

Sistema de Aquisição de Dados para Laboratórios

Paulo André Oliveira Libonati

Dissertação de Mestrado submetida à Coordenação do Curso de Mestrado em Engenharia Elétrica da Universidade Federal da Bahia, como parte dos requisitos necessários para obtenção do grau de Mestre em Ciências no domínio da Engenharia Elétrica.

Área de concentração: Processamento de Informação e Energia

Amauri Oliveira, Dr.
Orientador

Salvador, Bahia, Brasil

Sistema de Aquisição de Dados para Laboratórios

Paulo André Oliveira Libonati

Dissertação de Mestrado apresentada em Fevereiro de 2002

Amauri Oliveira, Dr.

Orientador

Caiuby Alves da Costa, Dr.

Componente da Banca

José Felício da Silva, Dr.

Componente da Banca

Salvador, Bahia, Brasil, Fevereiro de 2002

Resumo

Neste trabalho, é apresentado um sistema de aquisição de dados para automação de ensaios em laboratórios. Desenvolvido a partir da necessidade de automatizar os ensaios em amostras de solos no Laboratório de Geotecnia da Universidade Federal da Bahia, o sistema foi projetado de forma a permitir a realização de novos tipos de ensaios, podendo ser usado para aquisição de dados em outras áreas da engenharia, tais como: elétrica, mecânica, etc.

Em função do número de equipamentos a serem automatizados e da distância entre eles, optou-se por uma topologia distribuída. Foram usadas Unidades Remotas de Aquisição de Dados (Remota), interligadas por uma rede serial RS485 multiponto, usando protocolo mestre-escravo. A Remota é dotada de microcontrolador e memória RAM, o que lhe confere autonomia para a realização de ensaios com até 1280 leituras de oito sensores, com intervalo de tempo configurável a partir de 1 segundo. Um microcomputador central é usado apenas para configurar os parâmetros do ensaio, visualizar as leituras, traçar gráficos e gravar os dados, podendo até mesmo ser desligado após o início do ensaio. No caso de falta de energia durante um ensaio, os dados são mantidos na memória da Remota, até o retorno da energia, por até duas semanas. A Remota possui sete entradas analógicas de 1 a 5 Vdc (ADC de 12 bits) e uma entrada digital para transdutores tipo "encoder" (16 bits). Até dezesseis Remotas podem ser ligadas em uma mesma linha RS485.

A Remota prevê uma saída para motores de passo, usados para atuar válvulas reguladoras de pressão, de forma a aplicar, ao ensaio, um valor constante de pressão, ou um valor crescente/decrescente ao longo do tempo. Os motores são acionados pela Remota de forma a posicionar uma válvula reguladora para o valor de pressão desejado, sendo usado controle tipo ON-OFF com duas velocidades de acionamento. Para aplicação de cargas, pode ser usada uma célula de carga em conjunto com um pistão pneumático; o *firmware* da Remota irá monitorar o valor indicado pela célula de carga e regular o valor da pressão de ar de forma a obter a carga desejada. Os motores de passo também podem ser usados para posicionamento de um objeto, usando o sinal de um transdutor de deslocamento como variável a ser controlada. Para laboratórios que possuem computadores interligados em rede (usando protocolo TCP/IP), o sistema permite o acompanhamento dos ensaios a partir de qualquer um dos computadores da rede.

Abstract

In this work is presented a data acquisition system to automate laboratory testing. Developed from the need to automate soil testing at Laboratório de Geotecnia from Universidade Federal da Bahia, the system was designed to allow new kind of tests to be performed, so it can be used to automate tests in other engineering area, such as: electrical, mechanical, etc.

A distributed topology was used, due to the number of equipments to be automated and the distance between them. Remote Data Acquisition Units (Remote), interconnected by a RS485 multidrop serial network, using master-slave protocol, were used. Each Remote unit have a microcontroller and RAM memory, that enable it to perform stand-alone data acquisition of up to 1280 readings from eight sensors, at time intervals configurable from 1 second. A central microcomputer is used only to configure the test parameters, view readings, plot graphs and record data, so it can be turned off after the test begins. If a power supply failure happens, when the test is running, the data are preserved in the Remote unit memory for up to two weeks, until the supply restores. The Remote unit has seven 1 to 5 Vdc analog inputs (12 bits ADC) and a digital input for encoder-like transducers (16 bits). Up to sixteen Remote units can be connected to the same RS485 serial line.

The Remote unit has an output port to drive step-motors, used to actuate pressure regulator valves in order to apply, to the test, constant pressure values, or values that increase/decrease with time. Using ON-OFF control with two motor speeds, the step-motor position the valve axis to get the desired pressure value at the valve output. To apply load instead of pressure, a load cell and a pneumatic cylinder can be used; the Remote unit firmware reads the load cell and adjusts the air pressure to make the cylinder apply the desired load. The step-motors can also be used for positioning, using the signal from a displacement transducer as control input. At laboratories where computers are networked (using TCP/IP protocol), the system allows remote operation through network.

Sumário

1. Introdução	1
1.1. Aplicações	2
1.1.1. Ensaio de Compressão Triaxial	3
1.1.2. Ensaio de Cisalhamento Direto	4
1.1.3. Ensaio de Compressão Edométrica (Adensamento).....	5
1.1.4. Índice de Suporte Califórnia (CBR – California Bearing Ratio).....	6
1.1.5. Monitoramento da Tensão da Rede Elétrica.....	7
1.1.6. Monitoramento do Consumo de Equipamentos Elétricos	7
1.2. Organização do Trabalho.....	7
2. Barramento para Instrumentação	9
2.1. HART	12
2.2. GPIB/IEEE488	13
2.3. Fieldbus	15
2.4. Barramento RS485	20
2.4.1. Protocolo ASCII	23
2.4.2. Protocolo Binário.....	27
3. Interface Homem Máquina (IHM)	29
3.1. Acompanhamento do Ensaio	29
3.2. Disparo Automático.....	32
3.3. Configuração	32
3.4. Programa de Calibração.....	35
3.5. Acesso Remoto	37
4. Hardware da Remota	39
4.1. Microcontrolador	40
4.2. Memória RAM	42
4.3. RS485	42
4.4. Endereço	46
4.5. ADC.....	46
4.6. TDD	48
4.7. Interface para motor de passo	49

5. Conclusões	53
5.1. Futuro.....	54
Referências Bibliográficas	56

Lista de Tabelas

Tabela 1: Comandos HART	13
Tabela 2: Tabela ASCII.....	24
Tabela 3: Composição do nome do arquivo de dados	33
Tabela 4: Comparação entre ADC de 12 bits e ADC de 16 bits	55

Lista de Figuras

Figura 1: Desenho esquemático do ensaio de Compressão Triaxial	3
Figura 2: Desenho esquemático do ensaio de Cisalhamento Direto	5
Figura 3: Desenho esquemático do ensaio de Adensamento.....	6
Figura 4: Desenho esquemático do ensaio de CBR.....	6
Figura 5: Formas de ligação dos transdutores ao computador	10
Figura 6: Modulação FSK sobre o sinal 4-20 mA.....	12
Figura 7: Estrutura do barramento GPIB.....	14
Figura 8: Estrutura dos padrões GPIB.....	15
Figura 9: Modelo OSI aplicado ao Fieldbus.....	16
Figura 10: Codificação Manchester Biphase-L.....	16
Figura 11: Ciclo de execução dos Function Blocks	20
Figura 12: Transmissão Full-duplex e Half-duplex.....	21
Figura 13: RS485 Multiponto.....	21
Figura 14: Protocolo Mestre-Escravo.....	22
Figura 15: Exemplo de comando Optomux.....	25
Figura 16: Tela do Ensaio Triaxial (Etapa de cisalhamento)	30
Figura 17: Gráfico do ensaio	30
Figura 18: Leituras do ensaio	30

Figura 19: “Zoom” do valor de um sensor	31
Figura 20: Visão geral dos ensaios	31
Figura 21: Arquivo ASCII com os dados do ensaio	31
Figura 22: Ensaio em modo de espera	32
Figura 23: Ensaio em andamento	32
Figura 24: Sensibilidade do disparo automático	32
Figura 25: Configuração do ensaio	33
Figura 26: Configuração dos sensores	34
Figura 27: Configuração do ensaio genérico	34
Figura 28: Tela de Calibração	35
Figura 29: Gráfico de Calibração	36
Figura 30: Polinômio do 5º grau	37
Figura 31: Polinômio do 3º grau	37
Figura 32: Disponibiliza ensaios na Internet	38
Figura 33: Acessa ensaios pela Internet	38
Figura 34: Diagrama de blocos da Remota	39
Figura 35: Firmware	41
Figura 36: Capacitor de backup	42
Figura 37: Ligação das Remotas	43
Figura 38: Rede do Laboratório de Geotecnia da UFBA	45
Figura 39: Painel traseiro da Remota	46
Figura 40: Circuito de entrada do ADC	47
Figura 41: Painel frontal da Remota	48
Figura 42: Sinais do TDD	49
Figura 43: Sistema tradicional de aplicação de carga	50
Figura 44: Sistema pneumático de aplicação de carga	50
Figura 45: Interface para motor de passo	52

Lista de Símbolos

∅ - diâmetro;

A - altura;

ADC - Analog to Digital Converter;

ANSI - American National Standards Institute;
ASCII - American Standard Code for Information Interchange;
ATN - Attention;
CBR - California Bearing Ratio;
CLP - Controlador Lógico Programável;
DAV - Data Valid;
DC - Direct Current;
EOI - End or Identify;
EPROM - Erasable Programmable Read Only Memory;
FSK - Frequency Shift Keying;
GPIB - General-Purpose Interface Bus;
HART - Highway Addressable Remote Transducer;
IHM - Interface Homem Máquina;
IFC - Interface Clear;
ISA - Industry Standard Architecture;
LED - Light Emitting Diode;
LVDT - Linear Variable Differential Transformer;
NDAC - Not Data Accepted;
NRFD - Not Ready for Data;
NRZ - Not Return to Zero;
OSI - Open Systems Interconnect;
PCI - Peripheral Component Interconnect;
PID - Proporcional Integral Derivativo;
RAM - Random Access Memory;
REN - Remote Enable;
RTC - Real Time Clock;
SCPI - Standard Commands for Programmable Instruments;
SDCD - Sistema Digital de Controle Distribuído;
SRQ - Service Request;
TCP/IP - Transmission (Transport) Control Protocol/Internet Protocol;
TDD - Transdutor Digital de Deslocamento;
TTL - Transistor Transistor Logic;

Capítulo 1

1. Introdução

A informática, a cada dia, ganha maior aplicação nas engenharias de um modo geral, e os sistemas digitais para aquisição de dados são um exemplo disto. Em diversas áreas da engenharia é necessária a realização de ensaios para determinar propriedades físicas de materiais como madeira, solo, concreto, etc, onde diversas grandezas físicas são lidas e armazenadas ao longo do tempo, à medida que uma amostra do material é submetida a esforços mecânicos ou a mudanças das condições ambientais (ex: umidade, temperatura).

Os sistemas de aquisição de dados microprocessados são ferramentas poderosas na realização desses ensaios, pois permitem realizar essas leituras automaticamente. Isso proporciona maior confiabilidade dos dados, uma vez que a leitura manual é propícia a erros. Permite também um acompanhamento do ensaio através de gráficos gerados à medida que as leituras são feitas, além de liberar o técnico para a realização de outras atividades no laboratório.

Este trabalho tem como objetivo descrever o Sistema de Aquisição de Dados para ensaios de laboratório, desenvolvido pelo autor, e contém informações úteis a aqueles que pretendam desenvolver um sistema semelhante. Como alguns tipos de ensaios possuem características bem distintas dos demais, não se pode pensar em um sistema que atenda a todos os tipos. Por esse motivo, o sistema desenvolvido não objetiva a automação de qualquer tipo de ensaio de laboratório, mas de um grupo de ensaios com características comuns: até 8 sensores, máximo de 1280 leituras e taxa de amostragem máxima de 1 leitura/segundo.

Como cada área de aplicação (ex: petroquímica, indústria automobilística, manufatura) possui características próprias [5], o uso de um sistema comercial desenvolvido para uma outra área pode implicar na necessidade de uma série de adaptações no sistema. Muitas vezes, pode ser necessário adaptar o ensaio de forma a se adequar ao sistema de aquisição de dados adquirido. O custo de desenvolvimento de um sistema de aquisição de dados é, em alguns casos, compensado pelo fato de se ter um sistema dedicado mais simples e, portanto, mais barato.

Adicionalmente, há um ganho em produtividade por se ter um sistema adequado às necessidades, e mais simples de operar. Além disso, é possível alterar as características ou adicionar novos recursos, de acordo com a necessidade, não ficando preso às limitações de um sistema comercial proprietário.

O mesmo vale para as diversas topologias existentes. Um sistema de aquisição de dados pode ser montado usando placas de aquisição de dados ISA/PCI (Industry Standard Architecture / Peripheral Component Interconnect) ligadas diretamente ao barramento do microcomputador, equipamentos de medição (ex: multímetros, osciloscópios) ligados à porta serial RS232 de um microcomputador, unidades remotas interligadas em rede (ex: Fieldbus, RS485), equipamentos com interface GPIB/IEEE488, etc. Nesse trabalho é feita uma descrição dos barramentos mais utilizados atualmente, mostrando as vantagens e desvantagens do uso de cada um deles.

Em função das características dos ensaios a serem automatizados, optou-se pelo desenvolvimento de um sistema baseado em unidades remotas de aquisição de dados, interligadas por um barramento RS485. Isso possibilitou a obtenção de um sistema simples, compacto, de baixo custo, sobre o qual se tem domínio para novas implementações ou alterações de suas características.

1.1. Aplicações

As aplicações aqui descritas são exemplos dos tipos de ensaio nos quais a Remota pode ser usada. Isso não significa que sua aplicação esteja restrita aos ensaios aqui relacionados, pois todo o sistema foi desenvolvido de forma que possa ser configurado para novos tipos de ensaios com características similares às dos ensaios aqui descritos. Inicialmente são descritas algumas aplicações na área de Geotecnia (ensaios de laboratório para determinação das propriedades mecânicas de solos), seguido de exemplos de aplicações em outras áreas.

1.1.1. Ensaio de Compressão Triaxial

O ensaio de compressão triaxial [1] é feito usando-se um corpo de prova cilíndrico, moldado a partir de uma amostra de solo, o qual é colocado dentro de uma câmara própria envolto por uma membrana de borracha (Figura 1). A câmara é preenchida geralmente com água, à qual se aplica uma *pressão confinante*, que é uma tentativa de simular o estado de confinamento do solo em campo. O ensaio pode ser realizado aplicando-se uma carga axial constante ao corpo de prova, que irá se deformar em função dessa carga, ou pode-se controlar a deformação da amostra, fazendo com que o pistão penetre na câmara com velocidade constante, medindo-se a força de reação do solo. Durante o ensaio, pode-se drenar a água do corpo de prova por meio de pedras porosas colocadas no topo e na base do mesmo.

Ao longo do ensaio são realizadas leituras, em intervalos de tempo constantes, das seguintes grandezas: carga axial, deformação axial (deslocamento do pistão), pressão confinante e pressão neutra (pressão exercida pela água contida nos poros do solo). Pode-se determinar o momento de parar o ensaio traçando-se o gráfico da *carga axial x deslocamento axial* à medida que as leituras são realizadas. Quando o valor da carga estabiliza, começa a decrescer, ou aumentar mais lentamente, o ensaio pode ser finalizado. A depender do tipo de solo, o ensaio pode ter uma duração de 1 hora com intervalo entre leituras de 2 minutos (ensaios rápidos), podendo chegar a 72 horas com intervalo de 1 hora (ensaios lentos).

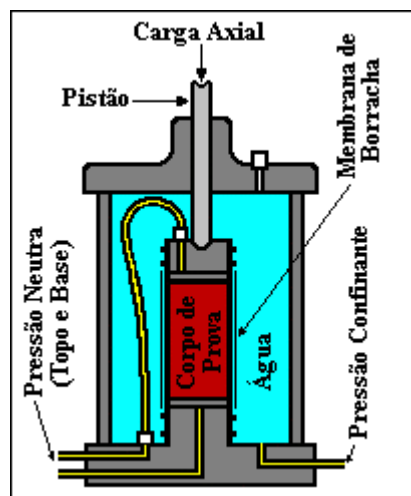


Figura 1: Desenho esquemático do ensaio de Compressão Triaxial

A partir dos dados obtidos no ensaio, é possível traçar o círculo de Mohr [1] correspondente à situação de ruptura. Com os círculos de Mohr determinados em ensaios feitos em outros corpos de prova (três, no máximo), é possível traçar uma envoltória linear de resistência conforme o critério de Mohr. Através deste ensaio é possível se ter uma idéia do comportamento tensão x deformação de um dado solo.

1.1.2. Ensaio de Cisalhamento Direto

O ensaio de cisalhamento direto [2] foi desenvolvido para a determinação da resistência ao corte de um corpo de prova de solo pouco espesso, geralmente prismático e de seção quadrada ou circular. O corpo de prova é colocado em uma caixa de cisalhamento composta de duas metades (superior e inferior), que serão forçadas em direções opostas, de forma a cortar o corpo de prova ao meio (Figura 2). Com uma carga vertical aplicada ao corpo de prova, a metade inferior da caixa de cisalhamento é movida com velocidade constante em relação à parte superior fixa, medindo-se os deslocamentos horizontal e vertical, e a força suportada pelo corpo de prova. O deslocamento vertical medido indicará se houve variação de volume do solo durante o ensaio.

Durante o ensaio são realizadas leituras, em intervalo de tempo constante, das seguintes grandezas: carga vertical, deslocamento vertical, carga cisalhante e deslocamento horizontal. Pode-se determinar o momento de parar o ensaio traçando-se o gráfico da *carga cisalhante x deslocamento horizontal* à medida que as leituras são realizadas. Como no ensaio de Compressão Triaxial, quando o valor da carga estabiliza, começa a decrescer, ou aumentar mais lentamente, o ensaio pode ser finalizado. A depender do tipo de solo, o ensaio pode ter uma duração de 1 hora com intervalo entre leituras de 2 minutos (ensaios rápidos), ou de até 72 horas com intervalo de 1 hora (ensaios lentos).

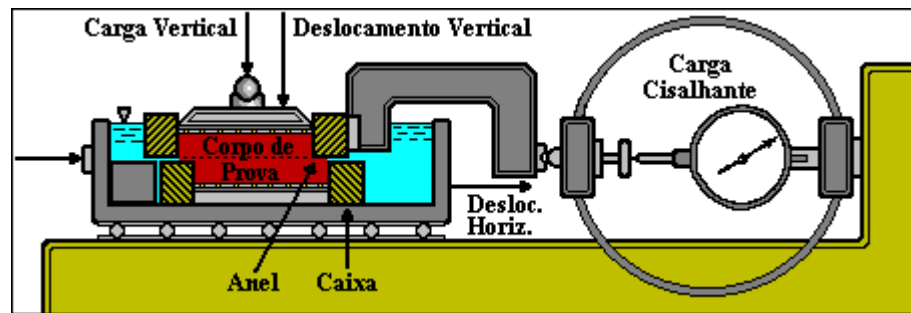


Figura 2: Desenho esquemático do ensaio de Cisalhamento Direto

Pode-se compor a envoltória de resistência do solo fazendo ensaios com diferentes tensões normais. O ensaio de cisalhamento direto é mais aplicado ao estudo da resistência ao cisalhamento em solos estratificados ou com xistosidades pronunciadas.

1.1.3. Ensaio de Compressão Edométrica (Adensamento)

O *ensaio de compressão edométrica* [3], também conhecido como *ensaio de compressão confinada* ou *ensaio de adensamento*, é usado para determinar o grau de compressibilidade de um determinado solo. Ao sofrer um carregamento, o solo tende a reduzir de volume, à medida que diminui os vazios entre suas partículas. Essa redução de volume ocorre de forma mais acentuada logo após a aplicação da carga, decaindo de forma exponencial com o passar do tempo. Por esse motivo, durante um ensaio de Adensamento, as leituras são feitas em intervalos de tempo que dobram a cada leitura.

Este ensaio é realizado em etapas, onde à cada etapa é adicionada uma carga a um corpo de prova confinado lateralmente (Figura 3). A carga é mantida constante ao longo de cada etapa, e é medida a deformação do solo até ele parar de adensar. Desse ensaio são interpretados parâmetros fundamentais para o cálculo de recalques por adensamento de um determinado depósito de solo.

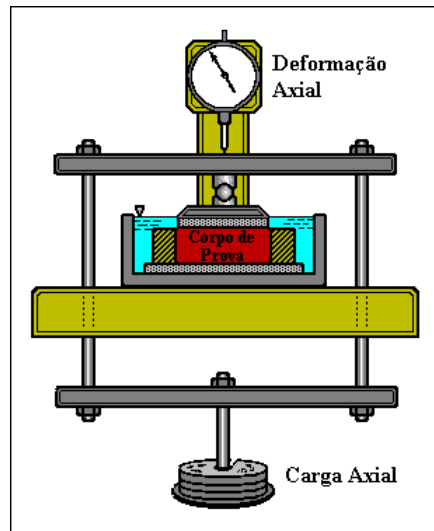


Figura 3: Desenho esquemático do ensaio de Adensamento

1.1.4. Índice de Suporte Califórnia (CBR – California Bearing Ratio)

O Índice de Suporte Califórnia [4] é útil em estudos de pavimentos, cujo valor, expresso em percentual, fornece um comparativo de resistência entre o solo testado e um solo de brita graduada, tido como padrão. A partir de uma amostra de solo, um corpo de prova é compactado com uma determinada energia e umidade, dentro de um cilindro metálico chamado "Cilindro Proctor" (Figura 4). Mede-se então, em intervalos de tempo constantes de 30 segundos, a resistência oferecida pelo solo à penetração, com velocidade constante, de um pistão com seção transversal padrão. A duração do ensaio é de aproximadamente 10 minutos.

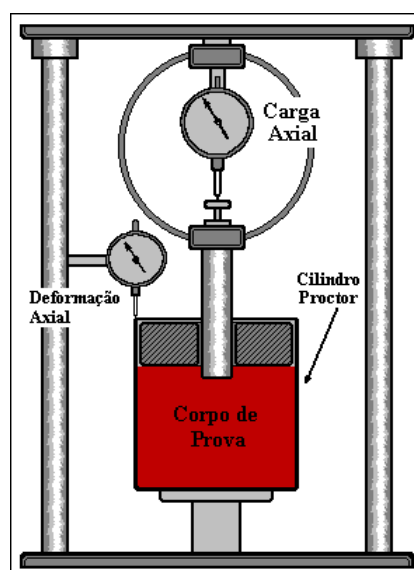


Figura 4: Desenho esquemático do ensaio de CBR

1.1.5. Monitoramento da Tensão da Rede Elétrica

A energia elétrica distribuída pela concessionária deve estar dentro de uma certa faixa de tensão, pois uma sobretensão, ou uma subtensão, pode causar danos a equipamentos elétricos. Uma forma de determinar se a energia fornecida está dentro da faixa aceitável, é registrar, ao longo do dia, o valor da tensão em um ponto da rede elétrica. As medições podem ser feitas, por exemplo, em intervalos de tempo de 1 minuto, permitindo registrar o valor da tensão, usando a Remota, por até 19 horas. Usando-se um intervalo de tempo maior entre as leituras, é possível registrar a tensão por períodos mais longos.

1.1.6. Monitoramento do Consumo de Equipamentos Elétricos

Muitas vezes é necessário determinar o consumo de um equipamento elétrico, que pode variar ao longo do dia em função da temperatura ambiente ou em função da demanda (ex: ar condicionado, geladeira). Medindo-se a tensão da rede e a corrente consumida pelo equipamento, é possível calcular o consumo total e o consumo nos diversos períodos do dia. No caso de um aparelho de ar-condicionado, por exemplo, pode-se usar transdutores para medir também a temperatura de entrada e de saída, e a vazão de ar, de forma a determinar o rendimento do mesmo.

1.2. Organização do Trabalho

Este trabalho começa com esta introdução e tem mais cinco capítulos, que tratam dos seguintes assuntos:

- No capítulo dois, é feita uma rápida descrição dos barramentos seriais para instrumentação, sendo comparados alguns padrões existentes no mercado. É mostrada também a estrutura de um protocolo de comunicação mestre-escravo, normalmente utilizado com os barramentos RS485;

- No capítulo três, é apresentado o programa de IHM desenvolvido para configuração, acompanhamento e gravação dos dados dos ensaios. O programa foi desenvolvido para ambiente Windows, com telas gráficas, de forma a tornar a operação simples e amigável;
- No capítulo quatro, são descritos os blocos funcionais que compõem o hardware da Remota;
- No capítulo cinco, são apresentados comentários e conclusões a respeito do uso de um sistema de aquisição de dados para realização de ensaios em laboratórios.

Capítulo 2

2. Barramento para Instrumentação

A forma mais simples de se montar um sistema de aquisição de dados, é usando uma placa de aquisição de dados ISA/PCI instalada dentro de um microcomputador. Nessa configuração, o cabo que conduz o sinal analógico de cada transdutor é ligado diretamente à placa de aquisição de dados (Figura 5.a), sendo necessário, muitas vezes, condicionar o sinal (normalmente usando o padrão 4-20 mA) para diminuir a interferência do ruído e da impedância do cabo. Além disso, a manutenção ou a alteração do sistema normalmente implica na parada de todos os processos ligados ao sistema de aquisição de dados.

Ramos et al. [6], devido aos requisitos e à disponibilidade de placas de aquisição de dados de baixo custo, usaram uma placa de aquisição de dados ligada ao barramento ISA de um PC. O sistema foi desenvolvido para automatizar até oito ensaios de compressão edométrica (adensamento), onde sensores a base de *strain-gauge* foram usados para medir a deformação de amostras de solo. Devido ao baixo sinal de saída dos sensores utilizados (20 mV max.), a distância das prensas de adensamento não pode ser muito grande, pois, senão, o ruído induzido compromete a exatidão das leituras. Além disso, para a troca de um sensor ou para manutenção do computador, é necessário desligar o sistema, o que implica em parar todos os ensaios em andamento.

Para evitar esses problemas, pode ser utilizado um barramento digital, onde o sinal é transmitido por meio de uma linha de transmissão serial, ou paralela, e vários dispositivos estão ligados à mesma linha de transmissão (Figura 5.b). O fato da informação ser transmitida de forma digital elimina interferência do ruído induzido durante a transmissão do sinal, e permite acrescentar funções ao dispositivo (ex: diagnóstico de falha, configuração, calibração). A manutenção e a expansão do sistema são também simplificadas.

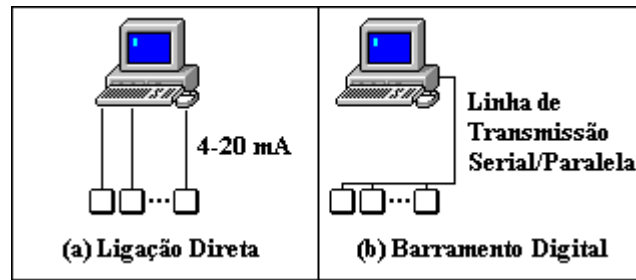


Figura 5: Formas de ligação dos transdutores ao computador

Apesar da linha de transmissão RS232 não ser um barramento, pois o padrão RS232 só permite que dois dispositivos sejam ligados a uma mesma linha, alguns equipamentos de medição (ex: multímetros, osciloscópios) já vêm equipados com uma interface serial RS232 para conexão com microcomputadores. Junto com um programa apropriado, é possível montar um pequeno sistema de aquisição de dados adequado para bancadas experimentais. Porém, para permitir a leitura do sinal de vários sensores e fornecer a tensão de alimentação apropriada para cada sensor, seria necessária uma série de adaptações externas. Para se ter, por exemplo, no mesmo sistema, entradas analógicas, entradas tipo “encoder” e saída para motores de passo, teriam que ser usados vários equipamentos, ligados a várias portas seriais do computador.

Existem vários padrões de barramento digitais em uso atualmente, cada um com características que o torna apropriado para uma determinada aplicação [7]. O GPIB (General-Purpose Interface Bus), também conhecido como IEEE 488 ou HPIB (Hewlett-Packard Interface Bus), é um barramento paralelo de comunicação para sistemas de instrumentação. Esse barramento normalmente é usado para interligar equipamentos de teste e medição, como multímetro digital, gerador de funções, fonte de alimentação programável, e outros aparelhos e acessórios necessários para montar um sistema de instrumentação. Apesar de ser um padrão muito usado em laboratórios para sistemas de medição, para aplicações distribuídas, onde a distância entre os equipamentos a serem interligados é maior, um barramento serial é mais apropriado.

O padrão Fieldbus [14], conforme visto a seguir, possui um grau elevado de complexidade, em função da necessidade de atender a uma grande variedade de transdutores, atuadores e controladores, e a um grande número de aplicações. Na indústria, utiliza-se um número elevado de transdutores, instalados à distâncias consideráveis. A grande diversidade de

fornecedores que, muitas vezes, adotam padrões proprietários, faz com que seus equipamentos sejam incompatíveis com os equipamentos dos demais fornecedores. Nesse cenário, o custo e a complexidade do Fieldbus são compensados pela economia decorrente da redução do número de cabos, do custo de manutenção, e pela interoperabilidade dos instrumentos.

Em laboratórios, como as distâncias são menores, e o número de transdutores e fornecedores é reduzido, torna-se oneroso adotar o padrão Fieldbus, podendo-se adotar um barramento mais simples. Portanto, apesar do Fieldbus estar se consolidando como padrão de rede local para instrumentação, foi adotado um barramento mais simples e mais apropriado para a aplicação em questão.

O RS485 é um padrão mais flexível, pois ele define apenas as características elétricas do barramento. Com isso, é possível especificar um barramento sob medida para uma aplicação específica, evitando a complexidade de padrões criados para atender um grande número de aplicações.

O HART é um barramento misto pois transmite um sinal digital modulado em frequência, sobreposto a um sinal analógico de 4-20 mA. Isso permite adicionar capacidade de processamento aos instrumentos, aproveitando o “cabeamento” de 4-20 mA existente. Porém, a taxa de transmissão de dados (1200 bps) é bem inferior à dos barramentos puramente digitais (até 2,5 Mbps).

Em locais remotos, de difícil acesso, onde os meios de comunicação são escassos, outras formas de transmissão de dados também podem ser usadas. S. Ameer et al. [8] implementaram um sistema de aquisição de dados onde os dados eram armazenados, diariamente, em um cartucho removível de memória RAM não-volátil. Mensalmente, o cartucho era substituído, e transportado para um local onde os dados eram transferidos para um computador. Esse sistema foi usado para monitoramento das condições ambientais de uma plantação de bananas em uma estufa, em uma fazenda perto de Tizi Ouzou (Algeria).

A seguir, é feita uma breve descrição dos barramentos digitais.

2.1. HART

O HART (Hihgway Addressable Remote Transducer) é um padrão misto para comunicação com instrumentos inteligentes. Um sinal digital modulado em frequência (FSK) é sobreposto a um sinal analógico de 4-20 mA, de forma a permitir uma comunicação bilateral com o sensor. Os dois sinais são transmitidos simultaneamente sem que haja interferência entre eles, pois o valor médio do sinal digital é zero.

Normalmente, o valor indicado pelo sensor (ou o sinal de controle para uma válvula) é transmitido pelo sinal de 4-20 mA, enquanto medidas adicionais, configuração, calibração e diagnóstico do instrumento são transmitidos usando o sinal digital FSK. A taxa de transmissão do sinal digital é de 1200 bps (bits por segundos), usando uma frequência de 1200 Hz para o nível lógico "0" e 2400 Hz para o nível "1" (Figura 6) [9].

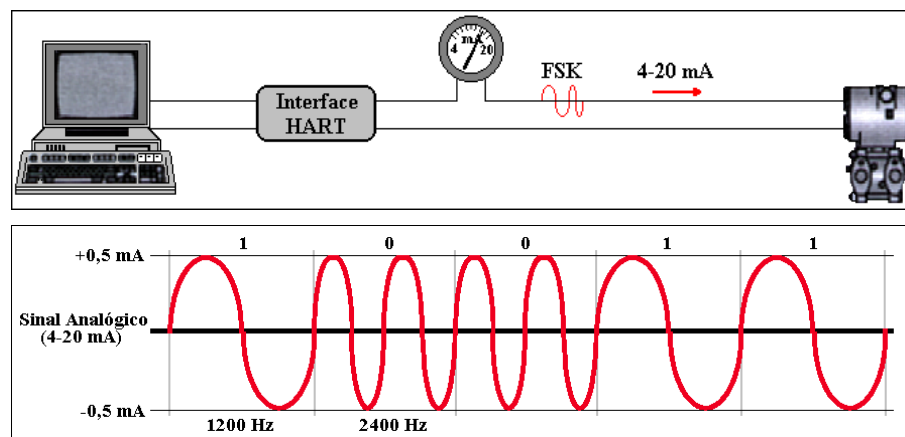


Figura 6: Modulação FSK sobre o sinal 4-20 mA

O HART é um protocolo mestre/escravo que prevê a existência de até dois mestres. Normalmente o mestre primário é um SDCD, um CLP ou um computador, enquanto que um terminal portátil é usado como mestre secundário para configurar o instrumento. A comunicação do "mestre" com o(s) instrumento(s) ("escravo") pode se dar de três modos:

Master/Slave - Esse é o modo normal de operação, onde o sinal de 4-20 mA é transmitido de forma contínua, simultaneamente com o sinal digital transmitido duas vezes por segundo. Nesse modo, o instrumento ("escravo") responde às mensagens do "mestre".

Brust - Modo de comunicação opcional em que um dispositivo "escravo" envia continuamente uma mensagem de resposta, sem que o "mestre" precise ficar solicitando-a. É usado para atualizar freqüentemente a informação sobre uma variável do processo, a uma taxa de 3 a 4 vezes por segundo.

Multidrop - Vários dispositivos são ligados em paralelo em uma configuração multiponto. Apenas o sinal digital (FSK) transmite informação, sendo o sinal de 4-20 mA usado apenas para alimentar os instrumentos.

O protocolo HART define Comandos Universais (Tabela 1), de forma a garantir interoperabilidade entre uma larga, e crescente, base de produtos de diferentes fornecedores [9]. Os Comandos de Uso Geral acessam funções que são implementadas em muitos dispositivos. Esses comandos são opcionais, porém, se implementados, devem ser feitos de acordo com a especificação. Os Comandos Específicos do Dispositivo, acessam características exclusivas de um dispositivo. Usados principalmente para configuração de parâmetros do dispositivo, esses comandos podem, por exemplo, enviar um novo *setpoint* para o algoritmo PID do dispositivo.

Tabela 1: Comandos HART

Comandos de Uso Geral	Comandos Específicos do Dispositivo
Ler variáveis	Funções específicas do modelo
Mudar faixa superior & inferior	Opções especiais de calibração
Ajuste (zero, span)	Iniciar, parar, ou limpar totalizador
Iniciar auto-teste	Selecionar variável primária
Número de série	Habilitar PID, mudar <i>setpoint</i>
Valores de constantes de tempo	Ajustar parâmetros de sintonia

2.2. GPIB/IEEE488

O GPIB (General-Purpose Interface Bus), também conhecido como IEEE488 ou HPIB (Hewlett-Packard Interface Bus), é uma interface paralela de comunicação de 8 bits, com uma taxa de transferência de até 1 Mbytes/s. O padrão IEEE488.1 [10] define as características

mecânicas e elétricas da interface. Até quinze dispositivos podem ser ligados a um barramento GPIB, sendo que o comprimento total dos cabos não pode exceder 20 metros.

O barramento é composto por dezesseis linhas agrupadas da seguinte forma (Figura 7):

- 8 linhas para o barramento de dados (DIO1 a DIO8)
- 3 linhas para o barramento de controle da transferência do *byte* de dados (DAV, NRFD e NDAC)
- 5 linhas para o barramento de gerenciamento da interface (ATN, IFC, SRQ, REN e EOI)

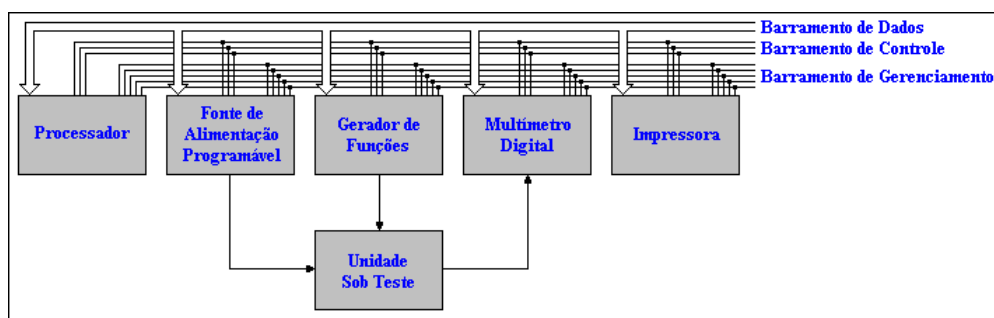


Figura 7: Estrutura do barramento GPIB

O barramento de controle da transferência do *byte* de dados é usado para efetuar a transferência de cada *byte* no barramento de dados. A linha DAV (Data Valid) é usada para indicar a condição (disponibilidade e validade) da informação no barramento de dados, a NRFD (Not Ready for Data) é usada para indicar a condição de disponibilidade do(s) dispositivo(s) para aceitar dados, e a NDAC (Not Data Accepted) é usada para indicar a condição de aceitação dos dados pelo(s) dispositivo(s).

O barramento de gerenciamento da interface é usado para gerenciar o fluxo ordenado de informações através da interface. A função de cada linha desse barramento é:

- ATN (Attention) é usada para especificar como os dados no barramento de dados devem ser interpretados e quais os dispositivos devem responder aos dados
- IFC (Interface Clear) é usada para colocar o barramento em um estado conhecido
- SRQ (Service Request) é usada por um dispositivo para indicar a necessidade de atenção e requisitar uma interrupção na atual seqüência de eventos

- REN (Remote Enable) é usada, em conjunto com outras mensagens, para habilitar ou desabilitar um ou mais controles locais que têm controles remotos correspondentes
- EOI (End or Identify) é usada para indicar o fim de uma seqüência de transferência de *bytes* ou, em conjunto com ATN, executar uma seqüência de varredura

Os padrões IEEE488.2 [11] e SCPI [12] definem o protocolo de comunicação e os comandos do GPIB (Figura 8) [13].

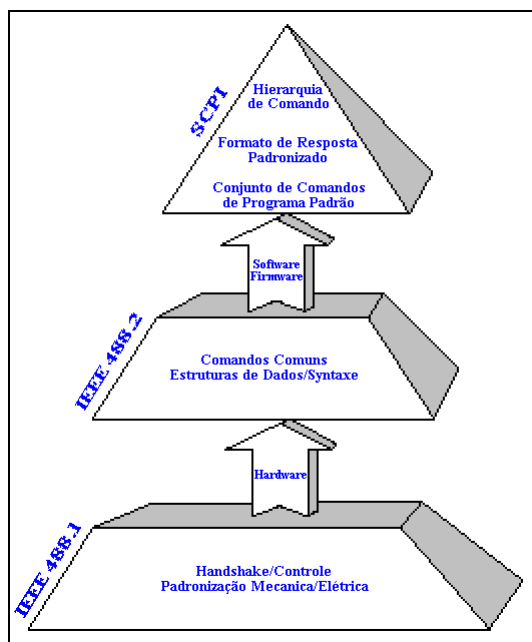


Figura 8: Estrutura dos padrões GPIB

2.3. Fieldbus

O Fieldbus é um padrão de rede local para instrumentação, que usa um barramento digital com comunicação serial, e permite distribuir o processamento entre os dispositivos (sensores, atuadores e controladores). Baseado no modelo OSI (Open Systems Interconnect), as etapas de comunicação são separadas em sete níveis, dos quais o 1, 2 e 7 são usados e o 3, 4, 5 e 6 são substituídos pelo Subnível de Acesso Fieldbus (Figura 9).

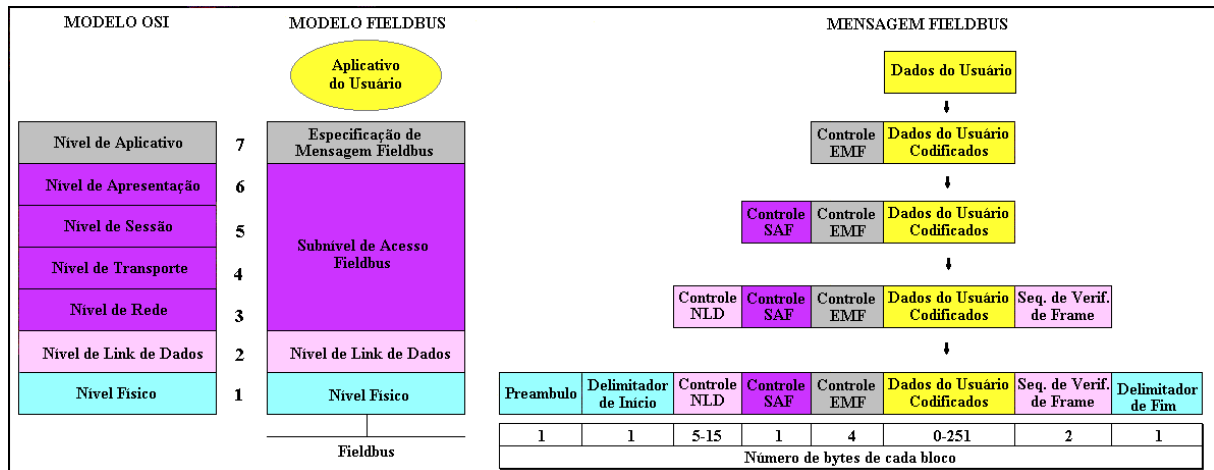


Figura 9: Modelo OSI aplicado ao Fieldbus

No nível Físico, definido pela norma ANSI/ISA-50.02 - Parte 2 [14] Anonimus. *Understanding Foundation Fieldbus Technology*, Fisher-Rosemount Systems, 1997.

[15], a codificação é feita usando a técnica Manchester Biphase-L (Figura 10). No início da mensagem, são enviados 8 bits de preâmbulo para o receptor sincronizar seu relógio interno com o "clock" do sinal. O receptor interpreta uma transição negativa no meio de um ciclo de "clock" como um bit de dados "1", e uma transição positiva como um bit "0".

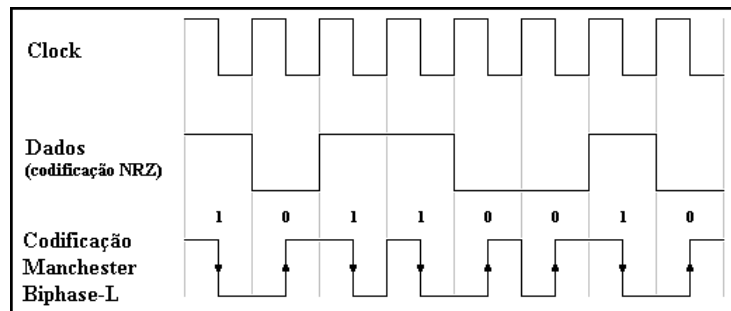


Figura 10: Codificação Manchester Biphase-L

Foram previstos três meios de transmissão: fio, fibra ótica e rádio. A especificação para transmissão usando fios já foi aprovada pela Instrumentation, Systems, and Automation Society (ISA-S50.02). Até 32 dispositivos podem ser ligados em um mesmo barramento serial, sendo que a alimentação dos mesmos pode ser realizada a partir do barramento, ou usando uma fonte própria ou um barramento de alimentação a parte. Três velocidades de transmissão são usadas. A mais lenta (31,25 Kbps) é prevista pelo H1 Fieldbus, e pode ser

usada aproveitando a fiação para 4-20 mA existente. As outras duas (1,0 Mbps e 2,5 Mbps) são previstas pelo H2 Fieldbus, e se destinam a aplicações que requerem uma maior velocidade. Na maioria das aplicações H2 Fieldbus, os dispositivos não são alimentados a partir do barramento serial. O comprimento máximo do cabo é função da velocidade de transmissão, tipo do cabo, bitola do fio, forma de alimentação dos instrumentos e do uso de barreira de segurança intrínseca, sendo de 1900 metros para 31,25 Kbps e 500 metros para 2,5 Mbps.

O nível Link de Dados, definido pela norma ANSI/ISA-50.02 - Partes 3 e 4 [16][17], controla a transmissão de mensagens. Dos três tipos de dispositivos que podem ser ligados a uma rede Fieldbus (Link Master, Basic Devices, ou Bridges), apenas um Link Master pode assumir o controle da rede, gerenciando o acesso ao barramento de forma determinística e centralizada. A transmissão de mensagens pode se dar de forma programada ou não-programada. A comunicação programada, usada para transferências regulares de dados entre dispositivos do Fieldbus. A partir de uma lista de atividades, é enviada uma mensagem Compel Data para cada dispositivo da lista, autorizando-o a usar o barramento para enviar sua mensagem. No intervalo de tempo entre as transmissões de mensagens programadas, o Link Master executa as seguintes tarefas:

- Sincroniza o relógio dos dispositivos, enviando uma mensagem Time Distribution,
- Atualiza a lista de dispositivos ativos (Live List) enviando uma mensagem Probe Node para cada endereço que não conste da lista,
- Dá oportunidade aos dispositivos de executar uma comunicação não-programada enviando uma mensagem Pass Token para cada um dos dispositivos da Live List.

O Subnível de Acesso Fieldbus define três tipos de serviço, através dos quais o nível Especificação de Mensagem Fieldbus acessa o nível Link de Dados:

Client/Server: é usado para comunicação não-programada, iniciada pelo usuário (ex: mudança de *setpoint*, reconhecimento de alarme), entre dois dispositivos. As mensagens são enviadas e recebidas na ordem em que foram submetidas.

Report Distribution: é usado para comunicação não-programada (ex: envio de notificação de alarme ao console de operação, envio de mensagens decorrentes de eventos), ou iniciada pelo

usuário, de um para vários dispositivos. As mensagens são enviadas e recebidas na ordem em que foram submetidas.

Publisher/Subscriber: é usado para comunicação programada, ou não programada, de um para vários dispositivos (ex: atualização dos dados das variáveis do processo). Cada mensagem recebida sobrescreve o valor armazenado em um *buffer*, que corresponde, por exemplo, ao valor de uma variável do processo sendo atualizado em intervalos de tempo regulares. Portanto, apenas o último valor recebido é mantido.

O nível Especificação de Mensagem Fieldbus, definido pela norma ANSI/ISA-50.02 - Partes 5 e 6 [18][19], define os serviços de comunicação, os formatos de mensagens e o comportamento do protocolo. Os serviços de comunicação provêm uma forma padronizada para os aplicativos do usuário se comunicarem através do Fieldbus; eles estão divididos em seis grupos:

- Gerenciamento de Contexto (ex: Initiate - estabelece comunicação, Status - Lê o status de um dispositivo)
- Dicionário de Objeto (ex: GetOD - Lê um dicionário de objeto, PutOD - Carrega um dicionário de objeto em um dispositivo)
- Acesso de Variável (ex: Read - Lê uma variável, DefineVariableList - define uma lista de variáveis)
- Evento (Ex: EventNotification - Reporta um evento, AlterEventConditionMonitoring - Habilita/desabilita um evento)
- Upload/Download (ex: RequestDomainUpload - Requisita um "upload", UploadSegment - Lê dados do dispositivo)
- Invocação de Programa (ex: CreateProgramInvocation - Cria um objeto de programa, Start - Inicia um programa)

O formato das mensagens é definido pela linguagem Abstract Syntax Notation 1 (ASN.1), desenvolvida pela International Telegraph and Telephone Consultative Committee (CCITT). A seguir, é dado um exemplo parcial para o serviço Read [14]:

```
Read_Request ::= SEQUENCE {
    Access-specification CHOICE {
        index [0] IMPLICIT Index
        variable name [1] IMPLICIT Name
        variable-list-name [2] IMPLICIT Name
    },
    sub-index [3] IMPLICIT Subindex OPTIONAL
}
```

Os dados transmitidos pelo Fieldbus são descritos por Object Descriptions agrupados em um Object Dictionary. Para que um dispositivo possa acessar os dados armazenados no Object Dictionary de um outro dispositivo, são usados os Virtual Field Devices. Certos tipos de objetos possuem regras especiais de comportamento, que podem ser controladas remotamente a partir de outro dispositivo.

O "nível" Aplicativo do Usuário, definido pela norma ISA-TR50.02 - Parte 9 [20], é baseado em blocos, sendo eles de três tipos: Resource Block, Function Block e Transducer Block. Cada dispositivo tem um Resource Block que descreve suas características: nome, fabricante, número de série. Um Aplicativo do Usuário pode ter vários Function Blocks (ex: Analog Input - AI, Proportional/Integral/Derivative - PID, Analog Output - AO, Discrete Input - DI), cujos parâmetros de entrada e saída podem estar conectados a outros blocos do mesmo dispositivo, ou a blocos de outro dispositivo ligado ao Fieldbus. As ligações entre os Function Blocks são definidas nos objetos chamados Link Objects. O Virtual Field Device também é usado para acessar remotamente as funções que os dispositivos executam, que são definidas pelas interconexões dos blocos. O Transducer Block executa as funções necessárias para ler sensores (entrada) e comandar atuadores (saída).

A execução dos Function Blocks (e a transmissão de seus parâmetros) de cada dispositivo ocorre de forma cíclica, em intervalos de tempo contados a partir do Absolute Link Schedule Start Time (Figura 11) [14]. Essa execução ocorre de forma sincronizada, pois o Link Master sincroniza periodicamente o relógio dos dispositivos.

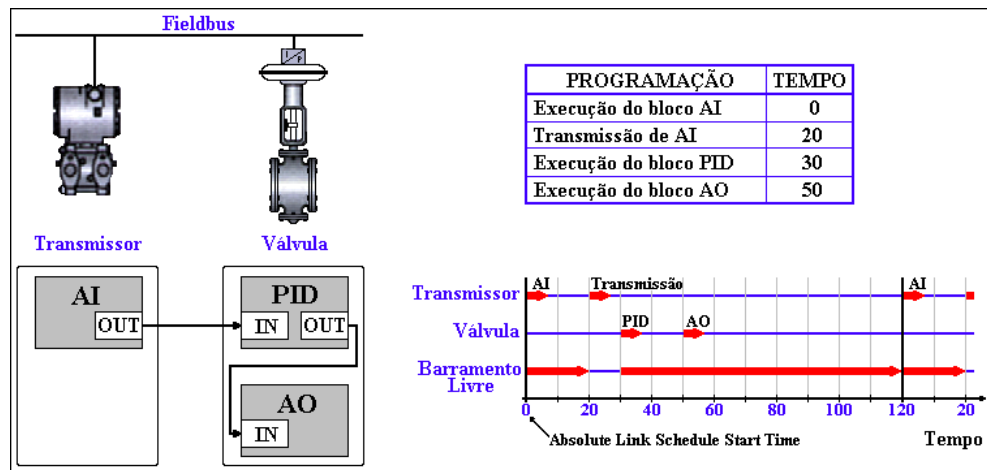


Figura 11: Ciclo de execução dos Function Blocks

Para permitir interoperabilidade, são usados os Device Description, que fornecem informações adicionais como: relação entre os parâmetros, faixa de valores, número de casas decimais, unidade de engenharia. Além dos Device Description para os Function Blocks e Transducer Blocks padrão, existem os Device Description que são disponibilizados pelos fabricantes dos equipamentos, e são chamados de “incrementais”.

2.4. Barramento RS485

O padrão RS485 define apenas as características elétricas de uma interface serial balanceada, deixando em aberto detalhes como a codificação dos bits de dados, taxa de transmissão. A grande imunidade a ruído, aliada a possibilidade de usar linhas de transmissão de baixo custo (par trançado), com comprimento de até 1200 metros, taxa de transmissão de até 10 Mbps, e vários dispositivos conectados a uma mesma linha, fez com que o padrão RS485 seja muito utilizado em aplicações industriais (ex: Profibus, Arcnet, Bitbus, Optomux).

O sinal é transmitido de forma diferencial, podendo ser usado um par de fios, no modo half-duplex, ou dois pares, no modo full-duplex (Figura 12). No modo full-duplex, um dos pares de fios é usado para transmissão e o outro para a recepção, permitindo transmitir e receber dados ao mesmo tempo. No modo half-duplex, o mesmo par de fios é usado para transmitir e receber dados, não permitindo transmissão e recepção simultânea de dados. Como, na área de instrumentação, normalmente é usado um protocolo mestre-escravo, o modo full-duplex não representa ganho na velocidade de transmissão de dados. Isso ocorre pois os dispositivos

“escravos” só transmitem a mensagem de resposta após receber e processar o comando enviado pelo dispositivo “mestre”, que fica aguardando a resposta do dispositivo antes de enviar um novo comando.

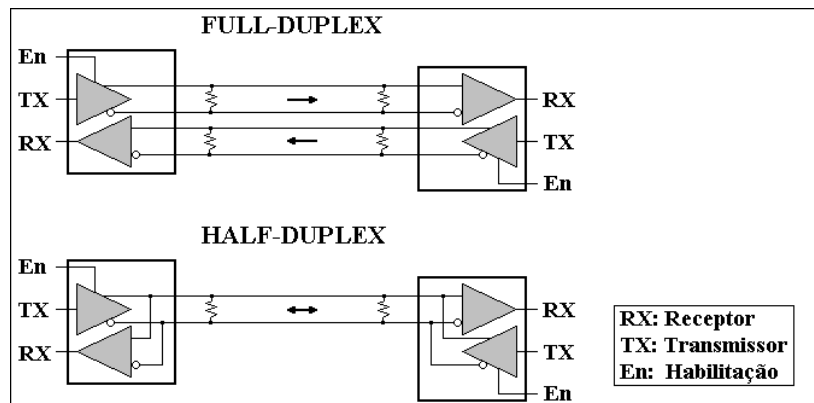


Figura 12: Transmissão Full-duplex e Half-duplex

No modo multiponto, até 32 dispositivos podem ser ligados no mesmo barramento serial (Figura 13). Para isso, a saída de cada transmissor fica desabilitada (em um estado de alta impedância), só sendo habilitada na hora de transmitir dados. Para evitar a reflexão do sinal devido a descontinuidades elétricas na linha de transmissão, são usados resistores terminadores nas extremidades da linha. Normalmente é usado cabo par-trançado, que reduz o ruído induzido na linha.

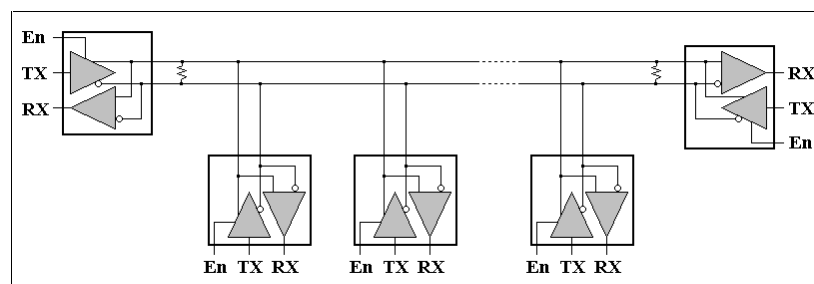


Figura 13: RS485 Multiponto

Padrões como o HART e o Fieldbus incluem a definição do protocolo de comunicação em suas especificações, chegando a detalhar os comandos usados. Como o RS485 define apenas as características elétricas do barramento, detalhes como a codificação dos bits de dados e o protocolo de comunicação ficam em aberto, devendo ser especificados de acordo com a aplicação. Para que todos os dispositivos ligados a um barramento RS485 possam se

comunicar, eles devem adotar a mesma codificação dos bits de dados e o mesmo protocolo de comunicação.

Um protocolo de comunicação define como os dispositivos ligados a uma linha de comunicação serial irão se comunicar. Ele define uma série de regras para que os dispositivos entendam os dados que estão sendo transmitidos e saibam como e quando agir. Fazendo uma analogia com a comunicação via rádio entre pessoas, pode-se observar uma série de "regras" que permitem uma comunicação eficiente. Por exemplo, se duas pessoas começarem a transmitir ao mesmo tempo, provavelmente não se entenderá o que elas estão falando. Quando várias pessoas estão usando uma mesma frequência de transmissão, é necessário informar o nome da pessoa para a qual está se transmitindo a mensagem (Ex: "Atenção Maurício. Qual o valor indicado pelo medidor de vazão? Câmbio."). Da mesma forma, deve-se informar quando acabou a transmissão da mensagem, liberando o canal para outra pessoa usar; isso normalmente é feito transmitindo a palavra "câmbio" no final da mensagem.

Um dos protocolos de comunicação mais utilizados em instrumentação é o Mestre-Escravo. Neste protocolo, um dos dispositivos do barramento faz o papel de Mestre (normalmente o computador de supervisão ou um CLP) enquanto os demais dispositivos se comportam como Escravos do Mestre. O Mestre detém o controle do barramento, e apenas ele pode iniciar uma comunicação; os demais dispositivos (Escravos) se limitam a responder os comandos do Mestre (Figura 14).

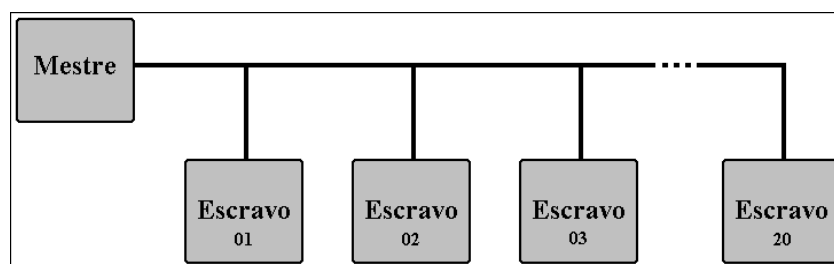


Figura 14: Protocolo Mestre-Escravo

Quando se tem diversos dispositivos ligados em um barramento serial, é necessário atribuir um número diferente a cada dispositivo, que será usado como "endereço" desse dispositivo no barramento. Ele equivale, na analogia feita acima, ao nome da pessoa para a qual se deseja transmitir a mensagem.

O formato de um protocolo de comunicação pode variar muito em função das características do barramento serial, do tipo de acesso e dos dispositivos ligados ao barramento. Diversos protocolos já foram desenvolvidos para as mais diversas aplicações, pois cada um tem características que o tornam mais apropriados para uma determinada aplicação. Para a transmissão de grandes volumes de dados por uma linha telefônica, por exemplo, é desejável que o protocolo usado permita compressão de dados e tenha um bom sistema de detecção e correção de erros de transmissão. Já para pequenos volumes de dados, a compressão aumenta o tempo e a complexidade do processamento, sem apresentar ganho significativo na compactação dos dados.

Ao se transmitir valores numéricos, os dados podem ser simplesmente inseridos no corpo da mensagem, ou podem ser codificados na base decimal, hexadecimal ou octal, usando os caracteres definidos pelo código ASCII (American Standard Code for Information Interchange) [21]. A seguir, serão analisadas as vantagens e desvantagens de cada método.

2.4.1. Protocolo ASCII

O ASCII define os códigos usados para representar cada caractere em uma transmissão de dados (Tabela 2). Essa padronização é desejada quando se está transmitindo mensagens alfanuméricas entre dispositivos, de forma a garantir que os dois dispositivos usarão o mesmo código para representar, por exemplo, a letra "A".

O protocolo ASCII leva esse nome pois todos os *bytes* que compõem o corpo da mensagem são caracteres imprimíveis da tabela ASCII (ex: "A", "5", ">"), e o *byte* que indica o fim da mensagem costuma ser um caractere de controle (ex: CR, LF) também definido na tabela ASCII. A vantagem do protocolo ASCII é que ele é mais fácil de ser implementado, e pode ser testado usando um programa emulador de terminais. Porém, como não pode ser transmitido qualquer valor no corpo da mensagem, o número 200, por exemplo, não pode ser transmitido em um único *byte*. Para transmiti-lo, pode-se usar os caracteres "2", "0" e "0" em seqüência (3 *bytes*), ou pode-se converter o número para hexadecimal (200 = C8h) e transmitir os caracteres "C" e "8" (2 *bytes*).

Tabela 2: Tabela ASCII

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

O protocolo Optomux da Opto 22 [23] será apresentado como exemplo de um protocolo ASCII Mestre-Escravo:

> EE C [DD DD ... DD] CC cr

- “>”: Caractere de início de mensagem
- EE: Endereço em hexadecimal (00-FF)
- C: Comando
- DD: Dados adicionais ao comando
- CC: Checksum
- cr: Caractere de fim de mensagem

O caractere de início de mensagem, como o próprio nome já diz, delimita o início da mensagem. Deve ser usado um caractere que se tenha certeza que não aparecerá no meio da mensagem. No caso do Optomux, os demais campos (Endereço, Comando, Dados e Checksum) usam os caracteres correspondentes às letras (“A”, ..., “Z”) e aos números (“0”, ..., “9”), e o caractere “?”. Portanto, esses caracteres não podem ser usados para delimitar o início da mensagem podendo, a princípio, ser usado qualquer outro caractere imprimível da tabela ASCII. No caso do Optomux, a Opto 22 optou por usar o caractere “>”. Se fosse escolhido o caractere “5”, por exemplo, ao tentar enviar uma mensagem para o dispositivo de endereço 5, o caractere “5” que compõe o endereço seria interpretado como sendo o início de

uma nova mensagem, fazendo com que o dispositivo Optomux de destino não recebesse a mensagem.

Após o caractere de início de mensagem, são enviados dois caracteres com a representação hexadecimal do endereço do dispositivo Optomux. Cada um dos caracteres pode ser um número (“0”, ... ,”9”) ou uma letra (“A”, ... ,”F”), de forma a representar os endereços de 00h a FFh (0 a 255 na base decimal).

O caractere de comando especifica o comando a ser executado pelo dispositivo Optomux. São usados os caracteres correspondentes às letras (“A”, ... , “Z”, “a”, ... , “z”), sendo que o comando “A” (maiúsculo) é diferente do comando “a” (minúsculo). O comando “M”, por exemplo, lê o status do dispositivo. Determinados comandos necessitam de dados adicionais, que são transmitidos no campo Dados. Portanto, o campo Dados tem tamanho variável em função do comando, podendo até não existir em alguns comandos.

O campo Checksum equivale ao dígito verificador de um número de cartão de crédito ou de um CPF. Ele é usado para verificar se não houve erro na transmissão dos dados. O Optomux usa um *checksum* de oito bits que é calculado da seguinte forma:

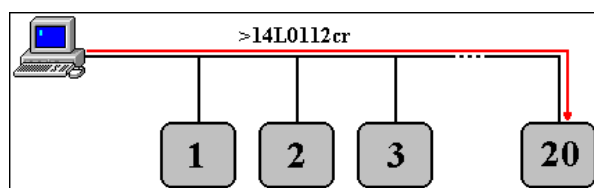
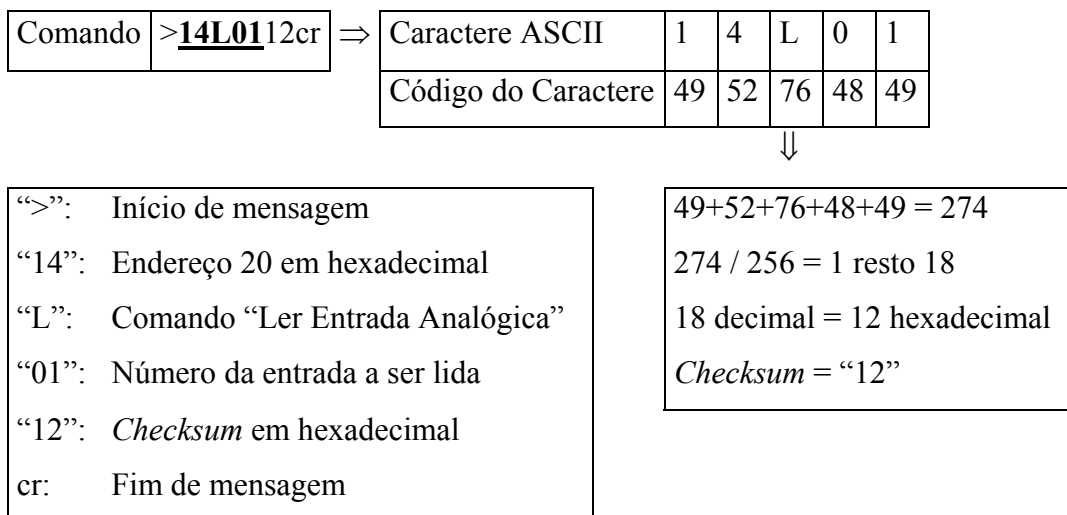


Figura 15: Exemplo de comando Optomux

No exemplo acima (Figura 15), foi usado o comando “L” que lê uma entrada analógica do dispositivo Optomux. Note que apenas os campos Endereço, Comando e Dados são usados para calcular o *checksum*. Os códigos dos caracteres que compõem esses campos são somados, e o valor final da soma é dividido por 256; o resto dessa divisão é então convertido para uma seqüência de dois caracteres que representam o valor do resto em hexadecimal. No exemplo acima, o campo Checksum é composto pelos caracteres “1” e “2” formando o número 12h, que equivale a 18 na base decimal.

Por fim, é enviado o caractere de fim de mensagem, que deve ser diferente do caractere de início de mensagem e dos caracteres usados nos campos de endereço, comando, dados e *checksum*. Como normalmente é usado um programa emulador de terminal para testar o protocolo, o caractere CR (Carriage Return) costuma ser usado para finalizar a mensagem, pois esse caractere é transmitido ao se pressionar a tecla ENTER. Os caracteres de início e fim de mensagem permitem detectar se uma mensagem chegou incompleta. Se for recebido um novo caractere de início de mensagem sem que tenha sido recebido um caractere de fim de mensagem, a mensagem que estava sendo recebida deve ser descartada pois está truncada.

Após receber e processar o comando, o dispositivo Optomux transmite uma mensagem de resposta no seguinte formato:

Comando: >14L0112cr

Resposta: A1000C1cr

“A”: Início de mensagem de resposta (Acknowledgment)

“1000”: Valor da entrada lida (1000-1FFF)

“C1”: Checksum

cr: Fim de mensagem

Caso haja algum erro, a mensagem de resposta tem o seguinte formato:

Comando: >14L0112cr

Resposta: N02cr

“N”: Início de mensagem de erro

“02”: Código do erro (02=Erro de *checksum*)

cr: Fim de mensagem

2.4.2. Protocolo Binário

No protocolo binário, os *bytes* dos campos de endereço, comando, dados e *checksum* podem assumir qualquer valor. Isso significa que, quaisquer que sejam os caracteres que se escolha para delimitar o início e o fim da mensagem, não têm como se garantir que eles não aparecerão no meio da mensagem. Por esse motivo, no lugar de usar caracteres delimitadores de mensagem, o protocolo binário transmite o tamanho da mensagem logo no início da mesma:

E T C [D D ... D] Cs

E: Endereço (0-15)

T: Tamanho da mensagem

C: Comando

D: Dados adicionais ao comando

Cs: Checksum

Apesar da maior dificuldade para implementar, e de não poder usar programas emuladores de terminais para depurar o protocolo, os valores numéricos a serem transmitidos não precisam ser convertidos em uma seqüência de caracteres que representem o valor na base decimal ou hexadecimal. Isso implica que o número de *bytes* a serem transmitidos nos campos Endereço, Dados e Checksum é duas a três vezes menor. Com isso, o tempo de processamento e transmissão da mensagem diminui significativamente.

Como não há caracteres delimitadores de início e final de mensagem, é necessário implementar um temporizador no programa de recepção dos dados, que conte o tempo decorrido a partir da chegada do primeiro *byte* de cada mensagem. Assim, cada mensagem terá um tempo máximo para ser transmitida, e se, ao final desse tempo, o programa não tiver recebido o número de *bytes* especificado no campo Tamanho da Mensagem, a mensagem será descartada. Isso é necessário pois um erro de transmissão no *byte* Tamanho da Mensagem faz com que o programa de recepção não consiga determinar corretamente o ponto de início da

próxima mensagem. Caso não houvesse o temporizador, o programa esperaria por um número de *bytes* diferente do tamanho real da mensagem, e consideraria o *byte* seguinte como sendo o início da próxima mensagem. Ao determinar incorretamente o início da próxima mensagem, o *byte* que será considerado com o tamanho dessa mensagem será, na verdade, um *byte* do campo de endereço, comando, dados ou *checksum*. Isso fará com que sejam cometidos novos erros na determinação do início das mensagens subseqüentes.

Capítulo 3

3. Interface Homem Máquina (IHM)

A Interface Homem Máquina (IHM) de um sistema de aquisição de dados é o dispositivo através do qual o operador interage com o sistema, podendo ser usado, por exemplo: botões, lâmpadas e/ou display de LEDs (ou de cristal líquido) no painel do equipamento, ou um computador com um programa específico. O uso do computador como IHM, além de possibilitar o ajuste de parâmetros do sistema e a visualização dos dados (como em um equipamento com IHM composta de botões, lâmpadas e display), permite compor uma interface com mais recursos. É possível, por exemplo: mostrar os dados sobrepostos a um desenho esquemático do equipamento (ou do processo), mostrar dados de diversos equipamentos em uma mesma tela, apresentar os dados de formas diversas (ex: gráficos, tabelas), transmitir os dados à distância para outros computadores.

Para utilizar os programas de IHM disponíveis no mercado (ex: Elipse 21, The Fix), seria necessário desenvolver um *driver* de comunicação com a Remota, específico para o programa a ser utilizado. Além do custo e da dificuldade de desenvolver esses *drivers* de comunicação, os programas de IHM mais simples disponíveis no mercado (Ex: Elipse 21, The Fix) não permitiam, por exemplo, traçar o gráfico do valor de um transdutor (eixo Y) em função do valor de outro transdutor (eixo X). Foi então desenvolvido um programa de IHM para o ambiente Windows, usando o Borland C++.

3.1. Acompanhamento do Ensaio

A tela principal do programa mostra um desenho esquemático do ensaio, onde são indicados os pontos onde as leituras são realizadas (Figura 16). Ao lado do desenho há pequenos mostradores onde é mostrado o nome da grandeza sendo medida (Carga Axial, Deformação Axial, Pressão Confinante, Pressão Neutra), o nome do arquivo de calibração do sensor (CC005, Res002, Press007, Press008), o canal da Remota onde está ligado o sensor (1, 2, 3, 4) e o valor atual da grandeza medida (14,9 Kgf, 15,65 mm, 105,0 kPa, 50 kPa). Esses mostradores permitem que o operador posicione os sensores antes de começar o ensaio,

aplique as cargas iniciais e acompanhe os valores medidos de forma a saber o momento de parar o ensaio.

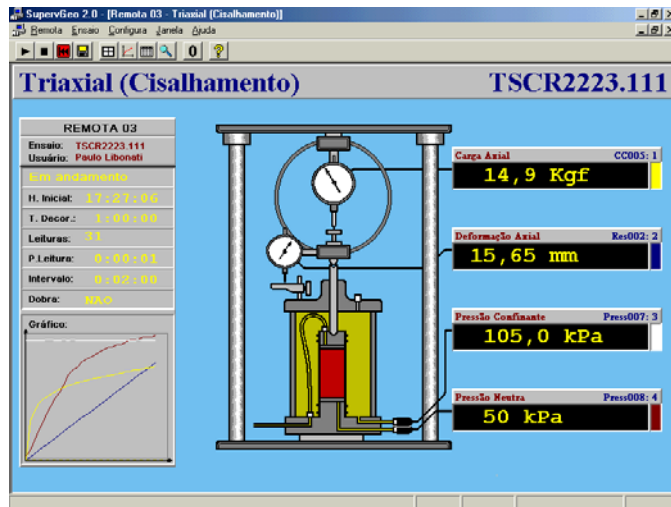


Figura 16: Tela do Ensaio Triaxial (Etapa de cisalhamento)

A tabela posicionada no lado esquerdo da tela mostra os dados relativos à configuração e ao andamento do ensaio. Mostra, também, um pequeno gráfico das leituras realizadas, usado para acompanhar o andamento do ensaio, que pode ser ampliado caso se deseje observar algum detalhe (Figura 17). Outra opção é acompanhar o ensaio através de uma tabela com os valores lidos (Figura 18), de forma a observar a variação do valor de uma grandeza entre duas leituras consecutivas. Em alguns ensaios, o momento de parar é determinado quando a variação de uma das grandezas lidas for menor que um valor determinado pelos parâmetros do ensaio.

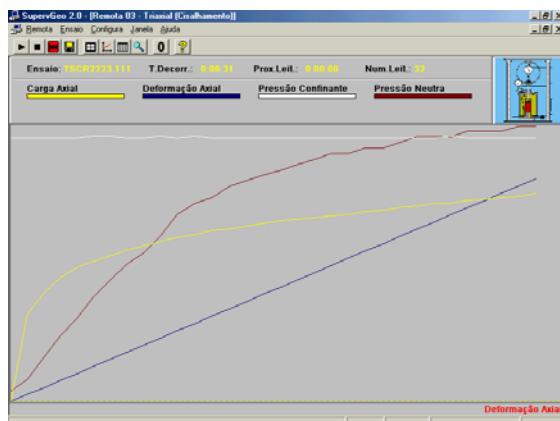


Figura 17: Gráfico do ensaio

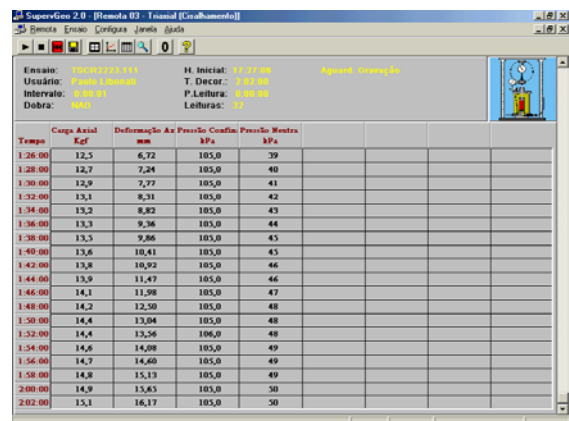


Figura 18: Leituras do ensaio

Quando a Remota está a uma certa distância do computador de supervisão, pode ser necessário ampliar o mostrador de um sensor para que se possa aplicar, por exemplo, a pressão confinante antes de começar um ensaio Triaxial. O programa possui um recurso de “zoom” que permite que o valor do mostrador possa ser observado à distância (Figura 19). Quando há vários ensaios em andamento, pode-se ter uma visão geral dos mesmos, abrindo várias telas de ensaio dentro do programa (Figura 20).



Figura 19: “Zoom” do valor de um sensor

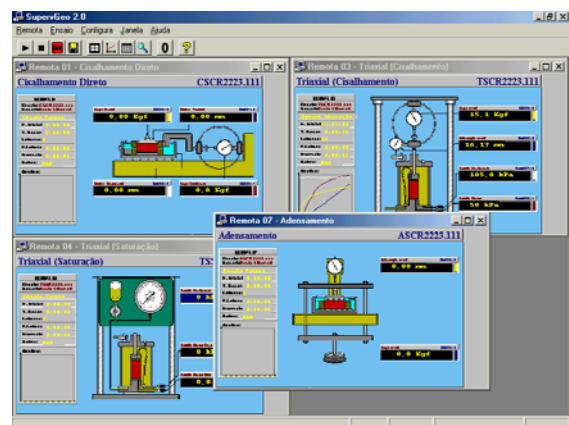


Figura 20: Visão geral dos ensaios

No final, ou mesmo durante o ensaio, os dados podem ser gravados em um arquivo ASCII (Figura 21) de forma que possa ser usada uma planilha eletrônica, ou programa específico, para o tratamento dos dados.

```
"Ensaio: Triaxial (Cisalhamento)"
"Arquivo: C:\REMOTA\TSCR2223.111"
"Usuario: Libonati"
"H.Inic.: 13:40:30"

"Leitura";"Hora (seg)";"Carga Axial (Kgf)";"Deformação Axial
(mm)";"Pressão Confinante (kPa)";"Pressão Neutra (kPa)"
0; 0; 0,0; 0,00; 105,0; 2
1; 1; 6,3; 0,51; 105,0; 4
2; 2; 7,9; 1,01; 105,0; 8
3; 3; 9,0; 1,53; 105,0; 12
4; 4; 9,7; 2,03; 105,0; 15
5; 5; 10,2; 2,55; 106,0; 19
6; 6; 10,6; 3,06; 106,0; 22
7; 7; 11,0; 3,58; 105,0; 25
8; 8; 11,3; 4,08; 105,0; 27
9; 9; 11,6; 4,61; 106,0; 30
10; 10; 11,9; 5,13; 105,0; 34
```

Figura 21: Arquivo ASCII com os dados do ensaio

3.2. Disparo Automático

Em ensaios como o de Adensamento, é necessário iniciar a aquisição dos dados no mesmo instante em que é aplicada a carga à amostra. Como a carga é aplicada através de pesos colocados sobre uma bandeja, o operador não consegue colocar os pesos e disparar o ensaio simultaneamente. Para contornar esse problema, a Remota pode ser configurada para aguardar a variação no valor de um determinado sensor antes de começar a armazenar as leituras. Com isso, quando o operador disparar o ensaio, a Remota entrará em modo de espera (Figura 22), só disparando efetivamente o ensaio depois que detectar uma variação no sensor especificado (Figura 23). O valor da variação necessária para disparar o ensaio (sensibilidade do disparo automático) pode ser configurado no programa (Figura 24).



Figura 22: Ensaio em modo de espera

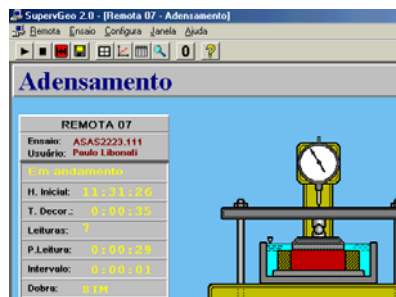


Figura 23: Ensaio em andamento

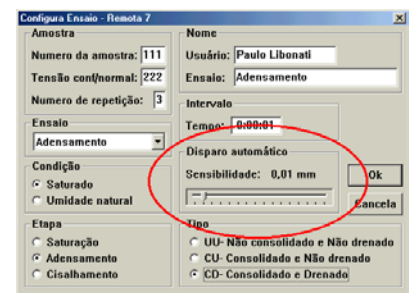


Figura 24: Sensibilidade do disparo automático

3.3. Configuração

Os diferentes tipos de ensaio (ex: Triaxial, Cisalhamento, Adensamento) possuem diferenças no que diz respeito ao número de sensores, intervalo entre as leituras e tipo de sensores. Antes de começar um ensaio, a Remota precisa ser configurada de forma a se adequar a essas diferenças (Figura 25).

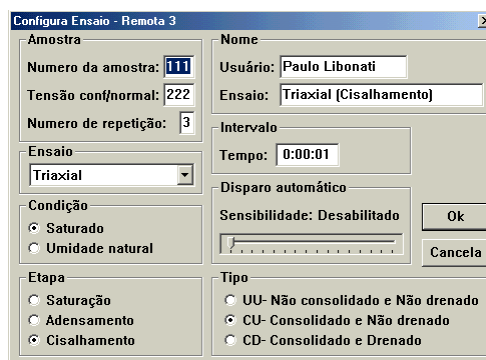


Figura 25: Configuração do ensaio

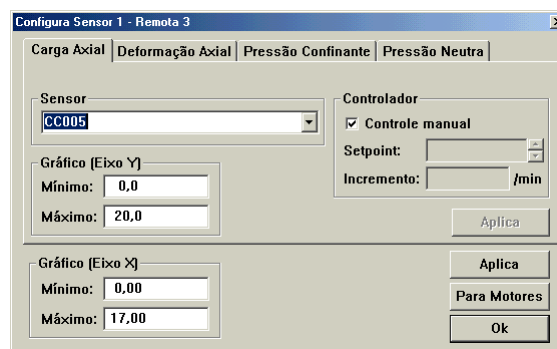
Ao seleccionar o tipo do ensaio (campo Ensaio), o programa automaticamente configura o número de sensores, o sensor usado para o disparo automático, o sensor em função do qual serão traçados os gráficos (eixo X), a forma de incremento do intervalo de tempo (dobra ou não dobra) e o desenho do ensaio. Os campos: Número da amostra, Tensão confinante, Número de repetição, Condição, Etapa e Tipo irão determinar o nome do arquivo no qual serão gravados os dados do ensaio. Foi adotado o seguinte padrão para o nome do arquivo: ①②③④⑤⑥⑦⑧. A B C (Tabela 3).

Tabela 3: Composição do nome do arquivo de dados

Posição	Campo	Significado
①	Ensaio	A - Adensamento B - CBR C - Cisalhamento T - Triaxial
②	Condição de Saturação	S - Saturado N - Não Saturado
③	Etapa do Ensaio	A - Adensamento C - Cisalhamento S - Saturação
④	Tipo do Ensaio	Q - UU (Não consolidado, Não drenado) R - CU (Consolidado, Não drenado) S - CD (Consolidado, Drenado)
⑤⑥⑦	Tensão Normal ou Confinante	Valor da tensão em kPa
⑧	Repetição	Número de repetição do ensaio
A B C	Registro	Número de registro da amostra

Os campos Nome do Usuário e Nome do Ensaio não interferem no andamento do ensaio. São apenas campos informativos para saber quem está realizando um determinado ensaio e qual o ensaio que está sendo realizado na Remota.

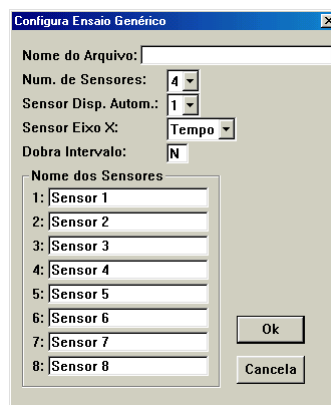
Além da configuração do ensaio, é necessário configurar cada um dos sensores usados no ensaio (Figura 26), de forma que o programa saiba qual o arquivo de calibração usar para converter a leitura do ADC (Analog to Digital Converter) no valor da grandeza física, e qual escala usar para traçar o gráfico. O arquivo de calibração contém os coeficientes da curva de calibração que será usada, além de informações sobre o sensor, a unidade de engenharia na qual ele foi calibrado (ex: mm, Kgf, kPa) e o número de casas decimais que deve ser usado para mostrar o valor lido. O campo Controlador define o *setpoint* e o incremento de pressão/carga a serem usados pela Remota para aplicação de pressão/carga, na amostra.



A janela 'Configura Sensor 1 - Remota 3' possui uma barra de menu com as opções: 'Carga Axial', 'Deformação Axial', 'Pressão Confinante' e 'Pressão Neutra'. O menu 'Carga Axial' está selecionado. O campo 'Sensor' contém o texto 'CC005'. O campo 'Controlador' possui uma caixa de seleção 'Controle manual' marcada com um checkmark. Abaixo, há campos para 'Setpoint:' e 'Incremento:' com uma unidade '/min'. Há dois grupos de campos para 'Gráfico [Eixo Y]' e 'Gráfico [Eixo X]', cada um com campos para 'Mínimo:' e 'Máximo:'. Botões 'Aplica', 'Para Motores' e 'Ok' estão localizados na parte inferior direita da janela.

Figura 26: Configuração dos sensores

Como o Laboratório de Geotecnia também realiza ensaios de pesquisa, muitas vezes é necessário realizar um ensaio que não era previsto pelo programa. Para esses casos, foi adicionada a opção de se configurar um ensaio genérico, que permite configurar os parâmetros que são selecionados automaticamente pelo tipo do ensaio (Figura 27).



A janela 'Configura Ensaio Genérico' possui um campo 'Nome do Arquivo:'. Abaixo, há campos para 'Num. de Sensores:' (valor 4), 'Sensor Disp. Autom.:' (valor 1) e 'Sensor Eixo X:' (valor Tempo). O campo 'Dobra Intervalo:' contém o valor 'N'. Há uma seção 'Nome dos Sensores' com oito campos de texto numerados de 1 a 8, cada um contendo o texto 'Sensor X'. Botões 'Ok' e 'Cancela' estão localizados na parte inferior direita da janela.

Figura 27: Configuração do ensaio genérico

3.4. Programa de Calibração

A Remota converte o sinal elétrico do transdutor usando um ADC de 12 bits. Para que esse valor possa ser mostrado em unidade de engenharia (ex: kPa, mm, Kgf), é necessário ter, para cada transdutor, uma curva de calibração relacionando valores lidos do ADC com os valores da grandeza física aplicada ao transdutor (Figura 28).

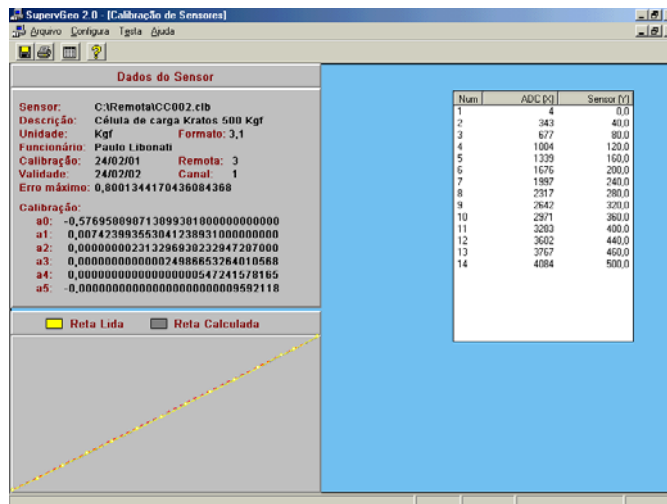


Figura 28: Tela de Calibração

A IHM desenvolvida possui um programa de calibração incorporado, que permite que o próprio usuário possa calibrar os transdutores (ou recalibrá-los com a periodicidade desejada), conhecendo-se apenas a faixa de operação do transdutor. Para isso, usando um equipamento padrão de medição, o usuário deve aplicar valores conhecidos ao transdutor que se deseja calibrar e, para cada valor aplicado, informar esse valor ao programa, que irá ler o ADC e montar a tabela de calibração. A partir da tabela de calibração, são calculados os coeficientes de um polinômio de até quinto grau (pelo método dos mínimos quadrados), é estimado o erro máximo, e é traçado o gráfico dos pontos da tabela (em amarelo) sobreposto ao gráfico do polinômio calculado (em cinza). Os gráficos são traçados sobrepostos de forma a facilitar a visualização da adequação do polinômio aos pontos da tabela; é traçada também uma reta (em vermelho, tracejada) unindo o primeiro ao último ponto da tabela, para que se possa ter uma idéia da linearidade do transdutor (Figura 29).

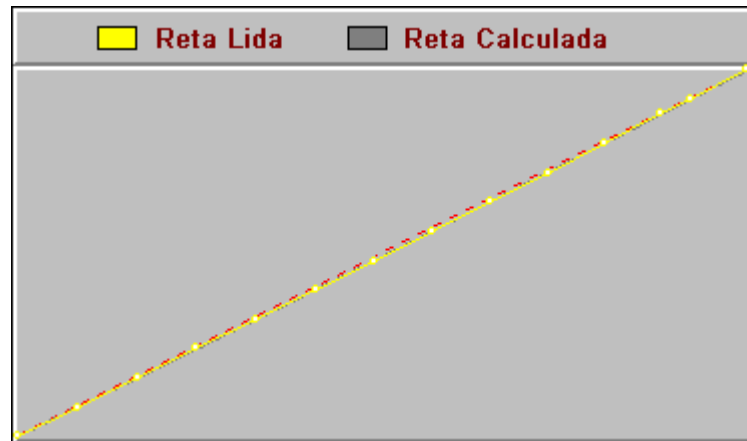
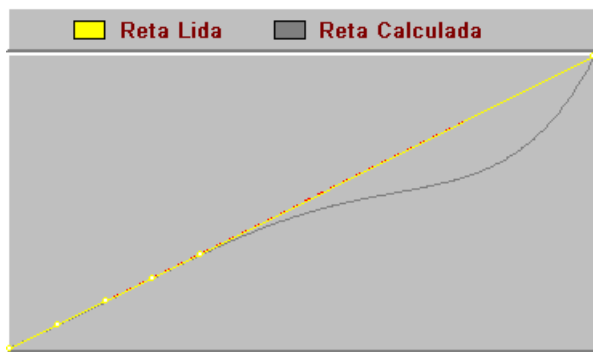
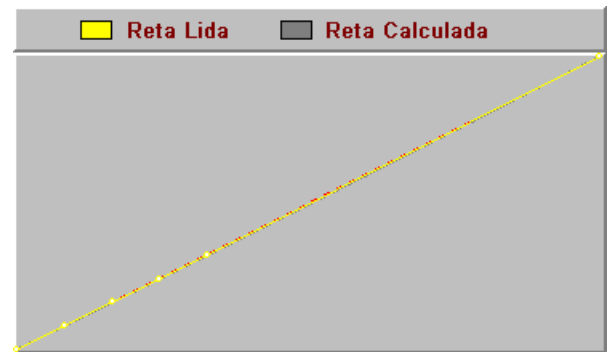


Figura 29: Gráfico de Calibração

O erro máximo é estimado calculando, para cada ponto da tabela de calibração, a diferença entre o valor padrão aplicado e o valor calculado usando o polinômio. Para os pontos interpolados, o erro pode ser maior que o erro máximo estimado, porém não é possível determinar esse erro, pois na tabela não tem informações sobre o comportamento do sinal de saída do transdutor para esses pontos.

Para que a curva de calibração seja a mais representativa possível, deve-se escolher pontos igualmente espaçados (ex: valor padrão aplicado de 20 em 20 kPa) ao longo de toda a faixa do transdutor. Para calcular os coeficientes de um polinômio do quinto grau, são necessários pelo menos seis pontos, porém, quanto maior o número de pontos, mais informações se terá sobre a curva do transdutor, permitindo um melhor ajuste do polinômio. O programa permite que sejam usados até 20 pontos para a determinação do polinômio. Um polinômio do quinto grau permite compensar pequenas não-linearidades do transdutor; porém, se necessário, o programa permite reduzir o grau do polinômio até o grau 1, ou seja, uma reta. Ao se reduzir o grau do polinômio, deve ser observada a linearidade do transdutor, e levado em consideração o erro causado por uma possível não-linearidade.

Se a faixa do equipamento padrão de medição for menor que a faixa do transdutor ou se os pontos não estiverem igualmente espaçados, o polinômio que melhor se ajusta aos pontos da tabela pode não ser o que melhor se ajusta à curva do transdutor (Figura 30). Nesses casos, deve-se reduzir o grau do polinômio de forma que ele melhor se ajuste à curva do transdutor (Figura 31).

**Figura 30: Polinômio do 5º grau****Figura 31: Polinômio do 3º grau**

A validade da calibração é configurada de acordo com normas ou protocolos internos adotados pelo laboratório. Após a data especificada nesse campo, o programa irá lembrar ao operador que o transdutor em questão precisa ser recalibrado. O programa continua funcionando normalmente, porém emitirá constantemente mensagens lembrando ao operador que a calibração do transdutor está vencida.

Todos os dados referentes à calibração (ex: tabela, coeficientes do polinômio, descrição) são guardados em um arquivo com o nome do transdutor. Na configuração do ensaio, ao configurar os sensores, é informado o nome do arquivo de calibração de cada sensor utilizado. O programa de IHM usa os dados contidos nesses arquivos para converter os valores em unidades do ADC, lidos pela Remota, em valores em unidades de engenharia.

3.5. Acesso Remoto

Muitas vezes é necessário acompanhar o andamento de um ensaio a partir de um local diferente do qual foi instalado o sistema de aquisição de dados. Usando o protocolo TCP/IP e a biblioteca WinSock [22], foi possível implementar o acesso remoto, permitindo acompanhar o ensaio a partir de qualquer ponto de uma rede local Ethernet, ou até mesmo de casa (ou de outro local fora do laboratório) pela Internet. Winsock é a implementação da interface de sockets da Berkley (usada no ambiente Unix) para o ambiente Windows. Ela consiste de um conjunto de funções que permite uma padronização no uso de diferentes implementações do TCP/IP.

Um processo que esteja “rodando” em um computador ligado a uma rede TCP/IP é identificado pelo endereço IP do computador onde o processo está “rodando” e o número da porta associada ao processo. Portanto, se um programa desejar disponibilizar um serviço em uma rede TCP/IP, ele deve reservar uma porta e ficar aguardando o acesso do computador remoto. Para usar o modo remoto, o programa deverá estar sendo executado no computador onde está ligado o sistema de aquisição de dados (Figura 32), e no computador do qual se deseja visualizar o ensaio (Figura 33).

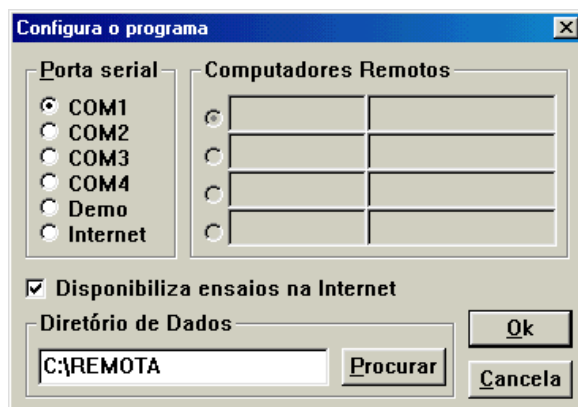


Figura 32: Disponibiliza ensaios na Internet

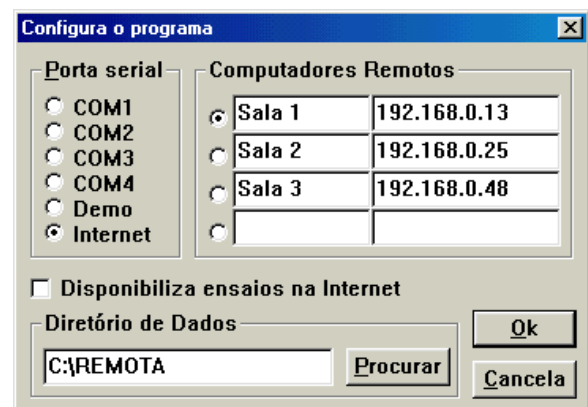


Figura 33: Acessa ensaios pela Internet

Capítulo 4

4. Hardware da Remota

Para desempenhar as funções necessárias à realização dos ensaios, a Remota possui algumas interfaces que podem ser descritas em termos de blocos funcionais (Figura 34). O microcontrolador é o bloco principal, responsável pelo processamento das leituras, temporização e gerenciamento dos demais blocos. A leitura do valor de um sensor pode ser feita através do bloco ADC (Analog to Digital Converter) ou TDD (Transdutor Digital de Deslocamento), a depender do tipo de sensor usado.

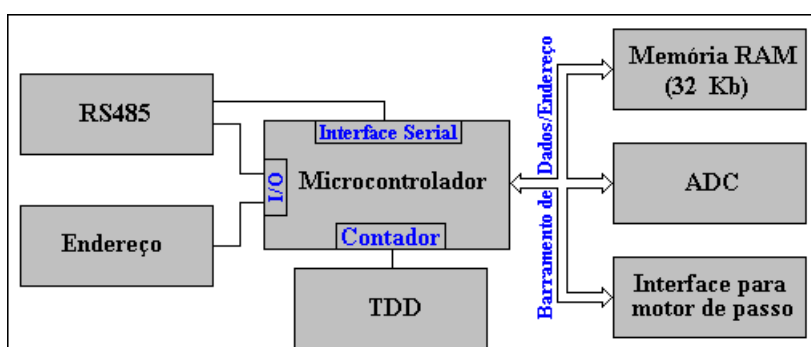


Figura 34: Diagrama de blocos da Remota

O bloco Memória RAM armazena a configuração do ensaio e os valores lidos dos sensores. Durante o ensaio, os dados lidos são armazenados nessa memória, sendo transferidos para o computador de supervisão quando requisitado. Durante o ensaio esses dados são requisitados para traçar os gráficos que permitem o acompanhamento da evolução do ensaio e, no final do ensaio, são transferidos para o computador para armazenamento em disco.

Para permitir a ligação da Remota no barramento de comunicação, o bloco RS485 converte os níveis de tensão TTL da saída do microcontrolador para os níveis de tensão do padrão RS485. O microcontrolador controla o estado de alta impedância da saída desse bloco, habilitando o *driver* RS485 antes de começar a transmitir dados, e colocando-o em alta impedância ao final da transmissão.

O bloco Endereço permite que o usuário defina o número (endereço) pelo qual a Remota será identificada no barramento RS485. É esse número que o programa de IHM usa para poder acessar uma Remota específica ligada ao barramento. Finalmente, o bloco Interface para Motor de Passo permite que a Remota atue no ensaio, controlando o valor de uma grandeza física como pressão, carga ou posição de um objeto.

4.1. Microcontrolador

No projeto foi usado o microcontrolador 89C52 da Atmel, compatível com a família MCS 51 da Intel [24]. O 89C52 é um microcontrolador de 8 bits de uso geral e baixo custo, que possui: uma interface serial, três contadores/temporizadores de 16 bits (T/C0, T/C1 e T/C2), 4 portas de I/O (P0 a P3), 8 Kb de memória FLASH (para armazenar o programa) e 256 *bytes* de memória RAM.

O T/C0 e o T/C1 são configurados como temporizadores de 8 bits com recarga automática (modo 2). O T/C0 é configurado para gerar 3.600 interrupções por segundo, sendo usado como base de tempo para o armazenamento de dados, movimentação dos motores de passo e para a transmissão serial; o T/C1 é usado para gerar a taxa de transmissão da interface serial. O T/C2 é configurado como um contador up/down de 16 bits (modo 1), usado para contar os pulsos gerados pelo TDD.

As portas P0 e P2 são usadas para o barramento de dados e endereço, através dos quais são transferidos dados entre o microcontrolador e os seguintes blocos: Memória RAM, ADC e Interface de motor de passo.

O *firmware* foi escrito em Assembly, pois as linguagens de alto nível disponíveis não permitem uma temporização tão precisa. Além disso, boa parte dos registradores e da memória interna do microcontrolador não fica disponível para o programa, sendo necessário o uso extensivo da memória externa. Há também um aumento da carga de processamento ao se usar uma linguagem de alto nível, limitando o número de tarefas que o *firmware* pode executar simultaneamente.

Ao ligar a Remota, o firmware faz um diagnóstico dos principais blocos (Figura 35). Para teste da memória ROM, todos os seus bytes são somados, e essa soma é comparada com um valor gravado na última posição de memória; caso alguma posição de memória seja corrompida, o valor da soma será afetado. A memória RAM é testada escrevendo-se valores alternados em todas as posições de memória, e verificando seu conteúdo em seguida. O bloco ADC é testado usando uma fonte de tensão de referência ligada a uma das entradas (vide bloco ADC). Caso algum problema seja detectado, o usuário receberá uma mensagem de aviso quando acessar a Remota através do programa IHM.

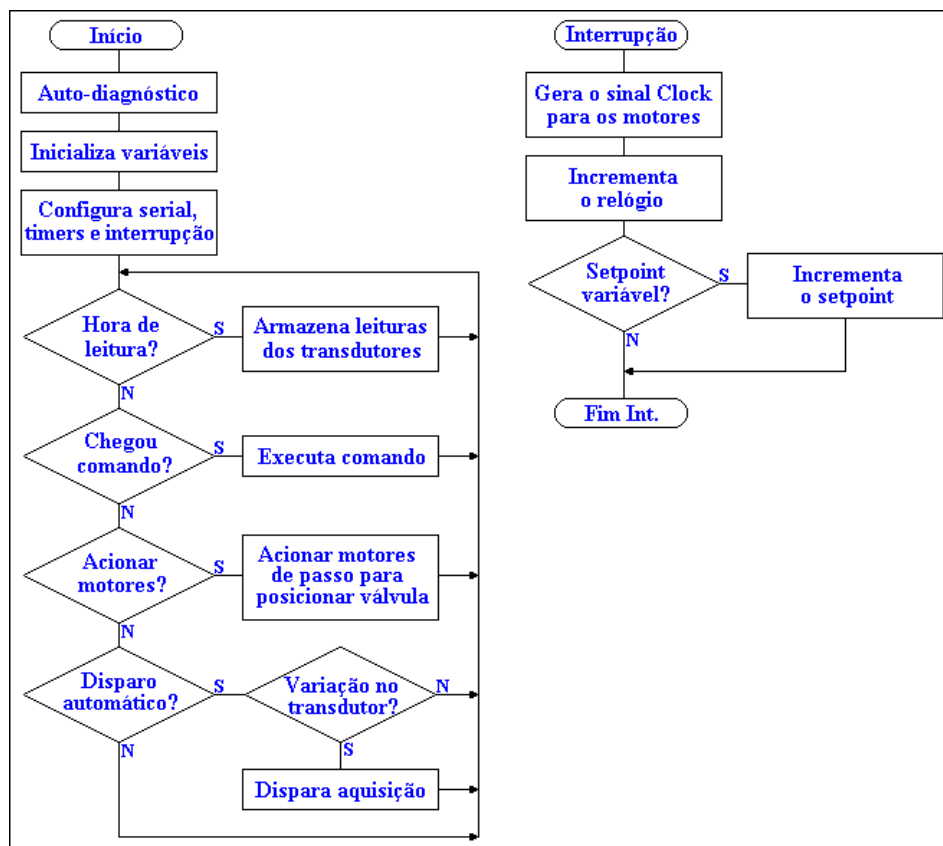


Figura 35: Firmware

Uma dificuldade que se encontra ao desenvolver o *firmware* de um equipamento é que, depois de testado com um simulador, o programa precisa ser gravado na EPROM do microcontrolador para ser testado. Durante o teste, à medida que se encontra erro ou que se aprimora o *firmware*, é necessário apagar e gravar a EPROM diversas vezes. Para contornar esse problema, foi escrito um programa que permite transmitir pela linha serial, para a memória RAM da Remota, o programa a ser testado.

4.2. Memória RAM

A Remota possui 32 Kb de memória RAM estática, usada para armazenar a configuração e os dados do ensaio. Para evitar perda de dados, no caso de falta de energia elétrica, pensou-se em utilizar uma bateria recarregável para manter os dados durante o período em que a Remota estivesse desligada. Porém, como é comum ocorrer vazamento nesse tipo de bateria (e o líquido que vaza acaba danificando o circuito impresso), optou-se por usar um capacitor de valor elevado.

Existe uma nova geração de capacitores de pequenas dimensões ($\varnothing=20\text{mm}$, $A=7\text{mm}$), com capacitância da ordem de 1 Farad, tensão de 5,5 Vdc, para baixas correntes, que são usados como alimentação de backup para memórias RAM estáticas (Figura 36). Em testes efetuados com a Remota, o capacitor manteve os dados por duas semanas.



Figura 36: Capacitor de backup

A leitura de um sensor ocupa dois *bytes* de memória, podendo ser usados até oito sensores em um ensaio. Com isso, tem-se que a memória da Remota é capaz de armazenar 1280 leituras:

$$1280 \text{ (leituras)} \times 8 \text{ (sensores)} \times 2 \text{ bytes} = 20480 \text{ bytes}$$

Portanto, são usados 20 Kb (1 Kb = 1024 *bytes*) para armazenar os dados dos ensaios, sendo que os 12 Kb restantes são usados para armazenar a configuração do ensaio.

4.3. RS485

O bloco RS485 converte os níveis de tensão TTL dos pinos TX e RX do microcontrolador para os níveis de tensão do barramento RS485 half-duplex. O bit 8 da porta P1 é usado para controlar o estado de alta impedância da saída RS485, permitindo que várias Remotas sejam ligadas ao mesmo barramento. O *firmware* da Remota controla o nível lógico desse pino,

ativando a saída RS485 no início da transmissão, e desativando-a ao término da transmissão da mensagem de resposta.

Para ligar o computador de supervisão ao barramento, é necessário o uso de um conversor RS232 para RS485 (Figura 37). Isso se deve ao fato das portas seriais do computador (COM1 a COM4) usarem o padrão RS232 que, apesar de ser um padrão altamente difundido, só permite a interligação de dois dispositivos, com um comprimento máximo de cabo de 10 metros; sendo, portanto, inadequado para um barramento serial. Para evitar a propagação do ruído elétrico do computador no barramento serial, o adaptador RS485 usa fotoacopladores e um conversor DC-DC.

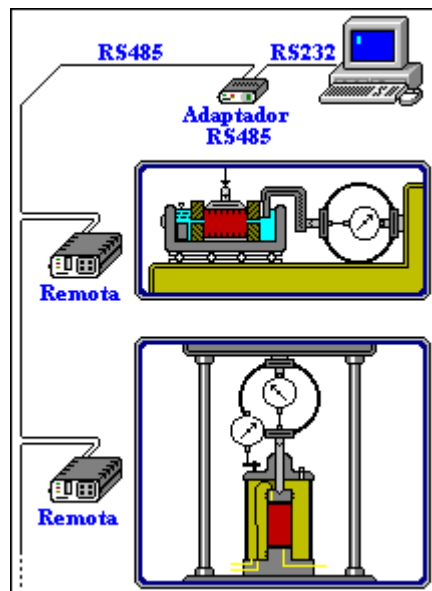


Figura 37: Ligação das Remotas

Com relação à codificação dos bits de dados, foi utilizada a codificação NRZ (apesar da codificação Manchester Biphase-L usada pelo Fieldbus ser superior), pois essa é a codificação usada pela porta serial RS232 dos microcomputadores e pela interface serial da maioria dos microcontroladores. Isso faz com que seja muito mais simples, e muito mais barato, utilizar a codificação NRZ. A taxa de transmissão escolhida para a comunicação serial foi de 9600 baud.

Foi implementado um protocolo mestre-escravo, com os dados sendo transmitidos de forma binária:

Comando:

E T C [D D ... D] Cs

- E: Endereço da Remota (0-15)
- T: Tamanho da mensagem
- C: Comando
- D: Dados adicionais ao comando
- Cs: Checksum

Resposta:

E T Er [D D ... D] Cs

- E: Endereço da Remota (0-15)
- T: Tamanho da mensagem
- Er: Código de erro
- D: Dados adicionais ao comando
- Cs: Checksum

Codigos de erro (Er):

- 00 - Não houve erro
- 01 - Faltou energia na Remota
- 02 - Comando inválido
- 03 - Erro de *checksum*
- 04 - Estouro do *buffer*
- 05 - Não definido
- 06 - Número de parâmetros incorreto
- 07 - Erro ao Disparar/Resetar a aquisição
- 08 - Limites especificados inválidos
- 09 - EPROM defeituosa
- 0A - RAM externa defeituosa

Comandos:

- 00 - Informar a versão e a data do *firmware*

- 01 - Configurar a Remota
- 02 - Ler a configuração guardada na RAM externa
- 06 - Disparar a aquisição
- 07 - Parar a aquisição
- 08 - Resetar a aquisição
- 09 - Zerar o contador do TDD
- 0A - Ler o status do ensaio
- 0B - Ler o valor dos 8 sensores
- 0C - Ler o canal X
- 0D - Ler a aquisição de numero XX
- 10 - Ajustar o *setpoint* da pressão controlada pelos motores
- 11 - Ler o *setpoint* da pressão controlada pelos motores

As mensagens de comando são enviadas pelo programa de IHM (mestre) para as Remotas (escravos) ligadas ao barramento RS485. O comando especificado no campo C deverá ser executado pela Remota cujo endereço é especificado no campo campo E. Caso um comando necessite de dados adicionais, esses dados são colocados no campo D, cujo tamanho é função do número de dados. O comando "Ler o canal X", por exemplo, transmite um *byte* no campo D para informar o número do canal a ser lido; já o comando "Configurar a Remota", transmite 16 *bytes* com os dados referentes à configuração. Uma vez executado o comando, a Remota envia uma mensagem de resposta contendo os dados solicitados ou um código de erro. Para permitir o acesso remoto aos ensaios, os computadores foram interligados através de uma rede Ethernet (Figura 38), que também está conectada à Internet.

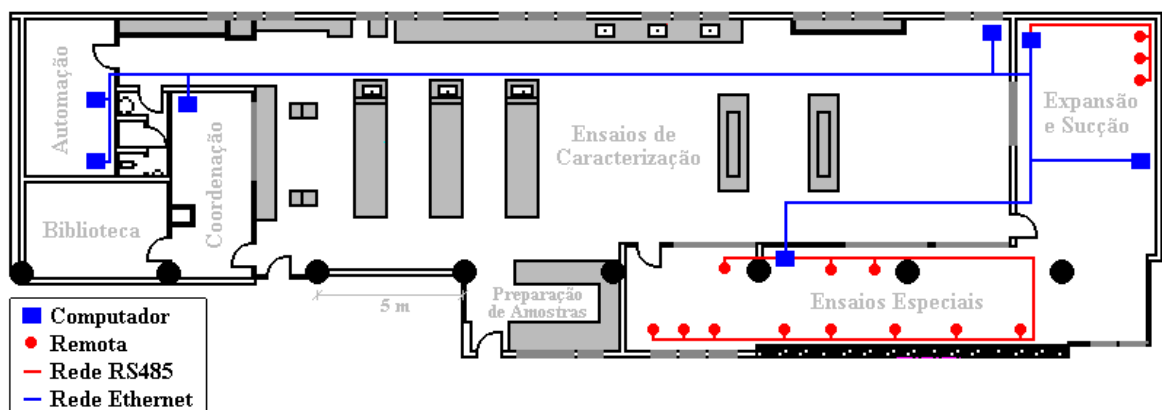


Figura 38: Rede do Laboratório de Geotecnia da UFBA

4.4. Endereço

O endereço da Remota é configurado por meio de uma chave rotativa fixada no painel traseiro da mesma (Figura 39). Esse endereço identifica a Remota no barramento serial, de forma que o computador de supervisão possa acessar cada Remota individualmente. A chave utilizada possui um rebaixo no disco central, de modo que é necessário o uso de uma chave de fenda para mudar o endereço, evitando que o usuário mude o endereço acidentalmente. Como a chave possui 16 posições (identificadas de 0 a F), até 16 Remotas podem ser ligadas a um mesmo barramento RS485.



Figura 39: Painel traseiro da Remota

4.5. ADC

A maioria dos transdutores disponível no mercado tem exatidão de 1% do valor máximo que ele é capaz de registrar (fundo de escala). Um ADC de 8 bits permite leituras com resolução de 1/256 (0,39%), enquanto que no ADC de 12 bits essa resolução é de 1/4096 (0,0244%), e no ADC de 16 bits é de 1/65536 (0,0015%). Levando em consideração a exatidão de 1% dos transdutores, a utilização de um ADC de 16 bits não representa ganho significativo em relação a um ADC de 12 bits (ao menos para transdutores com essa exatidão), pois o valor indicado pelo transdutor já tem um erro de 1%. Já o uso de um ADC de 8 bits, implicaria em um aumento de 39% no erro da leitura ($1\% + 0,39\% = 1,39\%$). Foi, portanto, utilizado um ADC de 12 bits, com conversão por aproximação sucessiva, e tempo de conversão de 35 μ S.

Para permitir a ligação de vários sensores, foi usado um multiplexador analógico de 8 canais, sendo que um dos canais é ligado a um valor conhecido de tensão (3 V), usado para teste da placa ADC (Figura 40). Ao se ligar a Remota, esse canal é lido para verificar se o bloco ADC está funcionando corretamente. Caso o valor lido neste canal esteja fora de uma faixa aceitável, o usuário é avisado através do programa de IHM.

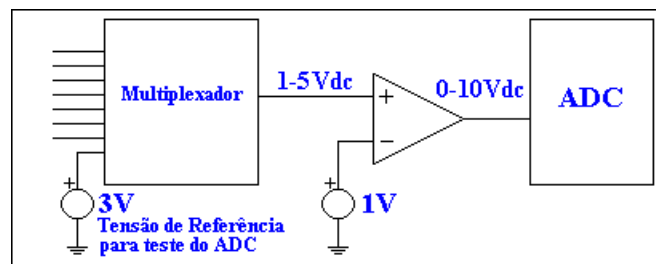


Figura 40: Circuito de entrada do ADC

Foi escolhida a faixa de tensão de 1 a 5 Vdc como padrão para as entradas analógicas, pois ela torna mais fácil a conversão dos diversos tipos de sinais (ex: 4 a 20 mA, 0 a 20mV). Como todas as entradas tem a mesma faixa de tensão, o usuário pode mudar o transdutor sem se preocupar com o tipo de sinal elétrico que ele gera. Para isso, todos os transdutores devem gerar um sinal de 1 a 5 Vdc; sendo que os transdutores que não vierem de fábrica para essa faixa de tensão, devem ter o sinal condicionado externamente.

A faixa de tensão de entrada do circuito integrado ADC utilizado é de 0 a 10 Vdc. Para a conversão do sinal de 1 a 5 Vdc em um sinal de 0 a 10 Vdc, foi usado um amplificador de instrumentação com alta impedância de entrada, ligado após o multiplexador (Figura 40). Foi analisada a possibilidade de utilizar um amplificador com ganho programável, porém só foram encontrados, para aquisição, amplificadores com ganhos correspondentes à potências de 2 (1, 2, 4, 8), que não permitem a conversão das faixas de tensão mais comuns (ex: 1-5 V, 0-10 mV, 0-1V) para 0 a 10 Vdc. Mesmo que houvesse disponibilidade do amplificador, ainda seria necessário um circuito adicional para ajuste do off-set. Por esses motivos, optou-se por uma faixa fixa para a tensão de entrada, sendo o sinal do sensor condicionado externamente quando necessário.

Visando também a facilidade do usuário mudar o transdutor utilizado, foram usados conectores nas entradas do ADC. A mudança de transdutor pode ser freqüente, a depender da

heterogeneidade das amostras a serem ensaiadas. Pode ser necessário, por exemplo, usar uma célula de carga de 50 Kgf para um solo mole, enquanto que, para solos mais resistentes, pode ser necessário usar células de carga de até 5.000 Kgf. A princípio, pode se achar que a célula de carga de 5.000 Kgf atenderia a ambos os casos; porém, a exatidão e a resolução de uma leitura são calculadas em função do valor de fundo de escala do transdutor.



Figura 41: Painel frontal da Remota

Outro detalhe importante, é que os contatos dos conectores devem ser banhados a ouro, de forma a reduzir a resistência elétrica do contato e minimizar a oxidação ao longo do tempo. Como muitos ensaios não usam mais do que quatro sensores, foram feitos dois layouts para o painel frontal da Remota: um com quatro entradas analógicas e uma digital (Figura 41), e outro com 7 entradas analógicas e uma digital. Para alimentação dos transdutores, foram disponibilizadas as tensões de 5 Vdc e 18 Vdc nos conectores de entrada do ADC.

4.6. TDD

Os Transdutores Digitais de Deslocamento (TDD) são dispositivos cujo princípio de funcionamento é o mesmo dos “encoders”. A saída desses transdutores consiste em dois sinais digitais “defasados de 90°”, que permitem detectar o sentido de deslocamento do transdutor. Para serem ligados à entrada do microcontrolador, esses sinais precisam ser convertidos em um sinal de CLOCK, que irá incrementar/decrementar o contador, e um sinal UP/DOWN, que determina o sentido de contagem (Figura 42). Uma vez convertidos, são então ligados às entradas do contador up/down de 16 bits do microcontrolador.

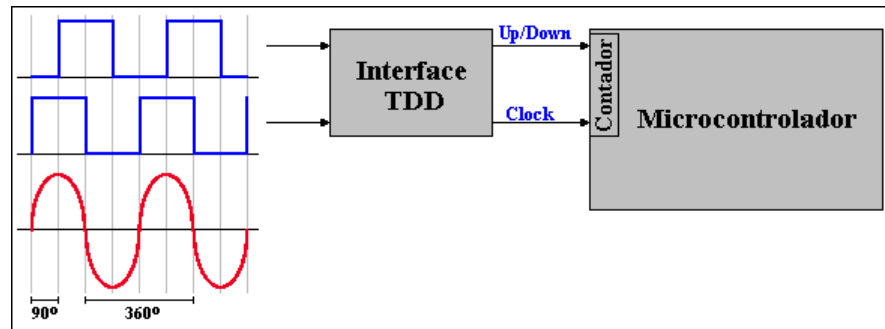


Figura 42: Sinais do TDD

Diversos transdutores têm esse tipo de saída. Como exemplo, temos a linha de apalpadores tipo METRO e os transdutores lineares da Diadur [25]. Os apalpadores tipo METRO são usados nos ensaios em que é necessária uma exatidão de centésimo a milésimo de milímetro.

4.7. Interface para motor de passo

Alguns ensaios em geotecnia necessitam que uma carga constante seja aplicada a uma amostra de solo. A forma usual de se aplicar essa carga é através de pesos conhecidos aplicados a uma bandeja (Figura 43), que pode ter um braço de alavanca para multiplicar a carga. Isso implica em cálculos que levam em consideração o braço de alavanca e o valor dos pesos disponíveis, além de haver o risco de esbarrar no eixo da bandeja ao aplicar a carga, prejudicando o ensaio. Muitas vezes, os pesos se equilibram de forma precária, principalmente quando há necessidade de aplicar cargas crescentes ao longo do tempo, onde pesos de tamanhos diferentes são adicionados a cada etapa do ensaio. Existe também a possibilidade de não se conseguir aplicar toda a carga de uma só vez por não se conseguir levantar todos os pesos ao mesmo tempo.

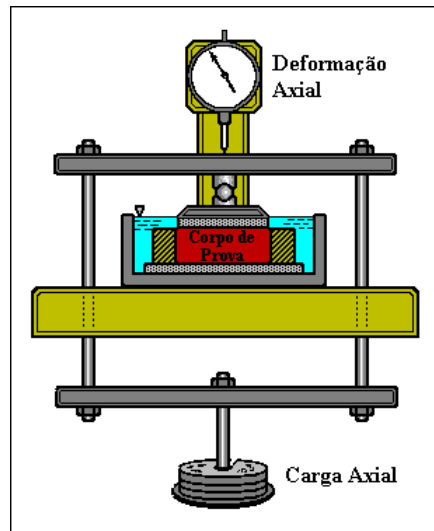


Figura 43: Sistema tradicional de aplicação de carga

Para resolver esse problema, foi desenvolvido um sistema pneumático para aplicação de cargas. Nesse sistema, a Remota aciona um motor de passo, que atua uma válvula reguladora de pressão de forma a abrir/fechar a válvula até se obter a carga desejada (Figura 44). O sistema prevê o acionamento de até três motores de passos, sendo que a saída da válvula reguladora pode, também, ser usada para aplicar pressão a um corpo de prova.

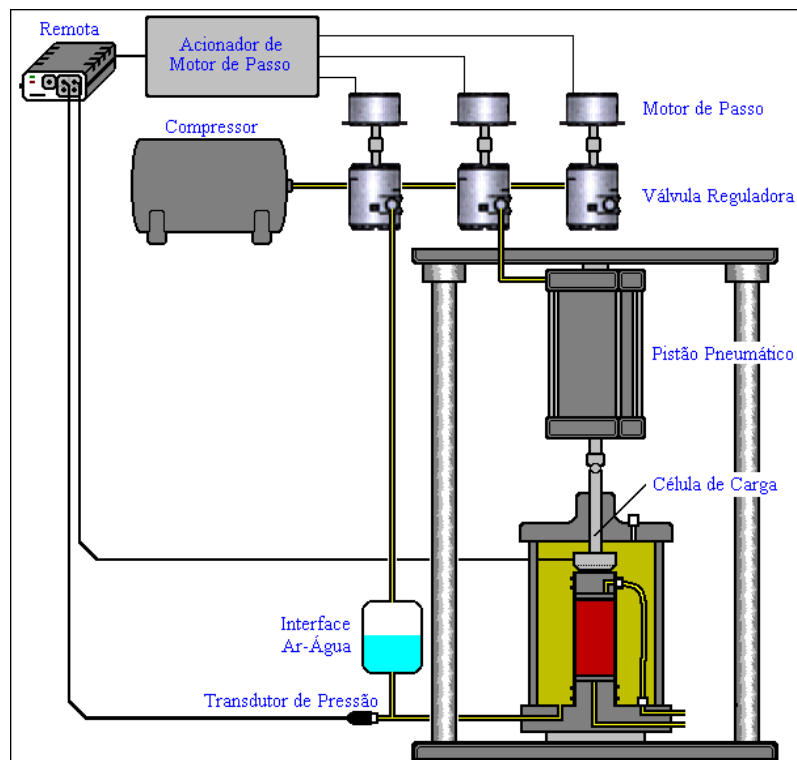


Figura 44: Sistema pneumático de aplicação de carga

Ao configurar os sensores do ensaio, o operador informa o valor da carga que se deseja aplicar à amostra (*setpoint*). A Remota irá, então, monitorar a célula de carga e abrir/fechar a válvula reguladora de pressão de forma a se obter a carga desejada. Como a resposta da válvula reguladora é lenta em relação ao motor de passo, foram usadas duas velocidades para evitar que o motor de passo ultrapasse o ponto desejado, e fique abrindo e fechando a válvula indefinidamente em torno de um ponto. É definida também uma “zona morta” em torno do ponto desejado, na qual a Remota não atua a válvula reguladora. No início do ajuste, o motor de passo atua a válvula reguladora com uma velocidade mais rápida, diminuindo a velocidade ao chegar próximo do valor desejado. Ao chegar no valor desejado, a Remota desenergiza as bobinas do motor de passo, de forma a evitar aquecimento e reduzir o consumo de energia.

Para uma maior velocidade e simplicidade, a Remota trabalha com o *setpoint* em unidades do ADC (valor entre 0 e 4095), o que implica que o *setpoint* em unidade de engenharia (ex: Kgf, kPa) tem que ser convertido para unidades do ADC. Essa conversão é o inverso do que é feito ao ler o valor de um sensor, quando o valor lido do ADC é convertido para unidade de engenharia usando a curva de calibração do sensor. Como, na calibração, é usado um polinômio do quinto grau, a função inversa não é facilmente obtida. Para resolver esse problema, foi usado o processo de aproximação sucessiva, onde o polinômio é calculado 12 vezes para se determinar o valor em unidades do ADC.

Inicialmente o polinômio é calculado para um valor de entrada de 800h (10000000000b) que corresponde a metade da faixa do ADC (0-FFFh). Se o resultado do cálculo do polinômio for maior que o valor do *setpoint*, e a curva correspondente ao polinômio for crescente, o novo valor de entrada para o cálculo do polinômio será 400h (010000000000b); senão, o novo valor será C00h (110000000000b). Esse processo é executado 12 vezes, uma vez para cada bit do ADC. O mesmo esquema de conversão é usado para o valor do incremento de carga, quando é necessário aplicar uma carga variável à amostra. O valor da carga é incrementado, do valor desejado, uma vez por minuto.

A etapa de potência do circuito acionador dos motores de passo é acondicionada em um painel, junto com as válvulas e os motores de passo. A Remota gera os sinais de CLOCK e Sentido de Rotação (disponíveis através do conector “Controlador” localizado no painel traseiro da Remota - Figura 39), e os circuitos integrados L297 e L298, a partir desses sinais,

acionam um motor de passo bipolar [26] com passo de $1,8^\circ$. Um monoestável é usado para desenergizar os motores na ausência do sinal de CLOCK (Figura 45). Para evitar propagação do ruído gerado no acionamento dos motores, foram usados foto-acopladores nas linhas de CLOCK e Sentido de Rotação.

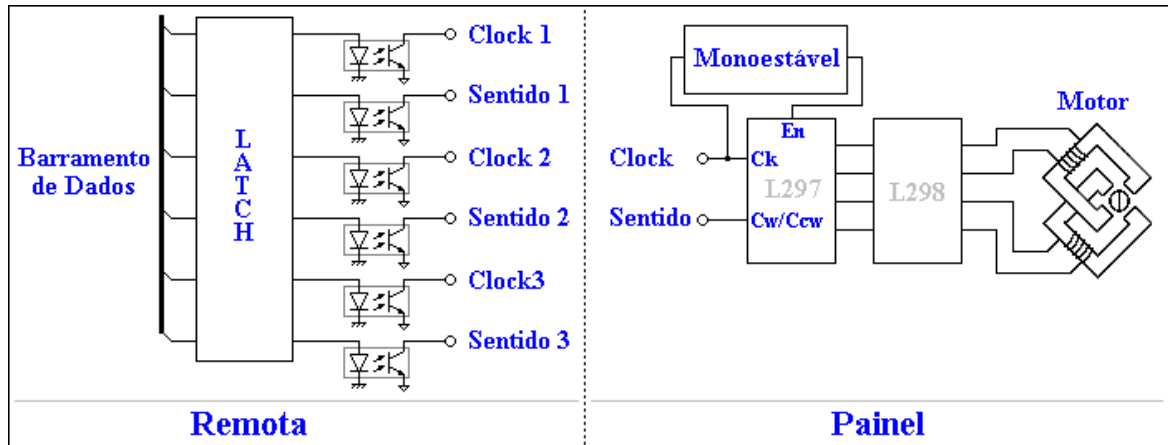


Figura 45: Interface para motor de passo

Capítulo 5

5. Conclusões

Padrões de instrumentação como o HART, GPIB e o Fieldbus, são de grande utilidade quando é necessário interligar equipamentos de diversos fornecedores, ou acrescentar novas funções ao sistema; porém, podem aumentar a complexidade e o custo do sistema, principalmente em aplicações pequenas. O mesmo vale para os sistemas de aquisição de dados, onde, nem sempre o equipamento mais sofisticado é o mais apropriado, pois ele pode ter sido desenvolvido para atender uma determinada área de aplicação (ex: petroquímica, indústria automotiva), não se adequando às necessidades de outras áreas. A depender da aplicação, mesmo os sistemas genéricos, projetados para atender a um grande número de aplicações, podem não ser apropriados por implicar em um custo e uma complexidade elevados, devido à necessidade de usar vários módulos. Pode, também, ser necessário usar equipamentos de mais de um fornecedor, nem sempre compatíveis entre si.

Existem, no mercado, diversos pacotes para desenvolvimento de IHM (Ex: FixDmacs, Genesis, Eclipse 21), que simplificam bastante o trabalho de elaboração das telas e de implementação de funções como log de dados, gráficos de tendência, intertravamento e controle PID. Porém, às vezes é necessário implementar recursos não previstos por esses pacotes, sendo necessário desenvolver a IHM em uma linguagem de alto-nível como C++ ou Delphi.

Um sistema de aquisição de dados devidamente projetado faz com que haja um ganho em produtividade por se ter um sistema adequado às necessidades, e mais simples de operar. O técnico de laboratório tem seu tempo otimizado podendo, inclusive, conduzir diversos ensaios simultaneamente. O desenvolvimento de equipamentos, em alguns casos, se justifica pelo fato de se ter um sistema dedicado, mais simples e, portanto, mais barato. Além disso, por haver um domínio sobre o sistema, é possível alterar as características ou adicionar novos recursos, de acordo com a necessidade, não ficando preso às limitações de um sistema comercial proprietário.

No projeto de um sistema de aquisição de dados, além da escolha dos equipamentos apropriados, deve-se levar em consideração aspectos como: comprimento dos cabos, ruído elétrico, exatidão e faixa dos transdutores e calibração dos transdutores. Os cabos devem ser blindados, para evitar a indução de ruído, e o sistema deve ser aterrado em uma malha de terra separada do aterramento usado para os demais equipamentos elétricos (principalmente motores e outros equipamentos de carga elevada). Deve-se observar, também, a disposição de equipamentos que gerem campos eletromagnéticos, procurando manter-los afastados do sistema de aquisição de dados. O uso de um sistema distribuído, com barramento digital para a transmissão dos dados, reduz bastante o ruído elétrico. Isso ocorre devido à diminuição do comprimento dos cabos dos sensores, que são ligados a unidades remotas localizadas próximo a cada ensaio.

Os sensores devem ser especificados de acordo com a faixa de valores a serem medidos, pois sua exatidão é função do máximo valor que ele é capaz de medir. Muitas vezes é necessária a troca do sensor em função das propriedades do material a ser ensaiado. É importante, também, que se defina uma política de calibração, onde seja prevista uma calibração periódica dos sensores, a partir de um equipamento padrão de medição.

O sistema desenvolvido, descrito nesse trabalho, é hoje usado pelo Laboratório de Geotecnia da UFBA, que utiliza quatorze Remotas para realização de ensaios com fins de pesquisa, ensino e extensão. Os laboratórios de Geotecnia Ambiental e de Madeira, da UFBA, e o Laboratório de Geotecnia da UFPE também possuem uma Remota (cada um deles) para a realização de ensaios de Pesquisa. O Laboratório de Geotecnia da PUC-Rio usa três Remotas para ensaios de pesquisa, e a empresa LCL, de Salvador, usa uma Remota para ensaios de laboratório em amostras de solo.

5.1. Futuro

Em uma versão futura da Remota, se pretende usar um ADC de 16 bits, não para aumentar a resolução das leituras, mas para aumentar a faixa da tensão de entrada da Remota. Na versão atual, a Remota é capaz de medir tensões de 1 a 5 Vdc com uma resolução de 12 bits. A idéia é usar um ADC de 16 bits com uma faixa de -10 a 10 Vdc:

Tabela 4: Comparação entre ADC de 12 bits e ADC de 16 bits

ADC	12 bits	16 bits
Faixa de entrada	1 a 5 Vdc	-10 a 10 Vdc
Resolução	$\frac{5-1}{4096} = 0,9765625mV$	$\frac{10-(-10)}{65536} = 0,30517578125mV$

Na Tabela 4, pode-se observar que, mesmo com uma faixa de entrada maior (-10 a 10 Vdc), o ADC de 16 bits apresenta uma resolução maior que o ADC de 12 bits. Isso significa que pode-se continuar a usar os transdutores de 1 a 5 Vdc (ou 4 a 20 mA) com uma resolução 3 vezes maior que a do ADC de 12 bits, e ainda ter a flexibilidade de poder ligar transdutores com outras faixas de tensão de saída (ex: LVDT – Linear Variable Differential Transformer com saída de -10 a 10 Vdc, transdutores de temperatura com saída de 0 a 1 Vdc).

Outra implementação que se pretende fazer, é dotar as Remotas com relógio em tempo real (RTC – Real Time Clock) de forma que a aquisição de dados possa continuar após uma falha no fornecimento de energia elétrica. Esse recurso não será útil para todos os ensaios pois alguns deles dependem de motores elétricos para a realização do ensaio, e uma parada dos motores implica em perda do ensaio.

É desejável também, adequar a Remota para uso em ensaios de campo, onde há a necessidade de uma maior robustez mecânica (contra choques mecânicos, umidade e poeira), menor interferência da temperatura nas leituras realizadas (pois, no campo, a variação de temperatura é maior que no laboratório) e alimentação por meio de baterias. Para isso, é necessário o uso de ADCs com auto-calibração, de forma a compensar os efeitos da variação de temperatura. Para reduzir o consumo da Remota, permitindo uma maior duração das baterias, é necessário implementar um gerenciamento de energia, colocando o microcontrolador em modo de baixo consumo (stand-by) sempre que possível, e desligando os sensores e a parte analógica no intervalo entre duas leituras.

Aproveitando a capacidade de processamento da Remota, podem ser acrescentados transdutores de temperatura, de forma que a Remota possa compensar a variação do sinal dos transdutores a ela ligados, em função da variação da temperatura.

Referências Bibliográficas

- [1] HEAD, K. H. *Manual of Soil Laboratory Testing*, Pentech Press, 1984. v.3, cap. 20.
- [2] HEAD, K. H. *Manual of Soil Laboratory Testing*, Pentech Press, 1984. v.2, cap. 12.
- [3] HEAD, K. H. *Manual of Soil Laboratory Testing*, Pentech Press, 1984. v.2, cap. 14.
- [4] HEAD, K. H. *Manual of Soil Laboratory Testing*, Pentech Press, 1984. v.2, cap. 11.
- [5] SERRANO, J., RAVIER, D., FRAGA, J., ORJALES, V., MOLANO, A. *GEODAS: An Industrial Experience with Component Frameworks for Data Acquisition and Analysis Systems*. In: Euromicro Conference, 2001. Proceedings 27th, 2001. New York: Institute of Electrical and Electronics Engineers, 2001, p.72-79
- [6] RAMOS, H. G., TRINDADE, M., SERRA, A. C. *A Measurement System to Automate Laboratory Testing in Soils*. In: IEEE Instrumentation and Measurement Technology Conference, 1997, Ottawa. New York: Institute of Electrical and Electronics Engineers, 1997, p.1048-1051
- [7] JORDAN, J. R. *Serial Networked Field Instrumentation*, John Willey & Sons, 1995.
- [8] AMEUR, S., LAGHROUCHE, M., ADANE, A. *Monitoring a Greenhouse Using a Microcontroller-based Meteorological Data-acquisition System*. *Renewable Energy* 24, 2001, p.19-30
- [9] HELSON, R. *The Benefits of HART Protocol Communication in Smart Instrumentation Systems*, HART Communication Foundation.
http://www.smar.com/PDFs/Catalogues/Hart_Tutorial.pdf
- [10] ANSI/IEEE Standard 488.1-1987. *IEEE Standard Digital Interface for Programmable Instrumentation*. Institute of Electrical and Electronics Engineers, Inc

- [11] ANSI/IEEE Standard 488.2-1987. *IEEE Standard Codes, Formats, Protocols, and Common Commands*. Institute of Electrical and Electronics Engineers, Inc
- [12] DEHNE, T. e GOLLA, L. *Increasing Test System Compatibility and Productivity with IEEE488.2*, National Instruments Corporation, 1993
- [13] Anonimus. *GPIB Hardware and Software Specifications*, National Instruments Corporation, 2001
- [14] Anonimus. *Understanding Foundation Fieldbus Technology*, Fisher-Rosemount Systems, 1997.
- [15] ANSI/ISA-50.02, Part 2-1992 - *Fieldbus Standard for Use in Industrial Control Systems Part 2: Physical Layer Specification and Service Definition*
- [16] ANSI/ISA-50.02, Part 3-1997 - *Fieldbus Standard for Use in Industrial Control Systems Part 3: Data Link Service Definition*
- [17] ANSI/ISA-50.02, Part 4-1997 - *Fieldbus Standard for Use in Industrial Control Systems Part 4: Data Link Protocol Specification*
- [18] ANSI/ISA-50.02, Part 5-1998 - *Fieldbus Standard for Use in Industrial Control Systems Part 5: Application Layer Service Definition*
- [19] ANSI/ISA-50.02, Part 6-1998 - *Fieldbus Standard for Use in Industrial Control Systems Part 6: Application Layer Protocol Specification*
- [20] ISA-TR50.02, Part 9-2000 - *Fieldbus Standard for Use in Industrial Control Systems: User Layer Technical Report*
- [21] ANSI X3.4-1986, *American National Standard Code for Information Interchange Coded Character Set – 7-bit*.
- [22] DUMAS, A. *Programando WinSock*, Axcel Books, 1995, cap. 3.

[23] OPTO 22. *Optomux Protocol User's Guide*, Temecula: Opto 22, 2000.

[24] SILVA JR., V. P. *Microcontrolador 8051 Hardware e Software*, Érica, 1994.

[25] DIADUR. *Catálogo Geral*, São Paulo: Diadur Indústria e Comércio Ltda, 1993.

[26] SAX, H. *Stepper Motor Driving*, SGS-Thompson Microelectronics, 1995.