sistema remotamente se dá por meio de um navegador web que carrega objetos ActiveX contendo os recursos para controle e visualização de dados.

No desenvolvimento do laboratório remoto foi implementado um circuito eletrônico, denominado placa de controle, para controle dos experimentos que se comunica com o computador recebendo e enviando comandos. Para realizar a comunicação entre o computador e placa de controle foi escolhido o padrão USB em razão de algumas vantagens descritas como flexibilidade, disponibilidade de controladores com baixo custo, documentação adequada e a tendência do mercado de substituir nos computadores pessoais as interfaces antigas como serial (RS232) e paralela (Centronics) pelo padrão USB. No capítulo III é apresentada a implementação da placa de controle cuja comunicação ocorre por meio do padrão USB.

A placa de controle tem como principais componentes o microcontrolador da empresa Microchip e um controlador USB da empresa FTDI. A escolha do microcontrolador levou em consideração a documentação disponível, facilidade de

aquisição no mercado nacional, preço e disponibilidade de ferramentas de desenvolvimento gratuitas. O controlador USB da empresa FTDI foi escolhido pela facilidade de aquisição, ferramentas de desenvolvimento e drivers bem documentados e gratuitos.

O capítulo IV descreve a implementação do software cliente no host para comunicação com a placa de controle.

Para o desenvolvimento dos aplicativos executados no computador PC foram utilizados as ferramentas de programação Microsoft Visual Studio, bibliotecas e outros recursos do Labview, drivers e ferramentas específicas para diagnósticos e testes no barramento USB. Na implementação do firmware do microcontrolador foi utilizada a ferramenta MPLAB.

Capítulo 2 - Arquitetura do Laboratório Remoto

De maneira geral, para desenvolver um experimento que envolva controlar dispositivos de hardware remotamente é necessário dispor de uma interface de hardware conectada ao computador, e um sistema de software. Este sistema de software normalmente é composto por um cliente e um servidor. No módulo cliente, para permitir a interação do usuário, pode existir uma interface gráfica ou os comandos são enviados por meio de comandos no modo texto. Outra função do cliente é de comunicação com a rede Internet necessária para encaminhar os comandos que serão executados no servidor. O módulo servidor deve ter funcionalidades para comunicação com a rede Internet necessária para o recebimento dos comandos originados no cliente e comunicação com o hardware para permitir que as ações do cliente possam ser executadas no dispositivo de hardware.

Para definição dos requisitos da arquitetura e funcionalidades que foram implementadas utilizou-se a técnica de caso de uso que busca mapear as ações que serão executadas em determinado sistema (Griver , Arnheiter e Gellis , 2000).

Caso de uso para o desenvolvimento de um sistema de experimento

Com o propósito de comparar o processo de desenvolvimento de um sistema para experimentos remotos utilizando a abordagem adotada nos laboratórios estudados no capítulo I e arquitetura proposta nesse trabalho, foi elaborado o caso de uso no qual são demonstradas as ações necessárias para implementar um sistema desse tipo nas duas situações. Nos laboratórios estudados o desenvolvedor deve ter o domínio das ferramentas de hardware e software para implementar os aplicativos que se comunicam com a rede e acessam o hardware. Na arquitetura proposta os aspectos técnicos relacionados à comunicação com a rede, interface gráfica, acesso ao hardware são providos pelo sistema, criando condições para que o desenvolvedor possa se concentrar na exploração dos aspectos pedagógicos do experimento que ele planeja realizar.

O caso de uso demonstrado a seguir descreve passos para desenvolvimento de um sistema de experimentos adotando a abordagem dos laboratórios estudados no capítulo I.

- Selecionar uma interface de hardware ou projetar uma que atenda as necessidades do experimento.

- Implementar os circuitos eletrônicos para controle de chaves, potenciômetros, entre outros.
- Selecionar uma ferramenta para o desenvolvimento do sistema de aplicativos
- Conhecer o modo de comunicação da interface de hardware com o computador e adquirir por meio de estudo o domínio para implementar o aplicativo que irá acessar o hardware.
- Desenvolver aplicativos para checar e avaliar a operação da interface de hardware.
- Estudar as formas de comunicação com a rede Internet e desenvolver um aplicativo que integre o acesso ao hardware a partir de dados recebidos via Internet.
- Estabelecer um protocolo de comunicação básico para troca de dados entre a interface de hardware e os aplicativos remoto e local
- Desenvolvimento de um aplicativo com interface gráfica que demonstre o experimento e ações associadas a ele de forma clara e com recursos de comunicação em rede para operação remota.
- Integração dos sistemas.
- Testes avaliação e depuração dos vários sistemas.
- Disponibilizar para o usuário.

Os passos descritos acima não se aprofundam em detalhes e, portanto não pretende esgotar todas as ações necessárias para implementação de um sistema para experimento remoto.

Com a arquitetura proposta nesse trabalho é possível elaborar os experimentos e executá-los remotamente. Em uma situação típica o professor ou laboratorista irá elaborar o experimento e o aluno irá executá-lo. A seguir descrevemos os passos necessários a elaboração de experimento e execução do mesmo.

- Planejar experimento:
- Montagem do circuito na placa do experimento já disponível.
- Conexão dos Blocos Eletrônicos ao circuito do experimento.
- Execução do aplicativo Editor de Experimentos.
- Inserir a imagem com o esquemático do circuito eletrônico, selecionar na barra de ferramenta os Blocos Eletrônicos que foram associados ao circuito do experimento e posicioná-los sobre a área de trabalho. Clicando sobre o objeto Bloco Eletrônico pode-

se mudar suas propriedades como estado, valor inicial, em qual porta o mesmo está conectado, entre outras propriedades.

- Após repetir o passo 5 para todos os Blocos eletrônicos é possível gravar o projeto e avaliar o mesmo com o Cliente do Experimento Remoto.

A descrição dos itens do caso de uso para o desenvolvimento do experimento remoto possibilitam constatar que com esta abordagem o número de ações é superior e essas ações supõem o domínio de conhecimentos técnicos que podem demandar um tempo longo de aprendizado. No desenvolvimento a partir da arquitetura proposta o número de passos é menor e dispensa conhecimentos técnicos relacionados à comunicação com o hardware e rede, implementação de interfaces gráficas e outros aplicativos. Um aspecto a considerar é que trabalhando com um sistema previamente avaliado e estável o tempo de diagnóstico e depuração pode ser reduzido de maneira significativa.

Na arquitetura proposta um caso de uso possível para utilização pelo usuário/cliente é descrito a seguir.

- O aluno deve abrir o navegador e apontar para o endereço do Servidor do Experimento.
- Na tela inicial é solicitado o nome de usuário e a senha para efetivar o login.
- Após o login é apresentada a interface gráfica contendo a figura com esquemático do circuito eletrônico e a representação gráfica dos Blocos Eletrônicos que fazem parte do experimento. Atuando sobre os Blocos Eletrônicos os comandos são enviados ao Servidor de Experimentos e executados na Placa de Experimento.
- A interação com os Blocos Eletrônicos se dá por meio de objetos gráficos como botões que mudam a imagem, no caso de chaves indicando a situação de ligada ou desligada, sliders para variar o valor de potenciômetros e caixas de seleção para definir o canal do osciloscópio utilizado.
- Um botão na interface gráfica permite abrir uma janela com o aplicativo Cliente GPIB que fornece a visualização dos sinais e controle do osciloscópio conectado a Placa de Experimento.

A interface gráfica do cliente executada em um navegador WEB é familiar nos aspectos de apresentação e interação com o usuário, eliminando o esforço e tempo necessários ao aprendizado de um novo aplicativo.

Interfaces gráficas

Houve especial cuidado na implementação da interface gráfica com o usuário, pois é por meio dela que ocorrerão todas as atividades do aprendiz no laboratório remoto. Como a ação do usuário ocorrerá por meio de um navegador para web alguns princípios de usabilidade foram seguidos incluindo a clareza na arquitetura da informação, facilidade de navegação, simplicidade, relevância no conteúdo, manutenção da consistência e foco no usuário (Nielsen, 1999) e serão discutidos de forma breve a seguir.

- **Clareza na Informação** - É importante que o usuário ao acessar o conteúdo de uma página tenha condições de discernir qual o propósito e o que é prioridade. Para tanto as informações devem ser organizadas e apresentadas de forma concisa e objetiva.
- **Facilidade de Navegação** - O usuário deve obter a informação desejada com o menor número de ações (cliques do mouse) possíveis. Neste sentido a organização da informação tem importante papel.
- **Simplicidade** - Ao acessar uma página o usuário deseja encontrar a informação a mais rápida possível. Portanto qualquer recurso que desvie a atenção do usuário deve ser evitado. No entanto deve-se cuidar para que a busca da simplicidade não resulte em omissão de informação.
- **Relevância no conteúdo** - Diferente de outros meios de comunicação como revistas impressas e a televisão, na Web o foco deve estar na informação. Os textos e imagens têm que ser o mais concisos e objetivos, implicando na adoção de um estilo de redação e apresentação otimizado para os usuários on-line.
- **Manutenção da consistência** - Um aplicativo assim como uma página na Web devem dar respostas as ações do usuário de forma consistente e uniforme de modo que com o uso ele não precise se preocupar com o que vai acontecer.
- **Foco no usuário** - O projeto de um sistema para a Web deve estar focado nas atividades que o usuário irá desempenhar.

Aspectos considerados no desenvolvimento buscam tornar a realização das atividades uma experiência mais próxima possível da que ocorre presencialmente no modo real.

A interface gráfica desenvolvida representa os circuitos eletrônicos e componentes por meio de simbologia padrão e portanto facilmente reconhecível pelo usuário.

A atuação no experimento ocorre por meio de uma interface de manipulação direta na qual não existem operações escondidas, sintaxes ou nomes de comandos para aprender. O único conhecimento requerido é no próprio domínio da tarefa (Rocha e Baranauskas, 2003). Assim para ligar ou desligar uma chave no circuito eletrônico experimental basta clicar no símbolo eletrônico que representa chave. A realimentação da ação do usuário ocorre pela mudança do desenho na interface gráfica.

Arquitetura do laboratório de acesso remoto

A arquitetura foi modelada tendo como foco o conjunto de informações que trafegam entre os diversos módulos. A adoção desse modelo permite planejar o desenvolvimento dos diversos aplicativos independente da tecnologia empregada. A figura 2.12 apresenta de forma geral a arquitetura do laboratório remoto.

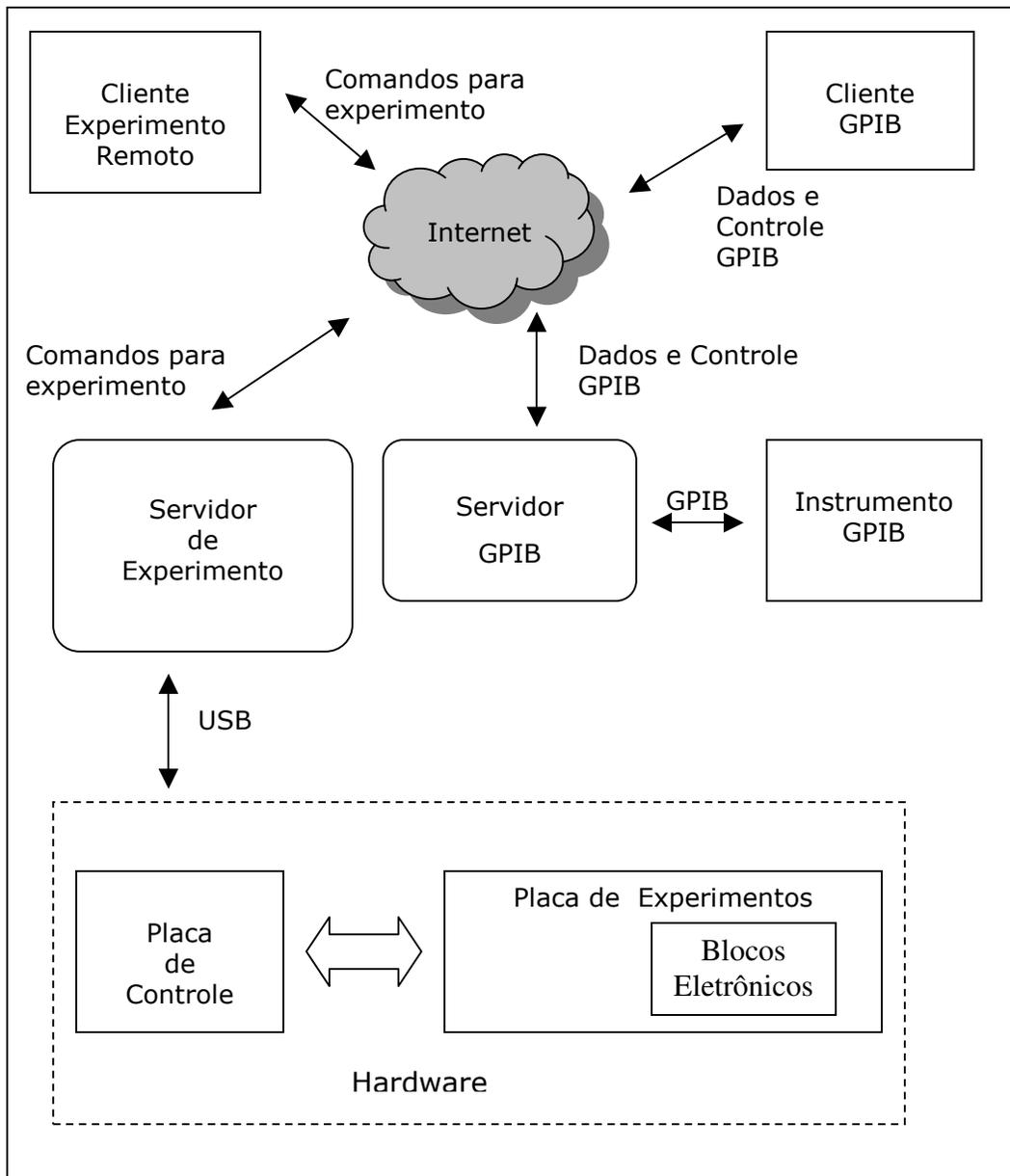


Figura 2.12 - Diagrama da arquitetura do Laboratório Remoto

A seguir são apresentados os componentes de hardware e software que compõem o laboratório de acesso remoto. Esses itens terão suas implementações discutidas em detalhes nos próximos capítulos.

- **Placa de Experimento** - Placa eletrônica contendo o circuito eletrônico que será disponibilizado para o experimento, dispõe de conectores para inserção dos Blocos Eletrônicos que permitem a variação dos diversos parâmetros do circuito.
- **Blocos Eletrônicos** - São circuitos eletrônicos com funções dos principais componentes utilizados em eletrônica como potenciômetros, chaves eletrônicas, banco

de resistores, entre outros. Os blocos eletrônicos recebem os comandos originados no computador e que convertidos na placa de controle variam seus valores e estados.

- **Placa de Controle** - Circuito eletrônico conectado ao computador pela porta USB é responsável por receber os comandos de acionamento e controle originados no aplicativo do PC para atuar nos blocos eletrônicos da placa de experimentos.
- **Editor de Experimentos** - É um aplicativo que será utilizado pelo desenvolvedor do experimento remoto. Utilizando uma interface gráfica constituída de uma área de trabalho e por uma barra de ferramentas possibilita ao usuário selecionar os Blocos Eletrônicos e objetos gráficos que serão utilizados no experimento, e por meio do recurso arrastar e soltar posicioná-los na área de trabalho. Clicando com o botão direito abre-se um formulário no qual é possível atribuir valores às propriedades disponíveis em cada Bloco Eletrônico e objeto gráfico.
- **Servidor de Experimentos** - Aplicativo responsável por receber os comandos originados no Cliente do Experimento Remoto e encaminhar esses comandos para a porta USB do computador para utilização pela Placa de controle.
- **Cliente do Experimento Remoto** - aplicativo executado em um navegador Web padrão, é responsável por disponibilizar uma interface gráfica com os elementos criados no editor de experimentos e permitir ao usuário atuar sobre esses elementos. Por exemplo, clicando em uma chave essa muda o gráfico para o estado de aberta ou fechada propiciando ao aluno uma visualização próxima ao real. Ao mesmo tempo envia um comando ao Servidor de Experimento para atuar no Bloco Eletrônico previamente configurado.
- **Servidor GPIB** - Geralmente nos experimentos eletrônicos instrumentos como osciloscópios são empregados para visualização gráfica dos sinais encontrados nos circuitos. A maioria dos osciloscópios comerciais se comunicam com o computador por meio do protocolo GPIB para enviar dados e receber comandos de controle. Para permitir ao usuário do experimento visualizar os sinais e controlar o osciloscópio foi desenvolvido o Servidor GPIB que se comunica com o hardware de um osciloscópio e com a rede Internet possibilitando o controle do mesmo remotamente pelo Cliente GPIB.

- **Cliente GPIB** - responsável por se comunicar com a rede, recebendo dados e enviando comandos de controle para o osciloscópio remoto. A interação com o usuário ocorre por meio de uma interface gráfica constituída dos botões de comandos e tela que representam um osciloscópio real.

Capítulo 3 - Projeto e implementação do Hardware

Este capítulo descreve a implementação do hardware dos diversos módulos que compõem a arquitetura do laboratório de acesso remoto. Em razão de muitos aspectos de implementação e funcionalidade dos módulos de hardware estar diretamente relacionada ao programa do microcontrolador, as rotinas implementadas para cada módulo serão apresentadas associadas à demonstração do desenvolvimento do circuito eletrônico.

O sistema de hardware é constituído pela Placa de Controle, Placa de Experimento e Blocos Eletrônicos.

A Placa de Controle tem como elementos principais o microcontrolador e o controlador para comunicação com o computador host via porta USB. A placa de controle é responsável por gerenciar a comunicação USB, os protocolos SPI e RS232, e gerar os sinais de strobe, clock e dados para os diversos componentes.

A Placa de Experimento é uma placa de circuito contendo uma área livre, para que o usuário monte o circuito que será usado como experimento, e os componentes responsáveis pelo controle dos blocos eletrônicos, ponto de medição do osciloscópio e comunicação com a Placa de Controle.

Os Blocos Eletrônicos são circuitos controlados pela Placa de Controle que reproduzem as funcionalidades de chaves eletrônicas, potenciômetros, seleção de componentes que poderão ser utilizados no circuito do experimento e chaveamento para pontos de medição empregando instrumentos de medida. Os Blocos Eletrônicos facilitam o desenvolvimento de um experimento, pois foram planejados para serem incorporados ao circuito eletrônico de forma semelhante ao componente de controle manual, dispensando mudanças significativas em circuitos já existentes.

Cada Placa de Experimento dispõe de recursos para controlar até 43 blocos eletrônicos sendo 8 potenciômetros digitais, 23 relês, 8 pontos para instrumento de medida e 4 bancos de componentes. Como cada Placa de Controle pode controlar até 8 Placas de Experimentos, é possível a um sistema atuar com até 344 Blocos Eletrônicos.

A seguir são descritos os principais componentes eletrônicos utilizados na implementação no sistema de hardware do laboratório de acesso remoto.

PIC16F877

O componente PIC16F877 (Microchip, 2004) é um microcontrolador de 40 pinos (figura 3.13) fabricado pela empresa Microchip empregando tecnologia de fabricação CMOS. Dispõe de memória FLASH para programação, o que permite a reprogramação sem a necessidade de utilizar luz ultravioleta para apagar a memória. Necessita de poucos componentes externos, o que reduz custos e o consumo de energia, aumentando a confiabilidade do conjunto eletrônico.

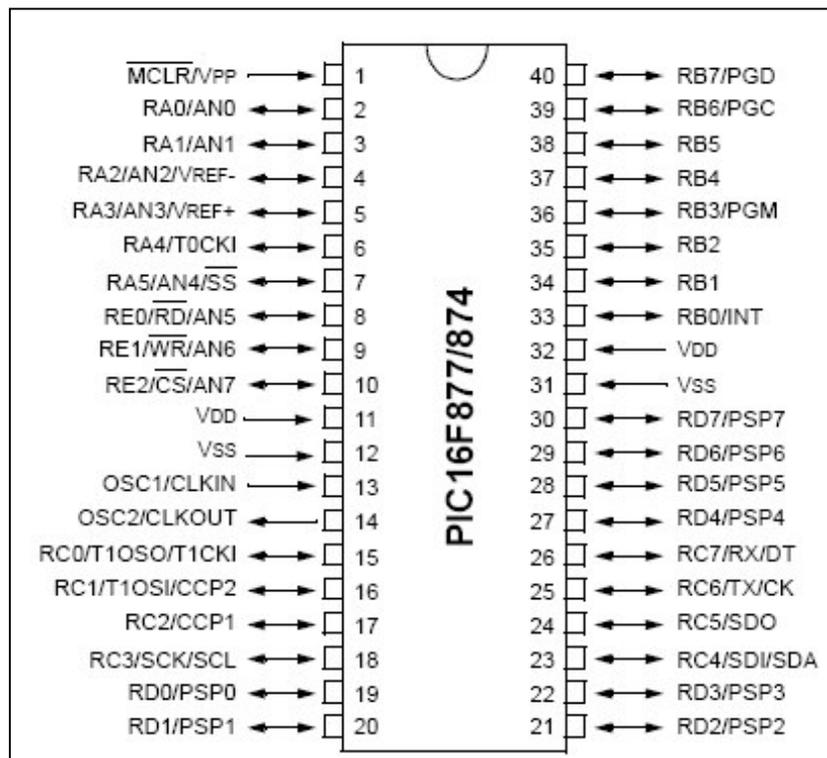


Figura 3.13 - Microcontrolador PIC16F877

Características do PIC16F877:

- Clock de até 20 Mhz
- 33 pinos configuráveis como entrada ou saída distribuídos nos registradores PORTA, PORTB, PORTC, PORTD e PORTE.
- 14 níveis de interrupção.
- 8 kbytes de memória Flash para programas
- Memória para dados RAM de 368 bytes.

- Memória EEPROM (não volátil) interna de 256 bytes.
- Via de programação com 14 bits e 35 instruções.
- 8 entradas com conversor analógico digital de 10 bits
- Comunicação serial SPI
- Comunicação I2C
- Usart Rs232
- 2 contadores de 8 bits e um de 16 bits
- 2 módulos de compare/capture e PWM (Pulse With Modulation)

O microcontrolador PIC16F877 adota a arquitetura do tipo Havard, enquanto grande parte dos microcontroladores tradicionais apresenta uma arquitetura tipo Von Neumann. A diferença entre as duas arquiteturas está na forma como dados e programas são processados pelo microcontrolador. Na arquitetura tradicional, tipo Von Neumann, existe apenas um barramento (bus) interno (geralmente de 8 bits), por onde passam as instruções e os dados. Já na arquitetura tipo Havard existem dois barramentos internos, sendo um de dados e outro de instruções (Tanenbaum, 1999). No caso dos microcontroladores PIC, o barramento de dados é sempre de 8 bits e o de instruções pode ser de 12, 14 ou 16 bits, dependendo do microcontrolador. Esse tipo de arquitetura permite que, enquanto uma instrução está sendo executada, outra instrução seja "buscada" da memória, tornando o processamento mais rápido. Além disso, como o barramento de instruções é maior que 8 bits, o OP CODE da instrução já inclui o dado e o local onde ela vai operar (quando necessário), o que significa que apenas uma posição de memória é utilizada por instrução, economizando assim memória de programa.

Desta forma, observa-se que dentro da palavra do OP CODE, que pode ser de 12, 14 ou 16 bits, existe uma limitação para o código da instrução propriamente dito. Por isso, os PICs utilizam uma tecnologia chamada RISC, (Reduced Instruction Set Computer - Computador com Conjunto de Instruções Reduzido). Desta forma, o PIC16F877 possui cerca de 35 instruções, muito menos que os microcontroladores convencionais (CISC), que chegam a possuir mais de 100 instruções.

A configuração mínima para operação do PIC16F877 é um cristal conectado nos pinos 13 e 14, e um resistor ligado no pino /MCLR e Vcc para eliminar a condição de reset (figura 3.14).

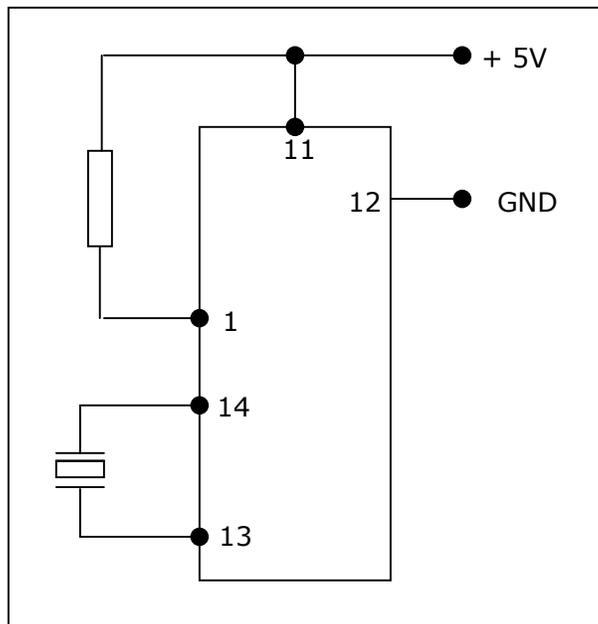


Figura 3.14 - Configuração mínima do PIC16F877

O microcontrolador PIC16F877 é um componente da placa de controle e atua com o componente FT245 gerenciando a comunicação USB, gera os sinais para controle dos registradores de deslocamento 4094, do potenciômetro SPI, comunicação serial RS232 e lógica auxiliar do circuito Placa de Experimento Remoto.

A opção pela utilização do microcontrolador PIC16F877 ocorreu em razão da ampla documentação disponível, facilidade de aquisição no mercado nacional, preço acessível e ferramentas de desenvolvimento disponibilizadas gratuitamente.

FT245

O componente FT245 é um circuito integrado desenvolvido pela empresa Ftdichip (FTDCHIP, 2003) que encapsula todo o protocolo USB em hardware, dispensando a necessidade de programação de firmware.

Para transferência de dados entre o Host USB e um periférico, como um microcontrolador, dispõe de um barramento de 8 bits com recurso de FIFO e pinos de controle para escrita, leitura e status.

A seguir são descritas as principais características do componente FT245:

- Taxa de transferência de até 8 Mbits .
- Compatível com as versões 1.1 e 2.0 do protocolo USB.
- Buffer de escrita com 384 bytes.
- Buffer de leitura de 128 bytes.
- Suporte para dispositivos de alto consumo alimentados pela porta USB
- Regulador de voltagem integrado de 3.3 volts
- Multiplicador de clock baseado em PLL
- Programação por meio de DLL e porta de comunicação virtual, dispensando em muitos casos o desenvolvimento de driver de hardware para USB

Na figura 3.15 pode ser vista a estrutura interna do FT245. No circuito desenvolvido foram utilizados os pinos de dados D0 a D7, os controles RXF#, RD#, TXE#, WR, e o transceptor USB USBDP e USBDM.

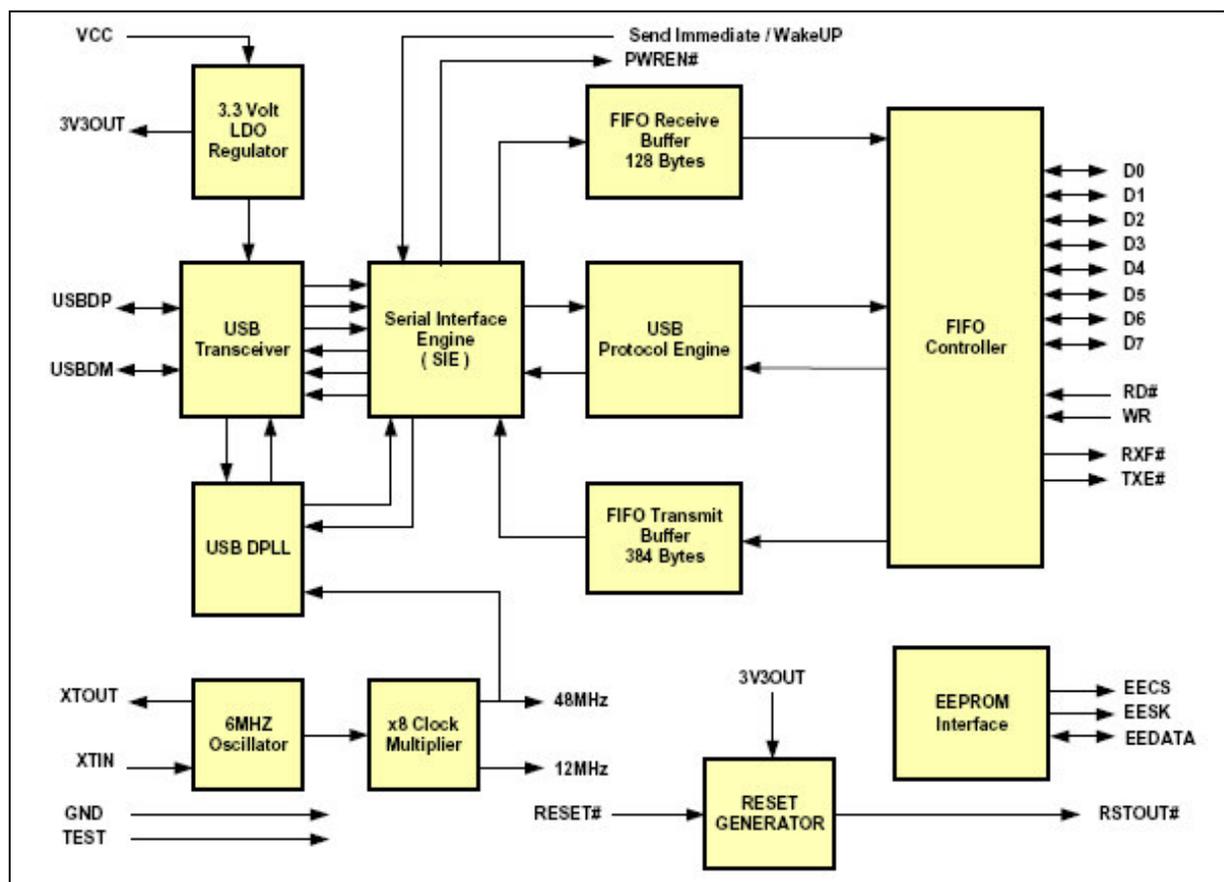


Figura 3.15 - Componente FT245 em blocos

Para enviar dados do periférico para o computador host os dados devem ser colocado nos pinos D0 a D7 quando o pino TXE# estiver em nível lógico baixo. O periférico deve colocar o pino WR em nível lógico alto para indicar que a operação será de escrita. A figura 3.16 ilustra o diagrama de sinais e tempo para operação de escrita no FT245.

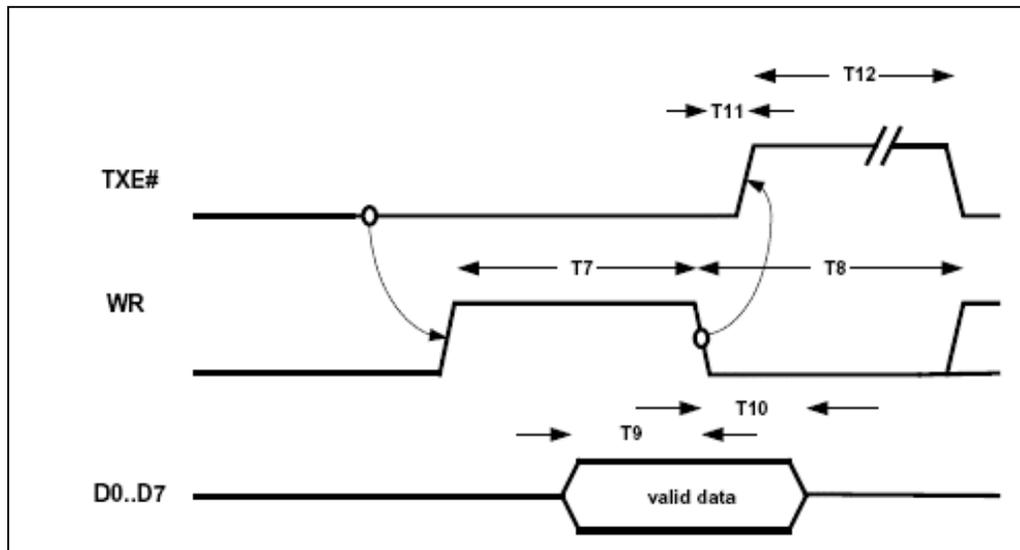


Figura 3.16 - Diagrama de tempo e sinais para operação de escrita do FT245

Se o buffer de transmissão estiver cheio ou ocupado armazenando os dados de alguma operação de escrita prévia o pino TXE# é mantido em nível lógico baixo.

Ao receber um dado enviado pelo computador host o FT245 tem o nível lógico do pino de controle RXF# levado ao valor baixo indicando ao periférico que existe ao menos um byte disponível para leitura nos pinos de dados D0 a D7. O periférico ao colocar o pino RD# no nível lógico baixo indica que irá realizar uma operação de leitura no barramento de dados. A figura 3.17 apresenta o diagrama de tempo e de sinais para uma operação de leitura do FT245.

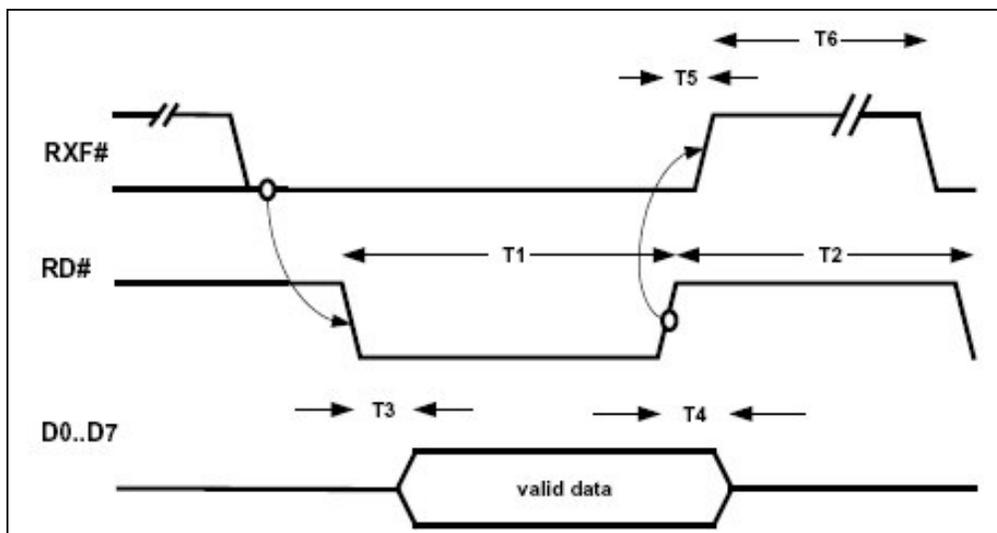


Figura 3.17 - Diagrama de tempo e sinais para operação de leitura do FT245

A programação do ponto de vista do microcontrolador compreende as ações de monitorar os pinos de controle para escrita e leitura (TXE# e RXF#), atuar sobre os pinos que habilitam a escrita e leitura (WR e RD#) e colocar ou ler os dados nos pinos D0 a D7.

A empresa fabricante do FT245 oferece dois recursos para desenvolvimento de aplicativos que realizam leitura e escrita de dados na porta USB a partir do computador host. Um desses recursos é o driver de dispositivo chamado VCP (Virtual Comm Port). Com o mesmo instalado, um dispositivo FT245 conectado será visto pelo sistema operacional como uma porta de comunicação serial padrão. Desse modo a programação no host poderá utilizar os mesmos recursos para programar a porta serial RS232. Este modo de programação facilita o desenvolvimento, pois dispensa qualquer conhecimento sobre o hardware USB, porém tem a desvantagem de limitar a taxa de transferência ao utilizar os padrões da comunicação serial.

Outra forma de programação é que foi adotada neste projeto é por meio do driver denominado FT2XX que permite o acesso direto ao dispositivo USB. A arquitetura do driver FT2XX consiste de um driver para o sistema operacional Windows que adota o modelo WDM (Windows Driver Model) que se comunica com o dispositivo via pilha de USB e realiza a interface com as aplicações escritas com ferramentas como C++, Visual Basic, Delphi, entre outras por meio de uma DLL.

Esta DLL inclui funções que permitem listar os dispositivos instalados, abrir e fechar a conexão, escrever e ler, e configurar diversos parâmetros da comunicação e do dispositivo.

No desenvolvimento dos aplicativos do computador host para a comunicação com o hardware USB foi utilizada o driver FT2XX, pois o mesmo permite obter a máxima taxa de transferência.

MAX232

Uma das funcionalidades desenvolvidas na Placa de Controle foi a comunicação serial RS232. Este recurso foi utilizado durante o processo de desenvolvimento para depuração do hardware, do aplicativo do computador host e do programa do microcontrolador. Conectando a placa de controle a um computador executando um programa de comunicação serial como o Hyperterminal foi possível visualizar o comportamento do sistema a partir de mensagens que eram enviadas a porta serial. O microcontrolador PIC16F877 possui uma USART acessível a partir dos pinos 25 e 26. Esses pinos são compatíveis com a especificação TTL, portanto para se compatibilizar os sinais com o padrão EIA-232 foi utilizado um circuito com o componente MAX232 (Maxim, 2005).

O MAX232 é um componente que contém um conjunto de dois receptores/transmissores para conversão de níveis entre TTL/CMOS e EIA-232. Contendo um gerador de tensão capacitivo gera as tensões com alimentação única de 5V. A figura 3.18 apresenta uma configuração típica para o MAX232 e que foi empregada no circuito da placa de controle.

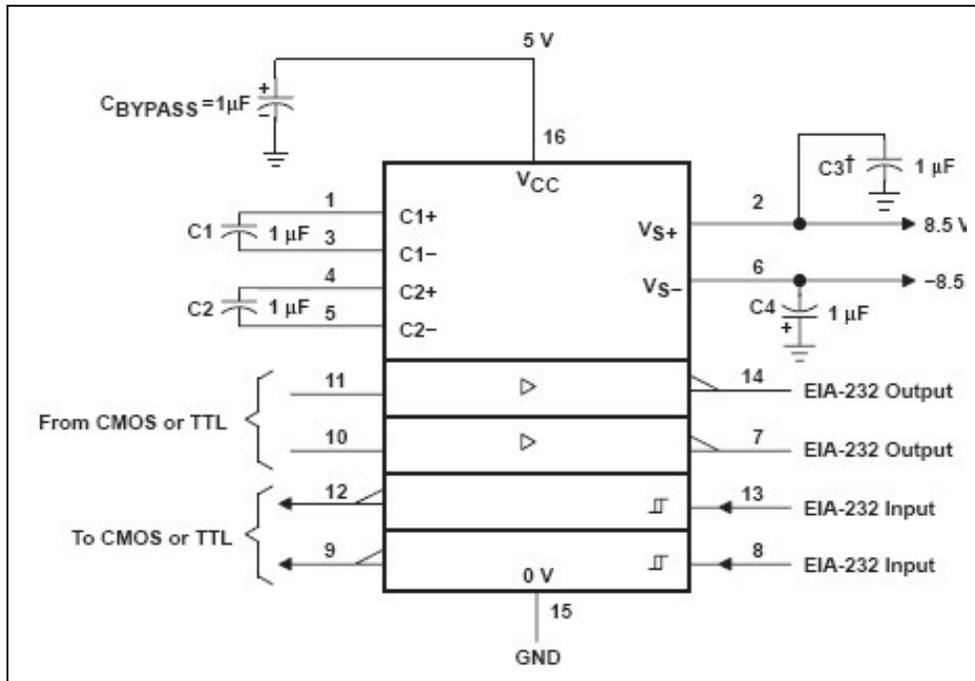


Figura 3.18 - Configuração típica do componente MAX232

Registro de deslocamento CD4094

O circuito integrado CD4094 consiste de um registro de deslocamento de 8 bits e um latch 3 State de 8 bits. Os dados inseridos no pino Data são deslocados serialmente no registrador de deslocamento na transição positiva do sinal de clock. A saída do ultimo estágio (pino Qs) pode ser usada para cascatear vários dispositivos. Os dados do pino Qs são transferidos para a segunda saída (pino Q's) do dispositivo seguinte na borda negativa seguinte do clock.

A figura 3.19 mostra a pinagem do componente CD4094 e a figura 3.20 a tabela verdade do mesmo.

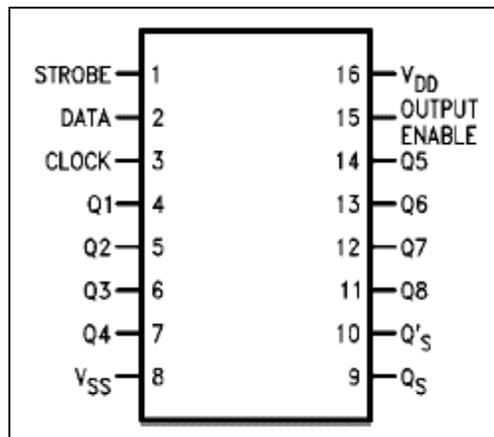


Figura 3.19 - Componente CD 4094

Clock	Output Enable	Strobe	Data	Parallel Outputs		Serial Outputs	
				Q1	Q _N	Q _S (Note 1)	Q' _Σ
↘	0	X	X	Hi-Z	Hi-Z	Q7	No Change
↘	0	X	X	Hi-Z	Hi-Z	No Change	Q7
↘	1	0	X	No Change	No Change	Q7	No Change
↘	1	1	0	0	Q _{N-1}	Q7	No Change
↘	1	1	1	1	Q _{N-1}	Q7	No Change
↘	1	1	1	No Change	No Change	No Change	Q7

Figura 3.20 - Tabela verdade do componente CD4094

O componente CD4094 é utilizado na placa de controle e placa de experimentos. Os sinais de clock, dados e strobe são gerados pelo microcontrolador PIC16F877 e encaminhados para os dispositivos CD4094 da placa de experimento para controle dos blocos eletrônicos chave, ponto de medição, potenciômetro digital e banco de componentes. A placa de controle utiliza um CD4094 para a lógica de seleção da placa de experimento.

Potenciômetro Digital AD7376

O componente AD7376 fabricado pela empresa Analog Devices (Analog,2005) é um potenciômetro digital que admite tensão para alimentação máxima de 30V em fonte simples e +-15V com fonte simétrica. Pode variar seu valor em 128 passos programáveis por meio de comunicação que adota o protocolo SPI. A figura 3.21 mostra a pinagem e os blocos internos do AD7376.

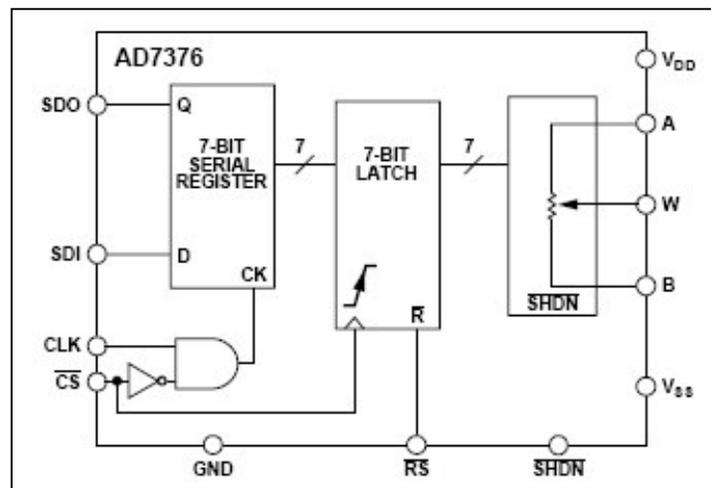


Figura 3.21 - Diagrama interno do AD7376

A variação da resistência ocorre por meio do envio de uma palavra de 7 bits contendo valores de 0 a 127. Os dados são enviados usando o recurso de comunicação SPI

do microcontrolador PIC16F877. O controle do pino de habilitação /CS do AD7376 é realizado pelo microcontrolador e lógica auxiliar. A figura 3.22 mostra o diagrama de tempo do AD7376.

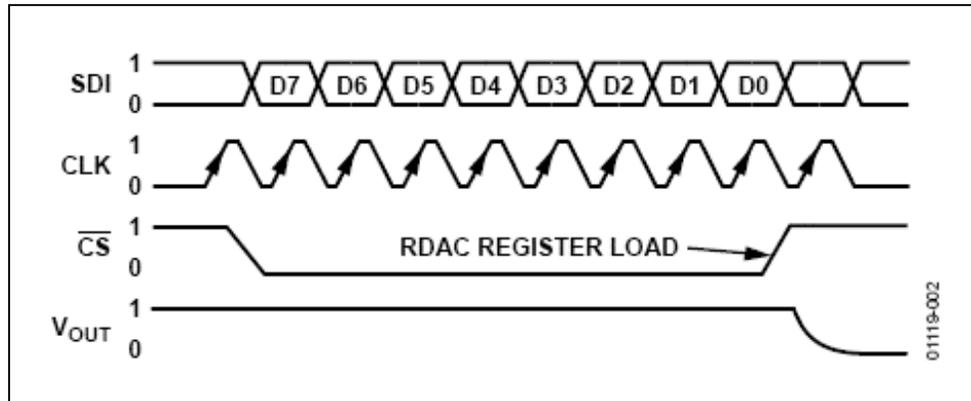


Figura 3.22 - Diagrama de tempo do componente AD7376

O componente AD7376 foi utilizado na implementação do bloco eletrônico potenciômetro disponibilizando o recurso de variar a resistência como um potenciômetro de controle manual normalmente usado em circuitos eletrônicos.

AD7501

O componente AD7501 produzido pela empresa Analog Devices (Analog, 2003) com tecnologia CMOS é um multiplexador analógico de oito canais que conecta uma de suas oito entradas a uma saída comum, dependendo do valor binário existente nos três pinos de endereçamento disponíveis e do estado do pino enable. A figura 3.23 mostra o diagrama em blocos do AD7501 e sua pinagem, a figura 3.24 mostra sua tabela verdade.

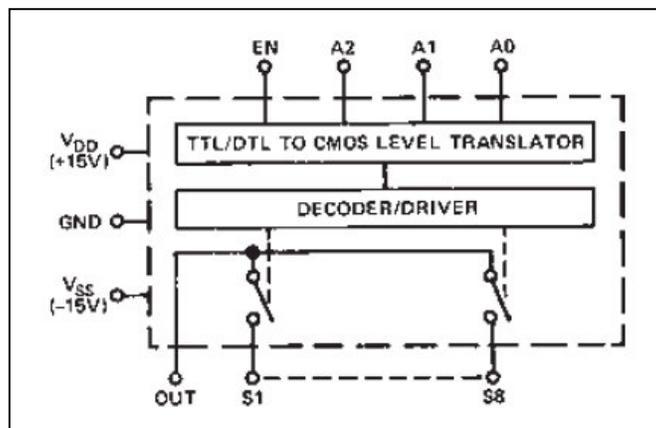


Figura 3.23 - Diagrama em blocos do componente AD7501

A ₂	A ₁	A ₀	EN	“ON”
0	0	0	1	1
0	0	1	1	2
0	1	0	1	3
0	1	1	1	4
1	0	0	1	5
1	0	1	1	6
1	1	0	1	7
1	1	1	1	8
X	X	X	0	None

Figura 3.24 - Tabela verdade do AD7501

Na posição ligada cada chave do AD75021 tem valor resistivo típico de 170Ohms e a potência de dissipação não deve exceder o valor de 30 μ W.

O AD7501 é utilizado no módulo Placa de Experimento para realizar as conexão das pontas de prova do osciloscópio ou outro instrumento aos pontos de medição.

Arquiterura do hardware

O sistema de hardware é constituído pela Placa de controle que se comunica com o computador host pela porta USB e por meio de um barramento com as placas de experimentos que por sua vez tem conectado os blocos eletrônicos e a área de experimento do usuário.

A figura 3.25 mostra o diagrama em blocos do hardware que compõe a sistema de hardware do laboratório de Experimento Remoto.

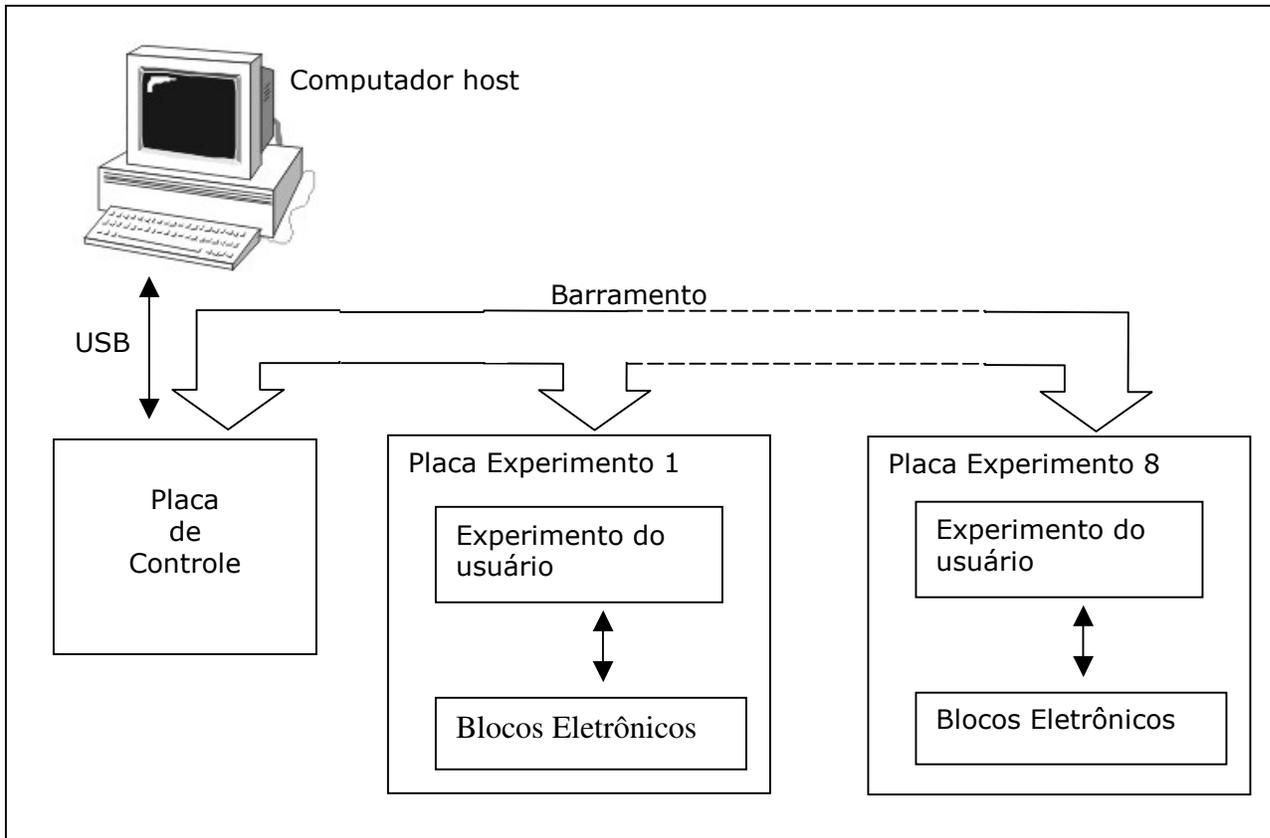


Figura 3.25 - Arquitetura do sistema de hardware

A foto da figura 3.26 exibe a montagem do sistema de hardware contendo os seus vários elementos.



Figura 3.26 - Sistema de hardware

Placa de Controle

A placa de controle se comunica com o computador via porta USB por meio do controlador USB FT245, o gerenciamento dessa comunicação é realizado pelo microcontrolador PIC16F877 que a partir dos comandos recebidos gera os sinais para controle dos registradores de deslocamento CD4094, da comunicação SPI para o controle do potenciômetro digital AD7376 e a comunicação serial RS232.

A figura 3.27 apresenta o diagrama de blocos da Placa de Controle com os seus principais componentes.

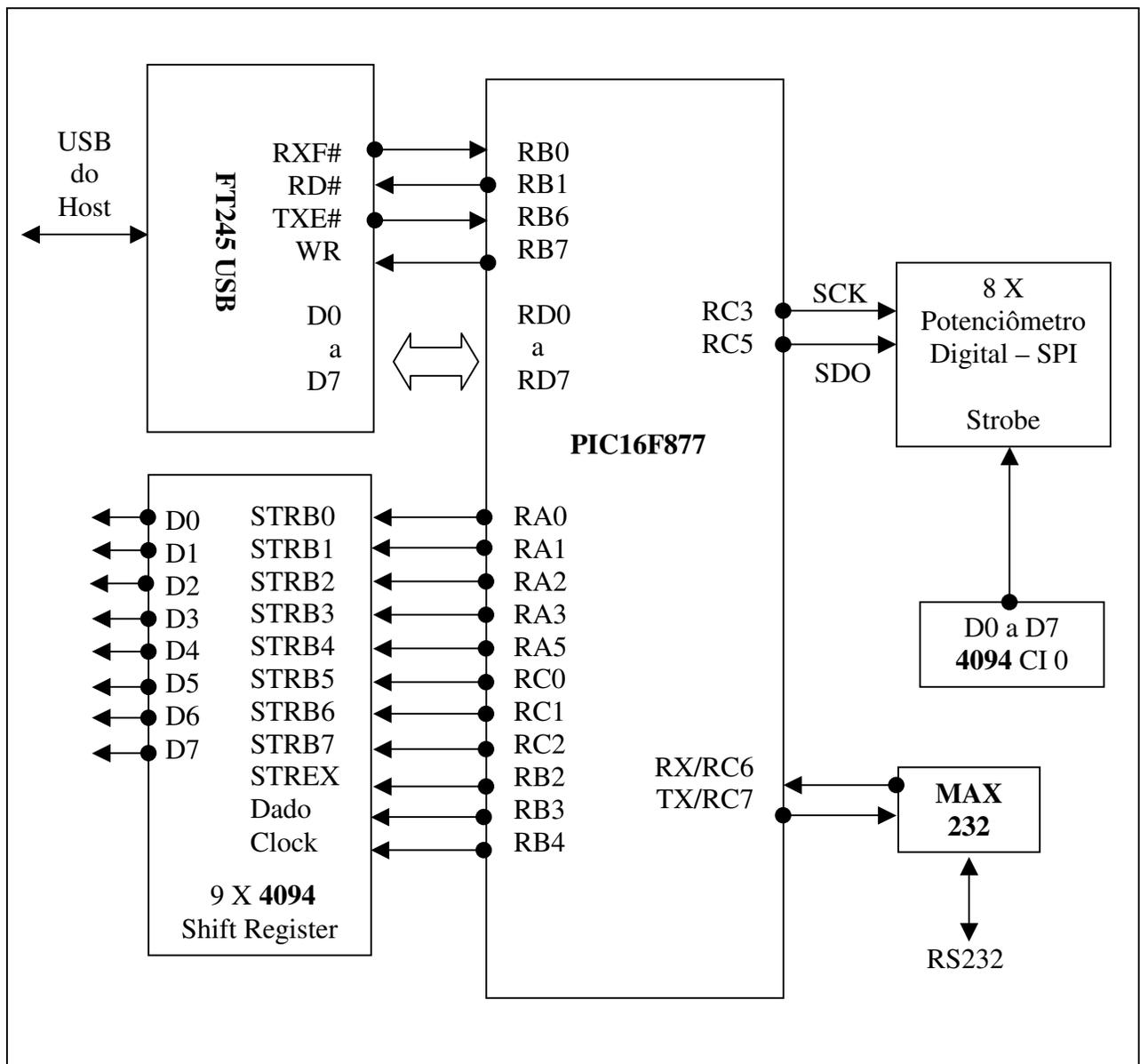


Figura 3.27 - Diagrama em blocos da Placa de controle

Para realizar a comunicação com o controlador USB FT245 o microcontrolador tem os pinos RB0, RB1, RB e RB7 conectado aos pinos de controle do FT245 e os pinos RD0 a RD7 aos pinos de dados do mesmo. Para a comunicação com os potenciômetros SPI são utilizados os pinos RC3 e RC5 do PIC16F877. A interface para a porta RS232 é implementada a partir dos pinos RC6 e RC7 do microcontrolador. Os pinos RA0 a RA5, RC0 a RC2 e RB2 a RB4 são conectados aos registradores de deslocamento CD4094 gerando os sinais de strobe, clock e dados. Cada Placa de Experimentos contém oito componentes CD4094, a habilitação de determinada Placa de Experimento se dá quando são habilitados os componentes CD4094 da respectiva placa. A Placa de Controle contém um CD4094 que recebe o dado de controle que coloca em nível alto o pino *output enable* do conjunto de CD4094 da placa de experimento habilitando que os comandos sejam aceitos somente por essa placa. A figura 3.28 ilustra o tráfego de sinais entre a Placa de Controle e as Placas de Experimento para atuar sobre os registradores de deslocamento CD4094.

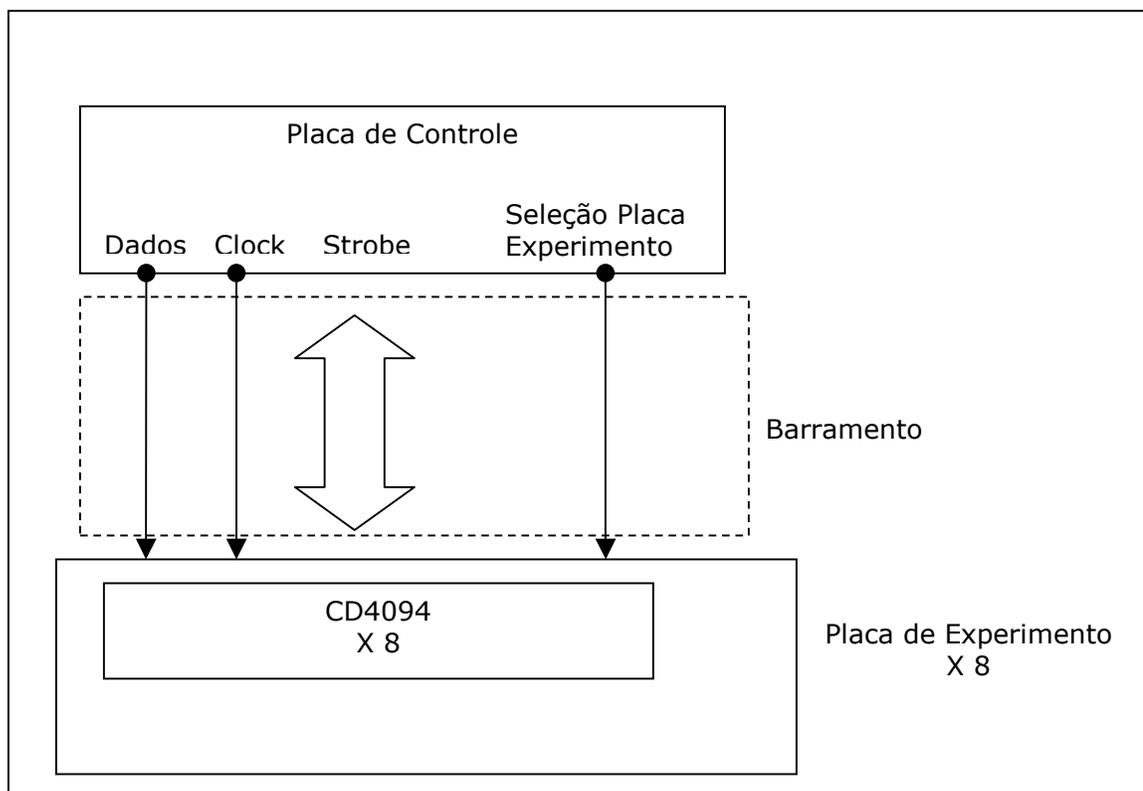


Figura 3.28 - Dados do CD4094 entre a Placa de controle e Experimentos

Caso de uso da Placa de Controle

O programa do microcontrolador da Placa de Controle foi desenvolvido a partir do levantamento do caso de uso em que foram planejadas as ações necessárias para a operação da Placa de Controle, integração com as Placas de Experimento e comunicação com os diversos periféricos e Blocos Eletrônicos. As ações para o caso de uso são descritas a seguir:

- Iniciar o hardware
- Ler controlador USB
- Escrever no controlador USB
- Decodificar comando recebido no controlador USB
- Enviar dados para registrador de deslocamento
- Gerar clock para registrador de deslocamento
- Gerar sinais de strobe para registrador de deslocamento
- Gerar sinais de habilitação para o potenciômetro digital
- Escrever no potenciômetro digital
- Escrever na Usart
- Ler Usart

A partir do caso de uso foram implementadas as funções que atendem aos requisitos do sistema e que são ilustradas no diagrama de arquitetura da figura 3.29.

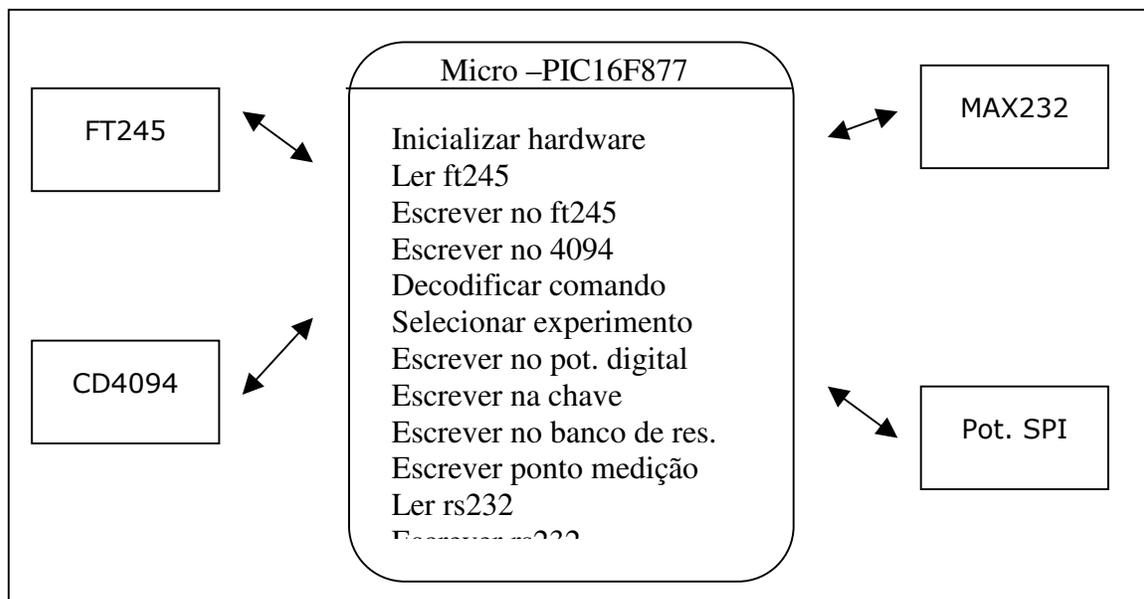


Figura 3.29 - Diagrama de arquitetura do programa do microcontrolador PIC16F877

A seguir será demonstrada a implementação das funções para o microcontrolador PIC16F877 desenvolvidas utilizando a ferramenta de software MPLAB e o compilador para linguagem C PICC.

Inicializar o hardware

O processo de inicialização do hardware da Placa de Controle compreende as ações realizadas para configurar os módulos Usart e SPI do PIC16F877, colocar o controlador USB FT245 em estado de pronto, e procedimento para colocar as saídas de todos os componentes CD4094 em nível lógico baixo garantindo a operação segura do sistema.

A Usart do PIC16F877 é configurada atribuindo valores aos registradores TXSTA e RCSTA. Nesses registros se define a velocidade e atributos da comunicação, o modo síncrono ou assíncrono, e a origem do clock.

A comunicação SPI é configurada por meio do registrador SSPCON no qual são configurados os parâmetros para habilitar os pinos da porta SPI, se a borda ativa do clock é de subida ou descida, e o modo do clock do módulo Master.

Os pinos do PIC16F877 utilizados como entrada e saída são tem seus parâmetros configurados nos registradores TRISA, TRISB, TRISC, TRISD e TRISE.

Os componentes CD4094 da Placa de Controle e das Placas de Experimento recebem dados que colocam suas saídas no nível lógico baixo evitando qualquer situação de funcionamento aleatório dos circuitos sob experimento.

Após a inicialização a placa de controle está pronta para receber e executar os comandos enviados pelo computador host via porta USB.

Ler ft245

O controlador de USB FT245 tem embutido todo o protocolo de comunicação USB. Os dados enviados pelo computador host são disponibilizados por meio de sua FIFO interna em um barramento de 8 bits. Os pinos de controle RXF# e RD# são responsáveis pela operação de leitura dos dados. O pino RXF# quando no nível lógico baixo indica que existem dados disponíveis na FIFO e que os mesmos podem ser lidos no barramento de dados. Para realizar a leitura o microcontrolador deve levar o pino RD# ao nível lógico baixo e em seguida ao nível alto. O algoritmo implementado no microcontrolador para

leitura do barramento de dados do FT245 é descrito por meio de pseudocódigo² na tabela 3.2.

```
Leia pino RXF#  
Se pino RXF# em nível baixo  
Colocar pino RD# em nível baixo  
Colocar pino RD# em nível alto  
Ler barramento de dados  
Fim Se
```

Tabela 3.2 -Procedimento Lerft245

Na implementação do PIC16F877 o pino RB0 é configurado como entrada e está conectado ao pino RXF# do FT245, o pino RB1 é configurado como saída e conectado ao pino RD#, os pinos RD0 a RD7 estão configurados como entrada e conectados aos pinos D0 a D7. Um laço infinito realiza a leitura do pino RB0, quando esse valor for para o nível baixo o pino RB1 é levado ao nível alto e em seguida ao nível baixo, a partir dessa ação os dados estão disponíveis e é realizada a leitura dos pinos RD0 a RD7, esses 8 bits contém o caractere enviado pelo computador e são retornados pela função ler_ft245. A linha de código da tabela 3.3 mostra a chamada a essa função e o retorno para uma variável.

```
comando = ler_ft245( );
```

Tabela 3.3 - Exemplo de utilização de ler_ft245

Sempre que houver o recebimento de caracteres pela porta USB esses serão lidos pela função ler_ft245 e após decodificação se forem válidos executarão as ações correspondentes.

Decodificar comando

Para que a Placa de Controle possa atuar sobre os diversos componentes do sistema de hardware atribuindo valores e ações aos mesmos é preciso que o computador host envie um conjunto de informações contendo os comandos com seus respectivos parâmetros. Para

² Uma forma híbrida de representar um programa que utiliza uma linguagem natural , no caso o português e a sintaxe de uma linguagem de programação estruturada (Pressman, 1999)

efetivar esse processo foi desenvolvido um protocolo constituído de dois caracteres de cabeçalho que indicam o início de um novo comando, um caractere que informa o comando que será executado e dependendo do comando um ou dois caracteres com parâmetros. A figura 3.30 ilustra a estrutura do protocolo de comunicação.

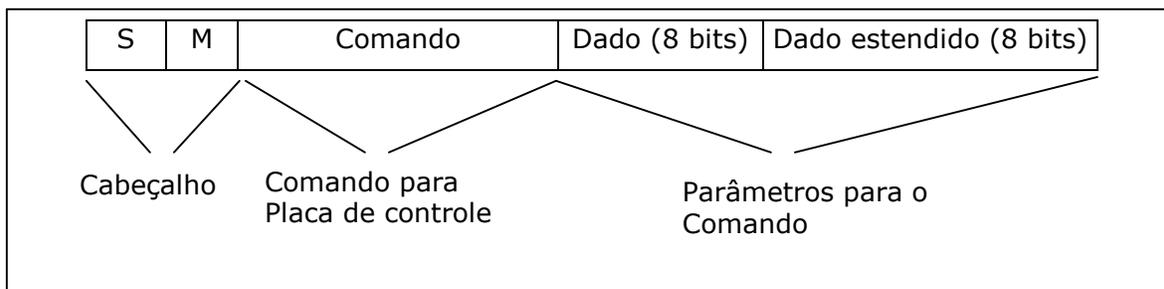


Figura 3.30 - Estrutura do protocolo de comunicação

O pseudocódigo da tabela 3.4 demonstra o processo implementado para decodificar os caracteres recebidos, detectar o comando e os parâmetros atribuídos, e encaminhar para execução.

```

Ler caracteres do cabeçalho
Se cabeçalho correto
  Ler comando
  Para cada comando
    ler o parâmetro
  Fim de Para cada comando
  encaminhar comando para execução
Fim de Ler comando
Fim de Se cabeçalho correto
  
```

Tabela 3.4 - Decodificação do protocolo de comunicação

Essa função está aninhada dentro de um laço infinito ativado após todas as inicializações do hardware. Desse modo após a inicialização todos os dados recebidos pelo controlador USB são tratados, se forem válidos executam o comando senão são ignorados.

Escrever no ft245

Para realizar a operação de escrita no controlador FT245 que significa enviar um caractere para o computador host são utilizados os pinos de controle TXE# e WR, nos pinos D0 a D7 são colocados os oito bits que formam o caractere a ser transmitido. Quando o

pino TXE# está em nível lógico baixo significa que a FIFO está habilitada para receber os dados que serão transmitidos. Os bits no barramento de dados são transmitidos quando o pino WR vai do nível lógico alto para o nível baixo. O pseudocódigo da tabela 3.5 descreve o processo de transmitir um caractere do microcontrolador para o computador host.

```
Ler pino TXE#
  Se pino TXE# em nível baixo
    Colocar dados no pinos D0 a D7
    Colocar pino WR em nível alto
    Colocar pino WR em nível baixo
  Fim Se
```

Tabela 3.5 - Pseudocódigo da função escreve_ft245

Esta função possibilita que dados gerados na Placa de Controle possam ser enviadas para o computador host que por meio de programação conveniente pode tratar essas informações.

Escrever no 4094

Todos o Blocos Eletrônicos e o módulo de seleção da Placa de Experimento fazem uso de sinais gerados por um conjunto de registradores de deslocamento CD4094, quer seja para ativar linhas de controle, strobe ou habilitação de outros componentes.

Os dados para o CD4094 são transmitidos serialmente e sincronizados por um sinal de clock. O pino RB3 do PIC16F877 transmite os dados na borda de subida do clock gerado pelo pino RB4. Após a transmissão dos 8 bits de dados um pulso de strobe habilita o latch interno que disponibiliza os dados na saída. A figura 3.31 ilustra os sinais e a temporização em uma operação de escrita no CD4094.

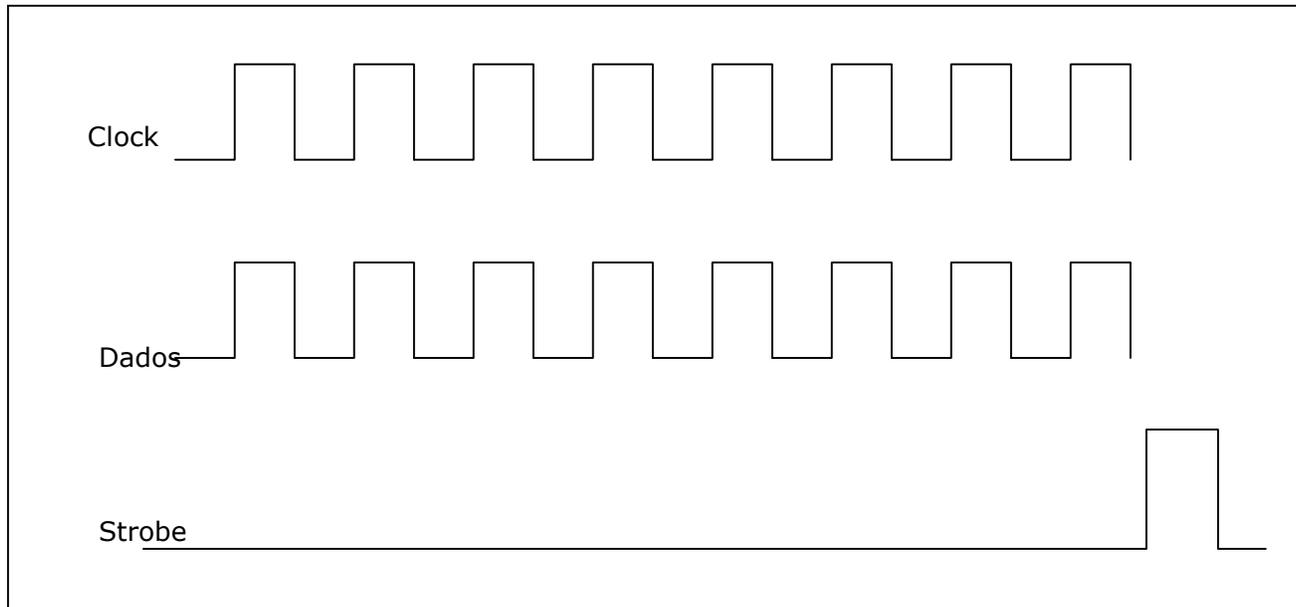


Figura 3.31 - Operação de escrita no CD4094

A função para escrita em determinado CD4094 recebe como parâmetros um valor que indica qual será o CD4094 que receberá os dados e 8 bits contendo os dados que serão escritos no registrador. A tabela 3.6 exibe a declaração da função de escrita no CD4094.

```
escreve_4094(8,0,0,0,0,1,0,1,0);
```

Tabela 3.6 - Declaração da função escreve_4094

O primeiro parâmetro informa em qual CD4094 atuar e pode variar de 0 a 8, os parâmetros seguintes declarados como variáveis do tipo bit são os dados para a saída do registrador. O pseudocódigo da tabela 3.7 demonstra a implementação dessa função.

```
Colocar todos pinos de strobe em nível baixo
Repetir 8 vezes
    Colocar o pino de clock em nível alto
    Escrever dado (0 ou 1) no pino de dado
    Colocar o pino de clock em nível baixo
Fim Repetir
Colocar em nível alto o pino de strobe do CD4094 selecionado
Colocar em nível baixo o pino de strobe do CD4094 selecionado
```

Tabela 3.7 - Pseudocódigo da função escreve_4094

Com essa função e a configuração de hardware adotada é possível atuar sobre 64 componentes CD4094 distribuídos em oito Placas de Experimento. A placa de controle contém um CD4094 responsável por gerar os sinais de habilitação das Placas de Experimento.

Selecionar experimento

O sistema de hardware contém um barramento no qual trafegam os sinais de clock e dados gerados pelo microcontrolador e que são aplicados a todos os componentes CD4094 das Placas de Experimento que estejam conectadas ao barramento. Em um dado momento apenas uma Placa de Experimento pode estar ativa e isso ocorre por um sinal aplicado ao pino enable dos componentes CD4094 dessa placa. Um registrador de deslocamento na Placa de Controle atua na geração desse sinal. O diagrama da figura 3.32 ilustra a lógica para seleção da Placa de Experimentos.

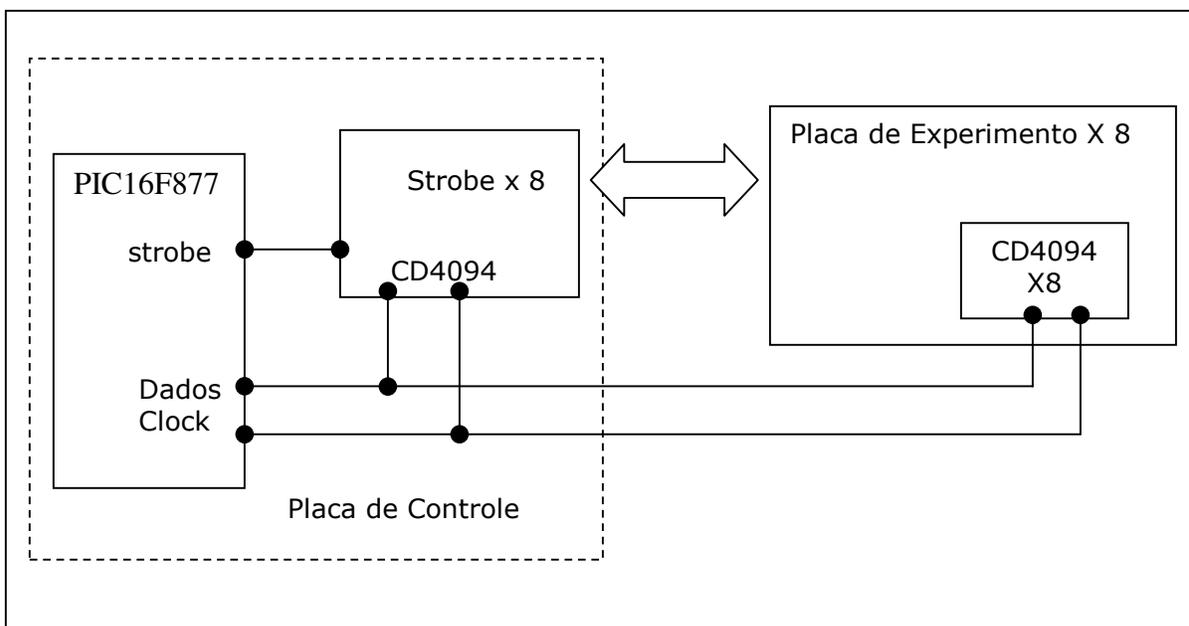


Figura 3.32 - Diagrama para seleção da Placa de Controle

A declaração da função para escrita no CD4094 com o parâmetro 8 (tabela 3.8) exemplifica a seleção da Placa de Experimento 4.

```
escreve_4094(8,0,0,0,0,1,0,0,0);
```

Tabela 3.8 – Função escreve_4094 usada para selecionar experimento

A configuração desenvolvida possibilita o controle de até oito Placas de Experimento.

Escrever no potenciômetro digital

A partir do componente AD7376 foi desenvolvido o Bloco Eletrônico potenciômetro digital. Os comandos para controle desse componente devem ser enviados por meio do protocolo de comunicação SPI. O microcontrolador PIC16F877 contido na Placa de Controle dispõe de recursos para comunicação por esse protocolo e é utilizado para enviar os comandos que irão variar o valor resistivo do AD7376.

Do ponto de vista de hardware a comunicação entre dispositivos SPI necessita dos sinais saída de dados (SD0), entrada de dados (SDI) e clock serial (SCK). Pelo protocolo SPI a comunicação ocorre entre dispositivos que são identificados como mestres ou escravos. Na operação para escrita no AD7376 o microcontrolador será o mestre e o potenciômetro será o escravo. Para efetuar a comunicação SPI o PIC16F877 dispõe dos registradores SSPCON e SSPSTAT que devem ser configurados para definir os seguintes parâmetros:

- Os modos mestre e escravo.
- A borda do clock que o dado estará disponível.
- A velocidade do clock.
- Polaridade do clock.

A figura 3.33 demonstra o diagrama de tempo dos sinais para comunicação SPI no modo mestre.

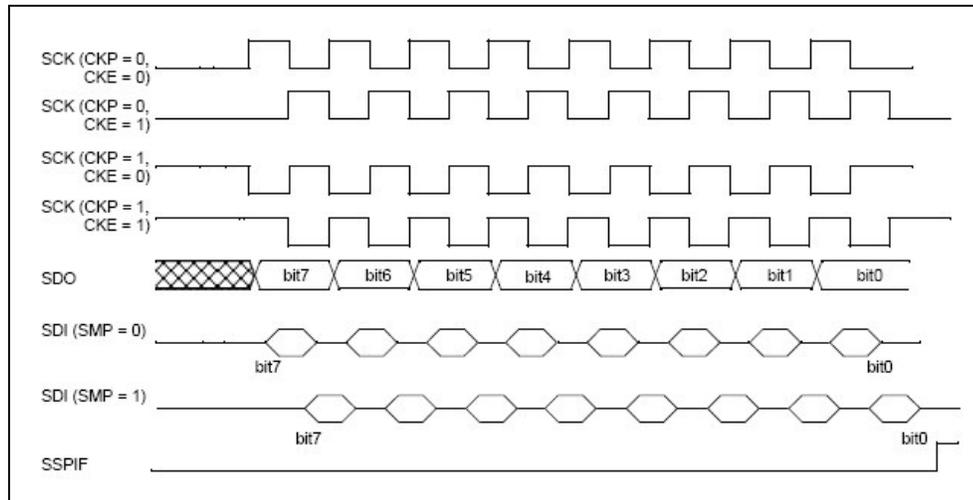


Figura 3.33 - Diagrama de tempo da comunicação SPI

Foi desenvolvida uma função para carregar nos registradores SSPCON e SSPSTAT as configurações necessárias para efetivar a comunicação entre o microcontrolador e o potenciômetro digital. Esta função deve ser invocada antes de qualquer comando de escrita ou leitura.

O microcontrolador dispõe do registrador SSPBUF que recebe dados no modo escravo e é utilizado para envio de dados no modo mestre. Na comunicação com o potenciômetro digital o PIC16F877 atua como mestre e, portanto SSPBUF recebe um valor que será transmitido para o AD7376. Após o envio do dado é possível acompanhar o estado da transmissão por meio da leitura do primeiro bit do registrador SSPSTAT. Se o bit estiver em 0 a transmissão está em andamento se o valor for 1 a transmissão foi finalizada. O pseudocódigo da implementação da função para escrita no protocolo SPI é mostrado na tabela 3.9.

<p>Descarregar o registro SSPBUF Atribuir valor para o registro SSPBUF Repetir até o primeiro bit do registro SSPSTAT conter o valor 1</p>
--

Tabela 3.9 - Pseudocódigo da função escreve_pot_digital

Na implementação de hardware desenvolvida, cada Placa de Experimento pode atuar em até oito Blocos Eletrônicos Potenciômetro Digital. Como cada potenciômetro após

receber o comando pela porta SPI deve ter o pino /CS ativado, o circuito da Placa de Experimento dispõe de um CD4094 (figura 3.34) que recebe da Placa de Controle dados para ativar o latch interno do AD7376 e disponibilizar o valor resistivo atribuído ao mesmo.

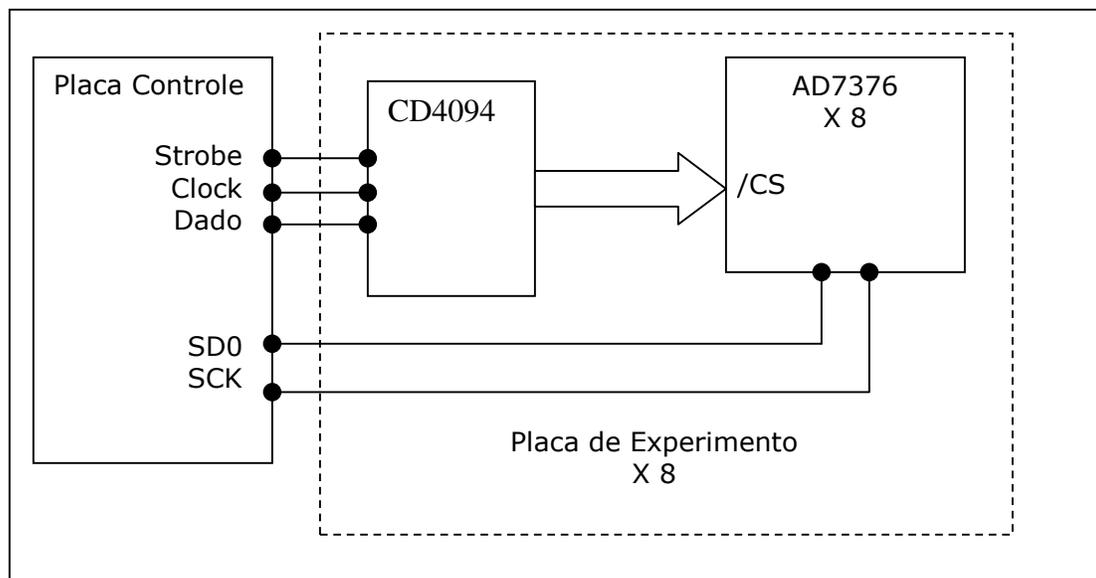


Figura 3.34 - Diagrama do hardware para controle do potenciômetro digital

Para realizar as operações de escrita no AD7376 e enviar dados para o CD4094 habilitador do potenciômetro foi implementada uma função que recebe como parâmetro qual potenciômetro será controlado e o valor que o mesmo deve assumir. Essa função encapsula a função para escrita na porta SPI e no CD4094. O pseudocódigo dessa função é demonstrado na tabela 3.10.

Receber valor para potenciômetro digital Receber valor de qual potenciômetro digital Escrever na porta SPI valor para potenciômetro digital Para o CD4094 que habilita o potenciômetro digital Colocar em nível alto o pino /CS do potenciômetro selecionado Colocar em nível baixo o pino /CS do potenciômetro selecionado Fim do Para

Tabela 3.10 - Pseudocódigo para função de escrita no potenciômetro digital

Escrever na chave

Cada Placa de Experimento pode controlar até 23 Blocos Eletrônicos de chave (relé). Para propiciar este controle são utilizados três componentes CD4094 sendo que uma saída do terceiro CD4094 é utilizada para o controle da lógica dos Pontos de Medição (figura 3.35).

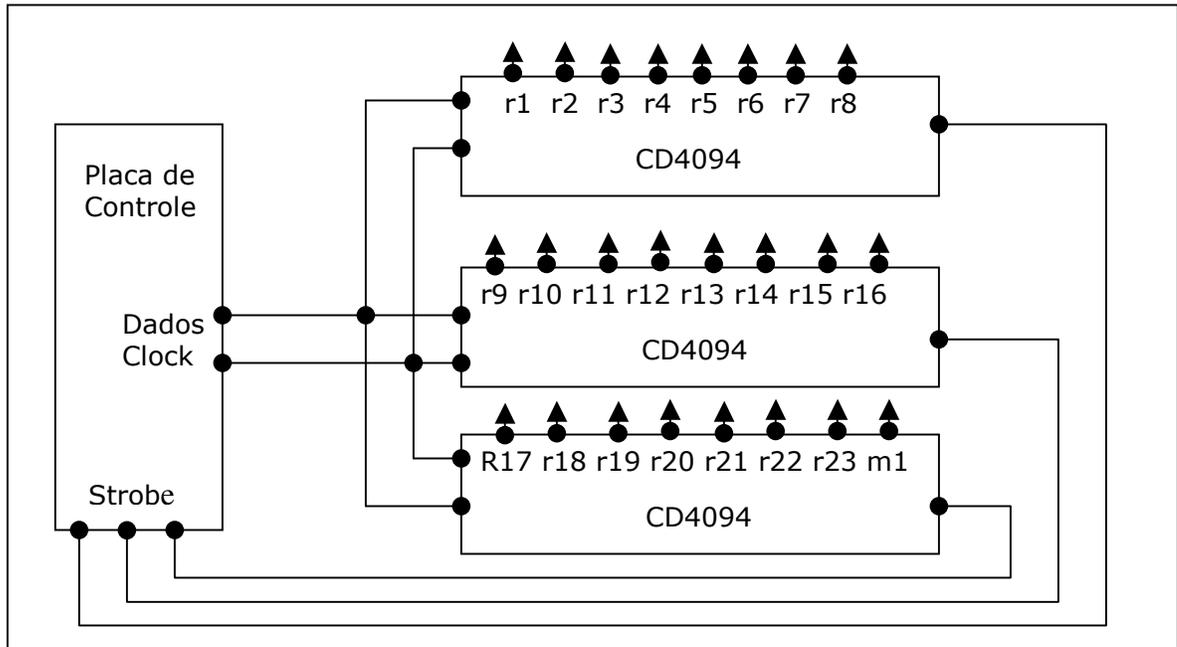


Figura 3.35 - Diagrama de controle para Bloco Chave

Os dados com informações para ligar ou desligar determinada saída são recebidos pela Placa de Controle do computador host e por meio da função *escreve_4094* que encaminha os dados contendo a informação para selecionar umas das 23 saídas com o nível lógico alto ou baixo. Como para cada operação de escrita todo o conteúdo do CD4094 precisa ser atualizado e deseja-se que o estado lógico das suas saídas seja mantido alterando somente a que recebeu o novo comando, antes de enviar um comando o estado anterior é gravado e realiza-se processamento para atualizar apenas a saída definida no comando. Na palavra de oito bits que contém os dados para acionamento da chave os primeiro quatro bits menos significativos são utilizados para indicar qual dos relês será controlado e quinto bit indicada a ação de ligar (nível lógico 1) ou desligar a chave (nível lógico 0) (figura 3.36).

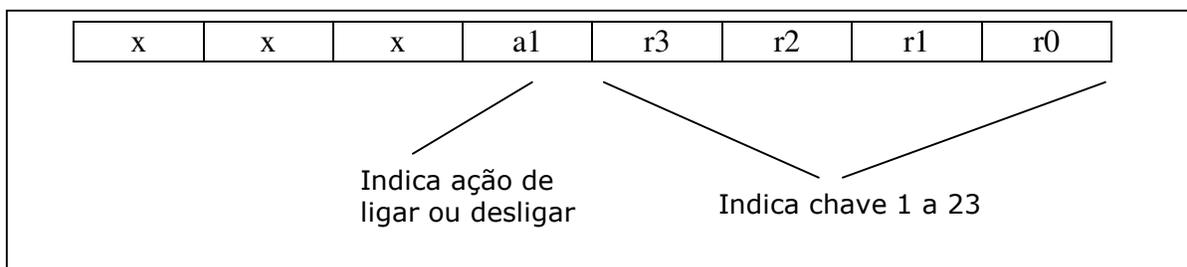


Figura 3.36 - Palavra de dados para Bloco Eletrônico chave

As chaves de 1 a 8 são controladas pelo primeiro CD4094 alocado para o Bloco Eletrônico Chave, o segundo controla as chaves de 9 a 16 e o terceiro de 17 a 23. O pseudocódigo com as ações necessárias para selecionar a saída acionada, o acionamento e a qual dos componentes CD4094 deve ser enviado o dado de controle é descrito a seguir (Tabela 3.11).

<p>Guardar estado atual das saídas dos 3 componentes CD4094 Ler dado para comando de Chave Extrair o número que indica a saída que será controlada Extrair o valor que será encaminhado para a saída selecionada. Selecionar em qual CD4094 está a saída que será controlada Atualizar as saídas do CD4094 selecionado com o novo valor para a saída acionada.</p>

Tabela 3.11 - Pseudocódigo para função escrever na chave

Este módulo foi desenvolvido para controle do Bloco Eletrônico chave, contudo devido às características do componente CD4094 o mesmo pode ser utilizado para acionamento de qualquer dispositivo compatível com a tecnologia TTL/CMOS.

Escrever no banco de resistor

O banco de resistor ou componentes pode ser utilizado para oferecer ao aluno que está realizando o experimento a possibilidade de trocar os componentes que fazem parte de determinado experimento. Cada Placa de Experimento pode conter até quatro Blocos Eletrônicos Banco de Componentes, que por sua vez pode conter até 24 componentes divididos em três blocos internos. O dado com instruções para controle do Banco de

Componentes é constituído de uma palavra de 8 bits permitindo seleccionar o banco utilizado, qual dos três blocos internos e o componente nesse bloco (figura 3.37).

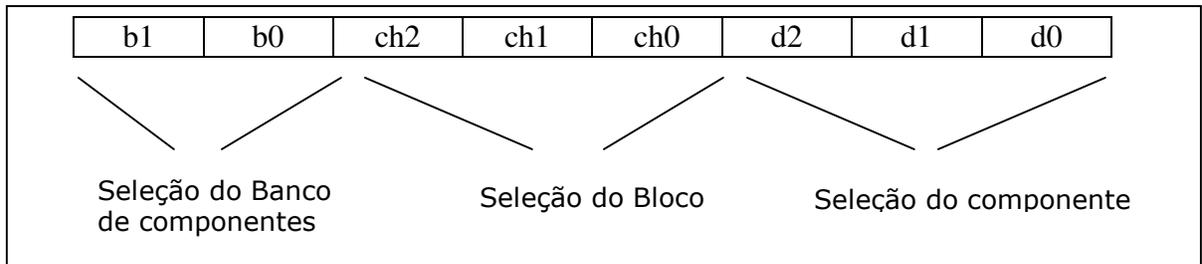


Figura 3.37 - Palavra de dados para Bloco Eletrônico Banco de Componentes

Para o controle de cada Banco de Componentes são necessários 8 bits de dados sendo utilizados quatro componentes CD4094 em cada Placa de Experimento configurados conforme mostrado na figura 3.38. Os seis bits que selecionam o componente e o bloco interno são aplicados às saídas de dados dos CD4094 e os dois bits para seleção do Banco de Componentes ativam o pino de strobe.

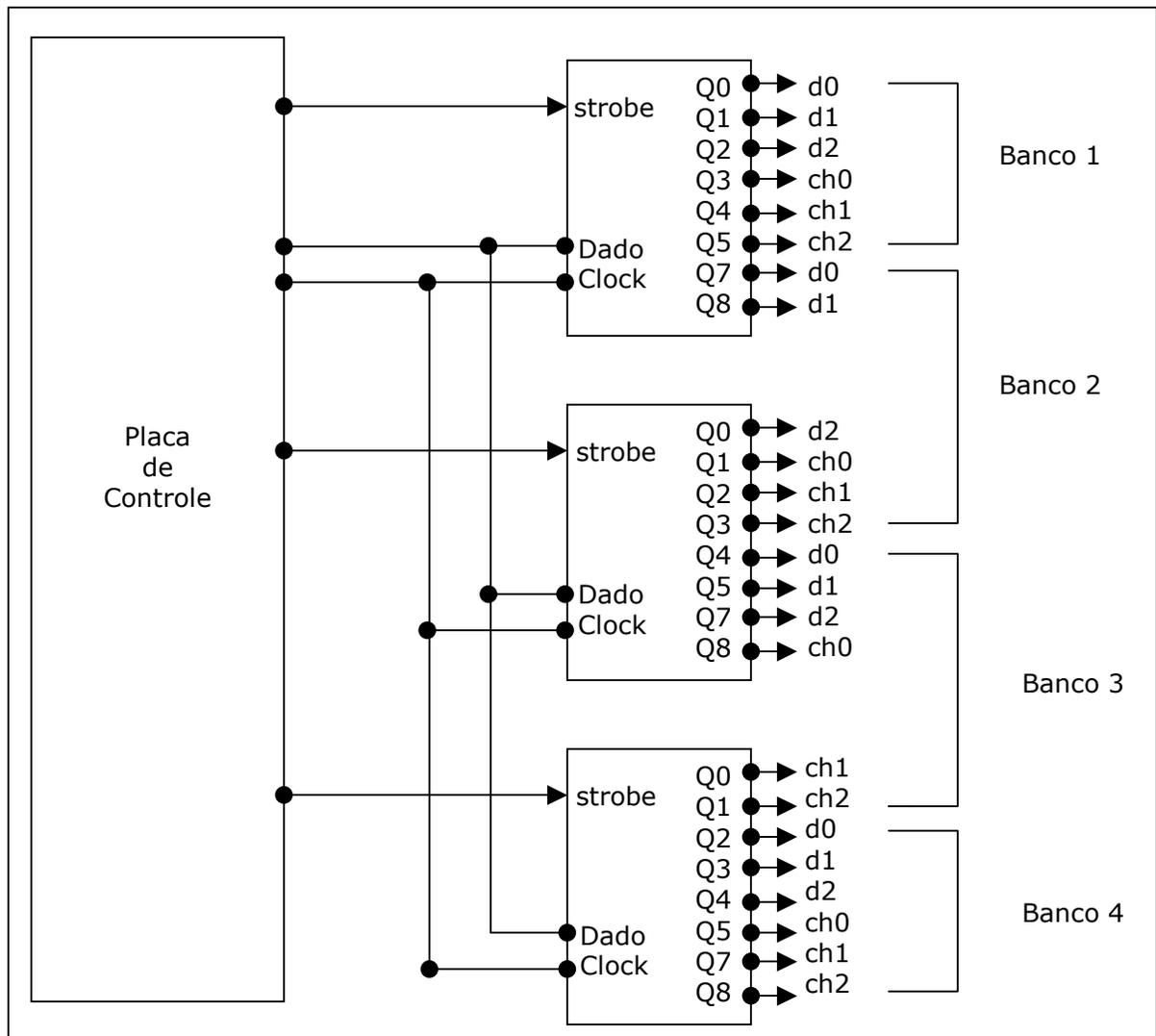


Figura 3.38 - Diagrama do circuito para controle de Banco de Componentes

Para o controle dos quatro Bancos de Componentes foi desenvolvido a função para o microcontrolador *escreve_banco_resistor*. Esta função é responsável por ler a palavra de dados do comando para Banco de Componentes e selecionar o banco que será acionado, a seguir o bloco interno e o componente dentro desse bloco. O pseudocódigo da tabela 3.12 demonstra a implementação para realização desse procedimento.

<p>Ler dado Extrair os dois bits mais significativos do dado Usar os dois bits para selecionar o banco que será acionado. Extrair os bits de seleção de bloco interno Extrair os bits de seleção de componente. Escrever os dados no CD4094. Ativar o strobe do CD4094 para selecionar o banco de componentes.</p>
--

Tabela 3.12 - Pseudocódigo para função escrever_banco_resistor

O Banco de componentes foi planejado inicialmente para permitir que resistores pudessem ser trocados durante a realização de um experimento, contudo esse bloco pode ser utilizado com qualquer componente de dois terminais como, por exemplo, capacitores, diodos, entre outros.

Escrever no ponto de medição

Uma das ações possíveis na realização de um experimento é a utilização de um instrumento como um osciloscópio para medidas e análise de sinais. Para que no experimento realizado remotamente seja possível realizar medidas de maneira próxima as que são realizadas no modo presencial é preciso prover meios para conectar as pontas de prova do instrumento nos pontos do circuito que se deseja examinar os sinais. O Bloco Eletrônico Ponto de Medição foi desenvolvido para permitir que dois canais e a referência de um osciloscópio possam ser conectados em até oito pontos do circuito eletrônico do experimento. O Bloco Eletrônico Ponto de Medição foi implementado com o componente multiplexador analógico AD7501. O componente AD7501 contém oito chaves analógicas multiplexadas cuja seleção ocorre por meio do valor binário presente nos pinos de endereçamento A0, A1 e A2 (figura 3.39)

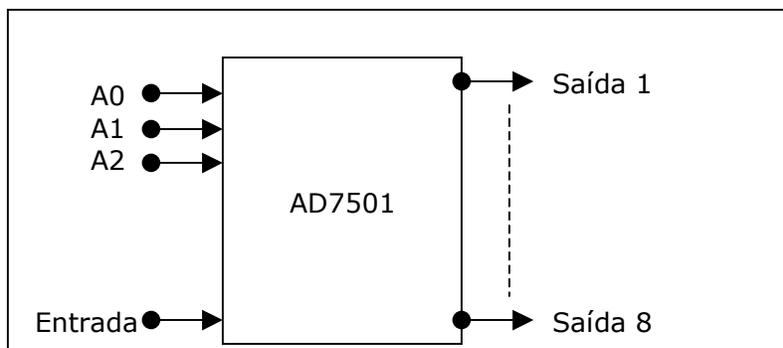


Figura 3.39 - Componente AD7501

Cada Placa de Experimento contém três componentes AD7501 que são usados para chavear o canal 1, 2 e a referência do osciloscópio. Os nove bits necessários para realizar a seleção de chaves dos três AD7501 são fornecidos por um bit de um CD4094 e oito bits de outro CD4094 (Figura 3.40)

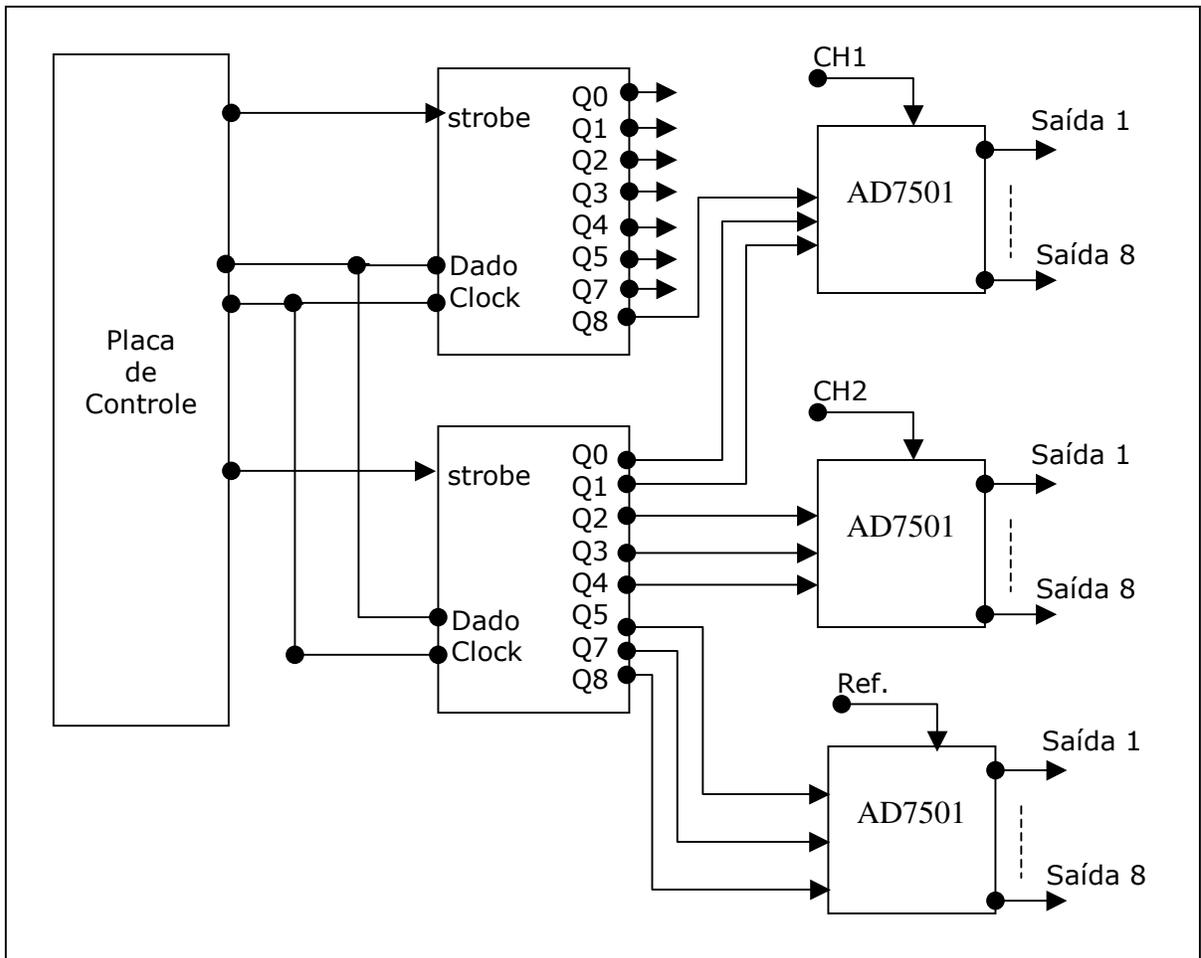


Figura 3.40 - Diagrama em blocos do Ponto de Medição

Como cada instrução para controle do Ponto de Medição é constituído de nove bits são utilizados dois bytes de dados sendo que um byte utiliza apenas um bit como dado válido (figura 3.41).

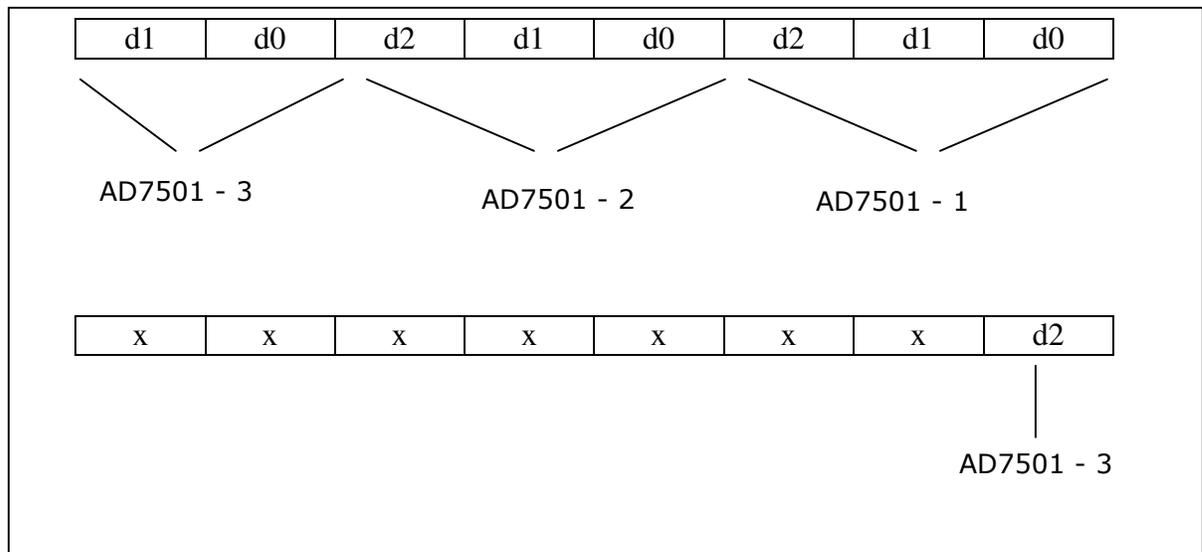


Figura 3.41 - palavra de dados para ponto de medição

Para seleção das chaves que estarão ativas no ponto de medição foi implementado no microcontrolador a função `escreve_pontos_med()`. Esta função após detectar que é um comando para o Bloco Eletrônico Ponto de Mediç o realiza duas leituras no controlador USB para obter os dois bytes com os dados de controle. Os dados novos s o atualizados nos dois componentes CD4094 respons veis pelo endereamento dos tr s componentes AD7501. O pseudoc digo da tabela 3.13 demonstra a implementa o da fun o para chaveamento dos pontos de medi o.

```

Se comando para ponto de medi o
  Ler dado no controlador USB
  Ler dado no controlador USB
  Escrever no 4094 numero 6 para atualizar dados
  Escrever no 4094 numero 7 para atualizar dados
Fim Se

```

Tabela 3.13 - Pseudoc digo para fun o ponto de medi o

O hardware desenvolvido para os pontos de medi o associado a interface gr fica no computador PC possibilitam ao usu rio uma experi ncia pr xima a obtida ao executar as medi es e conex es de um instrumento de medida em um circuito no modo presencial.

Ler Usart e Escrever na Usart

Com o prop sito de oferecer um meio de depura o para a Placa de Controle e propiciar recursos para controlar dispositivos que se comunicam por meio da porta serial

RS232 foi implementado o hardware e funções de software no microcontrolador para leitura e escrita nesta porta de comunicação. O PIC16F877 tem implementado em hardware o protocolo para comunicação serial. Os pinos 25 e 26 devidamente configurados atuam como transmissor e receptor dos sinais respectivamente. Como os sinais para o microcontrolador são eletricamente compatíveis com o padrão TTL/CMOS, para comunicação com um terminal, como a saída serial RS232 do computador PC, deve ser usado um adaptador de nível. Na implementação do hardware foi utilizado o componente MAX232 para essa função(Figura 3.42). As saídas desse componente são conectadas a um conector DB9, desse modo um cabo padrão para comunicação serial RS232 permite a conexão da Placa de Controle a um computador PC ou qualquer outro dispositivo com comunicação serial.

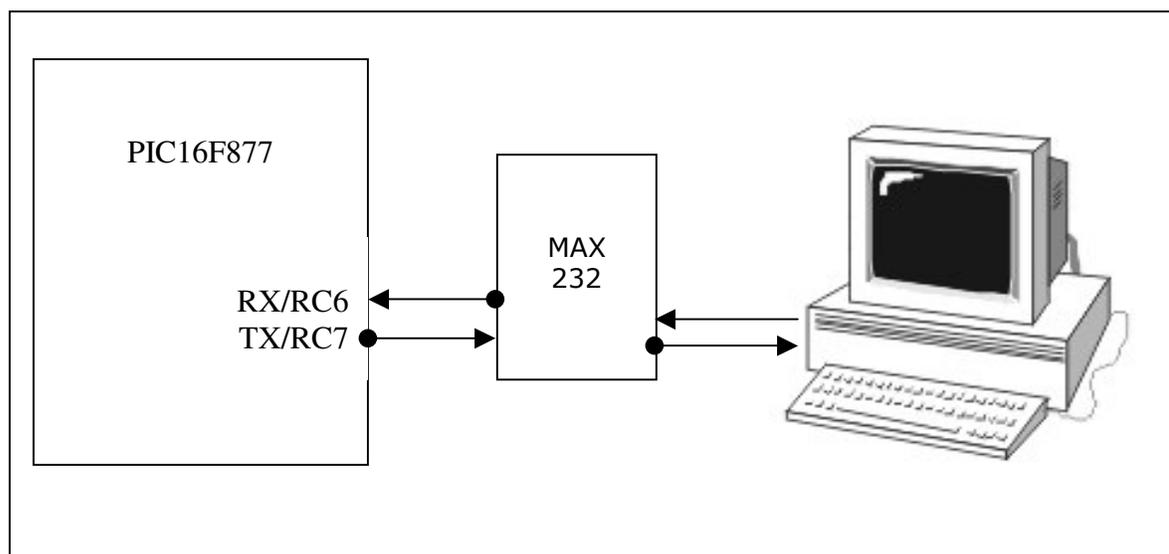


Figura 3.42 - Comunicação serial RS232 da placa de controle

Para a comunicação serial o PIC16F877 dispõe de um conjunto de registradores utilizados para escrever e ler dados, configuração e estado do módulo USART. O registrador TXSTA tem alguns bits utilizados para configurar parâmetros da comunicação e outros recebem o estado na transmissão de dados. O registrador RCSTA permite configurar parâmetros de transmissão. Para transmitir e receber dados são utilizados os registradores TXREG e RCREG. Os bits 4 e 5 do registrador PIR1, registrador para interrupção de periféricos, são utilizados para detectar o estado da transmissão e recepção de dados.

O compilador PICC não dispõe de bibliotecas para a comunicação serial, portanto foi implementada um conjunto de funções para inicialização e configuração do módulo USART, leitura e escrita de dados.

Para configuração foi desenvolvida a função *init_usart()* que por meio da escrita nos registradores TXSTA e RCSTA habilita o hardware do módulo USART, define a velocidade de comunicação, o tamanho da palavra de dados, entre outros parâmetros.

O envio de dados se dá com a função *putch(caractere)*, o parâmetro *caractere* passado como argumento da função é o dado transmitido. Na implementação foi utilizado o registrador TXREG que recebe o dado a ser transmitido. O bit TXIF do registrador PIR1 é testado para indicar se o buffer de saída está vazio e o dado pode ser enviado. O pseudocódigo da tabela 3.14 demonstra a implementação dessa função.

Recebe dado Loop até TXIF ir para nível lógico alto Escrever dado no registrador TXREG
--

Tabela 3.14 - Pseudocódigo para função de escrita da usart

Para receber dados pela porta serial foi desenvolvida a função *getch()* que não recebe parâmetros e devolve o caractere recebido. Na implementação foi empregado o registrador RCREG para receber o dado disponível na USART e o bit RCIF do registrador PIR1 é testado indicando que há dado no buffer de recepção. Na tabela 3.15 é demonstrado o pseudocódigo que implementa a função *getch()*.

Loop até RCIF for para nível lógico alto ler dado do registrador RCREG retornar o dado
--

Tabela 3.15 - Pseudocódigo para função de leitura da usart

A função *putch(caractere)* permite o envio de apenas um caractere e como na maioria das aplicações existe a necessidade do envio de uma cadeia de caracteres foi desenvolvida a função *putst(cadeia)*. Esta função recebe como parâmetro um ponteiro para uma cadeia de caracteres, que tem seu endereço incrementado e a função *putch(caractere)*

envia o valor do mesmo seqüencialmente. O pseudocódigo da tabela 3.16 demonstra a implementação dessa função.

```
Ler ponteiro da cadeia de caracteres
Enquanto endereço do ponteiro for diferente de 0
  Putch ( valor do ponteiro)
  Incrementar endereço do ponteiro
Fim do Enquanto
```

Tabela 3.16 - Pseudocódigo para escrita de cadeia na usart

O conjunto de funções desenvolvidas para a comunicação foi útil no processo de depuração do sistema, pois conectada ao um computador PC executando um software de terminal permitiu inserir mensagens que exibidas no PC possibilitam acompanhar os processos em andamento no microcontrolador. A utilização futura propiciará meios para o controle dos mais diversos dispositivos que se comunicam pela porta serial RS232.

Placa de Experimento

A Placa de Experimento é o módulo que conterà o circuito eletrônico que o usuário definiu para ser utilizado no experimento. A Placa de experimento é constituída pela área de experimento do usuário, pela lógica de controle dos Blocos Eletrônicos, conectores para inserção dos Blocos Eletrônicos e o barramento de dados para comunicação com a Placa de Controle (Figura 3.43). Os sinais com o clock e dados para os potênciômetros digitais chegam também à Placa de Experimento por meio do barramento.

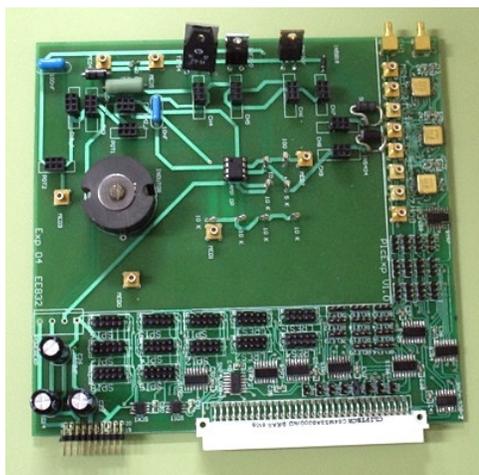


Figura 3.43 - Placa de Experimento

A opção pela forma de utilização da área de experimento fica a critério do usuário. Por exemplo, se o experimento for utilizado frequentemente ele poderá ser montado diretamente na placa de circuito impresso. Se houver necessidade constante da mudança do circuito para realizar um outros experimento pode ser acoplada a Placa de Experimento uma matriz de contato facilitando a implementação e modificação dos circuitos eletrônicos.

A Placa de Experimento dispõe de recursos para controlar um total de 71 Blocos eletrônicos, sendo oito potenciômetros digitais, quatro bancos de componentes, 23 chaves e 24 pontos de medição.

A lógica de controle dos Blocos Eletrônicos é realizada por oito registradores de deslocamento CD4094 que se comunicam com a Placa de Controle por meio do barramento para receber os sinais com dados, clock e strobe. O diagrama da figura 3.44 mostra como são utilizados os componentes CD4094 para controle dos Blocos Eletrônicos.

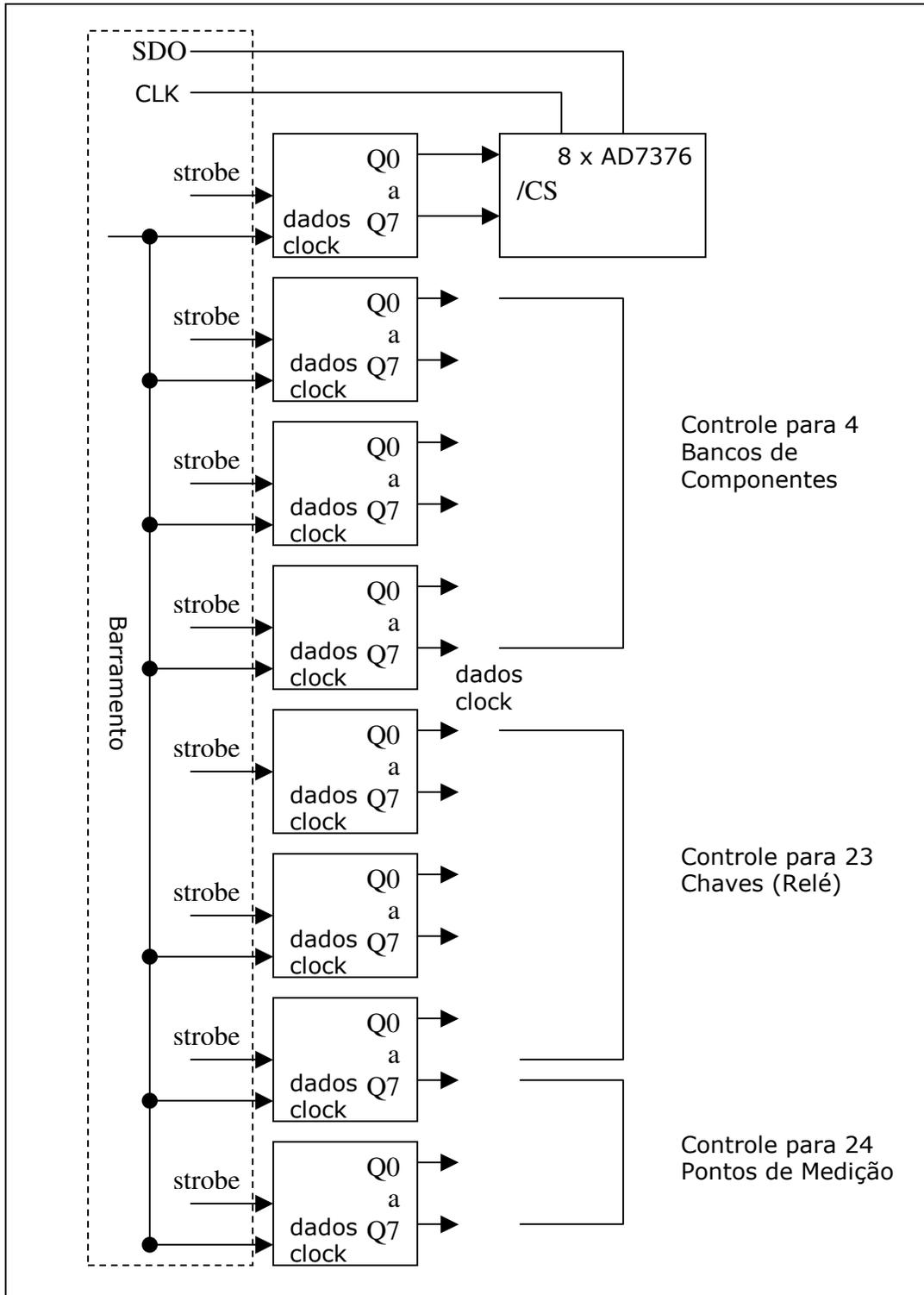


Figura 3.44 - Diagrama da Placa de Experimento

Para utilização dos Blocos Eletrônicos Banco de Componentes, Chave, Potenciômetro Digital existem conectores na Placa de Experimento que permitem inserir os mesmos na quantidade necessária ao circuito eletrônico. Os pontos de medição têm seu circuito, baseado no componente AD7501, montado diretamente na Placa de Experimento juntamente com os conectores para a entrada do instrumento de medida e os cabos que irão para os pontos do circuito nos quais serão realizadas as medições (figura 3.45).

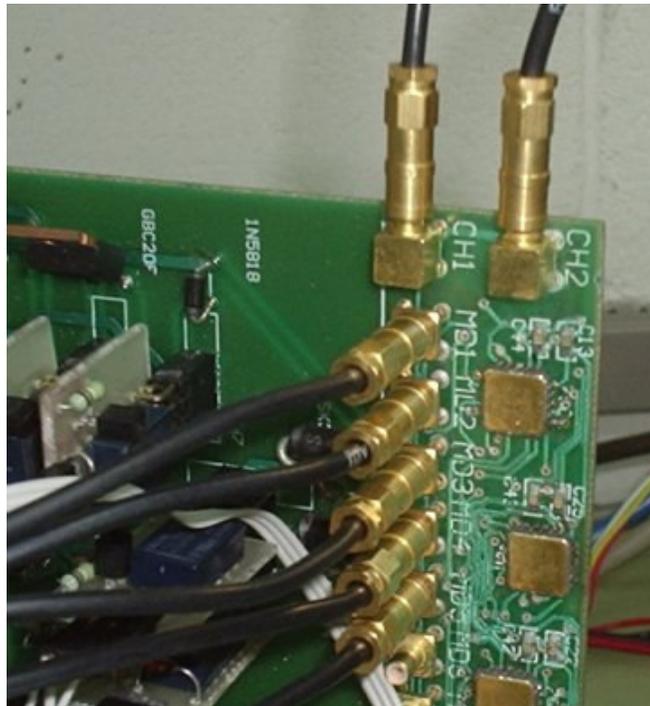


Figura 3.45 - Detalhe do ponto de medição com os cabos conectados

O conceito de modularidade adotado no desenvolvimento da Placa de Experimento propicia meios para que experimentos possam ser elaborados com facilidade e intercambiados rapidamente para atender as necessidades do professor na realização dos experimentos pelos alunos.

Bloco Eletrônico Chave Eletrônica

A chave eletrônica foi desenvolvida para utilização no chaveamento de pontos de medição com instrumentos como multímetros, osciloscópios entre outros. O circuito eletrônico foi implementado tendo por base o componente AD7501. Três desses componentes são montados na Placa de experimento. Para a entrada do instrumento

existem dois conectores e para saída oito conectores que podem ser conectados por meio de cabos nos pontos que se desejam realizar medidas (Figura 3.46).

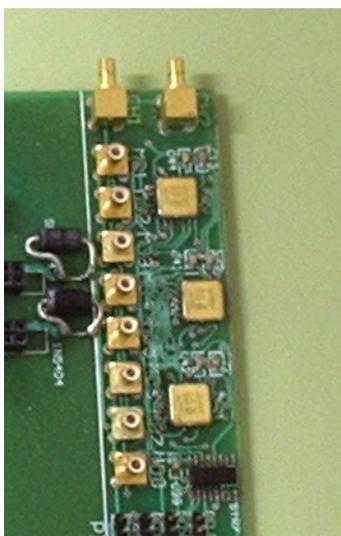


Figura 3.46 - Detalhe do circuito do ponto de medição

Um exemplo de utilização do Bloco Eletrônico é a conexão de dois canais e o ponto de referência de um osciloscópio nas entradas e as saídas no pontos de medida do circuito da Placa de Experimento.

Bloco Eletrônico Chave de Relé

Uma das ações normalmente realizadas em um experimento eletrônico é o chaveamento de componentes. Para atender esse requisito foi desenvolvido o Bloco Eletrônico Chave. O chaveamento é executado por relés e a justificativa para a utilização desse componente na implementação se deve ao fato de que os níveis de tensão e corrente podem variar muito em função do experimento, e a características elétricas dos relés atendem uma gama ampla de variações.

O circuito eletrônico é constituído do relé, um transistor excitador e um diodo em paralelo com a bobina do relé com a função de proteger o transistor da tensão reversa gerada no chaveamento (figura 3.47).

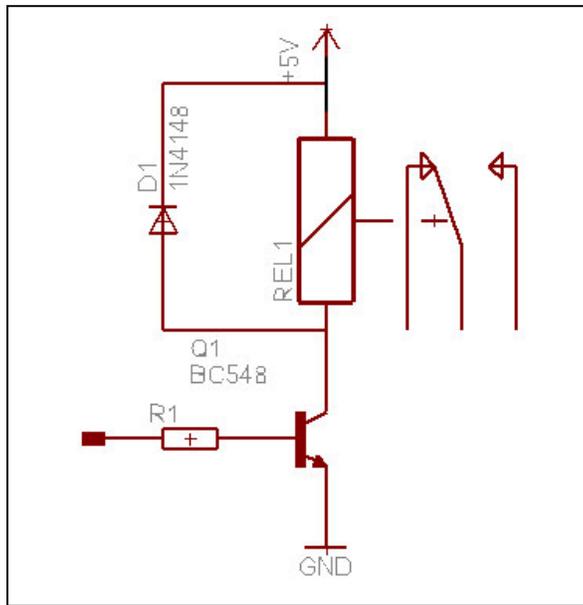


Figura 3.47 - Circuito do Bloco Eletrônico Chave de Relé

Ao se aplicar o nível lógico alto a entrada do Bloco Eletrônico o relé é acionado.

O Bloco Eletrônico Chave de Relé é montado em uma pequena placa de circuito impresso contendo os componentes eletrônicos e os conectores de entrada e saída. (figura 3.48). Os conectores de saída são conectados ao ponto do experimento em que ocorrerão o chaveamento de sinais ou componentes.

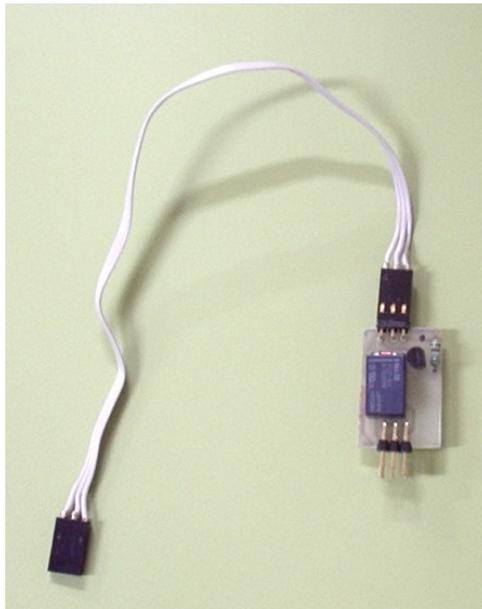


Figura 3.48 - Bloco Eletrônico Chave de Relé

Com o emprego de relés com características elétricas adequadas é possível utilizar o Bloco Eletrônico Relé nos mais diversos experimentos.

Bloco Eletrônico Potenciômetro Digital

Este Bloco eletrônico é implementado a partir do componente AD7376, um potenciômetro digital que se comunica por meio do protocolo SPI. Com esse bloco as ações do usuário no objeto potenciômetro digital envia comandos que decodificados na placa de controle e encaminhados ao Bloco Eletrônico Potenciômetro Digital varia sua resistência elétrica com atuação semelhante a um potenciômetro com controle mecânico tradicional.

Este Bloco Eletrônico é constituído por uma placa de circuito impresso que contém o componente AD7376 e os conectores de entrada para alimentação e sinais SPI, e a saída resistiva (figura 3.49).



Figura 3.49 - Bloco Eletrônico Potenciômetro Digital

Várias empresas têm em sua linha de produtos potenciômetros digitais com os mais diversos parâmetros e que se comunicam pelo protocolo SPI, podendo de modo geral atender os requisitos dos circuitos eletrônicos experimentais.

Testes do sistema de hardware

O sistema de hardware constituído pelos circuitos da Placa de Controle, Placa de Experimentos, Blocos Eletrônicos e placa de barramento, após montagem e integração, passou por um processo de testes para garantir a sua funcionalidade elétrica e a operação básica de comunicação USB e do protocolo SPI.

Com a utilização de oito Placas de Experimentos simultâneas, cada uma delas contendo oito componentes CD4094 que podem variar suas oito saídas, somados existem 512 saídas. Para avaliar o funcionamento dessas saídas foram desenvolvidos dois conjuntos de aplicativos para o computador PC com funcionalidade para comunicação com a Placa de Controle via porta USB.

Para o microcontrolador da Placa de Controle foi implementado um programa contendo as funções ler_ftdi, escrever_4094, escrever_usart e escrever_spi.

O primeiro aplicativo desenvolvido para o computador PC (figura 3.50) enviava comandos para a Placa de Controle que permitia controlar os diversos sinais lógicos gerados na Placa de Controle como strobe, dados, clock e a comunicação no protocolo SPI. Com esse recurso e utilizando um multímetro foi possível avaliar todos os sinais elétricos existentes na Placa de Controle e Placa de Experimentos.

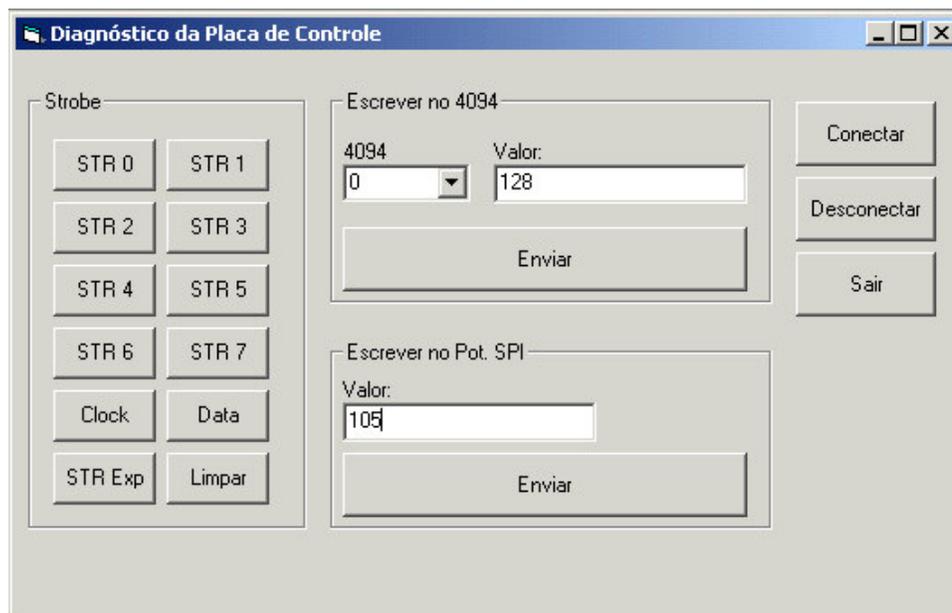


Figura 3.50 - Aplicativo inicial para diagnóstico do hardware

Com a avaliação inicial e a certificação de que o sistema estava operando corretamente foi desenvolvido um segundo aplicativo para o computador PC que possibilitou avaliar as várias funções desenvolvidas no microcontrolador para controle dos Blocos Eletrônicos, seleção dos experimentos e o desempenho da comunicação USB (figura 3.51).

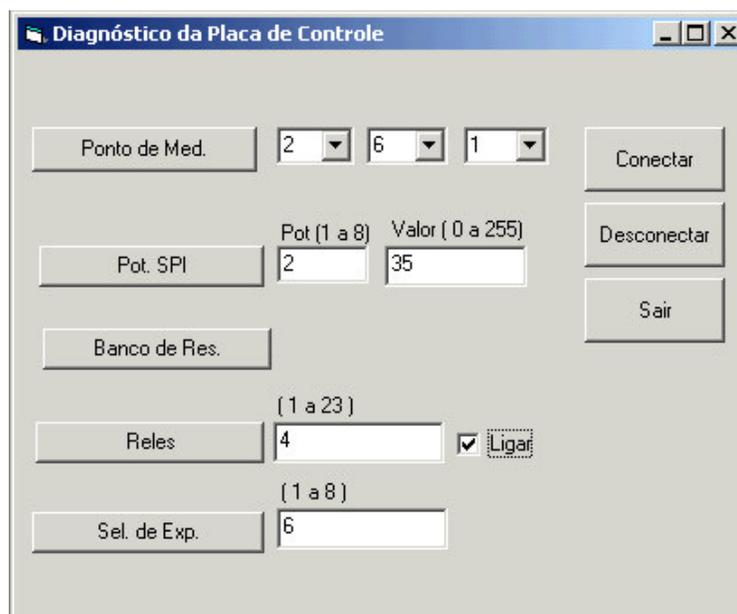


Figura 3.51 - Aplicativo para avaliar funções da Placa de Controle

Com o processo de testes do hardware efetuado e comprovado o funcionamento adequado dos seus vários componentes foi iniciado o desenvolvimento dos aplicativos para o computador PC que são descritos no próximo capítulo.

Capítulo 4 - Projeto e implementação do software

Neste capítulo é apresentada a implementação do Editor de Experimentos que possibilita ao usuário/professor criar um projeto no qual é possível especificar os elementos da interface gráfica do experimento e atribuir todas as funcionalidades relacionadas aos objetos de hardware e de informação.

Na seção Cliente de Experimento Remoto será descrita a implementação desse aplicativo responsável por executar o projeto que foi previamente criado no Editor de Experimentos.

A seção Servidor de Experimentos demonstra o desenvolvimento dos subsistemas necessários para comunicação com o Cliente do Experimento remoto e o hardware da Placa de Controle. O sistema para controle e exibição dos sinais do osciloscópio é apresentado nas seções Cliente e Servidor GPIB.

A integração dos aplicativos descritos estabelece um sistema completo para realização de experimentos remotos pela rede Internet.

Editor de Experimentos

A implementação do Editor de Experimentos contempla todas as funcionalidades necessárias do ponto de vista de software para o usuário elaborar o projeto do experimento. Após a montagem do hardware do experimento que compreende montar o circuito eletrônico e associar os vários Blocos Eletrônicos é com o Editor de Experimentos que serão inseridos os objetos que representam os Blocos Eletrônicos e atribuídos valores às propriedades de cada bloco. A atuação com o usuário ocorre por meio de uma interface gráfica (figura 4.52) composta por uma área de trabalho e pela barra de ferramentas. Cada objeto bloco eletrônico ou objeto de informação acessa uma janela de propriedades.

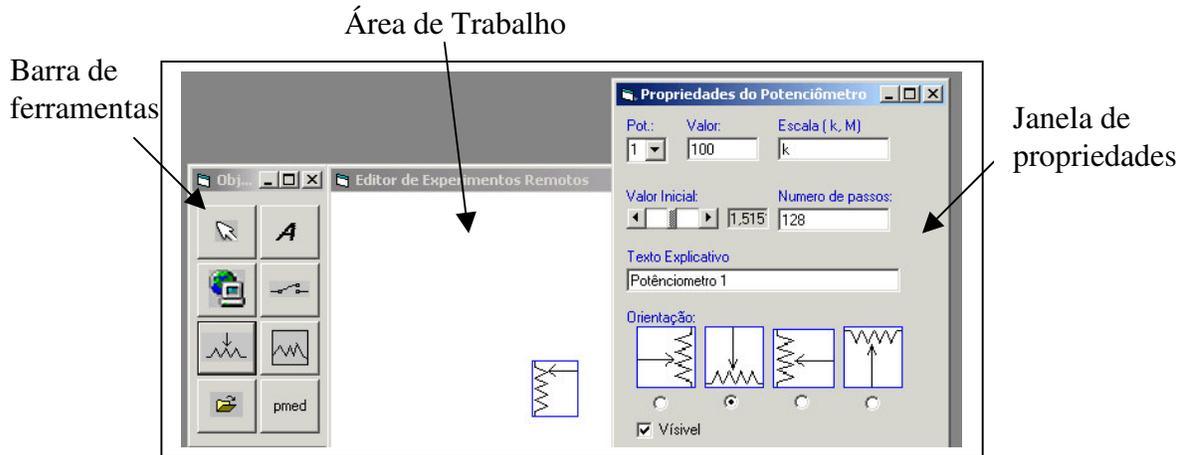


Figura 4.52 - Interface gráfica do editor de experimentos

Barra de ferramentas

A barra de ferramentas (Figura 4.53) é uma janela contendo botões que representam os elementos de hardware e de informação que podem ser selecionados e inseridos no projeto, também é possível importar a figura com o esquemático do circuito eletrônico.

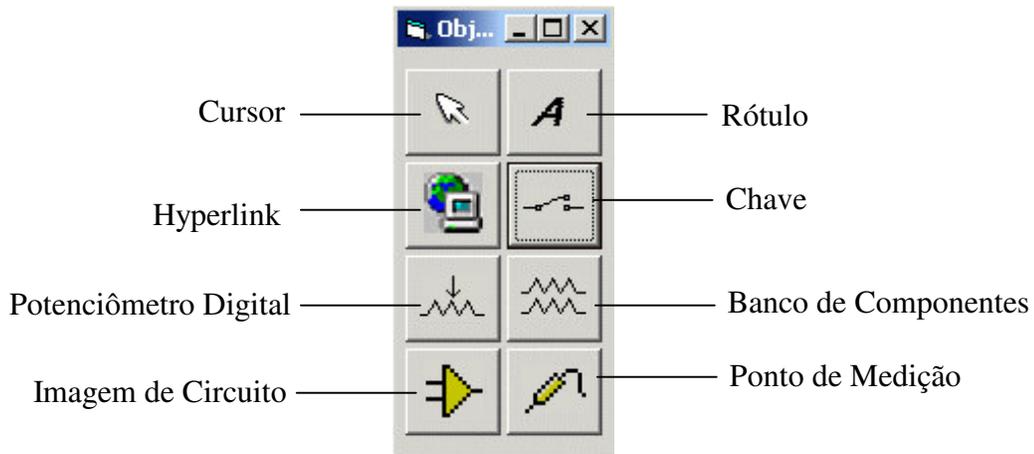


Figura 4.53 - Barra de ferramentas do Editor de Experimentos

Área de trabalho

A área de trabalho é constituída de uma janela na qual podem ser inseridos os objetos de hardware, de informação e a imagem com o desenho do circuito eletrônico. Clicando com o botão esquerdo do mouse e mantendo o mesmo pressionado é possível arrastar o objeto para a posição desejada. Com um duplo clique no mouse é habilitada uma alça de dimensionamento que permite a alterar a altura e largura do objeto. Clicando na área do objeto com o botão direito abre-se a janela de propriedades.

Janela de propriedades

Cada um dos objetos inseridos no projeto, quer sejam de hardware ou de informação, devem ter suas propriedades configuradas. Para cada objeto foi implementada uma janela de propriedades na qual podem ser inseridas os diversos valores específicos para cada objeto e que serão utilizados durante a execução do Cliente do Experimento.

Caso de uso do Editor de Experimentos

A seguir a demonstração por meio de um caso de uso descreve as ações necessárias para elaborar o projeto no Editor de Experimentos. Supõe-se que os componentes de hardware já estejam montados e configurados na placa de experimento.

- Criar o circuito esquemático em um editor de circuito e exportá-lo para um formato de figura como JPEG, Gif ou BMP.
- Executar o Editor de Experimentos
- Selecionar e inserir a figura com o esquemático
- Inserir os blocos eletrônicos que já estão configurados fisicamente na placa de experimento.
- Configurar as propriedades dos blocos eletrônicos.
- Inserir os blocos de informação como rótulos e hyperlink. Este passo não é imprescindível à operação do experimento, mas pode agregar informações úteis ao usuário no momento da execução do experimento.
- Salvar projeto
- Carregar o Cliente do Experimento com o projeto para visualização e avaliação.

O diagrama da figura 4.54 demonstra as funcionalidades implementadas no Editor de Experimentos e o fluxo de dados que o integra ao Cliente de Experimento Remoto.

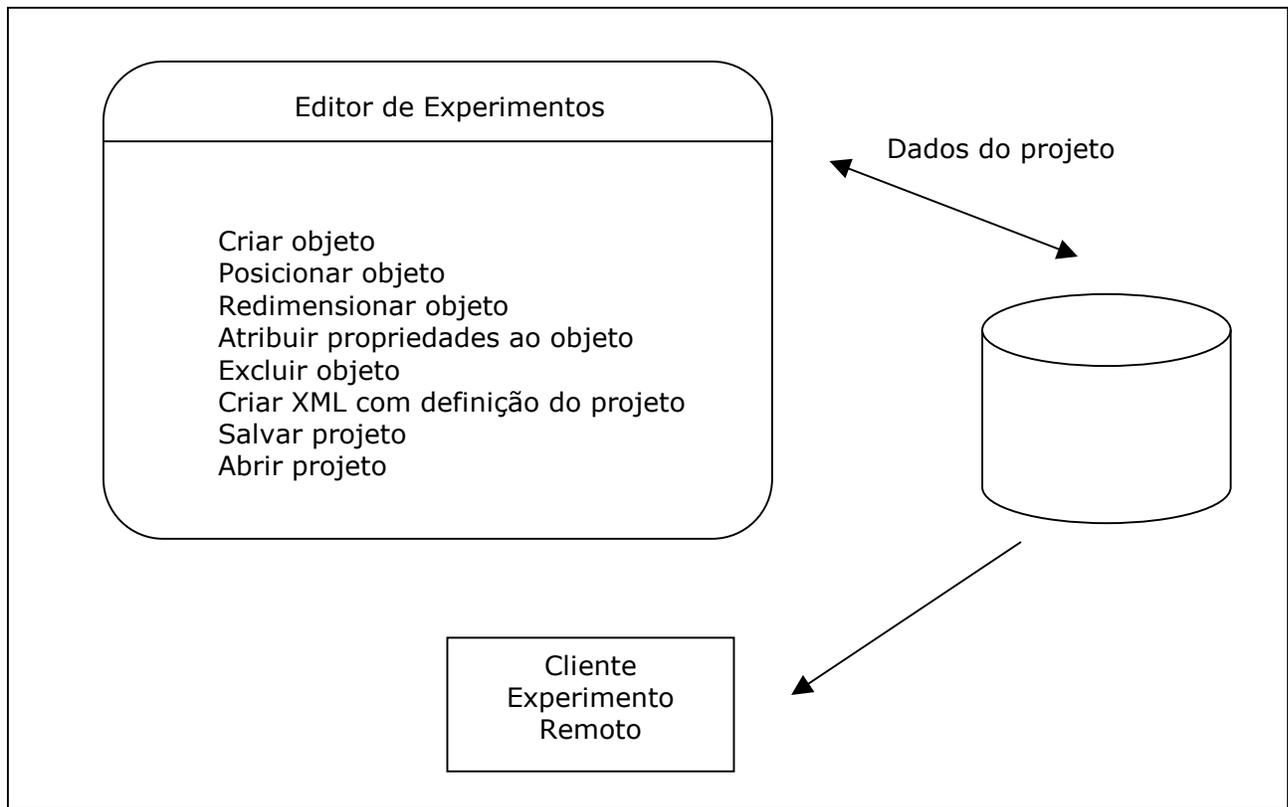


Figura 4.54 - funcionalidades do editor de experimentos

A seguir será descrita a implementação das funcionalidades desenvolvidas para o Editor de Experimentos.

Criar objeto

Para efeito de organização os objetos do Editor de Experimento foram classificados em objetos de informação e objetos Blocos Eletrônicos. Os objeto de informação são o rótulo e o link, eles não possuem correspondente em hardware, mas agregam informações textuais ao projeto. Os objetos bloco eletrônico são as representações em software dos Blocos Eletrônicos implementados em hardware e devem ter suas propriedades configuradas conforme a ação que se deseja obter no bloco eletrônico.

As propriedades de texto explicativo, posição e dimensão na área de trabalho e depois na visualização no Cliente do Experimento Remoto são comuns a todos objetos. As outras propriedades são exclusivas de cada objeto, pois representam suas particularidades do ponto de vista de hardware.

Para inserir um objeto é utilizado o comando *Load* que atua na criação de objetos em tempo de execução. O código da tabela 4.17 ilustra a implementação da criação do objeto Chave e a atribuição das propriedades de posicionamento na área de trabalho.

```
Dim NewIndex2 As Integer
NewIndex2 = Image2.Count
Load Image2(NewIndex2)
Image2(NewIndex2).Visible = True
Image2(NewIndex2).Left = X
Image2(NewIndex2).Top = Y
```

Tabela 4.17 - Código para a criação do objeto chave

A seguir a descrição dos objetos implementados e suas propriedades.

Potenciômetro Digital

O objeto potenciômetro digital é a representação do Bloco Eletrônico desenvolvido com o componente AD7376. Para a implementação foi utilizado o objeto do Labview *CWslide*, um controle do tipo deslizante. Ao controle foi atribuída a figura de um resistor e de uma seta compondo o desenho de um potenciômetro digital. O usuário no Cliente do Experimento remoto ao clicar no objeto pode variar a posição da seta (figura 4.55) o que gera um evento e a alteração do valor numérico da propriedade *Value*.

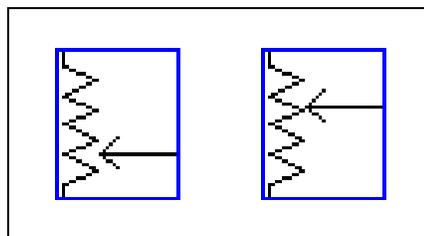


Figura 4.55 - Objeto potenciômetro digital representando dois valores

As seguintes propriedades foram definidas para o objeto potenciômetro digital:

- Potenciômetro ativo - Esta propriedade define qual dos oito potenciômetros possíveis de existir em um Experimento irá receber o comando de controle para variação de sua resistência.
- Valor Resistivo - Indica o valor em Ohms do potenciômetro utilizado.

- Escala - Propriedade que irá compor o texto explicativo indicando o valor do resistor.
- Valor Inicial - Valor que o potenciômetro deverá assumir no início do experimento.
- Número de passos - O número de passos do potenciômetro pode variar de acordo com o fabricante e com o componente, portanto para realizar os cálculos do valor do potenciômetro é preciso especificar o número de passos do componente utilizado.
- Texto Explicativo - O usuário pode elaborar um texto que será apresentado em uma caixa de texto sempre que o cursor do mouse passar sobre o objeto gráfico.
- Orientação - A figura do potenciômetro pode ser apresentada nas posições vertical à esquerda e à direita, e na horizontal para cima e para baixo. A seleção é feita clicando nos botões de opção da janela de propriedades.
- Visível - Em muitas situações pode ser necessário ter um potenciômetro no circuito, mas que não seja controlado pelo usuário, nesse caso pode se optar por selecionar o potenciômetro como invisível, ele receberá o comando inicial e o manterá já que não sendo visível no Cliente do Experimento Remoto não poderá ter seu valor alterado.

A figura 4.56 ilustra a janela de propriedades que possibilita a atribuição e alteração das propriedades do potenciômetro digital.



Figura 4.56 - Janela de propriedades do potenciômetro digital

Chave

O objeto chave atua sobre o Bloco Eletrônico chave. Implementado a partir do objeto *Image*, nativo da ferramenta de programação, Por meio da sua propriedade *picture* carrega um conjunto de imagens com o símbolo eletrônico para chave nas posições vertical e horizontal e ligada e desligada (figura 4.54).

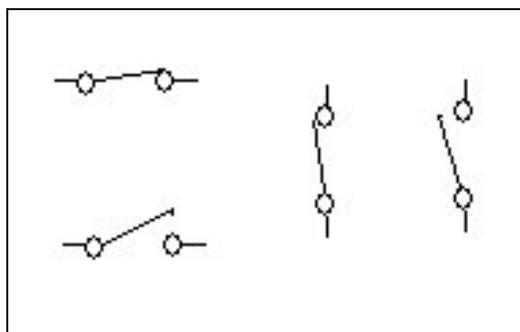


Figura 4.57 - Representação gráfica do objeto chave

No Cliente do Experimento Remoto ao clicar no objeto chave a imagem é mudada para o símbolo de chave aberta ou fechada e a rotina que trata do evento clicar envia o comando para atuar no Bloco Eletrônico Chave executando a ação solicitada.

O objeto chave dispõe das seguintes propriedades:

- Qual chave - Indica qual dos Blocos Eletrônico chave receberá o comando na Placa de Experimento.
- Texto Explicativo - Texto elaborado pelo usuário que será apresentado no Cliente do Experimento Remoto quando o cursor do mouse passar sobre o objeto chave
- Orientação - A apresentação do objeto chave pode ser na posição vertical ou horizontal e selecionada pelo botão de opção clicando em horizontal ou vertical.
- Estado Inicial - A posição ligada ou desligada pode ser configurada para ser assumida pelo Bloco Eletrônico ao ser carregado o Cliente do Experimento Remoto.
- Visível - Propriedade que permite configurar se o elemento gráfico do objeto chave será visível no Cliente do Experimento Remoto. O recurso de inserir chaves e torna-las visíveis ou não cria a possibilidade de se elaborar experimentos reconfiguráveis pelo software, pois com a combinação de chaves abertas e fechadas é possível criar circuitos diversos. Desse modo uma mesma placa de experimento pode oferecer mais de um experimento.

As propriedades do objeto chave podem ser configuradas durante a edição do experimento. Ao clicar com o botão direito sobre o objeto é aberta uma janela para atribuição das diversas propriedades (figura 4.58).



Figura 4.58 - Janela para atribuição das propriedades ao objeto chave

Ponto de Medição

O objeto ponto de medição permite abrir e fechar as chaves eletrônicas do componente AD7501 que tem as pontas de prova do osciloscópio conectadas. Este objeto é implementado a partir dos objetos *Label* e *Combobox*. O ponto de medição deve ser inserido no ponto do circuito eletrônico que se queira fazer a medição. Propriedades do objeto permitem selecionar o ponto, que está associado ao hardware da placa de experimento. Ao clicar sobre o ponto de medição é gerado um evento que dispara a rotina que posiciona o *Combobox* próximo ao ponto de medição e permite selecionar se o ponto será conectado ao canal 1, 2 ou ao ponto de referência (figura 4.59). A seguir é enviado um comando que irá atuar sobre o hardware da placa de Experimento.

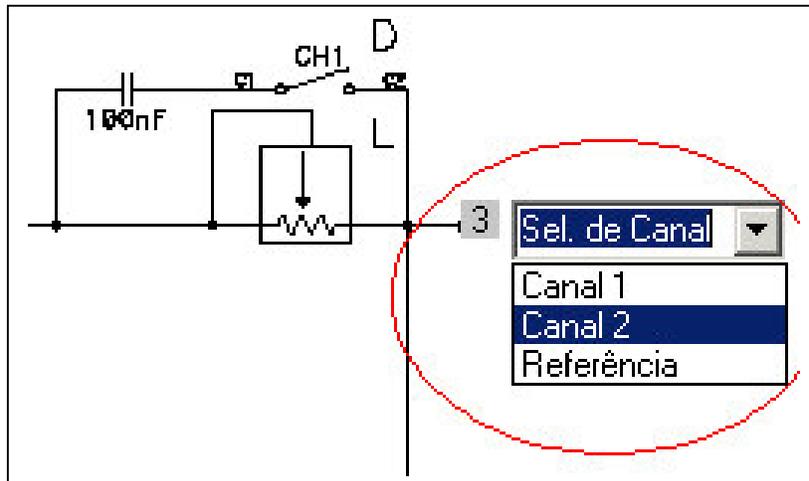


Figura 4.59 - Detalhe da representação gráfica do ponto de medição

A seguir são apresentadas as três propriedades que podem ser atribuídas ao objeto ponto de medição (figura 4.60):

- Ponto - permite selecionar o ponto de medição. Podem ser selecionados 8 pontos de medição por placa de experimento
- Texto Explicativo - Configura o texto que será exibido no Cliente do Experimento Remoto quando o cursor do mouse passar sobre o objeto.
- Visível - permite selecionar a opção de exibição ou não do objeto no Cliente do Experimento Remoto.



Figura 4.60 - Janela para atribuição das propriedades ao objeto ponto de medição

Banco de componentes

O Objeto Banco de Componentes permite configurar o banco de componentes que será usado, qual bloco e qual resistor será ativado (Figura 4.61).

A seguir são apresentadas as propriedades desse objeto:

- Selecionar Banco - Habilita um dos quatro bancos disponíveis.

- Selecionar Bloco - Permite selecionar qual bloco do banco configurado será utilizado. Pode variar de 1 a 3.
- Selecionar Resistor - Ativa um dos oitos resistores disponíveis no bloco e banco previamente configurado.
- Orientação - Exibe o resistor na posição vertical ou horizontal.
- Visível - Indica se o objeto será ou não visível no Cliente do Experimento Remoto.



Figura 4.61 - Janela para atribuição de propriedades ao objeto Banco de Resistores

Hyperlink

O objeto hyperlink é um objeto de informação, portanto não tem nenhum hardware associado (figura 4.62). É implementado a partir do objeto Hlink. Ao ser clicado é executado o navegador Web padrão do sistema que carrega a URL indicado na propriedade URL do objeto. Este recurso pode ser útil para inserir material de apoio. Por exemplo, o objeto HyperLink quando clicado pode apontar para o endereço que tem um documento com as especificações de determinado componente do circuito eletrônico que está no experimento.

As propriedades do objeto Hyperlink são:

- URL - Indica o endereço que o navegador web deverá carregar
- Texto do Hyperlink - Texto indicativo sobre o endereço que será carregado
- Texto Explicativo - Configura o texto que será exibido no Cliente do Experimento Remoto quando o cursor do mouse passar sobre o objeto.
- Visível - Indica se o objeto será ou não visível no Cliente do Experimento Remoto.

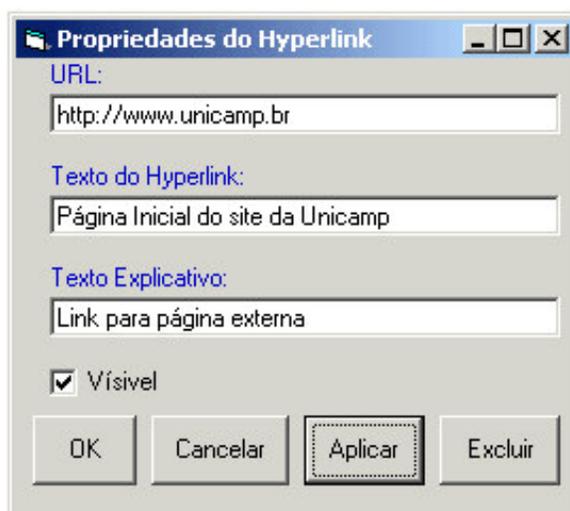


Figura 4.62 - Janela para atribuição de propriedades do objeto hyperlink

Rótulo

O objeto rótulo é outro objeto de informação e utilizado para inserir informações textuais que serão apresentadas no cliente do Experimento Remoto (figura 4.63).

O objeto Rótulo é implementado com o objeto Label e dispõe das seguintes propriedades:

- Texto do rótulo - permite ao usuário inserir o texto que ele desejar
- Visível - torna o objeto visível ou não no Cliente do experimento Remoto.

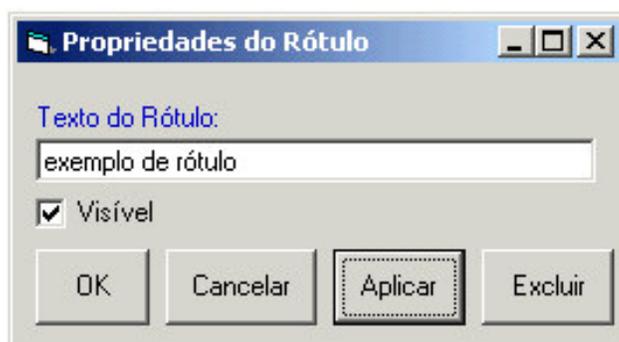


Figura 4.63 - Janela para atribuição de propriedades ao objeto rótulo

Posicionar objeto

Os objetos inseridos na área de trabalho podem ser posicionados para se ajustarem à posição que fique mais adequada a apresentação do esquemático do circuito eletrônico. O deslocamento do objeto pela área de trabalho ocorre quando o usuário clica o mesmo com o

mouse e sem soltar o botão desloca o mouse fazendo com que o objeto acompanhe o posicionamento do cursor (drag and drop), ao soltar o botão o objeto assume essa última posição.

Para esta implementação é associada uma rotina ao evento *Mouse_down* do objeto. Esse evento é gerado sempre que um botão do mouse é pressionado. Quando o evento ocorre a coordenada dos eixos X e Y do cursor do mouse na área de trabalho começa ser registrada e é atribuída as propriedades *Top* e *Left* do objeto que disparou o evento. Esta ação faz o objeto se deslocar pela área de trabalho a medida que o cursor do mouse muda de posição.

Redimensionar objeto

Para que os objetos inseridos na área de trabalho possam ter a sua dimensão ajustada foi desenvolvido o recurso para redimensionamento de objeto. Este recurso é importante, pois permite aos objetos de informação ter o texto exibido de forma correta e aos objetos Bloco Eletrônico um ajuste adequado quando sobrepostos a figura que contém o circuito eletrônico.

Para implementação o evento *DbClick* dos objetos, gerado quando o usuário dá um duplo clique no mouse, invoca uma rotina que torna visível um rótulo no extremo direito inferior do objeto (figura 4.64). Este rótulo atua como alça de redimensionamento. O rótulo ao ser clicado tem capturado o evento *Mouse_down* ao qual se atribuí uma rotina que registra a posição do cursor do mouse que soma ou subtrai as coordenadas X e Y obtidas às propriedades *Width* e *Height* que determinam a altura e largura do objeto que se deseja redimensionar.

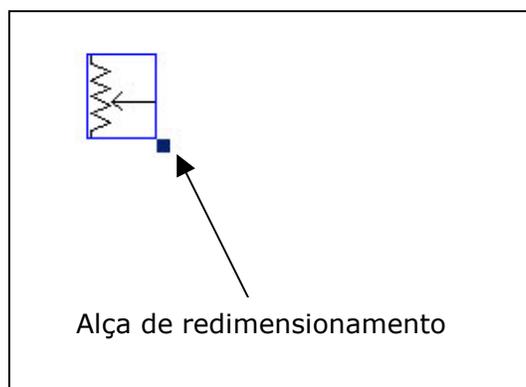


Figura 4.64 - Redimensionamento de objeto

Atribuir propriedades ao objeto

Cada objeto do Editor de Experimentos, quer seja de Informação ou Bloco Eletrônico, tem um conjunto de propriedades que deve ser configurada e pode ser alterada (figura 4.65). Para efetuar esse processo cada objeto tem um formulário com suas propriedades. Este formulário é carregado por uma rotina invocada ao ser gerado o evento `Mouse_down` que ocorre quando o usuário clica com o botão direito do mouse sobre um objeto. Cada objeto tem seu formulário, pois as propriedades são particulares. Esse mesmo formulário dispõe de botões para efetuar as ações de cancelar a alteração, excluir o objeto, aplicar a configuração e validar a alteração.

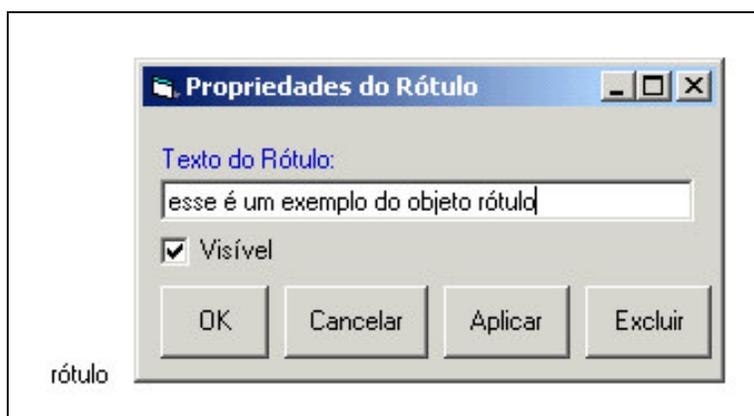


Figura 4.65 - Janela de propriedades do objeto rótulo

Os dados com as configurações são guardados em variáveis e atribuídos ao objeto que assume os parâmetros de forma imediata. Desse modo a visualização no modo de edição é a mesma que será exibida no Cliente do Experimento Remoto.

Excluir objeto

Os objetos que foram inseridos em um projeto podem ser excluídos a qualquer momento durante a edição. Para efetuar esse procedimento o usuário deve clicar com o botão direito do mouse no objeto que será excluído, no formulário que abre o acionamento do botão Excluir invoca uma rotina que executa o comando *Unload* nativo da ferramenta de programação que tem a função de excluir o objeto. Como pode haver vários objetos do mesmo tipo é atribuído um índice que identifica o objeto. A tabela 4.18 mostra a implementação da rotina para exclusão do objeto Hyperlink.

```
Unload Form1.HLink1(ind_controle8)
Unload Form4
```

Tabela 4.18 - Procedimento para excluir objeto no editor de experimentos

Salvar Projeto

Ao finalizar o projeto o mesmo deve ser gravado para posterior edição e para ser carregado pelo Cliente do Experimento Remoto. Ao Clicar no botão Salvar e após atribuir um nome ao arquivo inicia-se um procedimento que cria o arquivo XML e associa a estrutura desse arquivo todas as definições do projeto, os objetos e suas propriedades. Ao finalizar esse procedimento é chamado o método *Save* do objeto instanciado como documento XML. A seguir é demonstrada a implementação do método *save*.

```
objDOMDocument.save ("C:\exemplo1.xml")
```

Tabela 4.19 - Procedimento para salvar projeto no editor de experimentos

Abrir projeto

Para abrir um projeto o arquivo XML contendo a definição do projeto é carregado utilizando o método Load da instância do documento XML. A seguir cada nó do documento XML é lido. Ao encontrar a definição de um objeto este é criado e as propriedades recuperadas são aplicadas. Este processo é efetuado para todos os objetos. Desse modo ao finalizar a leitura a área de trabalho conterà todos os objetos criados anteriormente com suas respectivas propriedades.

A tabela 4.20 mostra fragmento de código que implementa a leitura do arquivo XML e atribuição das propriedades aos objetos.

```

Set objDOMDocument = New DOMDocument30
Dim objAttrib As IXMLDOMAttribute
Dim objvideosegment As IXMLDOMElement
objDOMDocument.Load ("C:\exemplo1.xml")
Dim ToNodes As IXMLDOMNodeList
Set ToNodes = objDOMDocument.selectNodes("//figura_fundo")
Set onodes = objDOMDocument.selectSingleNode("//figura_fundo/nome")
Call carrega_arq_ftp(onodes.Text)
UserControl.Picture = LoadPicture("c:\" & onodes.Text)

```

Tabela 4.20 - Procedimento para leitura do arquivo XML

Criar XML com definição do projeto

Ao elaborar um projeto no Editor de Experimentos objetos são inseridos, posicionados, redimensionados e tem suas propriedades definidas. Todas essas ações contêm informações que devem gravadas para posterior edição e para utilização no Cliente do Experimento Remoto. Entre as várias opções de persistências para essas informações o emprego de banco de dados foi uma opção considerada, porém como a Arquitetura para Experimento Remoto foi planejada para favorecer a integração de tecnologias em plataformas de software e hardware heterogêneas a tecnologia XML por estar se firmando como recurso para troca de dados com independência de plataforma foi a opção escolhida.

A implementação teve como base o modelo DOM, ou Document Object Model uma especificação para programação de interface, desenvolvida pelo World Wide Web Consortium (W3C, 2005). O DOM define interfaces que possibilitam ao programador navegar em documentos XML e manipular seu conteúdo e estrutura.

O modelo DOM permite que as aplicações trabalhem com as estruturas e informações do documento XML como estruturas de programa ao invés de cadeias de caracteres. Os interpretadores baseados no DOM carregam o documento como uma árvore hierárquica. Os nós da árvore representam o conteúdo e estrutura. As interfaces de programação possibilitam a aplicação acessar a árvore e manipular os nós. A instalação padrão de sistemas operacionais como Windows 2000 e XP disponibilizam a biblioteca dinâmica MSXML que implementa o modelo DOM, com os recursos desta biblioteca é possível carregar ou criar um documento, acessar e manipular a informação e estrutura

contida neste documento; e salvar em um arquivo XML. Na implementação a partir da instanciação da MSXML como objeto são utilizados os métodos, propriedades e eventos associados a ela para prover a manipulação dos arquivos XML. A tabela 4.21 mostra fragmento de código necessário para instanciação do objeto DOM e as operações de criação de um elemento na árvore XML e a gravação do arquivo.

```
Set objDOMDocument = New DOMDocument30
Set projeto = objDOMDocument.createElement("meuprojeto")
Set objNode = objDOMDocument.appendChild(projeto)
objDOMDocument.save ("C:\exemplo1.xml")
```

Tabela 4.21 - Procedimento para instanciar um objeto DOM

As informações dos projetos são estruturadas em um documento XML que grava as informações do projeto, como o nome do arquivo com a figura do esquemático e aninhados os objetos com suas propriedades. O diagrama da figura 4.66 apresenta a estrutura de dados do arquivo XML contendo a definição de projeto do Editor de Experimentos. A estrutura detalhada com a informação de todos objetos e suas propriedades está contida no anexo II.

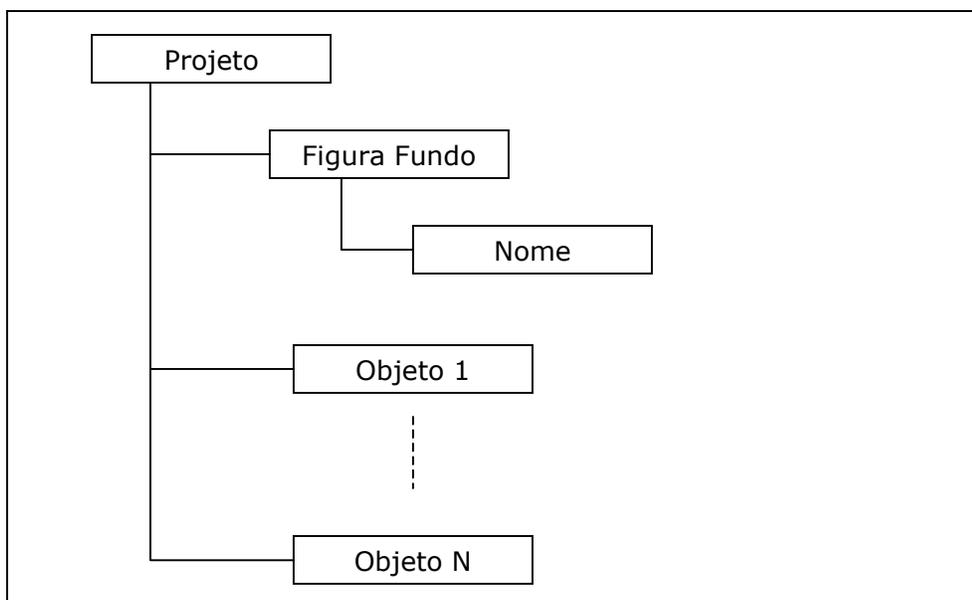


Figura 4.66 - Definição XML do projeto

Na tabela 4.22 pode ser visto um exemplo de projeto no qual está definido a figura de fundo, um bloco eletrônico chave e um objeto rótulo.

```

<meuprojeto>
  <figura_fundo>
    <nome>banco_h.bmp</nome>
  </figura_fundo>
  <chave>
    <top>1845</top>
    <left>2895</left>
    <height>855</height>
    <widht>810</widht>
    <orientacao>0</orientacao>

    <estado>1</estado>
    <tooltip>chave liga desliga</tooltip>
    <qualchave>1</qualchave>
    <visivel>1</visivel>
  </chave>
  <rotulo>
    <top>915</top>
    <left>1995</left>
    <texto>exemplo de rótulo</texto>
    <objeto>rotulo</objeto>
    <visivel>1</visivel>
  </rotulo>
</meuprojeto>

```

Tabela 4.22 - Exemplo de arquivo de projeto em formato XML

Cliente do Experimento Remoto

O aplicativo Cliente do Experimento Remoto desenvolvido para ser executado como um applet inserido em um navegador Web tem com função a exibição da interface gráfica com a representação do circuito, dos objetos eletrônicos e de informação que ao serem ativados enviam comandos para o Servidor Placa de Controle que os decodifica e encaminha para execução na Placa de Experimento.

O diagrama da figura 4.67 ilustra a arquitetura do aplicativo Cliente do Experimento Remoto e o fluxo de dados do sistema.

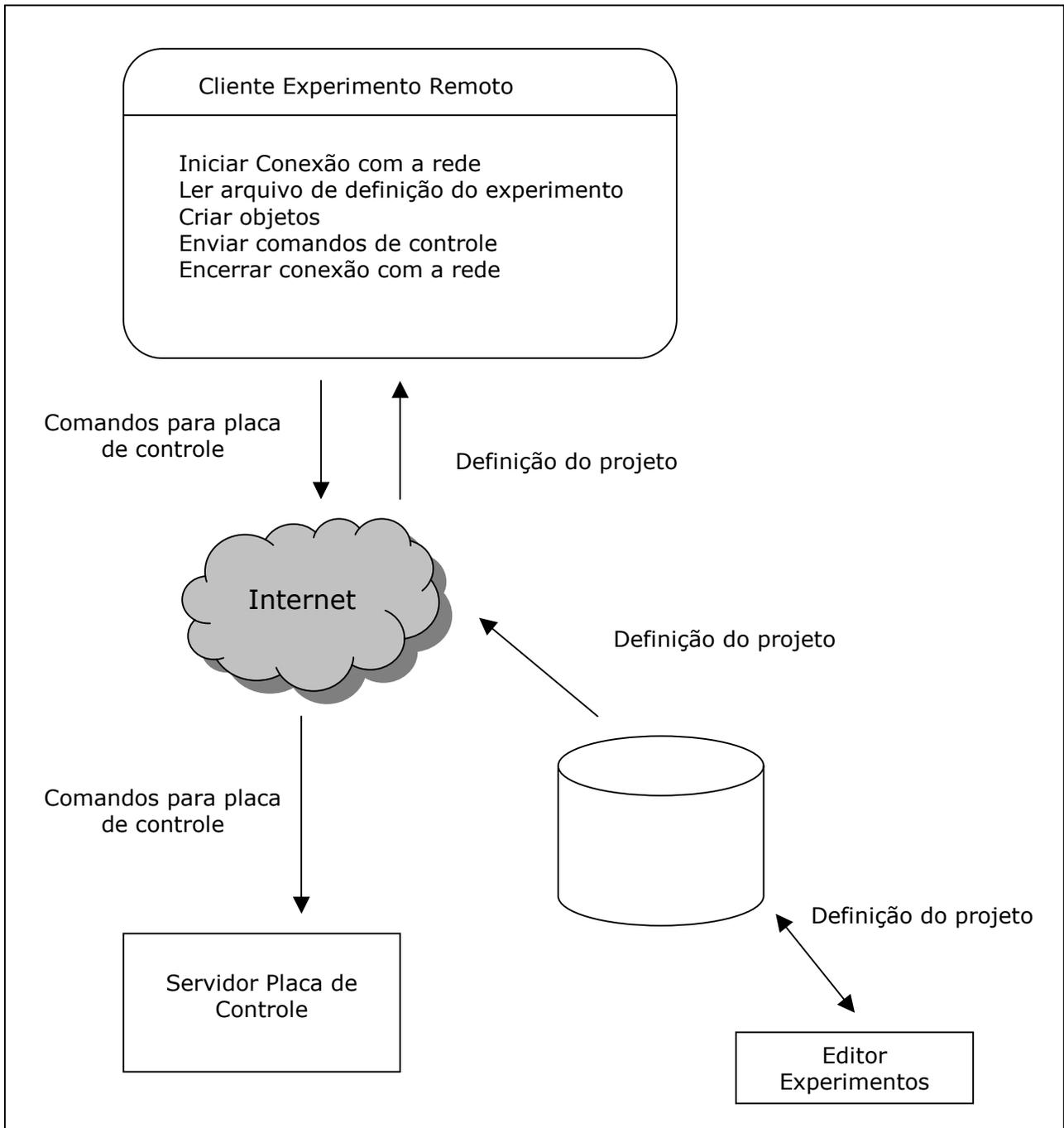


Figura 4.67 - Diagrama de funcionalidades do Cliente do Experimento Remoto

Para atender as funcionalidades previstas foi desenvolvido um conjunto de funções que são descritas a seguir.

Iniciar Conexão com a rede

O cliente deve se comunicar com a rede Internet para receber a definição do projeto e para encaminhar os comandos que são recebidos pelo Servidor da Placa de Controle. A implementação da comunicação utiliza o objeto do Labview *CWDataSocket1* invocando o método *ConnectTo* passando os parâmetros *para_servidor_plc* e *cwdsWriteAutoUpdate* definindo o endereço IP do computador que executa o aplicativo Servidor Placa de Controle e a ação de escrita respectivamente. A implementação é mostrada na tabela 4.23.

```
CWDataSocket1.ConnectTo para_servidor_plc, cwdsWriteAutoUpdate
```

Tabela 4.23 - Procedimento para conectar o cliente do experimento remoto ao servidor

Ler arquivo de definição do experimento

O arquivo de definição de projeto contém informações de como os diversos objetos previamente criados no Editor de Experimentos devem ser exibidos na interface gráfica do Cliente do Experimento Remoto e as propriedades que eles devem assumir. A função implementada *Carrega_Arquivo* encapsula as chamadas do objeto *CWDataSocket* utilizado para transportar arquivos do Servidor para a estação local.

Criar objetos

Após carregar o arquivo de definição de projeto na estação local a instância do objeto interpretador XML processa esse arquivo. Com as informações extraídas cria instâncias e atribui as propriedades aos objetos definidos previamente no Editor de Experimento Remoto. Nesse mesmo processo é enviado o comando para o Servidor da Placa de controle com a indicação do estado e valor inicial que os Blocos Eletrônicos devem assumir. A implementação demonstrada na tabela 4.24 exhibe o carregamento do arquivo de definição de projeto pelo objeto interpretador XML a instanciação e atribuição de propriedade ao objeto que carrega a figura com o circuito esquemático do experimento.

```
objDOMDocument.Load ("exemplo1.xml")  
Dim ToNodes As IXMLDOMNodeList  
Set ToNodes = objDOMDocument.selectNodes("//figura_fundo")  
Set onodes = objDOMDocument.selectSingleNode("//figura_fundo/nome")  
UserControl.Picture = LoadPicture(onodes.Text)
```

Tabela 4.24 - Procedimento para carga do arquivo de definição do projeto

Este processo se repete para todos os objetos contido no arquivo de definição de projeto. Ao final a interface gráfica está configurada e os objeto prontos para receber a atuação do usuário.

Enviar comandos de controle

Após a carga e processamento do arquivo de definição de projeto o Cliente está pronto para que o usuário atue com as ações necessárias ao desenvolvimento do experimento. Ao atuar em cada um dos objetos estes enviam para o Servidor Placa de Controle comandos que indicam seu estado e valor e que serão assumidos pelos Blocos Eletrônicos na Placa de Experimento. O objeto de informação Link irá abrir uma janela do navegador com a URL definida em sua propriedade quando clicado com o dispositivo apontador e o objeto rótulo ira exibir o texto definido. Como exemplo é demonstrado na tabela 4.25 a implementação para o Potenciômetro Digital.

```
a = "sms" & Chr(pot_qualpot(Index)) & Chr(Value)
CWDataSocket1.Data.Value = a
```

Tabela 4.25 - Procedimento para envio de comandos pela rede

Encerrar conexão com a rede

Ao terminar a interação para o desenvolvimento do experimento ou fechar a janela do navegador Web é encerrada a conexão com a rede Internet, liberando os recursos para que outros usuários possam se conectar. A implementação desse procedimento é demonstrada na tabela 4.26.

```
CWDataSocket2.Disconnect
```

Tabela 4.26 - Procedimento para encerrar a conexão de rede

Servidor Placa de Controle

O Servidor Placa Controle é o aplicativo executado no computador que tem instalado a Placa de Controle sendo responsável por duas funções: Estabelecer uma

conexão via rede internet para conexão ao Cliente de Experimento Remoto e iniciar a comunicação do controlador USB do computador com a placa de controle.

Com essas duas funcionalidades operando é possível que o usuário atue no experimento configurado no Cliente do Experimento remoto e os comandos enviados sejam executados na placa de controle.

A figura 4.68 demonstra a arquitetura do Servidor Placa de controle com suas funcionalidades e fluxo de dados.

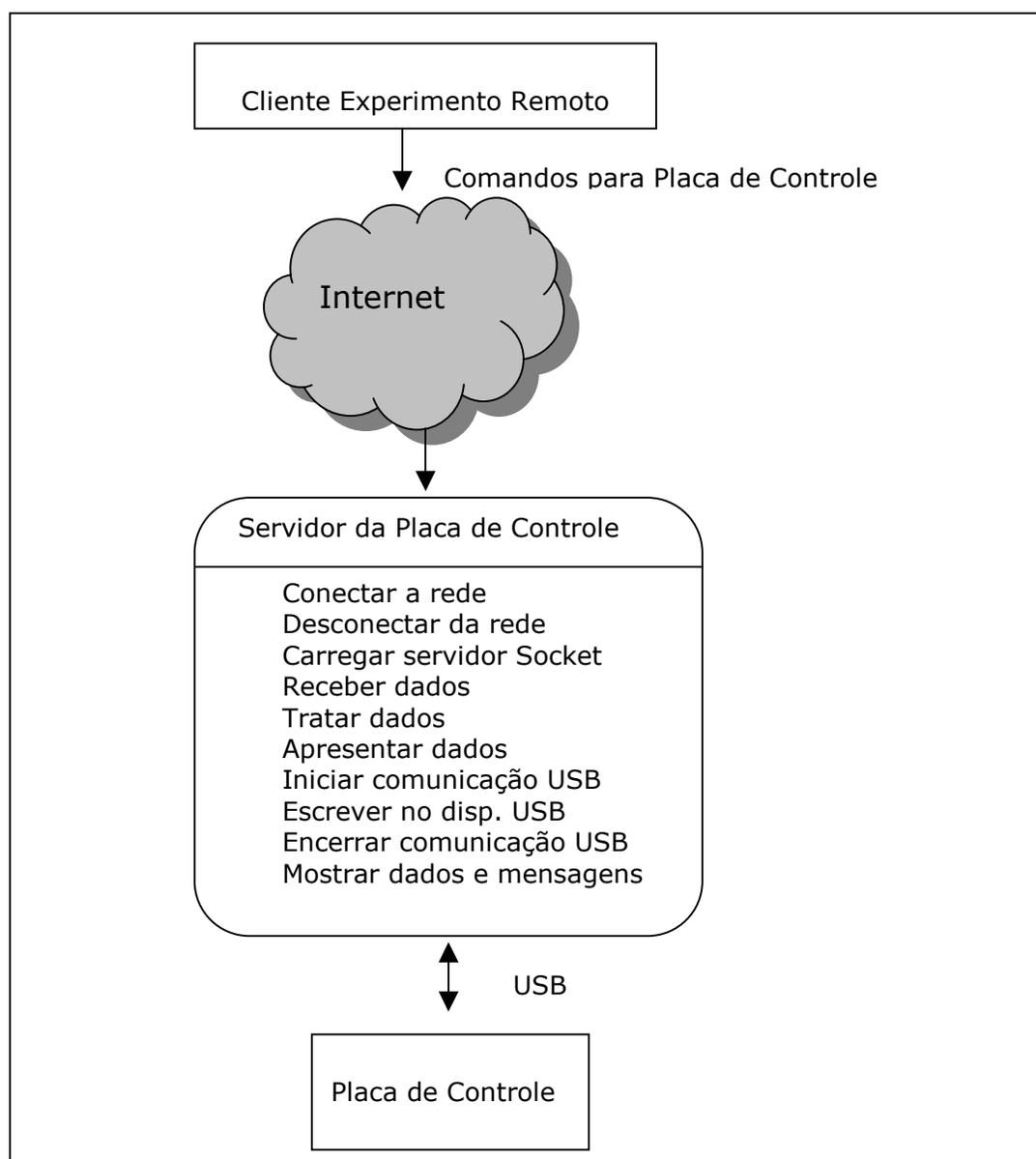


Figura 4.68 - Diagrama de funcionalidades para Servidor Placa de Controle

Por meio de uma interface gráfica (figura 4.69) o laboratorista pode visualizar os comandos que estão sendo enviados pelo usuário, as mensagens de estado do sistema e iniciar/encerrar as comunicações com a rede Internet e Placa de Controle.



Figura 4.69 - Interface Gráfica do Servidor Placa de Controle

A seguir é apresentada a implementação das funcionalidades desenvolvidas para o aplicativo Servidor de Experimento.

Conectar a rede

Para comunicação com a rede é utilizado o objeto do Labview cliente Datasocket. Este objeto dispõe de conjunto de propriedades, eventos e métodos que permitem estabelecer a conexão com o servidor de datasocket do Labview, para leitura e escrita de dados. Utiliza o esquema de URL para especificar as fontes e destino de dados.

Inicia-se a comunicação com a rede com a instrução *CWDataSocket1.ConnectTo para_servidor, cwdsReadAutoUpdate*, onde *CWDataSocket1* é o nome do objeto, *ConnectTo* é o método para conexão a rede, *para_servidor* é a variável que contém o endereço IP do computador utilizado como servidor e *cwdsReadAutoUpdate* é uma constante do sistema que sinaliza que o objeto irá realizar uma leitura sempre que um novo dado estiver disponível.

Desconectar da rede

Após a utilização das atividades a comunicação com de rede deve ser finalizada. Esta ação ocorre quando é chamado o método *Disconnect* do objeto *Datasocket* (Tabela 4.27).

```
CWDataSocket1.Disconnect.
```

Tabela 4.27 - Procedimento para desconectar da rede

Carregar servidor

O servidor de Datasocket é um aplicativo que faz parte do Labview. Ele habilita a troca de dados entre múltiplas aplicações que façam uso do objeto cliente Datasocket. O servidor recebe e armazena informações das fontes de dados e encaminha esses dados para outros destinos. A interface gráfica do Servidor da Placa de Controle dispõe de um botão que quando clicado ativa o servidor de Datasocket. Para habilitar essa funcionalidade é preciso criar um objeto que referencie o servidor e a seguir instanciar uma referência ao servidor. Este procedimento é demonstrado na tabela 4.28.

```
Dim cwdss As CWDataServer.CWDataServer  
Set cwdss = New CWDataServer.CWDataServer
```

Tabela 4.28 - Procedimento para carregar servidor

Receber dados

O objeto datasocket configurado para leitura e iniciado, ao receber um dado gera um evento e passa uma referência ao objeto *DataSocket CWData* que disponibiliza os dados recebidos. A Tabela 4.29 exibe o código que demonstra a implementação do recebimento de dados pelo cliente de datasocket.

```
Private Sub CWDataSocket1_OnDataUpdated(ByVal Data As CWDSLlib.CWData)  
Text1.text = Data  
End Sub
```

Tabela 4.29 - Procedimento para recepção de dados

Tratar dados

Após obter os dados disponíveis esta funcionalidade verifica a consistência das informações, separa o cabeçalho, o comando para USB e os dados do comando. Essa funcionalidade é implementada com as funções nativas da ferramenta de programação para operação com cadeia de caracteres.

Apresentar dados

Após os dados terem sido recebidos e tratados eles são apresentados na interface gráfica. Esta permite ao laboratorista acompanhar as atividades que estão sendo desenvolvidas pelo usuário remoto.

Iniciar comunicação USB

Para enviar os dados recebidos contendo os comandos e dados associados ao hardware da Placa de Controle existem duas possibilidades oferecidas pelo fabricante do controlador de USB FT245 utilizado no projeto. Uma é instalar um driver que emula uma porta de comunicação serial rs232. Do ponto de vista do programador a implementação é facilitada, pois se usam as mesmas ferramentas e procedimentos comuns para se desenvolver aplicativos que se comunicam com a porta serial rs232. A desvantagem dessa opção é que a velocidade máxima fica condicionada ao padrão da interface serial, que em termos práticos é operacional até 115200bps.

A segunda opção de implementação é utilizando o driver de dispositivo FT2XX também fornecido pelo fabricante do controlador USB.

A arquitetura desse driver consiste de um driver para Windows WDM que se comunica com o dispositivo por meio da pilha de USB do sistema operacional e com uma DLL que atua como interface entre o aplicativo do usuário e o driver WDM (Figura 4.70).

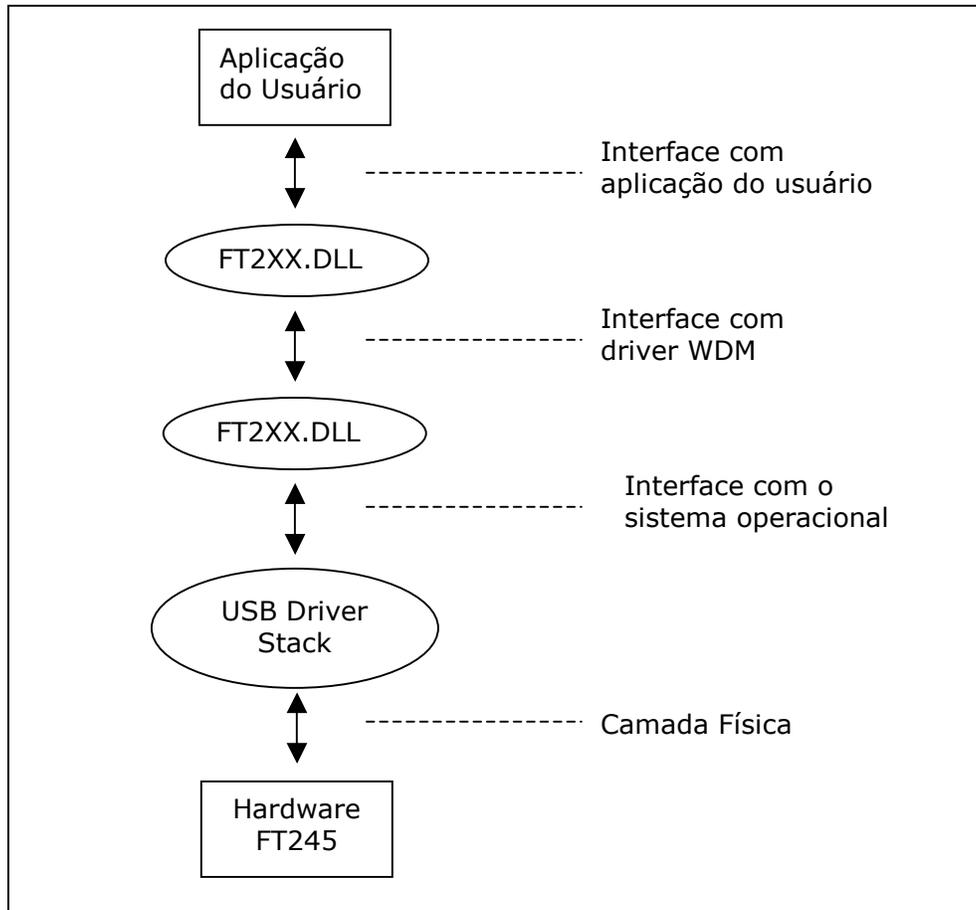


Figura 4.70 - Comunicação com o hardware USB

Para que a Placa de controle se torne acessível é preciso abrir a comunicação com a mesma. Para tanto na interface gráfica do Servidor Placa de controle existe o botão que aciona as funções para estabelecer a comunicação com o hardware USB (tabela 4.30)

```

If FT_Open(0, lnghandle) <> FT_OK Then
  Label1.Caption = "Falha ao conectar"
Else
  Label6.Caption = "FTDI Conectado"
End If
  
```

Tabela 4.30 - Procedimento para iniciar comunicação USB

A função *FT_Open* inicia a comunicação e retorna um handle que será usado por todas as funções de acesso ao dispositivo.

Escrever no dispositivo USB

Com a Placa de Controle corretamente iniciada podem ser iniciadas as operações de escrita e leitura. A função utilizada é a *FT_Write*, conforme mostrado na tabela 4.31.

```
If FT_Write(Inghandle, Data, tamanho, lngBytesWritten) <> FT_OK Then
    Label6.Caption = "Falha ao Escrever"
Else
    Label6.Caption = "Sucesso ao Escrever no FTDI"
End If
```

Tabela 4.31 - Procedimento para escrever no hardware USB

O parâmetro *Inghandle* deve ser o handle retornado pela função *FT_Open*, *Data* são os dados que serão enviados para a Placa de Controle, *tamanho* indica a quantidade de caracteres que serão enviados e é usado para comparar com o valor que é retornado pelo parâmetro *lngBytesWritten*. O teste de condição realizado com a constante *FT_OK* permite checar se o comando foi executado corretamente.

Exibir dados e mensagens de estado

O Servidor da Placa de Controle opera com dados que chegam da rede Internet e são encaminhados para a porta USB. Durante esse processo várias ações ocorrem e a falha em algum dos componentes envolvidos como rede internet sem comunicação, problemas no hardware, pode comprometer o desenvolvimento da atividade. Para que se possa detectar com facilidade qual dos componentes apresentou alguma falha o Servidor Placa de controle gera mensagens para todas as ações que ocorrem e as exibe na interface gráfica.

Encerrar comunicação USB e o Datasocket

Ao encerrar o Servidor da Placa de Controle é importante que seja encerrada a comunicação com o hardware da Placa de Controle e liberados os recursos de software utilizados pelo driver USB. Para tanto deve ser usada a função *FT_Close*. A conexão com a rede Internet deve ser fechada com o método do cliente datasocket *Disconnect* (tabela 4.32)

```
'fecha a conexão do datasocket  
CWDataSocket1.Disconnect  
  
'finaliza placa de controle  
If FT_Close(Inghandle) <> FT_OK Then  
  Label6.Caption = "Falha ao desconectar"  
Else  
  Label6.Caption = "FTDI desconectado"  
End If  
End
```

Tabela 4.32 - Procedimento para encerrar a comunicação com o hardware USB

Servidor GPIB

Este é o aplicativo instalado no computador configurado com o hardware GPIB constituído pela placa controladora GPIB e do osciloscópio HP 54503A. As funcionalidades básicas do Servidor GPIB são a comunicação com o hardware GPIB e comunicação via rede Internet com o Cliente GPIB para o recebimento de comandos de controle do osciloscópio e envio dos dados capturados no experimento. Para atender essas funcionalidades foi implementado um conjunto de funções que são descritas a seguir. A figura 4.71 demonstra a arquitetura do Servidor GPIB e seu fluxo de dados.

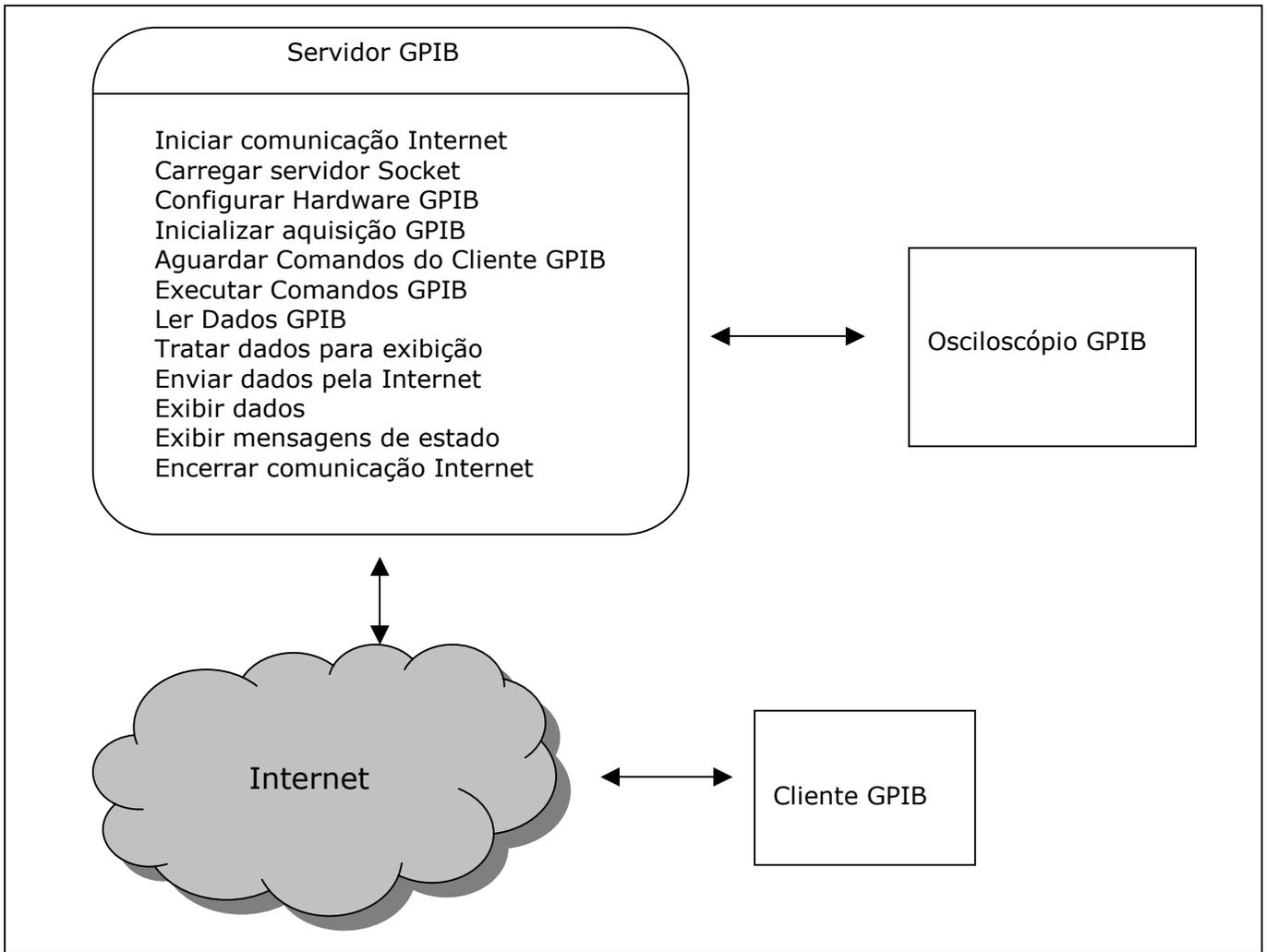


Figura 4.71 - Arquitetura do Servidor GPIB

Iniciar comunicação Internet

O Servidor GPIB estabelece comunicação com a rede para o envio dos dados que serão apresentados no Cliente GPIB, para receber os dados com comandos de controle do osciloscópio é estabelecida uma conexão para leitura com o cliente GPIB. Uma conexão de leitura é criada para receber as informações originadas no Cliente Experimento remoto com os dados dos pontos de medida do osciloscópio (tabela 4.33).

```
CWDataSocket1.ConnectTo ip_servidor, cwdsReadAutoUpdate
CWDataSocket3.ConnectTo ip_servidor, cwdsReadAutoUpdate
CWDataSocket2.ConnectTo ip_servidor, cwdsWriteAutoUpdate
```

Tabela 4.33 - Procedimento para conexão à rede

O método *ConnectTo* do objeto *datasocket* conecta-se ao servidor pelo endereço indicado na variável *ip_servidor*, as constantes *cwdsReadAutoUpdate* e *cwdsWriteAutoUpdate* indicam as operações de leitura e escrita respectivamente.

Carregar servidor Socket

Do mesmo modo que o Servidor da Placa de Controle, o Servidor GPIB disponibiliza na interface um botão que ao ser clicado ativa o servidor de *Datasocket*. Para habilitar essa funcionalidade é preciso criar um objeto que faz referência ao servidor e a seguir instanciar o objeto. Este procedimento é exibido na tabela 4.34.

```
Dim cwdss As CWDataServer.CWDataServer
Set cwdss = New CWDataServer.CWDataServer
```

Tabela 4.34 - Procedimento para iniciar servidor de *datasocket*

Configurar Hardware GPIB

Antes de iniciar a comunicação com o instrumento GPIB é preciso especificar no aplicativo o índice de acesso ao hardware GPIB, o endereço primário e secundário. Para realizar essas ações o objeto GPIB do Labview dispõe das propriedades *BoardNumber*, *PrimaryAddress* e *SecondaryAddress*. Após especificar as propriedades é necessário invocar o método *Configure* para que as configurações sejam assumidas. O código da tabela 4.35 demonstra a implementação da configuração GPIB.

```
CWGPIB1.BoardNumber = "0"
CWGPIB1.PrimaryAddress = "7"
CWGPIB1.SecondaryAddress = "0"
CWGPIB1.Configure
```

Tabela 4.35 - Procedimento para configurar comunicação com hardware GPIB

Inicializar aquisição GPIB

Para inicializar a aquisição de dados do instrumento GPIB é preciso configurar parâmetros nos subsistemas *ACQUIRE* e *WAVEFORM* que afetam o comando *DIGITIZE*.

O subsistema ACQUIRE configura o tipo de dado, o número de amostras que serão realizadas e a quantidade de pontos que serão capturados em uma operação de leitura.

O subsistema WAVEFORM é utilizado para transferir as leituras capturadas no instrumento. Um registro WAVEFORM é composto de duas partes, os dados da leitura e de um cabeçalho. Os dados da leitura contêm os pontos capturados na fonte especificada. O cabeçalho contém as informações para interpretação dos dados capturados, o que inclui o número de pontos capturados, o tipo do dado, formato, os incrementos dos eixos X e Y, origens e referências.

Com essas informações é possível converter os pontos capturados em valores de tensão e tempo. O código da tabela 4.36 configura os subsistemas ACQUIRE e WAVEFORM do instrumento GPIB.

```
CWGPIB1.Write ":ACQUIRE:TYPE AVERAGE"  
CWGPIB1.Write ":ACQUIRE:COMPLETE 100"  
CWGPIB1.Write ":WAVEFORM:FORMAT ASCII"  
CWGPIB1.Write ":ACQUIRE:COUNT 4"  
CWGPIB1.Write ":ACQUIRE:POINTS 150"
```

Tabela 4.36 - Configuração da aquisição de dados do hardware GPIB

Com essa configuração é selecionado o modo amostragem de 150 pontos, formato de dados ASCII e o número de amostra será quatro. Com essa configuração um comando DIGITIZE fará o subsistema WAVEFORM armazenar os dados após completar quatro amostragens de 150 pontos.

Ler Dados GPIB

Para efetuar a leitura dos dados capturados pelo osciloscópio é necessário inicialmente selecionar por meio do subsistema WAVEFORM a fonte do sinal, a seguir enviar um comando indicando que o sinal seja gravado na memória do instrumento, finalmente invocando o método de leitura do objeto GPIB se obtém os dados que são armazenados em uma matriz. O código da tabela 4.37 demonstra a implementação desse processo.

```
CWGPIB1.Write ":WAVEFORM:SOURCE CHANNEL1  
CWGPIB1.Write ":DIGITIZE CHANNEL1"  
Text1.Text = CWGPIB1.Read
```

Tabela 4.37 - Procedimento para aquisição de dados GPIB

A leitura e apresentação dos dados é realizada de forma contínua possibilitando ao usuário visualizar o sinal e suas variações no momento que ocorrem.

Executar Comandos GPIB

Os comandos GPIB podem ser executados por meio dos controles na interface gráfica do Servidor GPIB ou executados remotamente com os comandos recebidos do Cliente GPIB.

Cada botão ou controle da interface gráfica ao ser alterado gera um evento, que tem associado uma rotina que envia um comando de escrita para o instrumento GPIB. O Objeto do Labview *CWGPIB* ao invocar o método Write com a cadeia de caracteres que tem o comando GPIB como exemplificado no código da tabela 4.38 que seleciona a base de tempo do osciloscópio para 1 ms.

```
CWGPIB1.Write ":TIMEBASE:RANGE .01"
```

Tabela 4.38 - Procedimento para seleção da base de tempo

III. A lista com todos os recursos GPIB implementados no Servidor GPIB está no anexo

Tratar dados para exibição

Com a configuração especificada nos subsistemas ACQUIRE e WAVEFORM os dados enviados pelo osciloscópio contendo as informações adquiridas são do tipo inteiro variando de 0 a 32768. Estes dados devem ser convertidos no domínio do tempo e tensão para uma apresentação e visualização adequada.

Os valores usados para interpretar os dados são as referências, as origens e incrementos dos eixos X e Y. Estes valores são obtidos no cabeçalho do subsistema

WAVEFORM. A implementação (tabela 4.39) demonstra a aquisição desses dados para conversão dos níveis de voltagem.

```
CWGPIB1.Write ":WAVEFORM:YREFerence?"  
yref = Val(CWGPIB1.Read)  
CWGPIB1.Write ":WAVEFORM:YINCrement?"  
yinc = Val(CWGPIB1.Read)  
CWGPIB1.Write ":WAVEFORM:YORigin?"  
yori = Val(CWGPIB1.Read)
```

Tabela 4.39 - Obtenção do dados para ajuste da exibição

Para converter os dados capturados em valores de tensão é necessário aplicar uma fórmula para cada ponto (tabela 4.40).

$$\text{Tensão} = [(\text{dado_capturado} - \text{referênciaY}) * \text{incrementoY}] + \text{origemY}$$

Tabela 4.40 - Fórmula para ajustes dos dados exibidos

Os dados recebidos com os pontos capturados estão no formato ASCII e utilizam como delimitador o caractere “;”. Para que a manipulação matemática desses dados seja possível é preciso extrair os valores convertendo para o tipo inteiro e armazená-los em uma matriz. O objeto do Labview *CWSerial* permite a criação de objetos *Tasks* que podem executar a tarefa de decodificar os dados contidos em uma cadeia de caracteres extraíndo os dados relevantes diretamente para uma matriz de dados. Esse recurso é utilizado para obter os pontos capturados pelo osciloscópio.

O código da tabela 4.41 foi implementado para extrair os valores inteiros dos dados capturados e atribuí-los a uma matriz que a seguir é convertida em valores de tensão aplicando a fórmula demonstrada anteriormente.

```

Dados_capturados = CWGPIB1.Read
dados = CWSerial1.Tasks("ldado").Parse(Dados_capturados)
For f = 1 To 127
dados1(f) = ((dados(f) - yref) * yinc) + yori + v_pos_traco
Next

```

Tabela 4.41 - Conversão dos dados para exibição

Aguardar Comandos do Cliente GPIB

Após a inicialização do objeto *CWDataSocket* com o parâmetro *cwdsReadAutoUpdate* este fica lendo a conexão e ao receber um dado do Cliente GPIB gera um evento. A este evento está associada uma rotina que realiza a leitura dos dados. Cada pacote de dado recebido contém um cabeçalho de três caracteres que indica o comando mais os caracteres com o valor que deve ser passado como parâmetro ao comando GPIB. Após a interpretação de qual comando deve ser executado o método *Write* do objeto *CWGPIB* é invocado encaminhando o comando e o valor para o instrumento GPIB.

Enviar dados pela Internet

A matriz de dados com o sinal capturado pelo osciloscópio após tratamento para conversão em nível de tensão é enviada ao Cliente GPIB por meio da rede Internet. Como são capturados dois canais para que possam ser diferenciados pelo Cliente GPIB é adicionado um cabeçalho de três caracteres a matriz de dados.

A comunicação com a rede Internet se dá por meio do objeto *CWDataSocket* configurado para escrita com o parâmetro *cwdsWriteAutoUpdate*. O código da tabela 4.42 mostra a implementação para o envio dos dados do canal 2 do osciloscópio.

```

CWDataSocket2.data.Value = "ch2"
CWDataSocket2.data.Value = dados

```

Tabela 4.42 - Envio dos dados via rede Internet

Exibir dados

A matriz de dados com o sinal capturado pelo osciloscópio após ser enviado para o Cliente GPIB é exibido como gráfico na tela por meio do objeto do Labview *CWGraph*. Como é necessário exibir os sinais de dois canais a propriedade *Plots* indica a coleção de

CwPlots que será exibida por meio do método *PlotY*. A implementação demonstrada na tabela 4.43 exibe o sinal capturado pelo canal 2 do osciloscópio.

CWGraph1.Plots(2).PlotY dados

Tabela 4.43 - Exibição dos dados

Para diferenciar a visualização de um canal do outro são usadas cores diferentes para apresentação.

A interface gráfica dispõe de dois controles deslizantes do tipo *CWSlide*. Atuando nesses controles varia-se o seu valor numérico que multiplicado pela variável *v_pos_traco* utilizada na operação de transformação dos dados em nível de tensão, faz com que o dados exibidos no objeto *CWGraph* seja deslocado no eixo Y.

Exibir mensagens de estado

A operação do Servidor GPIB envolve a comunicação com o hardware GPIB, comunicação para escrita e leitura na rede Internet e alteração dos vários parâmetros do instrumento. Para permitir o acompanhamento do estado dos diversos processos que ocorrem de forma concorrente e em caso de alguma falha facilitar o diagnóstico os diversos eventos gerados pelos processos são monitorados e mensagens de textos são apresentadas.

As ações do usuário para controle do osciloscópio quer sejam executadas localmente ou remotamente no Cliente GPIB são exibidas de forma semelhante a que ocorre no instrumento real. Estas informações incluem a base de tempo de cada canal, a tensão, nível de trigger e borda de disparo (subida ou descida). A posição dos controles do tipo botão e deslizante também permitem visualizar a configuração do momento no osciloscópio.

Encerrar comunicação Internet

Após finalizar as atividades é necessário encerrar as conexões de escrita e leitura com a rede e com o Cliente GPIB. Para essa implementação é utilizado o método *Disconnect* do objeto *CWDataSocket* conforme mostrado na tabela 4.44.

CWDataSocket1.Disconnect

Tabela 4.44 - Procedimento para encerrar comunicação com a internet

Cliente Gpib

O Cliente GPIB é um applet executado no navegador Web do computador do cliente que se comunica com o Servidor GPIB. Disposto de uma interface gráfica que reproduz as funcionalidades de um osciloscópio real possibilita ao usuário visualizar os sinais do experimento e envio de comandos que alteram parâmetros como base de tempo, nível de tensão, trigger, entre outros (figura 4.72).

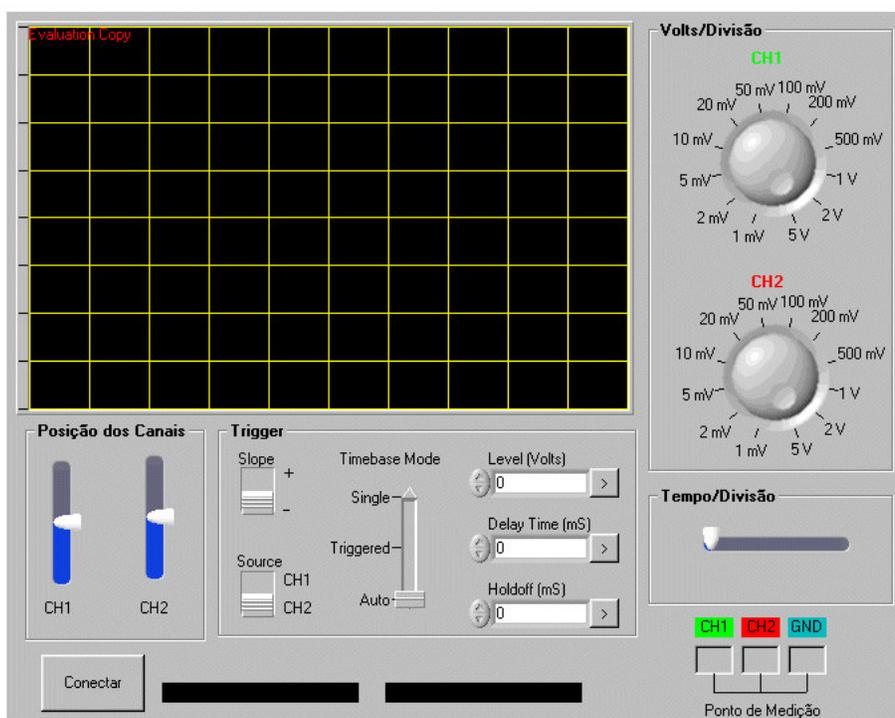


Figura 4.72 - Interface gráfica do Cliente GPIB

O applet foi desenvolvido com o modelo dcom, ao acessar o arquivo html este contém as informações de localização e a versão atual que será carregada e executada no navegador.

A figura 4.73 demonstra a arquitetura do Cliente GPIB e o fluxo de dados entre os módulos externos.

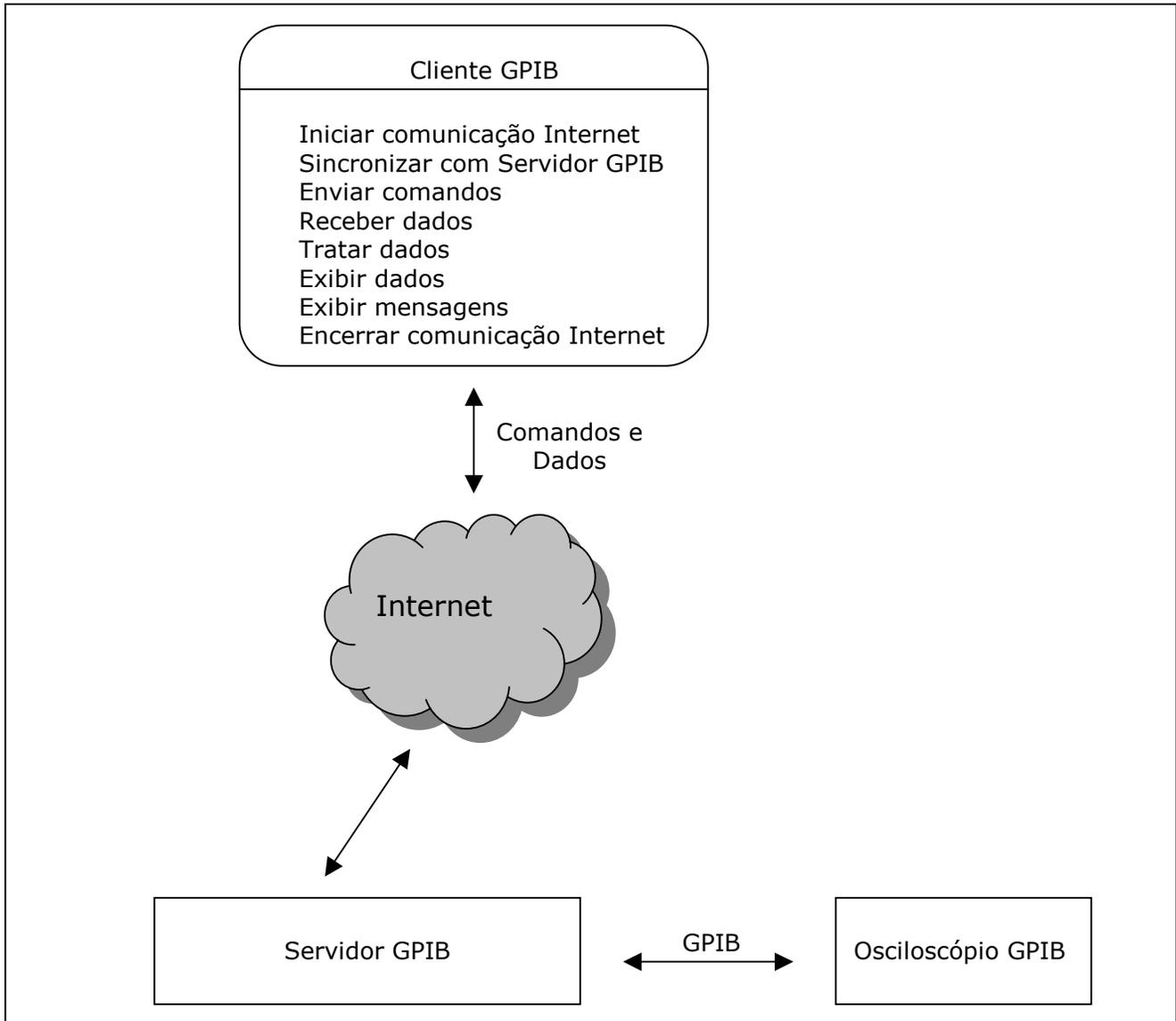


Figura 4.73 - Diagrama das funcionalidades do Cliente GPIB

A seguir é apresentada a implementação das funcionalidades desenvolvidas para o aplicativo.

Iniciar comunicação Internet

A comunicação como o servidor GPIB ocorre para o envio dos comandos de controle e o recebimento dos dados que serão apresentados graficamente, portanto são criadas duas instâncias do objeto datsocket, uma para leitura e outra para escrita (Tabela 4.45).

```
CWDataSocket1.ConnectTo ip_servidor, cwdsReadAutoUpdate  
CWDataSocket2.ConnectTo ip_servidor, cwdsWriteAutoUpdate
```

Tabela 4.45 - Procedimento para iniciar a comunicação com a internet

O método *ConnectTo* do objeto *datasocket* conecta-se ao servidor pelo endereço indicado na variável *ip_servidor*, as constantes *cwdsReadAutoUpdate* e *cwdsWriteAutoUpdate* indicam as operações de leitura e escrita respectivamente.

Sincronizar com Servidor GPIB

Ao ser carregado o applet para execução de uma atividade, existe a possibilidade do osciloscópio ter sido utilizado anteriormente e estar ajustado com um conjunto de parâmetros. Para que a interface gráfica seja sincronizada com a configuração do instrumento é enviada uma mensagem ao servidor GPIB que a interpreta enviando conjunto de comandos ao mesmo. Ao fim desse processo a interface gráfica do Cliente GPIB corresponde às configurações do osciloscópio.

Enviar comandos

A interação do usuário para o envio de comandos se dá por meio de objetos do Labview que representam graficamente botões, deslizantes e chave liga/desliga. Para cada comando existe um desses objetos. Ao atuar nesses objetos é gerado um evento e o valor dele varia dentro de uma faixa previamente estabelecida.

A figura 4.74 ilustra similaridade entre o botão criado na interface gráfica para o ajuste de Volts por Divisão e a botão de um osciloscópio real para a mesma função.

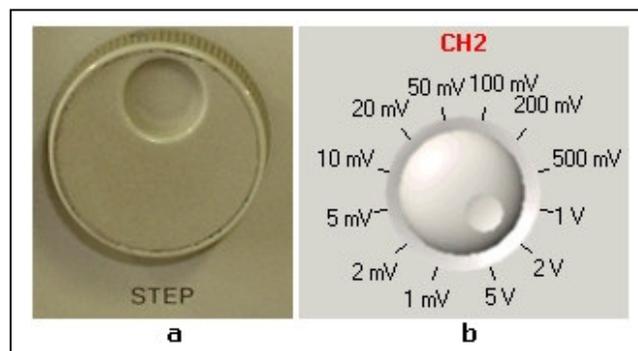


Figura 4.74- Botão de osciloscópio (a) e botão na interface gráfica (b)

Para que o Servidor GPIB possa identificar qual comando GPIB deve ser executado o Cliente GPIB envia um pacote composto de um cabeçalho de três caracteres indicador da instrução mais o valor que a mesma deve assumir (tabela 4.46)

Cabeçalho 1	Cabeçalho 2	Cabeçalho 3	Valor
-------------	-------------	-------------	-------

Tabela 4.46 - Pacote de dados do comando GPIB

O código da tabela 4.47 demonstra a implementação do envio do comando que ao ser recebido no Servidor GPIB ajusta o nível de Trigger do osciloscópio.

```
CWDataSocket2.Data.Value = "trl" & CWSlide4
```

Tabela 4.47 - Envio de comando GPIB

A cadeia de caracteres *trl* é o cabeçalho e o valor retornado pelo objeto *CWSlide4* contém o nível de trigger que deve ser ajustado no osciloscópio. O objeto *CWDataSocket2* envia esses dados pela rede Internet ao Servidor GPIB, onde será decodificado e enviado para o hardware GPIB.

O anexo III contém o conjunto comandos de GPIB implementados e os cabeçalhos identificadores associados.

Receber dados

Para reproduzir as funcionalidades de um osciloscópio é preciso exibir os sinais gerados pelo instrumento real. O Cliente GPIB recebe os dados com os sinais de dois canais que são gerados continuamente pelo Servidor GPIB. Para que os canais possam ser diferenciados são montados pacotes que contém um cabeçalho de três caracteres que identifica o canal e uma matriz de 128 bytes com os pontos que serão exibidos. Após processamento do cabeçalho que identifica o pacote como sendo o canal um ou dois a matriz com os dados é apresentada graficamente por meio do objeto Labview *CWGraph*.

Exibir dados

O objeto *CWGraph* dispõe de um conjunto de métodos e propriedades que proporcionam os recursos necessários para a exibição na tela do computador dos sinais capturados em um osciloscópio GPIB de forma bastante próxima a que é vista na tela do

instrumento real. Na interface gráfica do Cliente GPIB foram configuradas as propriedades *PlotAreaColor* que define a cor do fundo como preta, *ChartStyle* para indicar o modo de apresentação no qual os dados mais recentes são mostrados a direita e a exibição dos anteriores vai sendo deslocada para a esquerda, *MajorGrid* habilita a exibição das quadrículas.

Na figura 4.75 pode ser vista um detalhe da interface Gráfica do Cliente GPIB com o objeto CWGraph apresentando uma forma de onda do osciloscópio.

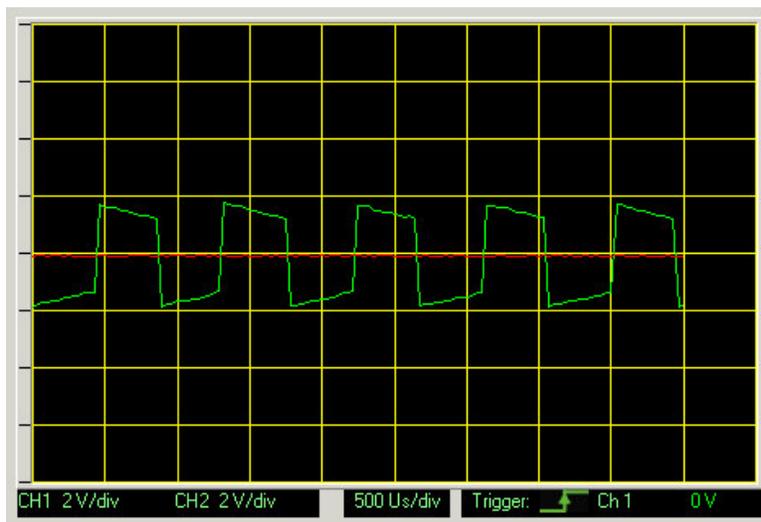


Figura 4.75 - Objeto CWGraph com forma de onda captura pelo Osciloscópio

Exibir mensagens

A fim de orientar o usuário quanto às configurações do osciloscópio, estado da conexão com a rede Internet e os pontos no circuito sob experimento que tem a ponta de prova do osciloscópio conectada são exibidas mensagens de texto com essas informações (figura 4.73).

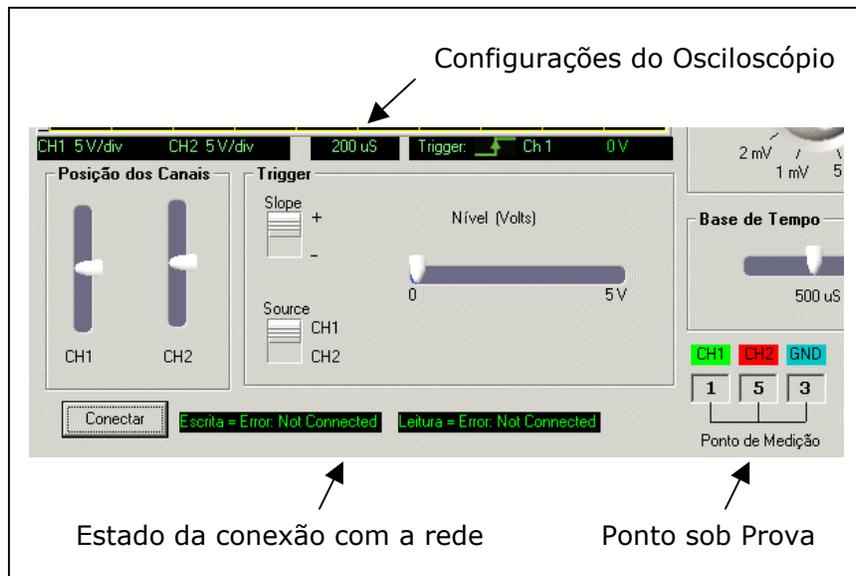


Figura 4.76 - Mensagens de estado e configuração

Abaixo da área de apresentação dos sinais encontram-se as informações sobre as configurações do osciloscópio incluindo Volts por divisão para cada um dos dois canais e a base de tempo. Para o trigger há indicações do canal usado, a borda de subida ou descida e nível de tensão selecionado.

Na parte inferior esquerda estão as mensagens de conexão com a rede, que indicam o estado da conexão ou erros relacionados a problemas na rede ou com o Servidor GPIB. Do lado direito na área inferior são indicados os pontos do circuito que tem as pontas de prova do osciloscópio conectadas. A indicação textual para os canais e referência está nas cores vermelho e verde e azul que correspondem as cores utilizadas na exibição dos sinais no objeto *CWGraph*.

Encerrar comunicação Internet

Após encerrar a atividade ou fechar a janela do navegador na qual está carregado o applet do Cliente GPIB são encerradas as conexões de escrita e leitura com a rede e com o Servidor GPIB. Para essa implementação é utilizado o método *Disconnect* do objeto *CWDataSocket* (tabela 4.48).

```
CWDataSocket1.Disconnect
CWDataSocket2.Disconnect
```

Tabela 4.48 - Fechar comunicação com a rede Internet

Capítulo 5 - Avaliação do sistema

Para avaliação do sistema quanto a funcionalidade e os objetivos propostos, foram utilizadas experiências executadas no laboratório convencional da disciplina de eletrônica industrial oferecido na graduação da Faculdade de Engenharia Elétrica da Unicamp.

Esse curso é constituído por um conjunto de experimentos nos quais os alunos têm a oportunidade de explorar várias configurações de circuitos eletrônicos, realizando medições com o osciloscópio, multímetro e outros instrumentos, e variando valores de componentes com o propósito de avaliar o comportamento do circuito. Ao final de cada experimento o aluno deve elaborar um relatório com os dados que obteve e suas conclusões.

Para demonstrar a utilização da Arquitetura desenvolvida na elaboração de atividades remotas foram selecionados dois experimentos, o experimento 1 que trata da polarização de transistores bipolares e o experimento 4 no qual é realizada a caracterização de dispositivos semicondutores de potência rápidos como diodo, transistor bipolar, MOSFET e IGBT.

O desenvolvimento do experimento 1 tem como objetivo demonstrar os passos para elaboração da atividade utilizando o editor de experimentos e sua integração com os Blocos Eletrônicos e de Informação.

A elaboração do experimento 4 envolveu o processo de implementação do circuito eletrônico a ser analisado na Placa de Experimento, a elaboração do projeto no Editor de Experimentos e a publicação do mesmo para acesso remoto.

No desenvolvimento do experimento o professor com conhecimento no conteúdo pedagógico a ser explorado irá utilizar o Editor de Experimentos, Blocos Eletrônicos e demais recursos da arquitetura para elaborar o experimento e disponibilizá-lo para acesso remoto.

Espera-se que o aluno ao acessar o Cliente do Experimento Remoto tenha a oportunidade de realizar as atividades de controle e obtenção dos dados de forma semelhante a realizada no modo presencial em que ele atua diretamente no circuito eletrônico.

Experimento 1

Nesse experimento o aluno tem como objetivo projetar uma estrutura de polarização com amplificador a transistor, de modo que, estabelecido um dado ponto de operação, a estrutura mantenha esse ponto, a despeito de variações nos parâmetros do transistor. O ponto de operação do transistor, também denominado ponto de trabalho ou quiescente, deve ser escolhido a fim de que satisfaça as restrições impostas pelas limitações físicas do dispositivo.

Para a escolha do ponto de operação vários critérios devem ser adotados, alguns tendo maior importância, em função da condição de trabalho desejada. Alguns desses critérios são:

- Consumo do circuito de polarização deve ser baixo,
- Linearidade de operação (se for o caso),
- Resposta em frequência,
- Deriva térmica,
- Excursão de tensão e correntes necessárias,
- Imunidade a ruídos.

Um projeto com boa linearidade, máxima excursão de tensão de saída, baixa deriva e alta imunidade a ruído, apresenta dificuldades para sua resolução.

Ao desenvolver essa atividade, o aluno por meio da variação dos componentes, troca de transistores, e realização de medições deve procurar uma solução de compromisso que satisfaça o maior número de critérios e em melhores condições.

As atividades realizadas em laboratório analisam as configurações polarização fixa (figura 5.77), polarização com resistor no emissor (figura 5.78) e polarização com divisor de tensão (figura 5.79).

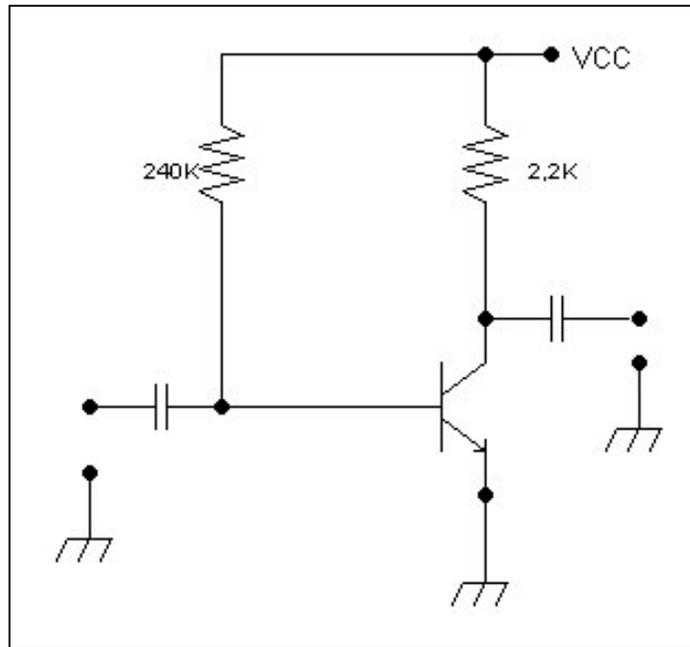


Figura 5.77 - Transistor com polarização fixa

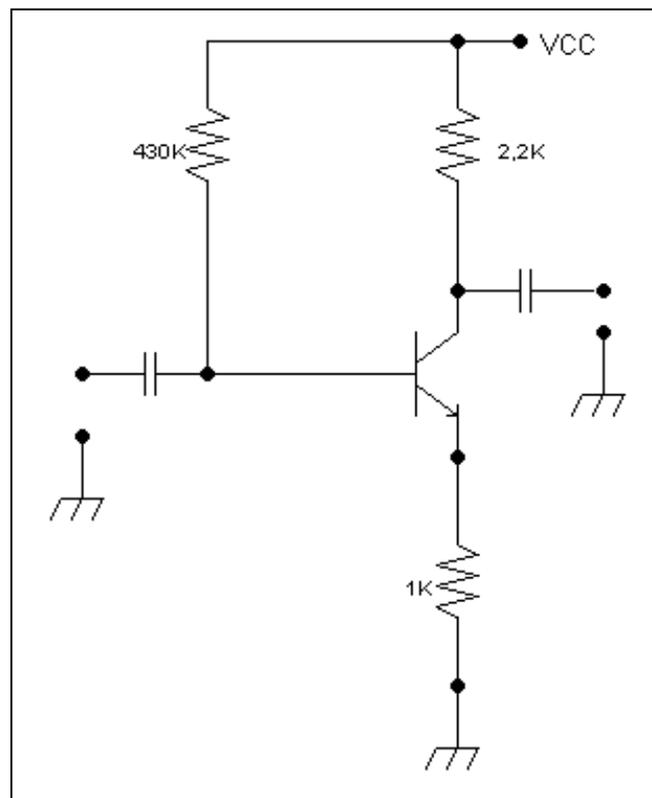


Figura 5.78 - Transistor polarizado com resistor no emissor

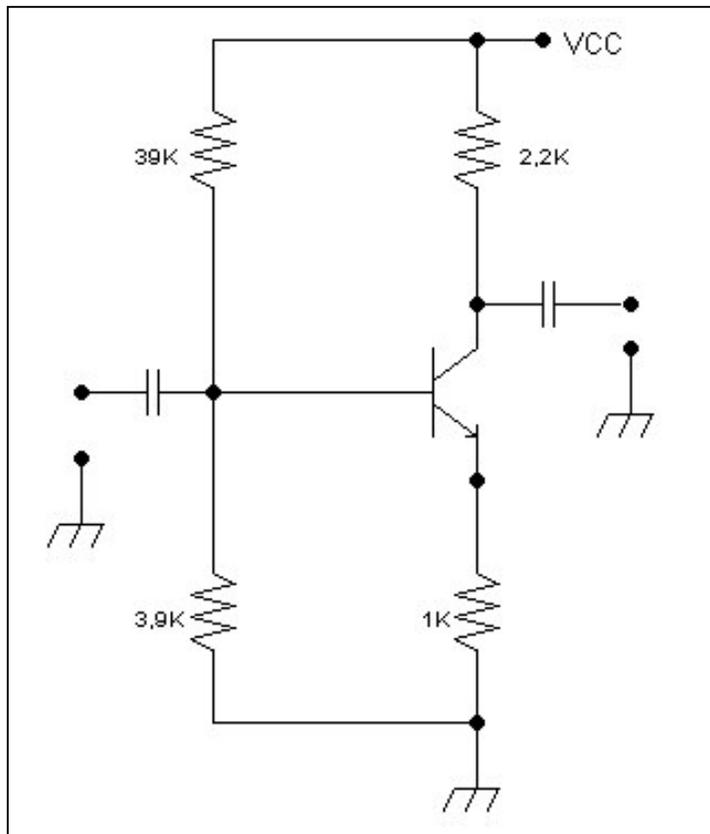


Figura 5.79 - Transistor polarizado com divisor de tensão

Com base nessas configurações o aluno deve realizar medições, trocar o transistor e comparar os valores obtidos pelas medições com os obtidos pela análise teórica do circuito.

Para implementação desse experimento contendo as três configurações de amplificador a transistor foi utilizado o Bloco Eletrônico Chave, Ponto de Medição e os objetos de informação Rótulo e Hyperlink. Por meio das propriedades que podem ser atribuídas no Editor de Experimentos ao Bloco Eletrônico Chave, como valor inicial que pode ser ligado ou desligado, e visível ou não na interface gráfica pode-se elaborar um circuito eletrônico na Placa de Experimento (figura 5.80) que pela combinação do estado das chaves permite obter os três circuitos que serão estudados. Cada chave no circuito representa um Bloco Eletrônico Chave.

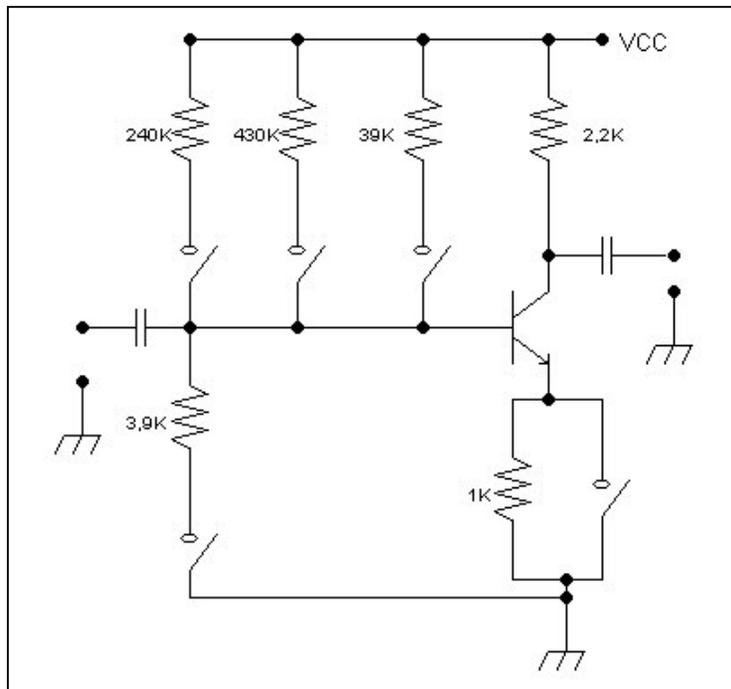


Figura 5.80 - Circuito para Experimento 1

Para disponibilizar as três configurações de polarização são elaborados três projetos nos quais varia-se o estado inicial do Bloco Eletrônico Chave ao ser carregado o projeto no Cliente do Experimento Remoto. Para medição são associados Blocos Pontos de Medição na base, coletor e emissor do transistor. Com a utilização de um Bloco Chave de dois pólos e duas posições associado ao transistor é possível trocar o mesmo durante a execução do experimento. Um Objeto Rótulo fornece informações adicionais. São usados dois Objetos Hyperlink, um deles remete ao site do fabricante do transistor permitindo acessar o arquivo com as especificações do dispositivo, o outro permite acessar um documento preparado pelo professor.

A figura 5.81 mostra como deve ficar o circuito na Placa de Experimento ao ser carregado o projeto com a configuração polarização fixa.

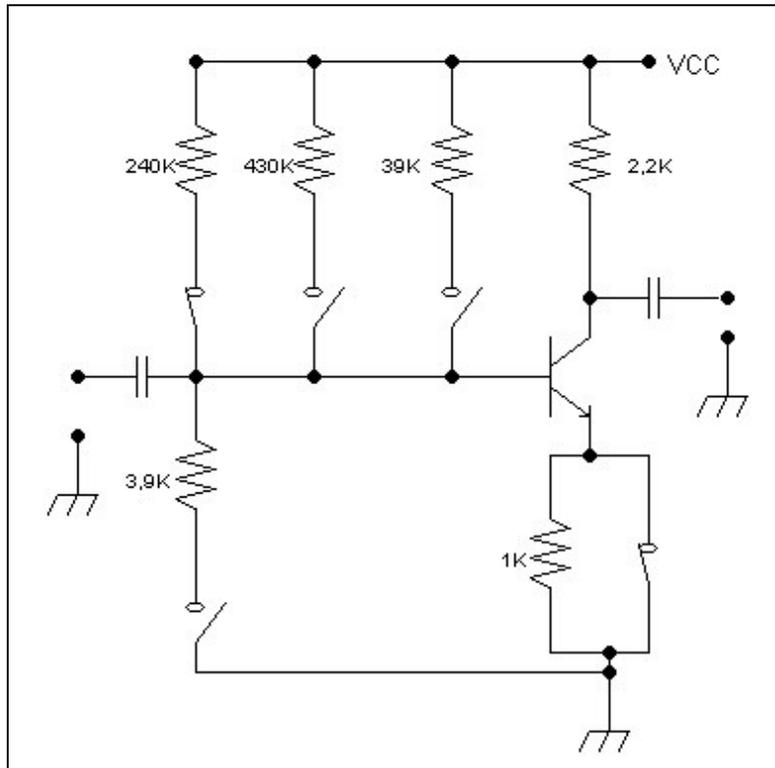


Figura 5.81 - Placa de Experimento para Experimento 1

A seguir serão descritos os passos necessários para elaborar o projeto. Do ponto de vista de hardware é montado na Placa de Experimento o circuito eletrônico com os Blocos Eletrônicos Chaves, Pontos de Medição e demais componentes. A tabela 5.49 mostra a atribuição dos Blocos Eletrônicos relacionados aos componentes do circuito.

Bloco Eletrônico	Componente associado
Chave 1	Emissor Transistor
Chave 2	Resistor 39K
Chave 3	Resistor 430K
Chave 4	Resistor 240K
Chave 5	Resistor 3,9K
Ponto de Medição 1	Base do transistor
Ponto de Medição 2	Coletor do transistor
Ponto de Medição 3	Emissor do transistor
Ponto de Medição 4	Referência
Chave 6	Coletor e base do transistor

Tabela 5.49 - Blocos eletrônicos do experimento 1

Após montar o circuito eletrônico deve ser executado o aplicativo Editor de Experimentos e iniciado um novo projeto.

Na barra de ferramentas deve-se clicar no botão figura para selecionar o arquivo de imagem com o circuito eletrônico. Nesse exemplo foi usado o esquemático do circuito por polarização fixa (figura 5.82).

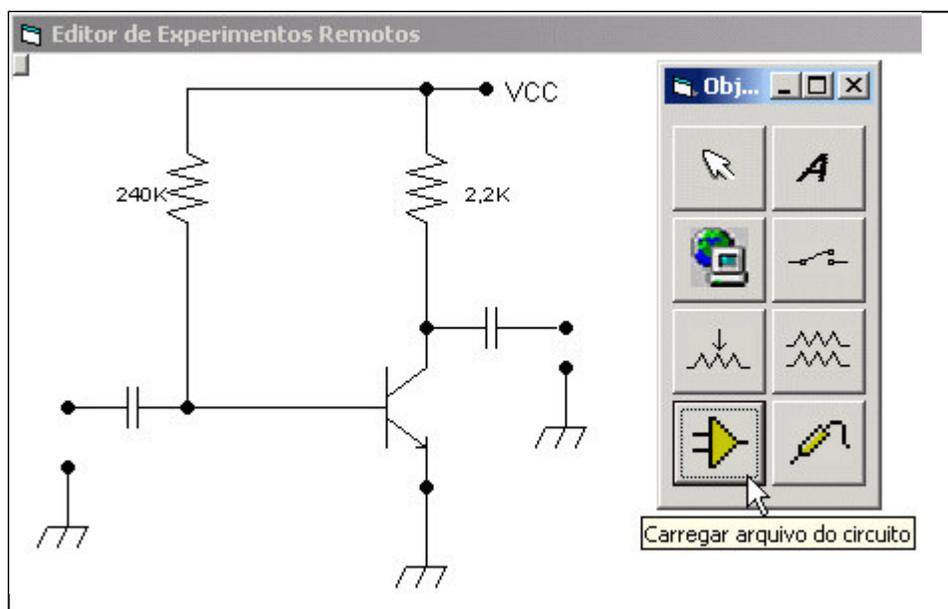


Figura 5.82 - Inserir circuito no editor de experimentos

Em seguida devem ser inseridos seis Blocos Eletrônicos Chave. Apenas a chave seis será visível ao usuário para permitir a troca do transistor e, portanto deverá ser posicionada de forma adequada, as outras chaves responsáveis por configurar o circuito não serão visíveis para o usuário podendo ser dispostas sem preocupação com a posição na área de trabalho (figura 5.83).

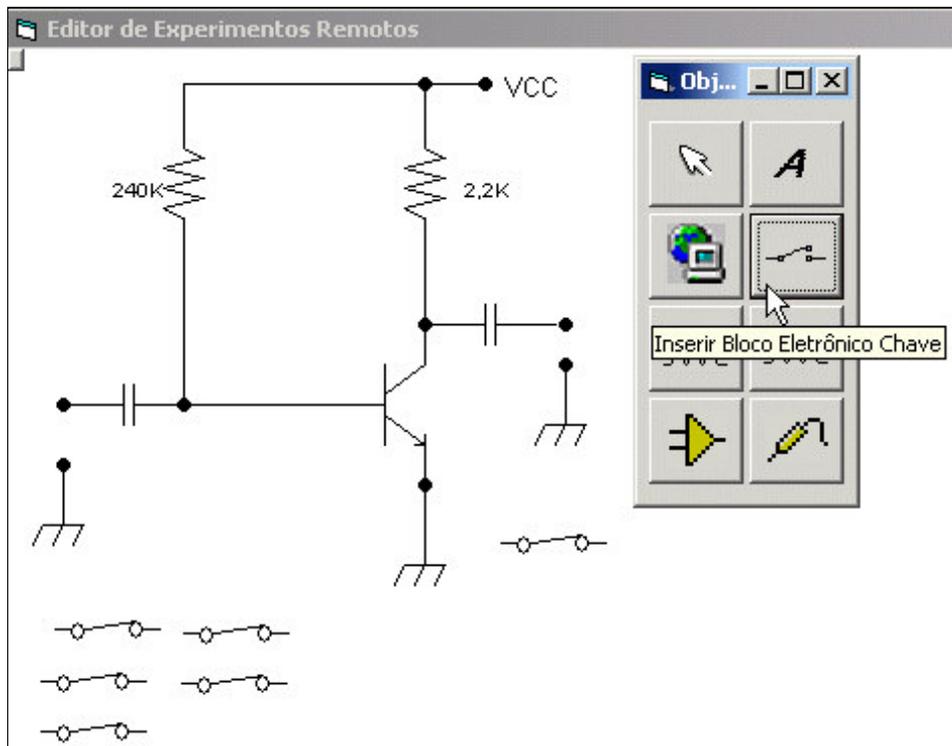


Figura 5.83 - Inserir Bloco Eletrônico Chave no Editor de Experimentos

Cada uma das chaves deve ter suas propriedades identificação, visibilidade, condição inicial (ligada ou desligada) configurada. Para alterar as propriedades clica-se com o botão direito do dispositivo apontador no desenho que representa a chave e na caixa de propriedades que abrir altera-se essas propriedades. No circuito exemplo a chave 1 deverá estar fechada e invisível (figura 5.84).

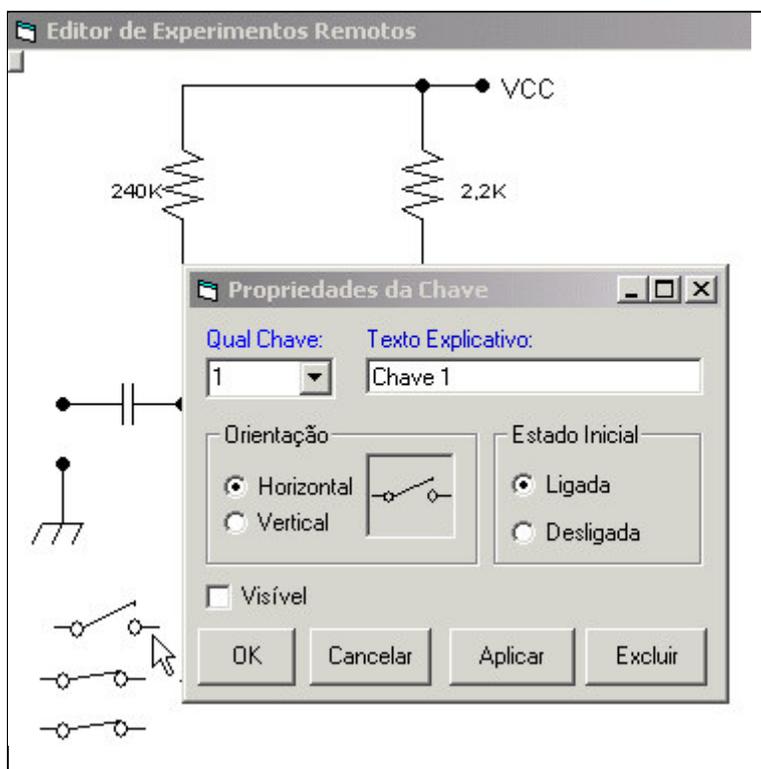


Figura 5.84 - Configurar propriedades do Bloco Eletrônico Chave

O mesmo procedimento deve ser repetido para todos os Blocos Eletrônicos Chave.

Os de número 1 a 5 devem ter a propriedade invisível marcada. Para que a configuração corresponda ao circuito da polarização fixa o estado inicial das chaves deve estar com os valores da tabela 5.50.

Bloco Eletrônico Chave	Estado inicial
1	Fechada
2	Aberta
3	Aberta
4	Fechada
5	Aberta

Tabela 5.50 - Propriedades das chaves para a configuração polarização fixa

A chave 6 deverá ser visível e conter um rótulo que indique que ao acionar a mesma o transistor será trocado por outro. O circuito da figura 5.85 mostra o circuito real para troca do transistor.

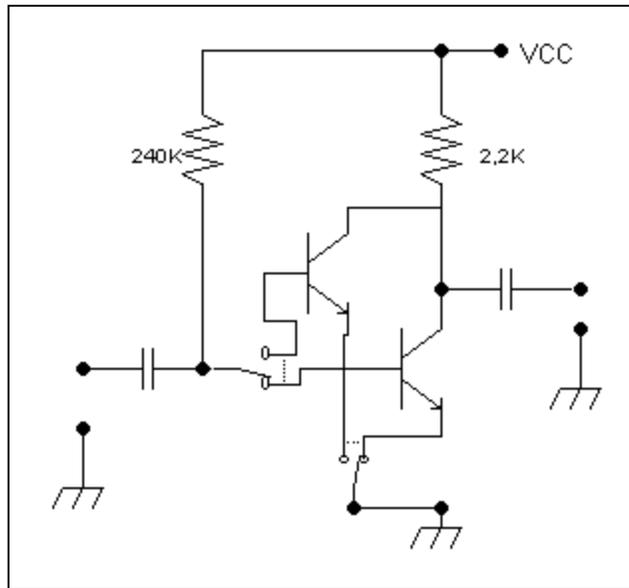


Figura 5.85 - Circuito com dois transistores e chaves para troca

Para configurar o objeto de informação rótulo o processo é o mesmo dos Blocos Eletrônicos, ao clicar sobre o objeto com o botão direito do mouse é aberta uma janela com as propriedades. No rótulo é possível inserir o texto que será exibido (figura 5.86).

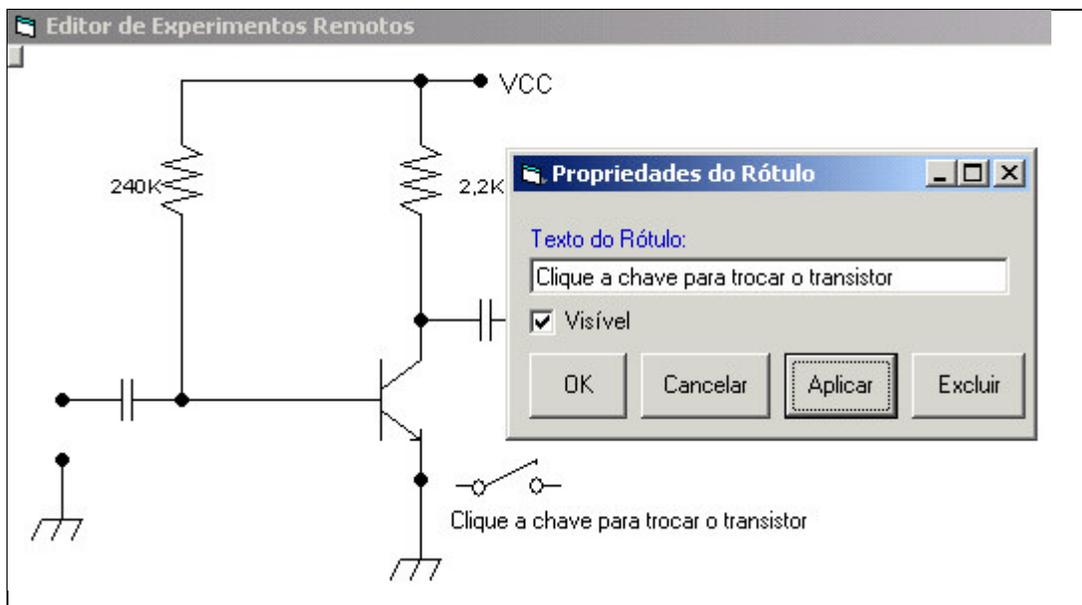


Figura 5.86 - Configuração do rótulo no editor de experimentos

O passo seguinte é inserir os Objetos Hyperlink. Um deles vai acessar um documento com a especificação do transistor utilizado e o outro um documento elaborado

peço professor com orientações sobre o experimento. A figura 5.87 ilustra a configuração do objeto hyperlink de acesso ao material do professor.

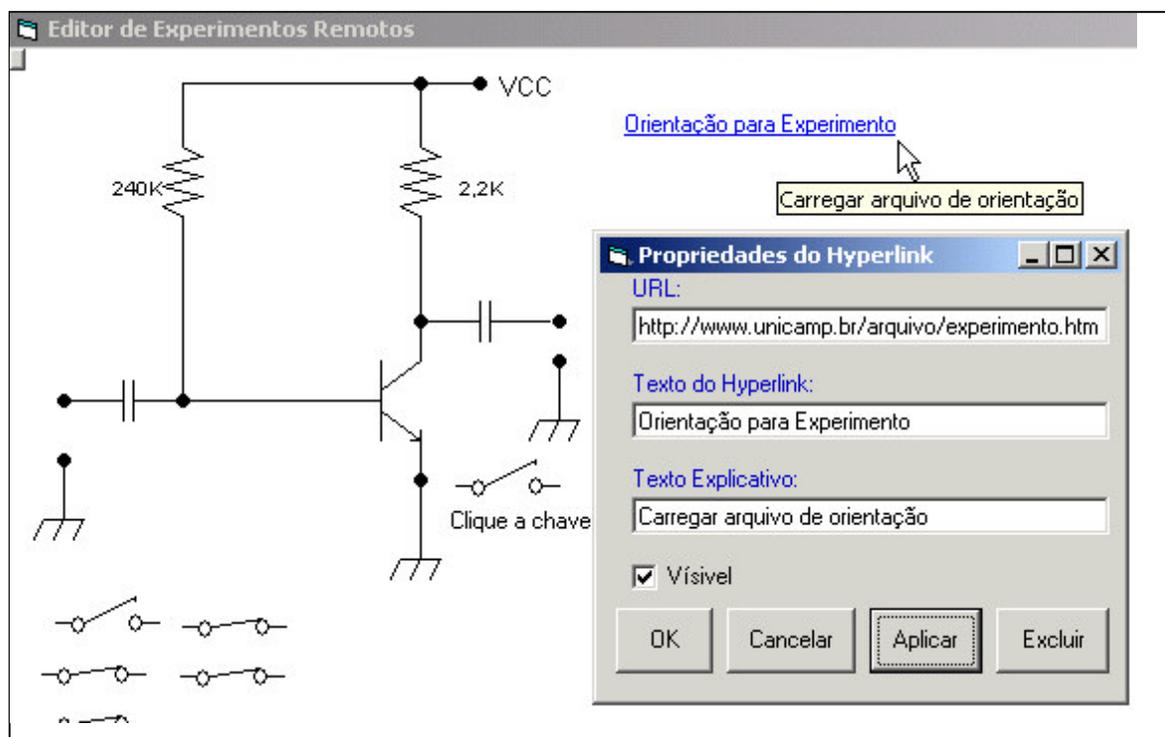


Figura 5.87 - Configuração das propriedades do Objeto Hyperlink

Para efetuar as medições no circuito são inseridos três Blocos Ponto de Medição, sendo um na base do transistor, outro no coletor e um terceiro no emissor do mesmo (Figura 5.88). A configuração permite selecionar os pontos que serão conectados ao circuito em um dado momento. O recurso proporcionado pelo Bloco Ponto de medição permite representar a situação de um experimento real no qual se coloca a ponta de prova de um instrumento em determinado ponto do circuito eletrônico.

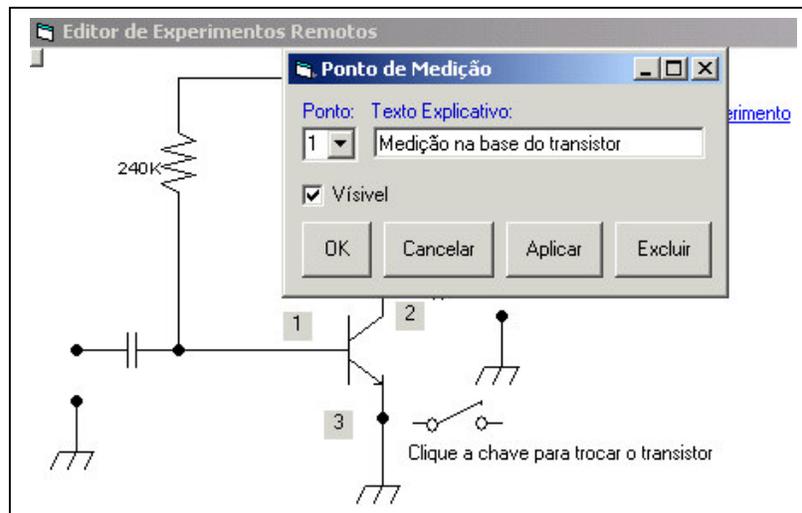


Figura 5.88 - Configuração das propriedades do Bloco Ponto de Medição

Após realizar todas as configurações basta salvar o projeto e carregá-lo no Cliente do Experimento Remoto (figura 5.89).

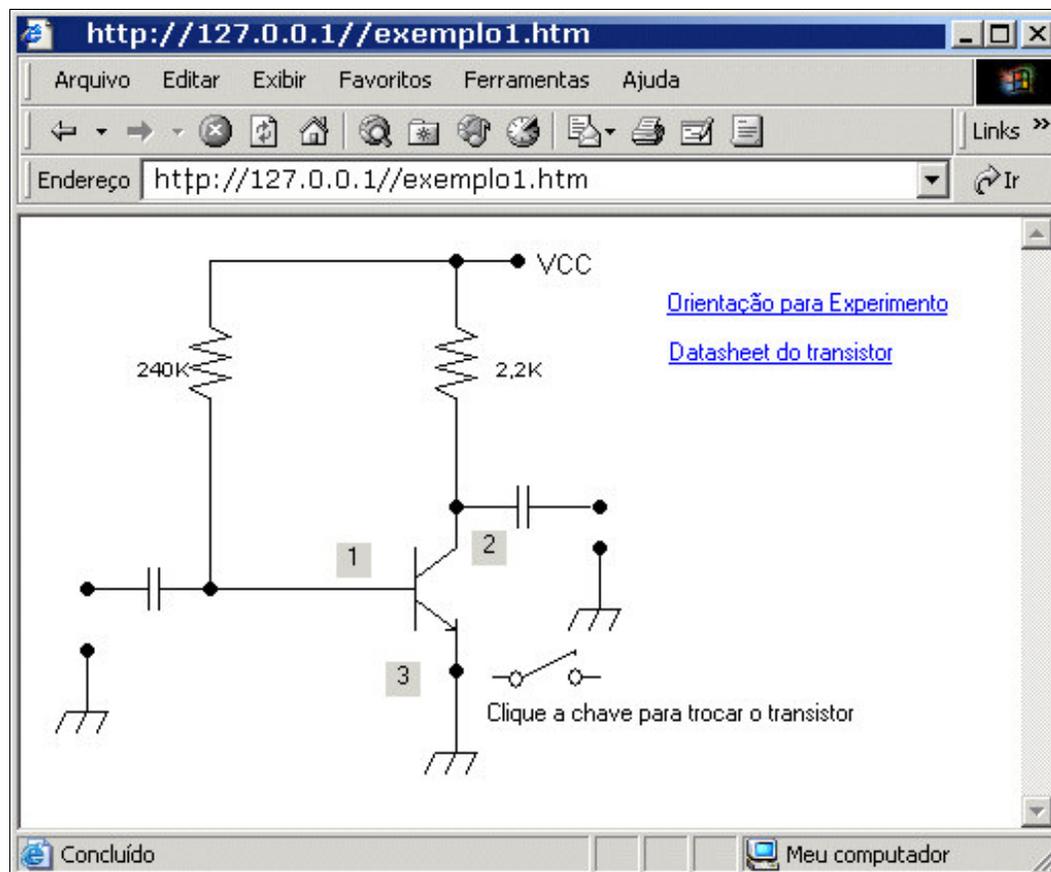


Figura 5.89 - Experimento de polarização no navegador web

Após desenvolver o experimento no Editor de Experimentos é necessário carregar os aplicativos Servidor da Placa de Controle e Servidor GPIB. Após executar esses passos o

experimento poderá ser controlado a partir de qualquer computador PC conectado à rede Internet.

Experimento 4

As atividades nessa experiência envolvem ligar e desligar chaves que alteram a configuração do circuito eletrônico, atuar sob os potenciômetros digitais e realizar leituras com o osciloscópio a fim de analisar os sinais. Para realizar as atividades o aluno é orientado por texto com um roteiro fornecido pelo professor

Para avaliação do sistema o desenvolvimento do experimento ficou sob a responsabilidade de um aluno de mestrado que realizou as seguintes ações:

- A partir do esquemático com o circuito eletrônico (figura 5.90) foi implementada uma placa de experimento contendo o circuito, e os blocos eletrônicos, como chaves, potenciômetro e pontos de medição.

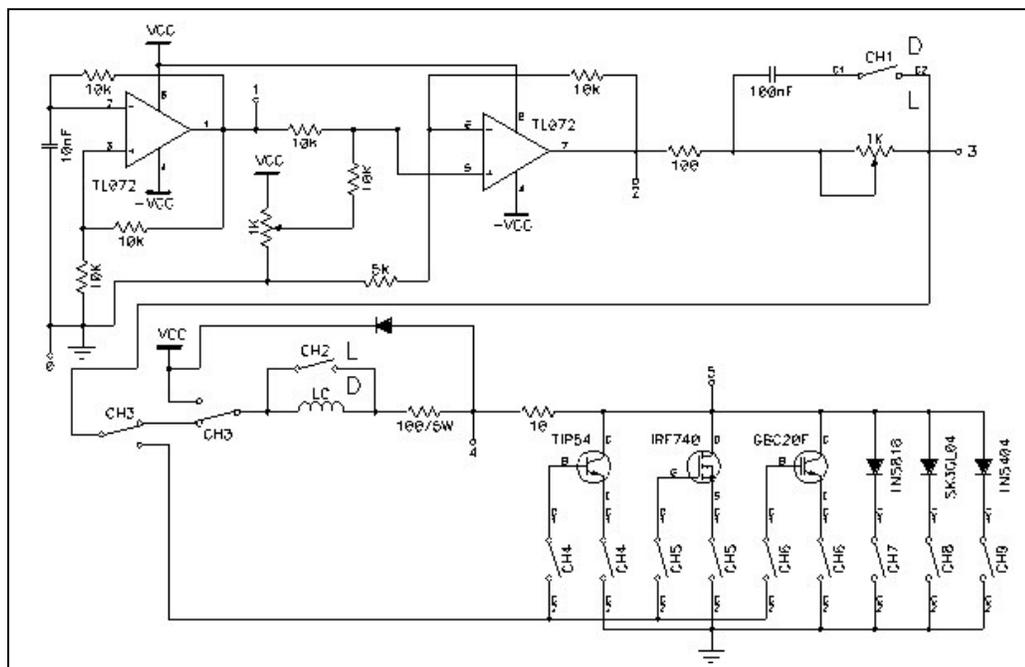


Figura 5.90 - Diagrama esquemático do experimento quatro

- Conversão do esquemático para um arquivo de imagem e a utilização do editor de experimentos para inserir a imagem do circuito e os objetos gráficos e blocos eletrônicos com suas respectivas propriedades configuradas.

- Execução do roteiro fornecido pelo professor utilizando o cliente do experimento remoto. Nessa etapa todos os procedimentos realizados no modo local foram realizados remotamente.

Para execução do experimento foi utilizado um computador com processador Celeron de 800Mhz e 256 Mb de memória RAM com sistema operacional Windows 2000.

Neste computador foi conectado pela porta USB a Placa de Controle executando o aplicativo Servidor da Placa de Controle. Em um computador com processador Pentium de 233Mhz e memória de 128Mb com placa GPIB da National instruments foi conectado o Osciloscópio HP modelo 54403 e foi executado o aplicativo Servidor GPIB. Os dois computadores estavam conectados a intranet da Unicamp.

A interface com o usuário no Cliente do Experimento Remoto é constituída pelo circuito esquemático e dos Objetos que podem ser alterados durante a execução dos procedimentos da experiência (Figura 5.91).

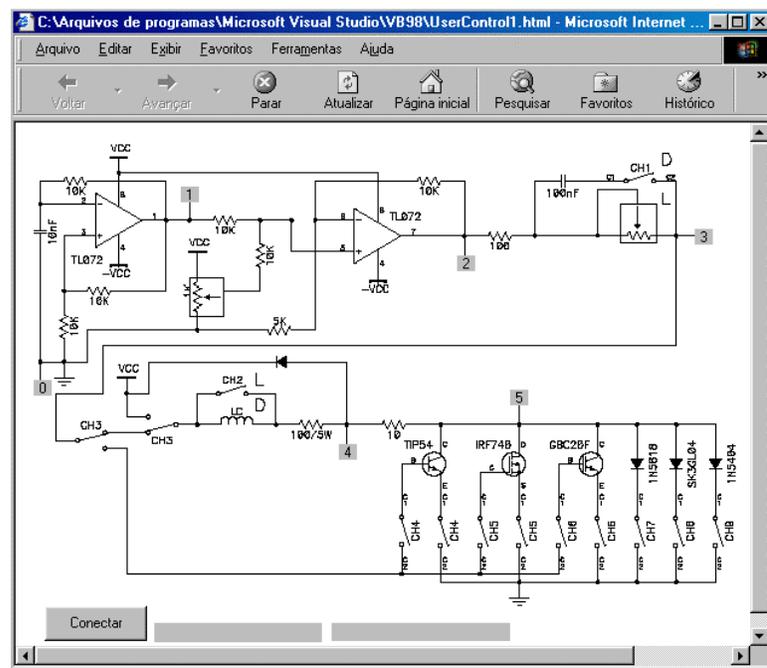


Figura 5.91 - Interface Gráfica do Experimento 4

As chaves e os potenciômetros da experiência real são representados no desenho esquemático do circuito eletrônico e ao serem clicados funcionam como se o aluno estivesse atuando diretamente sobre o experimento convencional, ou seja, os componentes serão fisicamente acionados, isto é, chaves serão ligadas ou desligadas, potenciômetros

terão seu valor alterado na Placa do Experimento. Para cada ação do usuário um retorno visual é apresentado, o que propicia uma representação fiel do estado do circuito que está sendo experimentado.

A figura 5.92 mostra o detalhe da interface gráfica do experimento remoto com uma chave de seleção na condição de desliga.

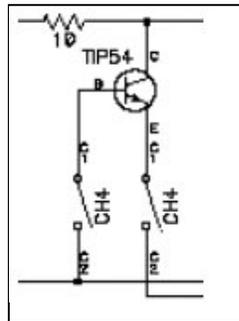


Figura 5.92 - Detalhe da interface gráfica com chave aberta

Ao clicar no desenho será enviado um comando para a Placa de Controle que fecha a chave na Placa de Experimento e o desenho da chave no Cliente do Experimento Remoto é mudado para condição de fechada (Figura 5.93).

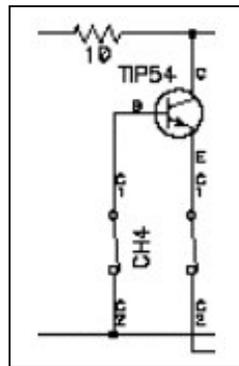


Figura 5.93 - Detalhe da interface gráfica com chave fechada

Do mesmo modo podem ser variados os pontos de medição que conectam as pontas de prova do osciloscópio nos pontos do circuito eletrônico que devem ter o sinal observado. No Cliente de Experimento Remoto ao clicar em um ponto de medição é aberta uma janela que permite a conexão do canal 1, 2 ou o ponto de referência do osciloscópio (Figura 5.94).

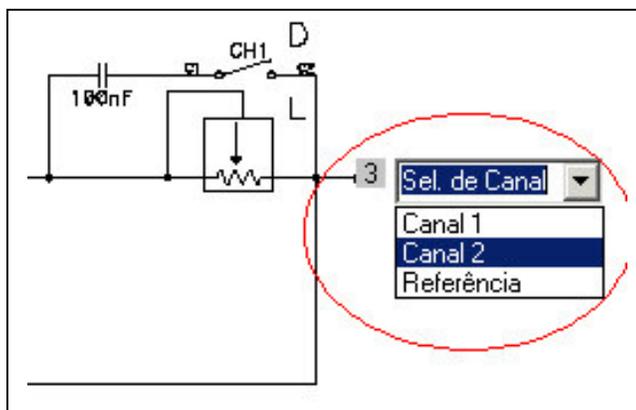


Figura 5.94 - Detalhe da seleção do ponto de medição

Para alterar o valor do objeto Potenciômetro Digital o usuário deve clicar o objeto potenciômetro na interface gráfica do Cliente do Experimento Remoto e arrastar o curso para que ele mude de posição e envie os comandos para a Placa de Controle. Na figura 5.95 pode ser vista a imagem com o potenciômetro em duas posições que foram modificadas pelo usuário.

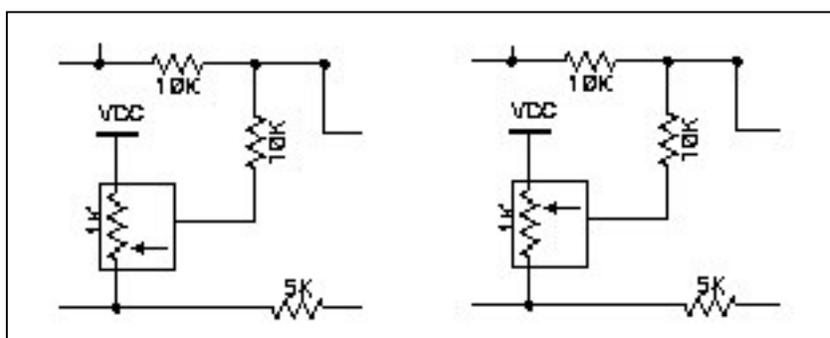


Figura 5.95 - Detalhe interface gráfica com potenciômetro digital

Com os recursos disponíveis foi possível a realização do roteiro do experimento remotamente, executando as ações de controle e visualização dos sinais no Cliente GPIB.

O trabalho do professor para desenvolver experimentos como os demonstrados envolve elaborar o hardware específico com as configurações do experimento e associar os Blocos Eletrônicos que fazem parte do circuito. Com o hardware configurado o Editor de Experimentos deve ser executado para criar a interface gráfica para o usuário, no caso o aluno, configurar as propriedades dos Blocos Eletrônicos e salvar o projeto para disponibilizar o acesso remoto por meio do Cliente do Experimento Remoto.

O processo de implementação do experimento remoto por meio do Editor de Experimentos é realizado pelo professor sem que ele precise escrever programas para

acesso ao hardware de controle, rede Internet e elaboração da interface gráfica. Desse modo ele pode empregar maior tempo nos aspectos pedagógicos da atividade planejada.

Conclusão

O conjunto de programas e dispositivos de hardware integrados atingiram o objetivo proposto de possibilitar a realização de experimentos remotamente utilizando a rede de computadores internet.

Os programas desenvolvidos ainda que executem funções como acesso a drivers de hardware, comunicação com a rede Internet e visualização gráfica de dados e controles gerados dispensam a necessidade de computadores com muitos recursos. Um computador com microprocessador de pelo 500Mhz, 256 Mb de memória RAM e 10Mb de espaço em disco já permite a utilização dos sistemas tanto servidores como cliente.

Como o sistema opera enviando informações em pacote de dados que pode variar de quatro a cinco bytes e recebe pacotes de dados com até 512 bytes não há demanda por uma largura de banda de rede elevada.

A utilização do Editor de Experimentos permitiu simplificar e reduzir de forma substancial o tempo necessário para a implementação de um experimento remoto já que os aspectos técnicos relacionados à comunicação com a rede, hardware e interação com o usuário é realizada pela integração do Editor aos demais aplicativos do sistema ocorrendo de forma transparente ao usuário.

O desenvolvimento do conceito de Bloco Eletrônico e representação do mesmo em software permite uma flexibilidade e escalabilidade no desenvolvimento dos sistemas. Tendo como referência os Blocos Eletrônicos já existentes o desenvolvedor pode criar novos blocos com outros componentes ou outras características.

Com um computador e uma Placa de Controle é possível controlar até oito Placas de Experimento. O número de experimentos desenvolvidos com esse sistema, contudo pode ser maior que oito, caso se elabore projetos em que o hardware possa ser reconfigurável. Um exemplo é o experimento 1 demonstrado no capítulo V em que por meio da elaboração dos projetos no Editor de Experimentos foi possível desenvolver três experimentos distintos com uma Placa de Experimentos.

Vários sistemas de experimentos que podem estar inclusive distantes geograficamente podem estar acessíveis de forma transparente pelo usuário, favorecendo a realização de trabalhos colaborativos.

A modelagem do sistema de software baseada no fluxo de dados e a adoção de tecnologias aceitas universalmente para troca de dados como XML e socket permite que diversas tecnologias de hardware e software possam ser utilizadas na implementação.

Como exemplo podemos vislumbrar alguns cenários possíveis para desenvolvimento:

- Cliente do Experimento Remoto desenvolvido em Java operando com os outros aplicativos originais. Esta situação é possível, pois a linguagem Java tem recursos para comunicação por socket e manipulação de XML.
- Servidores podem ser desenvolvidos para execução em sistemas operacionais como Windows, UNIX, Linux, já que ferramentas de programação como C/C++ possuem bibliotecas para acesso ao barramento GPIB, USB e comunicação com a rede Internet.
- A elaboração de Clientes com ferramentas para desenvolvimento de páginas dinâmicas pode permitir o acesso ao experimento remoto a partir de plataformas como assistentes pessoais, telefones móveis entre outros.

O emprego de componentes de hardware como microcontroladores e controladores USB e componentes digitais diversos, que podem ser adquiridos com facilidade no mercado nacional permite que o custo final dos sistemas de hardware seja reduzido. Também a possibilidade do compartilhamento de instrumentos com maior custo por diversas instituições podem favorecer a implantação dos sistemas de Laboratório de Acesso Remoto em instituições com recursos financeiros limitados.

Do ponto de vista educacional o professor ao utilizar o Editor de Experimentos e sistemas associados na preparação de experimentos pode se concentrar nos aspectos educacionais que ele pretende explorar, pois os recursos técnicos necessários para acesso ao hardware e comunicação com a rede são disponibilizados pela arquitetura do laboratório de acesso remoto.

A possibilidade de alterar, inserir, retirar Blocos Eletrônicos no experimento e configurar essas ações e funcionalidades por meio do Editor de Experimento permite que os experimentos possam ser modificados e disponibilizados sem a necessidade implementação de programas e montagem de dispositivos eletrônicos para controle, o que pode propiciar ganho em termos de tempo para disponibilização do experimento ao aluno.

As características indicadas para elaboração e alteração dos experimentos podem criar oportunidades para que o professor desenvolva uma quantidade maior de experimentos levando o aluno a ter contato com mais atividades de laboratório o que de certa forma pode colaborar em sua formação.

Trabalhos futuros

O trabalho desenvolvido admite aperfeiçoamento e acréscimos tanto do ponto de vista de software quanto de hardware.

Para que os experimentos do Laboratório de Acesso Remoto possam ser utilizados efetivamente com grupos de alunos é preciso que exista a integração com um sistema de gerenciamento. Este sistema de gerenciamento deve prover funcionalidades como agendamento de horário, controle de acesso incluindo recursos disponíveis e tempo para cada usuário.

A integração dos recursos de Laboratório Remoto a um sistema de ensino a distância poderia ser uma opção interessante. Nesse caso as atividades tradicionais como chat, fórum, disponibilização de materiais poderiam estar associadas as atividades de acesso ao laboratório.

Em termos de hardware podem ser desenvolvidos novos blocos eletrônicos estendendo os componentes disponíveis para utilização nos experimentos.

A oferta de componentes como microcontroladores com recursos para internet embutida possibilitará o desenvolvimento integrado de placas de controle com acesso a internet podendo dispensar o uso do computador PC com função de servidor.

A medida que o hardware para sistemas de rede sem fio se tornar mais acessível abre-se a possibilidade de que o sistema de experimentos possa ganhar mobilidade e aproveitando-se desse recurso novos experimentos podem ser planejados.

Do ponto de vista de software a tendência é independência de plataformas e mobilidade. Uma característica interessante para os próximos sistemas é que os mesmos possam ser acessíveis a partir de um computador de mesa, um computador de mão e mesmo de um telefone móvel com acesso a internet.

O desenvolvimento de atividades em laboratórios de acesso remoto pode propiciar materiais de estudo que se analisados podem propiciar meios para se estabelecer metodologias adequadas para aplicação eficaz desse recurso educacional.

Referências bibliográficas

- Aulanet (2003) Aulanet, <http://asgard.les.inf.puc-rio.br/aulanet/>. Consulta em 25/01/2004
- Analog (2005) AD7376,
http://www.analog.com/UploadedFiles/Data_Sheets/488716441AD7376_pra.pdf. Consulta em 30/08/2005
- _____. (2003) AD7501,
http://www.analog.com/UploadedFiles/Data_Sheets/57806281055232AD7501_2_3_b.pdf. Consulta em 30/08/2005
- Anderson, D. (2001), USB System Architecture (USB 2.0). Mindshare Inc., Colorado
- Axelsson J. (2001), USB Complete. Lakeview Research, USA
- Baker, A. e Lozano, J. (2000), The Windows 2000 Device Driver Book, A Guide for Programmers. Prentice Hall, New Jersey
- Borges, P.B. (2002), Instrumentação virtual aplicada a um laboratório com acesso pela Internet. Dissertação de Mestrado, Escola Politécnica da Universidade de São Paulo. São Paulo
- Casini M., Prattichizzo D. e Vicino A. (2002), Automatic Control Telelab: un Laboratorio Remoto per E-learning,
http://www.dii.unisi.it/~control/act/reports/act_c1.pdf . Consulta em 03/05/2005
- _____. (2004) The Automatic Control Telelab. A Web-based technology for Distance Learning IEEE Control Systems Magazine, 24(3):36-44.
- _____. (2003) The Automatic Control Telelab: a User-Friendly Interface for Distance Learning IEEE Transactions on Education, 46(2):252-257.
- Catsoulis, J. (2002), Designing Embedded Hardware. O'Reilly, USA
- Comer, D. E. (2000) Internetworking with TCP/IP Client-Server Programming and Applications. Prentice Hall, New Jersey
- Erl, T. (2004) Service-Oriented Architecture A field guide to integrating XML and Web Services. Prentice Hall, New Jersey
- Gruber Y.A., Arnheiter M. e Gellis M. (2000) Visual Basic para UML - Guia do programador e padrões de design. Market Books, São Paulo

Hua, J. e Ganz, A. (2003) "A new model for remote laboratory education based on next generation interactive technologies",
<http://researchers.conferenceexp.net/Lists/Research%20Papers2/Attachments/5/aseeivlab.pdf>,
f, Abril

Ilab(2005) Ilab MIT Service Broker, http://ilab.mit.edu/ServiceBroker6_0/home.aspx.
Consulta em 07/05/2005

Lucena C., Fuks H.(2000) A Educação na Era da Internet. Clube do Futuro, Rio de Janeiro

Maxim-ic (2005) MAX232, <http://pdfserv.maxim-ic.com/en/ds/MAX220-MAX249.pdf>.
Consulta em 30/08/2005

Microchip (2004) PIC16F877,
http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1335&dDocName=en010242. Consulta em 30/08/2005

Moultis N. P.,Kirk C., (2000) XML Black Book. Makron Books, São Paulo

National (2001) National Instruments, <http://www.ni.com/>. Consulta em 10/01/2004

Nielsen, J. (1999) Design Web Usability. New Riders Publish., Indianapolis, Indiana USA

Potts, A. e Friedel, D.H.(1996) Java Programming Language Handbook.Coriolis Group Books, USA

Pressman, R. S. (1999) Engenharia de software. Makron Books, Brasil

Rexlab(2003) Rexlab, <http://www.inf.ufsc.br/~jbosco/labvir.htm>. Consulta em 03/08/2004

Rocha, H. V. e Baranauskas, M. C. C. (2003), Design e avaliação de interfaces humano-computador, Nied/Unicamp, Campinas-SP

Rubin, W.e Brain, M.(1999) Understanding Dcom. Prentice Hall, New Jersey

Schafer, T., Seigneur, J. M. e Donnelly, A. (2002) PEARL: A Generic Architecture for Live Experiments in a Remote Lab, <http://iet.open.ac.uk/pearl/publications/icsee03.pdf>

Silva M. (2000) Sala de Aula Interativa. Quarter Ed., Rio de Janeiro

Shen H., Xu Z., Dalager B., Kristiansen V., Strøm Ø., Shur M. S., Fjeldly T. A ., Lü J. e Ytterdal T. (1999) " Conducting Laboratory Experiments over the Internet", IEEE transactions on education, vol. 42, no. 3, p.180-185.

Sun (200) JNI, <http://java.sun.com/j2se/1.4.2/docs/guide/jni/spec/jniTOC.html> . Consulta em 12/10/2005

Tanenbaum, A. S.(1999) Structured Computer Organization. 4^a ed. Prentice Hall, USA.

Teleduc (2003) Teleduc, <http://teleduc.nied.unicamp.br/~teleduc/>. Consulta em 28/01/2004

USB.ORG(2003) Universal Serial Bus Revision 2.0 specification, <http://www.usb.org/developers/docs/>. Consulta em 08/05/2005

W3C(2005) DOM, <http://www.w3.org/DOM//>. Consulta em 30/08/2005

_____.(1996) SGML, <http://www.w3.org/Markup/SGML/>. Consulta em 10/08/2005

_____.(1996) XML, <http://www.w3.org/XML/>. Consulta em 08/05/2006

Webct (2002) WEBCT, <http://www.webct.com/>. Consulta em 25/01/2004

Webster J. G., Tompkins, W. J. (1987) Interfacing sensors to the IBM PC. Prentice Hall, New Jersey

Anexo 1

Protocolo de comunicação da placa de controle

O protocolo é composto por 5 bytes, sendo dois bytes iniciais de cabeçalho, um byte de comando e dois bytes de dados.

Formato do protocolo

S	M	Comando	Dado (8 bits)	Dado estendido (8 bits)
---	---	---------	---------------	-------------------------

Byte 1 cabeçalho

S

Byte 2 cabeçalho

M

Byte 3 comando

- S para potenciômetro SPI
- R para reles
- B para banco de resistores
- M para ponto de medição
- E para seleção da experiência

Ponto de medição (5 bytes)

S	M	M	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1
			x	x	x	x	x	x	x	Bit 9

Byte 1 cabeçalho

S

Byte 2 cabeçalho

M

Byte 3 comando

M comando para ponto de medição

Byte 4 dados

Dados bit1 a bit8

Byte 5 dados

Dado Bit9

B1,b2,b3 seleciona pontos 1 a 8 do CH1

B4, b5, b6 seleciona pontos 1 a 8 do CH2

B7,b8,b9 seleciona pontos 1 a 8 da referencia

Potenciômetro Digital SPI (5 bytes)

S	M	S	x	x	x	x	x	E2	E1	E0
			D7	D6	D5	D4	D3	D2	D1	D0

Byte 1 cabeçalho

S

Byte 2 cabeçalho

M

Byte 3 comando

S comando para potenciômetro SPI

Byte 4 dado

Bits e2, e1, e0 selecionam o endereço do potenciômetro (1 a 8)

Byte 5 dado

Byte com dado contendo o valor a ser escrito no potenciômetro digital

Banco de Resistores (4 bytes)

S	M	B	B1	B0	Ch2	Ch1	Ch0	D2	D1	D0
---	---	---	----	----	-----	-----	-----	----	----	----

Byte 1 cabeçalho

S

Byte 2 cabeçalho

M

Byte 3 comando

B comando para banco de resistores

Byte 4 dados

Seleção do banco de resistor

Banco Selecionado	B1	B0
1	0	0
2	0	1
3	1	0
4	1	1

Seleção do Bloco

Bloco selecionado	Ch2	Ch1	Ch0
1	0	0	1
2	0	1	0
3	1	0	0

Seleção do resistor

Resistor selecionado	D2	D1	D0
0	0	0	0
1	0		1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Chave (4 bytes)

S	M	R	x	x	C0	R4	R3	R2	R1	R0
---	---	---	---	---	----	----	----	----	----	----

Byte 1 cabeçalho

S

Byte 2 cabeçalho

M

Byte 3 comando

R comando para reles

Byte 4 dados

Bit C0 se 1 liga o rele, se 0 desliga o rele.

R0 a R4 endereça o relê que será acionado de 1 a 23

Seleção de experimento (4 bytes)

S	M	E	x	x	x	x	x	D2	D1	D0
---	---	---	---	---	---	---	---	----	----	----

Byte 1 cabeçalho

S

Byte 2 cabeçalho

M

Byte 3 comando

E comando para seleção do experimento

Byte 4 dados

Bits D2, D1, D0 selecionam a experiência (1 a 7)

Anexo 2

Propriedades dos Blocos Eletrônicos e Objetos de Informação.

Potenciômetro Digital	
Propriedade	Descrição
top	Posição do objeto na área de trabalho no eixo X
left	Posição do objeto na área de trabalho no eixo Y
height	Altura do objeto
width	Largura do objeto
orientacao	Forma de exibição do objeto (esquerda, direita, cima ou para baixo)
npasso	Números de passos do potenciômetro (especificação do componente)
tooltip	Caixa de Texto exibido ao passar o mouse sobre o objeto
valor	Valor resistivo do potenciômetro
qualpot	Indica o potenciômetro a ser usado na placa de experimento
vinicial	Valor inicial que o potenciômetro deve assumir no experimento
escala	Escala para exibição (por exemplo Kohms)
objeto	Nome do objeto
visível	Define se o objeto será visível no cliente do Experimento Remoto

Rótulo	
Propriedade	Descrição
top	Posição do objeto na área de trabalho no eixo X
left	Posição do objeto na área de trabalho no eixo Y
texto	Texto que será exibido
objeto	Nome do objeto
visível	Define se o objeto será visível no cliente do Experimento Remoto

Hyperlink	
Propriedade	Descrição
top	Posição do objeto na área de trabalho no eixo X
left	Posição do objeto na área de trabalho no eixo Y
url	Url que será carregada na janela do navegador
texto	Texto que será exibido
tooltip	Caixa de Texto exibido ao passar o mouse sobre o objeto
visível	Define se o objeto será visível no cliente do Experimento Remoto

Chave	
Propriedade	Descrição
top	Posição do objeto na área de trabalho no eixo X
left	Posição do objeto na área de trabalho no eixo Y
height	Altura do objeto
width	Largura do objeto
orientacao	Forma de exibição do objeto (vertical ou horizontal)
estado	Condição inicial (ligada ou desligada)
tooltip	Caixa de Texto exibido ao passar o mouse sobre o objeto
qualchave	Indica a chave a ser usada na placa de experimento
visível	Define se o objeto será visível no cliente do Experimento Remoto

Banco de Componentes	
Propriedade	Descrição
top	Posição do objeto na área de trabalho no eixo X
left	Posição do objeto na área de trabalho no eixo Y
height	Altura do objeto
width	Largura do objeto
orientacao	Forma de exibição do objeto (vertical ou horizontal)
banco	Banco utilizado
bloco	Bloco no banco utilizado
resistor	Resistor no bloco utilizado
visível	Define se o objeto será visível no cliente do Experimento Remoto

Ponto de Medição	
Propriedade	Descrição
top	Posição do objeto na área de trabalho no eixo X
left	Posição do objeto na área de trabalho no eixo Y
ponto	Ponto de medição na Placa de Experimento
tooltip	Caixa de Texto exibido ao passar o mouse sobre o objeto
objeto	Nome do objeto
visível	Define se o objeto será visível no cliente do Experimento Remoto

Exemplo de definição de projeto que contempla todos os Objetos Eletrônicos e de Informação

```
- <meuprojeto>
  = <figura_fundo>
    <nome>ico_mouse.gif</nome>
  </figura_fundo>
  = <potenciometro>
    <top>2505</top>
    <left>6240</left>
    <height>585</height>
    <width>705</width>
    <orientacao>1</orientacao>
    <npassos>128</npassos>
    <tooltip>potenciometro no circuito</tooltip>
    <valor>100</valor>
    <qualpot>1</qualpot>
    <inicial>128</inicial>
    <escala>k</escala>
    <objeto>potenciometro</objeto>
    <visivel>1</visivel>
  </potenciometro>
  = <chave>
    <top>3570</top>
    <left>2535</left>
    <height>855</height>
    <widht>810</widht>
    <orientacao>0</orientacao>
    <estado>1</estado>
    <tooltip>chave liga desliga</tooltip>
    <qualchave>3</qualchave>
    <visivel>1</visivel>
  </chave>
  = <pmedicao>
    <top>2265</top>
    <left>2205</left>
    <ponto>5</ponto>
    <tooltip>ponto para medida do osciloscópio</tooltip>
    <objeto>ponto_medida</objeto>
    <visivel>1</visivel>
  </pmedicao>
  = <rotulo>
    <top>990</top>
    <left>2700</left>
    <texto>texto do rótulo</texto>
    <objeto>rotulo</objeto>
    <visivel>1</visivel>
  </rotulo>
```

```
= <banco_resistor>
  <top>3330</top>
  <left>5340</left>
  <height>330</height>
  <width>675</width>
  <orientacao>h</orientacao>
  <banco>0</banco>
  <bloco>4</bloco>
  <resistor>5</resistor>
  <visivel>1</visivel>
</banco_resistor>
= <link>
  <top>1605</top>
  <left>3105</left>
  <url>www.unicamp.br</url>
  <texto>site da unicamp</texto>
  <tooltip>carregar o site</tooltip>
  <visivel>1</visivel>
</link>
</meuprojeto>
```

Anexo 3

Comandos GPIB para osciloscópio HP modelo 54503A implementados no Servidor de GPIB

Identificador no Cliente	Comando GPIB	Descrição
vda	:CHANNEL1:RANGE <valor>	volts/divisão para canal 1
vdb	:CHANNEL2:RANGE <valor>	volts/divisão para canal 2
tib	:TIMEBASE:RANGE <valor>	base de tempo
trl	:TRIGGER:LEVEL <valor>	nível de trigger
trd	:TRIGGER:DELAY TIME <valor>	delay do trigger
trh	:TRIGGER:HOLDOFF TIME <valor>	trigger holdoff
tmm	:TIMEBASE:MODE <valor>	timebase mode
tsl	:TRIGGER:SLOPE POSitive ou NEGative	trigger slope
tso	:TRIGGER:SOURCE CHAN2 ou CHAN2	trigger source
pca	Deslocamento do eixo gráfico	Posição canal 1
pcb	Deslocamento do eixo gráfico	Posição canal 2