UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE FÍSICA E QUÍMICA DE SÃO CARLOS

"DESENVOLVIMENTO DE UM KERNEL
PROGRAMÁVEL DE CONTROLE, COM
APLICAÇÕES EM TELEPLUVIOMETRIA
E AUTOMAÇÃO DE EXPERIMENTOS
DE PCR"

João Eduardo Machado Perea Martins

*Dissertação apresentada ao Instituto
de Física e Química de São Car-
los, Universidade de São Paulo, para
obtenção do Título de Mestre em
Ciências - Física Aplicada, na sub-
área de Física Computacional*

ORIENTADOR: *Prof. Dr. Álvaro Garcia Neto*

DEPARTAMENTO DE FÍSICA E CIÊNCIA DOS MATERIAIS

SÃO CARLOS

1993

**IFQSC UNIVERSIDADE
DE SÃO PAULO**
Instituto de Física e Química de São Carlos

Fone (0162) 72-6222
Fax (0162) 72-2218

MEMBROS DA COMISSÃO JULGADORA DA DISSERTAÇÃO DE MESTRADO DE JOAO EDUARDO MACHADO PEREA
APRESENTADA AO INSTITUTO DE FÍSICA E QUÍMICA DE SÃO CARLOS, DA UNIVERSIDADE DE SÃO PAULO
01/10/1993

Av. Dr. Carlos Botelho, 1465
CAMPUS Postal 369
CEP 13560-970 - São Carlos - SP
Brasil

COMISSÃO JULGADORA:

Prof. Dr. Alvaro Garcia Neto

Prof. Dra. Maria Stela Veluco de Paiva

Prof. Dr. Noritsuna Furuya

SERVIÇO DE BIBLIOTECA E INFORMAÇÃO - IFQSC
FÍSICA

Agradecimentos

Ao meu orientador, Prof. Dr. Álvaro Garcia Neto, pela amizade, pelo apoio no desenvolvimento desta dissertação e pelo inegável exemplo de dedicação ao trabalho.

Aos meus pais, José Perea Martins e Ercília Machado Perea Martins, pelo apoio e incentivo dados durante todo o curso de mestrado.

Ao Dr. Roberto Vicente Calheiros, Diretor do Instituto de Pesquisas Meteorológicas (IPMET) - UNESP, pelo incentivo ao desenvolvimento do aplicativo em pluviometria.

Ao técnico Lírío Onofre B. de Almeida, do Laboratório de Instrumentação Eletrônica (LIE) do IFQSC-USP, pela atenção e pelas valiosas sugestões relativas à implementação prática do trabalho.

Aos técnicos do LIE: Marcos Roberto Gonçalves, Ivanilda H. Zucolotto, Ailton B. Alves e Maria Benedita S. G. Silva, pelo senso profissional com que colaboraram nas atividades de laboratório.

Ao eng. Luiz F. Matteo Ferraz, do grupo de crescimento de cristais, pela troca de idéias e pelas importantes sugestões práticas.

Ao técnico Ernandes Aparecido Saraiva, do IPMET, pelo apoio no desenvolvimento do projeto de pluviometria.

À secretária Lia Mara C. Chinaglia, do grupo de instrumentação e informática, pela atenção e pelo importante serviço de secretaria.

À Wladeres Aparecida G. Caiado, do setor de pós-graduação, pela constante atenção e espírito de colaboração.

Ao programador Marcos Ferrari dos Santos, do polo computacional da UNESP, campus de Bauru, pela colaboração na editoração gráfica.

Aos amigos: Elisabete Rubo, João A. Martini, Marcos Cavenaghi e Regina F. Eto, pela amizade e apoio durante o desenvolvimento deste trabalho.

A todos que direta ou indiretamente colaboraram para o desenvolvimento deste trabalho.

Conteúdo

Introdução	1
Capítulo I	3
Princípios de aquisição de dados	3
1.1 Introdução	3
1.2 A ciência da instrumentação	3
1.3 Profissionalização	5
1.4 Dispositivos de sensoriamento	6
1.4.1 Classificação	7
1.4.2 Sensibilidade e linearidade	9
1.4.3 Precisão e exatidão	10
1.5 Calibração e erros	11
1.5.1 Calibração	11
1.5.2 Tipos de erros	12
1.5.3 Erro de reprodutibilidade	12
1.5.4 Erros intrínsecos	13
1.5.5 Erros aleatórios	13
1.5.6 Erros por ruídos	13
1.5.7 Erros por fatores humanos	14
1.5.8 Erro de leitura	15
1.6 Aquisição de dados	15
1.7 Sistemas para aquisição de dados	17
1.8 Transferência de dados	18
1.9 Telemetria	19
Capítulo II	22
A estrutura de um computador genérico	22
2.1 Introdução	22
2.2 O conceito de computador	22

2.3 A organização de um computador.....	22
2.4 A Unidade Central de Processamento (UCP)	24
2.4.1 Descrição da UCP	24
2.4.2 Funcionamento da UCP	26
2.5 Microcontroladores	28
2.5.1 A escolha de um microcontrolador	30
2.6 O software.....	31
2.6.1 Ferramentas de software.....	33
2.7 Memórias	34
2.7.1 Memória flash.....	35
2.8 Interfaceamento.....	36
2.9 Interrupções	37
2.10 Dispositivos de temporização.....	40
2.11 Comunicação de dados	41
2.11.1 Modos de transmissão	42
2.11.2 A interface serial.....	43
 Capítulo III	 45
O kernel programável de controle	45
3.1 Introdução.....	45
3.2 O hardware do kernel	45
3.2.1 A arquitetura de Harvard	46
3.2.2 As memórias do kernel	47
3.2.3 Velocidade de processamento	49
3.2.4 O teclado.....	49
3.2.5 O visor	50
3.2.6 Entradas de sinais digitais	51
3.2.7 Entradas analógicas	52
3.2.8 Temporizadores.....	52
3.2.9 A interface serial.....	53
3.2.10 O microcontrolador usado no kernel	54
3.3 O software do kernel	57

Capítulo IV	60
Aplicações em pluviometria e PCR	60
4.1 Introdução	60
4.2 Pluviometria	60
4.2.1 Automação da pluviometria	61
4.2.2 Tipos de pluviômetros	62
4.2.3 Interfaceamento entre o kernel e o pluviômetro	63
4.2.4 Funcionamento do sistema	64
4.3 O processo de experimentos com PCR.....	66
4.3.1 O instrumento de automação de PCR	70
4.3.2 Medição térmica	71
4.3.3 Controle térmico	72
4.3.4 Sensoriamento e controle térmico em PCR	73
4.3.5 Testes do sistema desenvolvido	78
 Capítulo V	82
Conclusões e trabalho futuro	82
5.1 Conclusões	82
5.2 Avaliação crítica	83
5.3 Trabalho futuro.....	84
 Apêndice I	85
 Bibliografia	86

Lista de figuras

Figura 1.1 Número de transistores em microprocessadores INTEL	5
Figura 1.2 Transdutor passivo	8
Figura 1.3 Linearidade do sensor	10
Figura 1.4 Esquema de um controle de processo	17
Figura 1.5 Rede de aquisição de dados	19
Figura 2.1 A arquitetura de Von Neumann	23
Figura 2.2 Organização simplificada da UCP	25
Figura 2.3 Exemplo dos ciclos necessários para a execução de uma instrução	27
Figura 2.4 Aumento da frequência de operação em microprocessadores	27
Figura 2.5 Organização básica da linha MCS-51	29
Figura 2.6 Diagrama de um pino de E/S	37
Figura 2.7 Esquema de um dispositivo temporizador/contador	40
Figura 3.1 Diagrama lógico do kernel	46
Figura 3.2 Memórias do kernel	47
Figura 3.3 Configuração lógica do microcontrolador 80N535	54
Figura 4.1 Armazenamento de dados no sistema pluviométrico	65
Figura 4.2 Composição química do nucleotídeo	67
Figura 4.3 O processo de replicação do DNA	68
Figura 4.4 Variação da atividade enzimática	68
Figura 4.5 Ciclo térmico do processo de PCR	69
Figura 4.6 Sistema para automação de PCR	70
Figura 4.7 Variação da temperatura	74
Figura 4.8 Controle da fonte de calor	74
Figura 4.9 Amplificador de sinais	77
Figura 4.10 Controle em 30°C	78
Figura 4.11 Controle em 30°C	79
Figura 4.12 Análise de overshoot	80
Figura 4.13 Ciclo térmico de PCR	80
Figura 4.14 Erro do ciclo térmico de PCR	81

Lista de tabelas

Tabela 1.1 Exemplo de alguns tipos de sensores	07
Tabela 1.2 Classificação básica dos dispositivos de sensoriamento	08
Tabela 3.1 Pinos de controle das memórias externas	48
Tabela 3.2 Pinos de controle do visor	51
Tabela 3.3 Configuração da RS-232 no kernel	53
Tabela 4.1 Medidas do A/D com precisão de 5 mV	75
Tabela 4.2 Medidas do A/D com precisão de 19,6 mV	76
Tabela 4.3 Relação entre precisão e faixa de operação térmica	77

Resumo

Este trabalho descreve o projeto e implementação de um kernel computacional integrado e de baixo custo, utilizado para aplicações em coleta de dados e controle de processos. A filosofia do projeto é o desenvolvimento de um circuito que possa ser facilmente aplicado a várias atividades, com pequenas alterações no hardware ou simples troca de software.

O kernel foi desenvolvido com um microcontrolador programável que permite uma grande redução na complexidade do circuito. O kernel foi usado em duas aplicações diferentes: a aquisição automática de dados em pluviômetros e o controle de experimentos com PCR (Polimerase chain Reaction).

Abstract

This work describes the project and implementation of a low cost, highly integrated, microprocessor-based kernel for applications in intelligent data aquisition and process control. The project philosophy concerns the development of a circuit that could be easily applied to several fields with few hardware alterations or just software change.

The kernel was developed using a programmable microcontroller which enables large reductions in circuit complexity. The kernel was used in two different applications: the automatic data aquisition in pluviometers and in control of experiments of PCR (Polymerase Chain Reaction).

Introdução

Este trabalho consiste no desenvolvimento de um módulo computacional básico, chamado de *kernel*, e seu emprego em dois aplicativos diferentes, sendo o primeiro destinado à aquisição automática de dados em pluviômetros, e o segundo no sensoriamento e controle térmico na automação de experimentos com PCR (*Polimerase Chain Reaction*).

A filosofia do projeto é o desenvolvimento de um kernel que possa ser facilmente configurado para diversas aplicações de aquisição de dados e controle de processos, bastando que seja trocado o software de controle ou que sejam implementadas pequenas complementações de hardware, quando necessárias.

O kernel desenvolvido é um sistema que permite a coleta inteligente de dados, tendo sido projetado com um microcontrolador programável, que permitiu o desenvolvimento de um circuito eletrônico com reduzido número de componentes.

Muitas pesquisas científicas, ou mesmo controles de processos industriais, necessitam de dispositivos inteligentes de aquisição de dados ou controle, que tenham a capacidade de operar em função de critérios rigidamente pré-estabelecidos. Estes dispositivos inteligentes, como o kernel, devem permitir um pré-processamento dos dados, possibilitando uma aquisição, processamento e transferência de dados, de modo seguro e otimizado.

O primeiro aplicativo desenvolvido, na aquisição automática de dados em pluviômetros, permite o registro automático do volume de chuvas (*índice pluviométrico*), em determinada região. O sistema permite que estes dados sejam transmitidos em tempo real, via interface serial, possibilitando a formação de uma rede de telepluviometria. Caso os dados coletados não sejam usados em tempo real, o sistema garantirá o seu armazenamento, mesmo em caso de falhas no sistema de alimentação, uma vez que usa uma memória não volátil para este armazenamento. As informações relativas aos índices pluviométricos são úteis nas pesquisas

meteorológicas e podem ser utilizadas por diversos outros setores, como os setores agrícolas ou de construção civil.

O segundo aplicativo desenvolvido, foi o sensoriamento e controle térmico de experimentos com PCR. A técnica de PCR possibilita a replicação de uma amostra de DNA, *in vitro*, permitindo a obtenção de milhões de cópias idênticas à amostra original. Para a realização desta técnica, a amostra de DNA deve ser submetida a um ciclo repetitivo de temperaturas, cujo controle é feito pelo kernel desenvolvido. A técnica de PCR é uma importante ferramenta da biologia molecular e engenharia genética. Esta técnica pode, entre outras coisas, ser utilizada na análise do comportamento reativo a antibióticos ou no levantamento das características do genótipo, para fins judiciais.

Capítulo I

Princípios de aquisição de dados

1.1 Introdução

Este capítulo introdutório descreve alguns tópicos básicos relativos aos princípios de detecção, aquisição de dados e controle, utilizados em instrumentação.

Inicialmente será discutida a evolução da instrumentação, que permitiu sua utilização em inúmeras áreas da sociedade moderna, exigindo uma especialização cada vez maior dos pesquisadores. Como a medição de determinados fenômenos físicos exige o sensoriamento de diversas grandezas, será feita uma abordagem sobre o conceito de sensores e transdutores, bem como sobre princípios de calibração de instrumentos de medições e formas de erros encontradas nas mesmas.

Serão discutidas as técnicas mais modernas de aquisição de dados, que permitem o registro automático da medição de uma determinada grandeza. Além disto, é freqüente que estes dados coletados tenham que ser transferidos a computadores. Por isso, algumas técnicas de transferência de dados também serão descritas neste capítulo.

1.2 A ciência da instrumentação

Medir fisicamente uma determinada grandeza significa comparar o valor medido com uma grandeza homogênea, adotada como padrão de medida [65],

usando-se instrumentos específicos para esta tarefa. Uma medida será *direta* quando a comparação entre a grandeza medida é feita diretamente com o instrumento de medição, como na medição de um comprimento com uma régua. A medida será *indireta* quando for observada através de instrumentos que usem um modo associativo entre a grandeza medida e uma escala, como no caso de instrumentos com galvanômetros [44].

As ciências exigem a medição de diversas grandezas físicas como o comprimento, a massa, o tempo, a velocidade, a temperatura, a resistividade, etc. Toda grandeza física, ao ser definida, deve ter uma série de procedimentos para sua medição, cabendo ao Bureau Internacional de Pesos e Medidas, fundado em 1875, em Paris, padronizar estes procedimentos, mantendo contato com laboratórios de padrões de todo o mundo [24].

Os instrumentos de medição são constantemente aprimorados, uma vez que o estudo de fenômenos físicos, freqüentemente, requer instrumentos com características inovadoras [65]. Até a segunda guerra mundial, a instrumentação caminhou em passos relativamente lentos. A partir de então, o seu desenvolvimento cresceu aceleradamente, aproveitando-se do desenvolvimento em algumas áreas, como por exemplo, a mecânica, a eletrônica, a física dos semicondutores e a computação. Esta evolução causou um interesse comercial mais amplo pela instrumentação, que deixou de ser uma quase exclusividade das comunidades científicas e militares e passou a fazer parte da automação de automóveis, de indústrias, de equipamentos médicos, de sistemas de transportes e comunicações, entre outros. Atualmente, a instrumentação atingiu um grau de evolução que a torna presente, direta ou indiretamente, em inúmeros seguimentos da sociedade moderna.

Mesmo com todo o desenvolvimento recente, alguns instrumentos clássicos ainda apresentam amplo desempenho em algumas áreas, e continuam sendo usados de modo paralelo a instrumentos mais modernos. Os antigos galvanômetros, que sofrem uma concorrência direta de visores eletrônicos ou de cristal líquido, ainda são amplamente utilizados. Outro exemplo são os microscópios óticos, usados paralelamente aos sofisticados microscópios eletrônicos e aos recentes microscópios de rastreamento por tunelamento, que, usando o conceito de efeito túnel, permitem a

determinação das características físicas do relevo da superfície de um material, em escala sub microscópica [52, 53].

Atualmente, as linhas de instrumentação mais ativas incluem a instrumentação eletrônica, mecânica e ótica. No caso da instrumentação eletrônica, observa-se que, em apenas algumas décadas, os instrumentos apresentaram uma grande redução de tamanho, consumo de energia e custo, aliados a um aumento da confiabilidade e diversidade de aplicações. Atualmente, a densidade de integração dos circuitos integrados tem aumentado muito, conforme exemplifica a figura 1.1, onde é verificada a evolução da densidade dos microprocessadores da INTEL, iniciando pelo 4004, lançado em 1972 e chegando ao recente P5 [78]. A densidade de integração evoluiu ao ponto de um único circuito integrado conter um computador completo a um preço relativamente baixo [17], facilitando assim o uso de computadores em muitas aplicações dedicadas ao controle de processos ou medições físicas.

Fig. 1.1 Número de transistores em microprocessadores INTEL

1.3 Profissionalização

A evolução da instrumentação a tornou cada vez mais complexa, exigindo um grau de especialização cada vez maior dos seus pesquisadores [40], que passaram a se dedicar exclusivamente aos mais diferentes segmentos desta área.

Antes desta evolução, muitos pesquisadores eram obrigados a deslocar, para um segundo plano, a sua pesquisa principal, para tentar resolver problemas de instrumentação necessários ao andamento dos trabalhos [23]. Como estes cientistas não tinham uma formação sólida em instrumentação, acabavam perdendo um grande tempo no desenvolvimento de instrumentos, que eram freqüentemente de baixa qualidade.

A necessidade de desenvolvimento de instrumentos específicos para algumas aplicações, obriga freqüentemente o pesquisador da área de instrumentação a interagir com outras áreas. O mesmo deve ter, portanto, uma formação ampla, que lhe facilite uma interação eficiente [22], possibilitando o desenvolvimento de projetos de maneira interdisciplinar. A profissionalização permitiu a formação de pessoal com grande capacidade de análise dos problemas e implementação de soluções viáveis, dentro da realidade disponível. A solução de um problema pode ser o desenvolvimento de um novo produto ou mesmo a escolha correta de algum instrumento comercial, que deve ser feita em função de critérios rígidos e coerentes, evitando a malversação de recursos.

Caso a solução mais viável para um determinado problema seja o desenvolvimento de um novo produto, é de responsabilidade do especialista, baseado em critérios científicos, traçar critérios como níveis de precisão, dimensões físicas, consumo de energia, formas de calibração, implementação e testes [40]. Estes fatores são freqüentemente conflitantes, pois em alguns casos, não é fácil conciliar alta precisão e baixo custo. Além da capacidade de gerenciar diferentes fatores, o especialista em instrumentação deve ter uma sólida visão científica que lhe permita pesquisar criteriosamente as publicações mais recentes relativas à área de atuação e ainda estudar as considerações teóricas que se fizerem necessárias para cada projeto.

1.4 Dispositivos de sensoriamento

Nesta seção serão abordados alguns tópicos relativos à detecção de uma determinada grandeza, através de *dispositivos de sensoriamento*, chamados sensores e transdutores, que são importantes peças na área de instrumentação.

1.4.1 Classificação

Dois termos freqüentemente confundidos na área de instrumentação são os sensores e transdutores. Os **sensores** são dispositivos que possuem características que podem variar em função de alguma grandeza física. Um sensor fotoresistível, por exemplo, tem o valor de sua resistência alterado em função das variações da intensidade da luz que incide sobre o mesmo. Esta relação determina o tipo do sensor, conforme exemplifica a tabela 1.1. Os **transdutores** são considerados dispositivos que transformam um tipo de energia em outra. Um alto-falante, por exemplo, é considerado um transdutor, já que transforma oscilações elétricas em ondas acústicas [44].

tabela 1.1 exemplo de alguns tipos de sensores

<i>TIPO DO SENSOR</i>	<i>GRANDEZA MEDIDA</i>			
	temperatura	luminosidade	força	umidade
resistivo	X	X		
capacitivo				X
piezoelétrico			X	

Os transdutores são basicamente classificados em função do tipo ativo ou passivo e do princípio de funcionamento. Os **transdutores ativos** convertem diretamente um tipo de energia para outro, enquanto que os **transdutores passivos** não fazem a conversão e operam sob o princípio de controle de energia, necessitando obrigatoriamente de uma fonte auxiliar de alimentação e de um sensor, que forneça algum parâmetro referencial para o sinal de saída [40], conforme figura 1.2. Embora os transdutores passivos não convertam formas de energia e fujam do conceito

clássico de transdutores, eles são classificados como transdutores por caracterizarem uma classe específica de dispositivos de sensoriamento.

Fig. 1.2 Transdutor Passivo

O princípio de funcionamento determina a grandeza física que pode sensibilizar o transdutor e o tipo de sinal que o mesmo fornece em sua saída. Um transdutor piezoelétrico, por exemplo, fornece um sinal elétrico de saída em função de uma força mecânica que inside sobre o mesmo. O tipo de sinal fornecido na saída de um transdutor pode ser elétrico, acústico ou luminoso, entre outros, sendo que os transdutores elétricos são extremamente usados, pois o sinal elétrico pode ser facilmente amplificado, atenuado, modulado, etc.

tabela 1.2 classificação básica dos dispositivos de sensoriamento

<i>CLASSIFICAÇÃO</i>	<i>CARACTERÍSTICAS</i>
ativo	converte formas de energia
passivo	controla sinal de uma fonte de alimentação
sinal de saída	elétrico, acústico, magnético, etc
tipo de funcionamento	resistivo, capacitivo, indutivo, etc
tipo de sensibilidade	térmica, luminosa, baraométrica, etc

Embora haja uma distinção entre sensores e transdutores, existe uma relação direta entre ambos, uma vez que os mesmos podem ser utilizados na formação de dispositivos de sensoriamento, que podem ser genericamente classificados conforme tabela 1.2 [40, 42].

1.4.2 Sensibilidade e linearidade

Todo dispositivo de sensoriamento possui um fator que determina o grau de **sensibilidade** do mesmo às variações do sinal de entrada, ou seja, define o grau necessário de variação do sinal de entrada para que haja alguma variação no sinal de saída. Além da sensibilidade existe um fator de **reprodutibilidade** que determina o grau de variação do sinal de saída em função de uma determinada variação do sinal de entrada. Assim, a sensibilidade pode ser associada ao sinal de entrada, enquanto a reprodutibilidade pode ser associada ao sinal de saída. A reprodutibilidade é um fator que pode ser compensado através de amplificadores, enquanto a sensibilidade é um fator mais crítico que depende exclusivamente das características intrínsecas do sensor.

Além do problema de sensibilidade, a maioria dos dispositivos de sensoriamento só consegue produzir sinais de saída quando o sinal de entrada está dentro de certos limites. A partir daí, o sistema não produzirá resposta correta e confiável, entrando em um limite operacional chamado de *zona morta*.

Um dispositivo de sensoriamento também pode ter, em sua saída, um sinal que não varie na mesma proporção do sinal de entrada. Os dispositivos com essa característica são chamados de **não lineares**, enquanto que nos sensores **lineares**, há uma relação proporcional entre a saída e a entrada. Esta saída linear pode ser representada pela equação da reta $y = m \cdot x + c$ [40], onde x é o sinal de entrada e y é o sinal de saída. Embora o sensor possa ter sido idealizado para ter uma saída realmente linear, existem casos onde o resultado pode sofrer distorções em relação ao ideal. A figura 1.3 mostra as relações de linearidade.

Os sensores não lineares se tornam desejáveis quando manipulamos fenômenos naturalmente não lineares. Um exemplo de fenômeno não linear pode ser dado através do uso potenciômetros logarítmicos, utilizados no controle da amplitude do volume em equipamentos de áudio, pois a variação logarítmica se ajusta a resposta logarítmica (não linear) da audição humana. Isto permite uma "linearização" entre a variação do controle de som e a resposta auditiva.

Fig. 1.3 (A) Sensor Linear
(B) Sensor Linear Distorcido
(C) Sensor não Linear

1.4.3 Precisão e exatidão

A **precisão** de um dispositivo de sensoriamento representa a sua capacidade de repetir o mesmo valor de uma determinada medida, toda vez que a mesma for realizada, nas mesmas condições físicas. A **exatidão** é o fator que representa a diferença entre o valor medido e o valor real da medida.

A exatidão é um fator que pode ser compensado através de dispositivos amplificadores, atenuadores ou mesmo tabelas de ajustes, enquanto que a precisão torna-se um fator mais crítico, uma vez que não pode ser corrigido e pode comprometer toda confiabilidade do sistema.

1.5 Calibração e erros

As próximas sub seções apresentam conceitos sobre calibração de instrumentos de medição e sobre erros de medições, uma vez que estes fatores afetam diretamente a confiabilidade da medida.

Qualquer instrumento de medição, calibrado ou não, pode estar sujeito a erros, ocasionados por vários fatores como: erros de reprodutibilidade, erros humanos, erros intrínsecos, erros de leitura ou erros de interferência. É interessante que se conheça estas formas de erros para se determinar o possível grau de alterações que as mesmas podem causar no valor medido.

1.5.1 Calibração

A **calibração** é a operação que estabelece a relação entre uma grandeza medida e a escala usada pelo instrumento [44], podendo ser classificada como fixa ou variável. A **calibração fixa** supõe o ajuste inicial de determinados fatores, como escalas de leitura ou fatores de amplificação, que são feitos após a fabricação ou no ato de instalação, enquanto que a **calibração variável** é feita para compensar descalibrações que podem ocorrer naturalmente, em função do uso ou do tempo.

A calibração variável ocorre principalmente devido à interferência de fatores ambientais ou de fatores internos cujas características mudam em função do tempo, de forma conhecida, como é o caso do decaimento de materiais radioativos ou mesmo dispositivos do tipo *timer*, em sistemas computacionais. Nestes casos, que podem ser previstos, uma recalibração pode ser efetuada de modo automático, através de tabelas de ajuste ou sistemas de realimentação.

1.5.2 Tipos de erros

Em qualquer sistema de sensoriamento, medição e registro de dados, temos que considerar a possibilidade de ocorrência de erros, nas suas mais diferentes formas. Os erros podem ser classificados em sistemáticos e ocasionais [44]. Os **erros sistemáticos** decorrem de fatores intrínsecos ao processo de medição, podendo ser diminuídos com o aperfeiçoamento do mesmo. Os **erros ocasionais** ocorrem em função de fatores externos. Como o erro está diretamente associado a uma medida, diz-se que uma medida é **acurada** quando seu erro sistemático for pequeno e que ela é **precisa** quando seu erro ocasional for pequeno. Uma boa medida deve ser acurada e precisa [44].

Na seção 1.3.3 foram discutidos os conceitos de exatidão e precisão relativos a um dispositivo de sensoriamento, enquanto que nesta seção, estes conceitos são apresentados de forma mais genérica e associados aos erros ocorridos no processo de medição, que serão apresentados a seguir.

O sistema de medição pode ser dividido em partes individuais e cada uma destas partes pode ter sua margem de erro característico. Assim, o sistema, como um todo, terá uma margem de erro total representada pelo somatório das partes individuais [40]. Esta margem de erro pode ser expressada percentualmente, como uma relação entre o valor medido e o valor escolhido para representar a medida de forma exata [44].

1.5.3 Erro de reprodutibilidade

A resposta de qualquer dispositivo de sensoriamento ou medição pode variar, ou seja, alterações no sinal de entrada podem não ser reproduzidas com a proporção esperada na saída do sistema, ocorrendo um **erro de reprodutibilidade** [40]. Caso a medição apresente um erro de reprodutibilidade em caráter constante, haverá um problema de exatidão, caso contrário, haverá um problema de precisão.

1.5.4 Erros intrínsecos

Circuitos eletrônicos, ou os próprios componentes internos ao mesmo, podem apresentar falhas e gerar uma margem de erro nas medições. Estes **erros intrínsecos** podem surgir em função de erros no projeto ou pela qualidade dos componentes usados. Os erros de projeto podem ser evitados através de testes rigorosos que os detectem antes do produto final estar concretizado. Os erros originários por componentes são mais críticos, pois não são facilmente previsíveis e podem se tornar mais graves quando usamos dispositivos eletromecânicos.

1.5.5 Erros aleatórios

Os **erros aleatórios**, como o ruído, ocorrem quando várias medidas da mesma quantidade, nas mesmas condições gerais, apresentam diferentes valores. Este tipo de erro geralmente pode ser expressado de um modo estatístico, pois normalmente ocorrem de modo imprevisível. Assim, baseando-se em experiências anteriores, pode-se calcular a probabilidade de sua ocorrência. Nota-se, porém, que não se pode determinar o momento exato da ocorrência de um erro deste tipo.

Atualmente, existem muitos estudos a respeito de fenômenos que ocorrem de modo imprevisível, tentando determinar o mecanismo de sua formação e, desta forma, poder compensá-los. As pesquisas sobre estes fenômenos, como os erros aleatórios, tentam descobrir uma ordem na sua ocorrência [47], identificando uma ordem no caótico e determinando exatamente quando, onde e como estes fenômenos poderão ocorrer [48, 49].

1.5.6 Erros por ruídos

O **ruído** pode ser considerado um sinal indesejado presente em um sistema, cujo efeito, sobre o sistema, é chamado **interferência** [62]. Como o ruído é

o sinal interferente, o mesmo não deve ser expressado em termos absolutos, mas sim de modo relativo ao sinal original do sistema. Assim, a grandeza relacional **sinal/ruído** quantifica a proporção do sinal do ruído em relação ao sinal original do sistema [64].

A relação sinal/ruído é um conceito importante na análise de ruídos, influenciando até mesmo a capacidade de transmissão de dados de um sistema de comunicações [64].

Além do ruído causado por fatores externos, como motores elétricos, sistemas de ignição de automóveis, transformadores, linhas de transmissão de energia elétrica, transmissores de rádio, etc, o sistema de medição também pode apresentar ruídos de origem interna. Um exemplo é o ruído térmico, onde a energia de vibração dos elétrons do sistema, que varia em função da temperatura, pode produzir uma flutuação na diferença de potencial do mesmo. Também existe o ruído produzido pelo efeito Schottky, onde uma flutuação aleatória da emissão de elétrons por um cátodo provoca uma modificação na tensão do mesmo e uma conseqüente flutuação da corrente [44].

1.5.7 Erros por fatores humanos

Muitos erros podem ocorrer por falhas humanas e não dos dispositivos de medição. São comuns os erros de instalação, causados pela falta de atenção a detalhes como alimentação elétrica, temperatura ambiente, ligações de cabos, parâmetros de inicialização do sistema, etc. Além da instalação errada [40], o fator humano também pode provocar erros de ajustes, seleção de escalas inadequadas, paralaxe, leitura, etc. Estes erros realçam a importância do fator humano, que deve ser confiável e bem treinado.

Alguns instrumentos possuem tolerância relativa a falhas humanas, como por exemplo, usando visores digitais ao invés de galvanômetros ou possuindo proteção interna para prevenir sobrecargas devido à seleção errada de escalas.

1.5.8 Erro de leitura

Os erros de leitura podem ser causados por fatores humanos, conforme seção acima, ou por defeitos no dispositivo de leitura do instrumento de medição. Exemplos disto podem ser dados no caso de ponteiros indicadores tortos em função de choques mecânicos [44] ou posicionamento errado na hora de leitura, ocasionando erro de paralaxe, que é a focalização errada do ponteiro em relação à escala.

1.6 Aquisição de dados

Em função do rápido desenvolvimento tecnológico, foram desenvolvidos novos conceitos e técnicas para a coleta e armazenamento de dados. A coleta deve ser planejada de modo a não se perder os dados de entrada e o armazenamento deve ser efetuado sob uma estrutura que permita fácil recuperação das informações, e que seja compatível com os recursos computacionais disponíveis, tais como o tamanho de memórias e capacidade de processamento.

O conceito de coleta indiscriminada de dados foi substituído por um conceito mais moderno, cuja premissa é "coletar somente o que se precisa, quando se precisa". Isto implica em uma coleta seletiva de dados, onde os mesmos não são coletados continuamente, mas sim, sob critérios pré-definidos. Isto exige que o dispositivo de aquisição de dados tenha uma capacidade de processamento e, desta forma, realize uma coleta denominada *inteligente* [15].

Os dispositivos mais simples de coleta de dados fazem uma coleta chamada de *continuous history*, pois normalmente coletam os dados continuamente, enquanto os dispositivos com critérios seletivos (inteligentes) de coleta podem ser conceituados como *event history*. O termo "history" pode ser designado para descrever a coleta de dados. Acharmos pertinente esta colocação, pois encontramos diversas formas de se referenciar um dispositivo de aquisição de dados, sem que exista um padrão para isto [15].

Um sistema inteligente de aquisição de dados, além da seletividade na coleta, deve ter capacidade de pré-processamento. Como o processamento pode ser feito por um computador de maior porte, o pré-processamento pode diminuir o volume de dados transferidos, aliviando o sistema de comunicação. Além disto, ele permite especificar níveis de segurança que selecionem quais os usuários que podem ter acesso aos mesmos, ou ter ainda algum dispositivo de tolerância a falhas.

Nos sistemas inteligentes, a antiga idéia de "salve tudo que pode eventualmente ser necessário mais tarde", foi substituída, em muitos casos, pela filosofia "coleta-se o que precisar, quando precisar, e sabe-se como recuperar esta informação" [15]. Os sistemas inteligentes têm a vantagem de poderem ser programados e, assim, podem fazer uma coleta seletiva ou contínua, enquanto um sistema não inteligente faz uma coleta unicamente contínua.

Coletar, processar e armazenar dados em grandes quantidades não é uma tarefa trivial. Embora o conceito de "quantidade de dados" dependa dos recursos tecnológicos disponíveis, em termos práticos, um telescópio espacial, por exemplo, pode transmitir milhões de informações por segundo, ininterruptamente.

Em atividades que exigem um processamento intensivo de dados, como a astronomia, existe um grande problema de coleta e armazenamento dos mesmos, uma vez que são exigidos equipamentos que operem em alta velocidade e possuam grande espaço para armazenamento. Algumas correntes defendem que, em certas áreas, todas as informações adquiridas ou processadas devem ser salvas, pois o custo de aquisição e processamento destas informações é muito alto, podendo ser muito superior ao custo de armazenamento. Além disso, esta informação coletada pode representar um fenômeno ainda não detectado [21]. Este caso é um exemplo em que um sistema inteligente poderia ser utilizado para uma coleta não seletiva de dados.

Em muitos casos a filosofia de se salvar todos os dados é impossível, em função do volume gerado pelo sistema sensor e pelas limitações do sistema de aquisição e processamento. Nestes sistemas, não seletivos, torna-se obrigatória uma análise crítica para se determinar todos os possíveis níveis de perdas e qual a sua possível representatividade diante do todo.

1.7 Sistemas para aquisição de dados

Um sistema de aquisição de dados deve ser capaz de coletar e armazenar os dados oriundos de algum sistema. Um sistema de aquisição de dados pode ser desenvolvido com um computador pessoal dedicado a esta atividade. Porém, em algumas circunstâncias, pode haver um super dimensionamento do equipamento, uma vez que esta aquisição poderia ser realizada por um sistema especialmente projetado à aquisição de dados, que poderia apresentar igualdade de desempenho, a um custo e dimensões menores.

Um dispositivo eficiente para coleta de dados é o *data-logger*, que registra a ocorrência de determinado fenômeno físico, sua amplitude, horário, e ainda realiza atividades de processamento prévio ou controla um processo, formando, assim, um sistema inteligente de coleta de dados. Este instrumento pode ser encarado como um microcomputador dedicado à aquisição inteligente de dados, podendo ser empregado em muitas atividades. Atualmente, grandes processos industriais são eficientemente controlados por data-loggers de baixo custo [14], conforme figura 1.4.

Fig. 1.4 esquema de um controle de processo

Um dispositivo de aquisição de dados deve ter a capacidade de coletar, armazenar, processar e transferir dados da maneira rápida e segura [16, 81]. Como constantemente surgem componentes e técnicas de trabalho mais modernos, os data-loggers também devem ter seus projetos revistos para acompanhar a evolução geral. Esta evolução engloba problemas de conversão analógico-digital, recuperação de

informações, armazenamento, tratamento matemático das informações, novas arquiteturas, componentes, transmissão de dados, etc.

Com o desenvolvimento dos microcontroladores programáveis, os data-loggers tiveram uma significativa evolução, uma vez que estes sistemas embutidos em um único componente facilitam o projeto de hardware, requerendo poucos componentes adicionais para a formação de todo o circuito eletrônico do data-logger [07, 26, 27].

1.8 Transferência de dados

Atualmente podemos construir dispositivos dedicados à aquisição de dados com grande capacidade de armazenamento e processamento. Apesar disto, é comum que um computador de maior porte realize parte do processamento geral, uma vez que o mesmo pode dispor de mais recursos, como memória, velocidade de processamento ou terminais gráficos. Assim, os dados adquiridos devem ser transmitidos ao computador de processamento geral.

Por vezes, não é necessário que os dados sejam processados imediatamente após sua coleta, facilitando a transferência dos mesmos, pois o dispositivo de aquisição de dados pode ser fisicamente transportado e conectado diretamente ao computador de processamento geral. Este processo é muito simples pois é realizado a curta distância e não necessita de maiores aparatos, como modems e linha telefônica.

Outro tipo de aquisição de dados exige que os mesmos sejam transferidos para o computador principal tão logo sejam coletados, operando em tempo real. Como a distância entre o dispositivo de aquisição de dados e o computador de processamento pode ser grande, então a conexão entre ambos poderá ser feita através da rede telefônica, que é ligada aos mesmos, através de modems. Assim, os dados podem ser rapidamente transmitidos a qualquer distância [39]. Neste caso, o software que controla a comunicação deve ser extremamente criterioso, estabelecendo todas as regras necessárias para a transferência dos dados, através de

um protocolo de comunicações [51]. Este ambiente exige uma padronização dos equipamentos utilizados, permitindo o uso de equipamentos de diferentes fabricantes.

Se tivermos vários dispositivos de aquisição de dados interligados através de um computador central, conforme figura 1.5, poderemos formar uma rede de aquisição de dados, onde poderemos compartilhar recursos de hardware e software, além da troca de informações [51], sendo que o computador central poderia estabelecer novos parâmetros para a coleta inteligente de dados e fazer a transferência quando achasse conveniente.

Fig. 1.5 Rede de Aquisição de dados

1.9 Telemetria

A **telemetria** é um processo que permite o monitoramento remoto de determinado evento e está frequentemente associada a determinado tipo de controle. Este processo pode contar, inclusive, com um dispositivo programável de controle, como um data-logger, tornando o sistema mais eficiente. A primeira utilização comercial da telemetria ocorreu em 1912 [20]. Estes sistemas iniciais representavam a informação a ser transmitida através de voltagens, correntes ou baixas frequências.

Existiam dispositivos que recebiam estes sinais e indicavam o valor da informação recebida.

A telemetria evoluiu muito, pois se valeu dos avanços tecnológicos ocorridos nas técnicas de telecomunicações, informática e eletrônica. Este desenvolvimento, com aplicações comerciais, exigiu o estabelecimento de padrões que garantissem a confiabilidade do sistema. Assim o AIEE (*American Institute of Electrical Engineers*), que possui um sub comitê especial para telemetria, criou uma classificação para diferentes tipos de sistemas telemétricos, que levou em consideração itens como níveis de corrente, tensão, frequência e forma de pulso [20].

Um sistema de telemetria, no processo de comunicação, usa um dispositivo transmissor, um meio físico de transmissão e um dispositivo receptor. Estes sistemas normalmente usam ligações via cabo para transferência de dados, no entanto, existe uma variedade de processos que podem ser usados e que devem ser analisados criteriosamente [19].

A radiotelemetria é o processo de telemetria que usa a comunicação via ondas de rádio, ao invés da ligação via cabos. Os primeiros sistemas de radiotelemetria usavam transmissão em amplitude modulada a baixas frequências, apresentando resultados insatisfatórios. Com a utilização mais moderna de transmissão em frequência modulada e utilização de faixas de VHF ou UHF, pudemos observar resultados mais satisfatórios.

Apesar das pesquisas com radiotelemetria, alguns problemas ainda persistem como o congestionamento de faixas de onda, exigindo grande seletividade do receptor e deixando uma margem para interferências de sistemas que operam com frequências próximas. Além desta interferência, podemos encontrar ruídos causados por fatores externos como descargas elétricas ou ainda por motores elétricos. Outro grave problema da transmissão via rádio são as falhas de propagação. Como as informações são transmitidas por ondas eletromagnéticas, elas estão sujeitas à influência da topografia da região, condições atmosféricas, frequência de transmissão ou, condutibilidade do terreno [64].

Cada frequência de rádio comunicação apresenta um comportamento diferente, quanto à propagação. Assim um sinal transmitido em ondas curtas, por exemplo, poderia alcançar longas distâncias, porém, estaria extremamente sujeito a falhas de propagação. Frequências maiores, como na faixa de microondas, podem apresentar uma confiabilidade muito maior, porém, exigem equipamentos mais sofisticados, obrigando haver um caminho direto e sem obstáculos entre o foco transmissor e o receptor [64].

Embora a telemetria, via rede telefônica, seja eficiente, existem pesquisas com uso de transmissores via rádio, que podem permitir maior mobilidade física ao sistema, uma vez que não existem cabos entre os mesmos [18].

Capítulo II

A estrutura de um computador genérico

2.1 Introdução

Este capítulo aborda tópicos relativos à estrutura dos controladores eletrônicos, desenvolvidos com um modelo computacional, que normalmente são aplicados na medição de fenômenos físicos ou controle de processos.

2.2 O conceito de computador

Um computador é uma máquina dedicada ao processamento de dados, capaz de coletar dados, processá-los e fornecer resultados [59], de forma automática e dirigida [44]. O mesmo pode ser visualizado de diferentes formas, inclusive como um dispositivo totalmente mecânico. Um simples ábaco, por exemplo, é capaz de processar dados, embora não tenha a versatilidade de ser programado. Usualmente o conceito de computador se refere a uma máquina eletrônica, com lógica digital e implantada de acordo com uma arquitetura específica [31, 59].

2.3 A organização de um computador

A organização de um computador determina a maneira como o mesmo é desenvolvido, especificando suas partes básicas e o modo de funcionamento. A arquitetura de von Neumann é a mais comum, sendo organizada em cinco unidades básicas, conforme figura 2.1.

Fig. 2.1 A arquitetura de Von Neumann

As unidades desta arquitetura podem ser basicamente descritas da seguinte maneira [31]:

1 - Unidade de entrada e unidade de saída: fazem o interfaceamento entre as partes internas do computador e os dispositivos externos de entrada e saída como terminais de vídeo, teclados, memórias externas, mouses, plotters, etc.

2 - Unidade de lógica e aritmética (ULA): realiza operações lógicas e aritméticas, como soma, subtração, deslocamento, negação, "and", "or", etc.

3 - Unidade de controle (UC): gerencia o sincronismo de operações das outras unidades, fornecendo os sinais de temporização necessários para este fim. A unidade de lógica e aritmética e a unidade de controle podem ser fisicamente construídas juntas, formando uma unidade chamada **unidade central de processamento (UCP)**.

4 - Unidade de memória: armazena os programas e os dados que serão manipulados durante o processamento. Existem diferentes tipos de memórias e a sua escolha depende das características exigidas pelo sistema.

Todas estas unidades do computador devem estar fisicamente interligadas através de barramentos específicos, que podem ser divididos em três tipos diferentes: *Barramento de endereços*, que seleciona uma determinada posição de memória, ou porta de entrada e saída; o *barramento de dados* que permite o fluxo de dados entre as diferentes partes do computador e o *barramento de controle* que serve para conduzir os sinais que controlam os componentes, permitindo ou não a sua habilitação [31].

2.4 A Unidade Central de Processamento (UCP)

Esta seção descreve o funcionamento básico da UCP, que é um dispositivo fundamental no desenvolvimento de um computador.

2.4.1 Descrição da UCP

A unidade central de processamento (UCP) contém basicamente uma unidade de lógica e aritmética, uma unidade de controle e alguns registradores [59]. Quando fabricada em um único componente, é chamada **microprocessador** e sua organização básica pode ser analisada na figura 2.2 [31].

Segundo Tokheim [31], a UCP possui quatro funções básicas, que são:

- 1 - Buscar, decodificar e executar instruções do programa em ordem apropriada.
- 2 - Controlar o fluxo de dados da memória e dispositivos de entrada e saída.
- 3 - Responder aos pedidos de interrupção externos.
- 4 - Promover os sinais de temporização e controle para todo o sistema.

Fig. 2.2 Organização simplificada da UCP

A **unidade de lógica e aritmética (ULA)** realiza as operações lógicas e aritméticas e a **unidade de controle (UC)** realiza o sincronismo para controle das operações UCP. Os **registradores** são pequenas unidades de memória, que armazenam temporariamente os dados manipulados pela UCP. Entre os diferentes registradores possíveis, pode-se destacar o *acumulador*, que armazena os dados a serem processados pela ULA. Também temos registradores com funções especiais como o *registrador contador de programa*, que contém o endereço da próxima instrução a ser executada, um *registrador apontador de pilha* que contém o endereço do topo da pilha de dados, um *registrador de instruções* que armazena a instrução em execução na UCP e, finalmente, um registrador especial, chamado *flag*, que indica a ocorrência de determinados eventos. Além dos registradores, da ULA e da UCP, existe um outro dispositivo importante chamado *decodificador de instruções*, que determina os passos corretos a serem seguidos, para a execução da instrução armazenada no registrador de instruções [31, 59].

Como as características de um projeto de hardware dependem diretamente da UCP, é necessário que alguns de seus itens sejam criteriosamente analisados, como:

- 1 - A arquitetura da UCP
- 2 - A função de cada pino
- 3 - O conjunto de instruções
- 4 - O sistema mínimo de hardware exigido
- 5 - Os sinais de controle

2.4.2 Funcionamento da UCP

Logo que a UCP é inicializada, após ser ligada ou reinicializada, ela automaticamente endereça uma determinada posição da memória de programa, interpretando o conteúdo desta posição como sendo o início de um programa. Todo programa é seguido passo-a-passo pela UCP, assim, a mesma sempre realiza uma sequência de buscar (*fetch cycle*), decodificar e executar cada instrução do programa (*execute cycle*) [59]. Cada uma destas três fases, durante sua realização, pode, em muitos casos, ser dividida em uma série de pequenos passos chamados *microprogramas* [31]. Embora o processo de busca e execução das instruções, uma a uma, seja o mais convencional, atualmente existem processadores que realizam esta tarefa na forma de *pipeline*, onde o processador pode fazer a busca da próxima instrução, enquanto executa a instrução atual [78].

Como a UCP necessita de um sinal que promova a sincronização de suas operações internas, ela deve ser associada a um dispositivo oscilador que gera um sinal em uma frequência precisa. O conjunto mínimo de pulsos de sincronia exigidos pelo processador é chamado **ciclo de máquina** [58]. Como cada instrução requer um determinado número de ciclos de máquina para sua realização, formando um **ciclo de instrução**, e cada ciclo de máquina requer um número de sinais do oscilador, a velocidade de processamento depende diretamente da frequência do oscilador [59]. A figura 2.3 exemplifica as relações entre os ciclos e os sinais do oscilador (*clock*).

Fig. 2.3 Exemplo dos ciclos necessários para a execução de uma instrução

Cada UCP possui uma frequência máxima de trabalho, porém, com o desenvolvimento tecnológico, esta frequência máxima tem se tornado cada vez maior, permitindo um processamento cada vez mais rápido. A figura 2.4 mostra o aumento da frequência máxima de operação (*clock*) utilizada pelos microprocessadores da Intel e Motorola [79]. Em termos reais, um microcontrolador da linha MCS-51, por exemplo, trabalhando com um cristal de 12 MHz, realiza a maioria das suas instruções em 1 microsegundo [25, 57, 69].

Fig. 2.4 Aumento da frequência de operação em Microprocessadores

2.5 Microcontroladores

O avanço da tecnologia da fabricação de circuitos integrados, permitiu que surgissem circuitos integrados com um computador completo embutidos. Estes dispositivos são chamados de microcontroladores e sua organização pode ser exemplificada através da figura 2.5 [57]. Um microcontrolador, entre outras coisas, pode conter [25, 57, 69]:

- 1 - UCP
- 2 - Memórias de dados e de programas
- 3 - Temporizadores programáveis
- 4 - Tratamento de interrupções internas e externas
- 5 - Portas bidirecionais de entrada/saída
- 6 - Canais de comunicação serial
- 7 - Conversores analógico/digitais

Os microcontroladores vêm sendo usados nos mais diferentes sistemas dedicados ao controle de processos [02] e o seu uso se torna cada vez mais freqüente, principalmente em função do custo e conveniência de permitir um projeto de hardware relativamente simples [01, 06]. Os dispositivos desenvolvidos com microcontroladores vêm substituindo cada vez mais os microcomputadores em muitas aplicações dedicadas. Além do custo e facilidade de projeto, estes dispositivos são mais versáteis para manipulação física. Outra vantagem dos microcontroladores é que eles podem ser projetados especialmente para operações específicas, como, por exemplo, o processamento digital de sinais, apresentando um alto desempenho nestas tarefas [02].

Fig. 2.5 Organização básica da linha MCS-51

Os primeiros microcontroladores de 8 bits, como o 8048 da INTEL, lançado em 1976 [17], foram inicialmente usados para o controle de eventos simples. Hoje, porém, existem linhas de 16 e 32 bits, que atuam em áreas mais sofisticadas, que podem requerer grandes memórias externas, ciclos de máquina na ordem de nanosegundos ou operações com ponto flutuante, como, por exemplo, operações em tempo-real, processamento de sinais ou controle de redes de comunicação [04, 10, 12, 28]. Apesar disto, os microcontroladores de 8 bits não foram negligenciados, apresentando um número cada vez maior de aplicações, pois, em muitas destas aplicações, apresentam excelente performance a um custo menor [08, 68].

Um exemplo de evolução dos microcontroladores de 8 bits é o DS5000, da Dallas Instruments, que possui uma bateria interna, que alimenta uma memória interna de dados, podendo reduzir, ainda mais, o hardware do sistema [05, 03]. Estes sistemas híbridos, que incluem o semicondutor e a bateria, permitem que o armazenamento de dados seja garantido, mesmo na falta de alimentação por parte do sistema principal e ainda permitem que o dispositivo de processamento seja reprogramado a longas distâncias, via rede telefônica.

Entre os microcontroladores de 8 bits, a linha MCS-51, fabricada pela INTEL, é líder mundial de vendas [04]. Embora outros microcontroladores possam apresentar algumas vantagens isoladas, a linha MCS-51 apresenta um conjunto de características, como custo, suporte e facilidade de uso, que tem lhe garantido a liderança no mercado.

2.5.1 A escolha de um microcomputador

Na escolha de um microcontrolador devem ser considerados fatores como requisitos de entrada/saída, memória necessária, consumo de energia, tolerância à temperatura, capacidade de processamento (tipo, volume e velocidade), custo e suporte [66, 67].

O volume de dados e tamanho de programas determinam a *quantidade de memórias* que o microcontrolador terá que endereçar. Além de possuírem uma pequena memória interna de dados, alguns microcontroladores também podem possuir uma memória interna de programa. Além das memórias internas, alguns microcontroladores, com 16 linhas de endereços, podem ter até 64 Kbytes de memória de programas e 64 Kbytes de memória de dados, ambas externas, uma vez que existem diferentes pinos de controle para cada uma das memórias.

Os *sinais de entrada e saída*, requeridos do microcontrolador, devem ser analisados em função de sua quantidade, frequência, amplitude e tipo, onde são observados critérios como o formato digital, analógico, serial ou paralelo. Isto permite a ligação do microcontrolador a dispositivos externos, como visor, teclados, modem, chaves, sensores, etc [67].

Normalmente existe uma preocupação em se desenvolver dispositivos com pequeno *consumo de energia*, principalmente quando usamos baterias para alimentação. Alguns microcontroladores apresentam diferentes versões, quanto à tecnologia de fabricação, possibilitando diferentes valores relativos ao consumo de energia [68]. Algumas destas versões ainda permitem certos modos de operação

onde o processamento fica suspenso até a ocorrência de uma interrupção ou reinicialização do sistema, permitindo uma redução no consumo de energia [57].

O *suporte de desenvolvimento* engloba manuais, simuladores, montadores, artigos, exemplos de aplicações, consultoria técnica e fornecedores.

Outro critério importante é o *custo* de um componente. Embora este critério dependa de uma relação benefício-custo, nota-se que, de um modo geral, o custo de componentes vem diminuindo com o tempo. Atualmente um microcontrolador, como o 8031 da INTEL, pode ser encontrado no mercado brasileiro por aproximadamente cinco dólares.

2.6 O software

Na escolha de uma linguagem para desenvolvimento do software, temos que considerar alguns fatores, como a velocidade e tamanho do programa executável gerado e a facilidade de manutenção do software. Estes fatores não dependem somente da linguagem, mas também do compilador ou montador utilizados. As linguagens de alto nível normalmente apresentam mais facilidade na manutenção de programas, enquanto que a linguagem montadora, pode, em muitos casos, apresentar um programa executável menor e mais rápido.

Um bom programa montador é uma poderosa ferramenta no desenvolvimento de software em linguagem de baixo nível, permitindo cláusulas condicionais, inclusão de arquivos, uso de macros, geração de tabelas de símbolos e criação de sub-rotinas com rótulos locais [70]. As linguagens de alto nível, por sua vez, vêm ganhando um espaço cada vez maior, com o desenvolvimento de eficientes compiladores, principalmente para linguagem C, que geram programas executáveis cada vez mais rápidos e pequenos [77].

Como o software também depende do hardware disponível, alguns fabricantes têm procurado novas técnicas para o desenvolvimento de processadores, que realizem as operações de software de um modo otimizado, procurando maximizar

a velocidade de processamento. Um exemplo são os processadores do tipo RISC (Reduced Instructions Set Computers), que usam todas as instruções com um único ciclo de máquina, "pipelining" e com a maioria das instruções manipulando registros, ficando a memória com operações do tipo "load/store" [58].

O IEEE (*Institute of Electrical and Electronics Engineers*), dos Estados Unidos, dividiu o conjunto básico de instruções de um processador genérico em sete partes, que são [31]:

- 1 - Instruções aritméticas
- 2 - Instruções lógicas
- 3 - Instruções de movimentação de dados
- 4 - Instruções de desvio
- 5 - Instruções de chamada de sub-rotinas
- 6 - Instruções de retorno
- 7 - Instruções diversas

Todo dispositivo computacional necessita de um software de controle, chamado *software básico*, que atua diretamente sobre o seu hardware. Em máquinas de uso genérico, há um tipo especial de software básico, chamado **sistema operacional**, que funciona como uma interface entre o hardware e o programa aplicativo do usuário, liberando o programador de manipular dispositivos de hardware [58]. A complexidade de um sistema operacional varia em função do sistema que o mesmo controla, existindo um tipo mais simples de sistema operacional, chamado **monitor**, que visa interfacear o usuário com máquinas de uso genérico, porém, projetadas de modo relativamente mais simples [30].

Em muitos dispositivos de aplicações dedicadas, o software aplicativo, além das suas próprias funções aplicativos, realiza as funções de controle da máquina,

dispensando assim o uso de um sistema operacional específico. Neste caso, o programador normalmente desenvolve uma biblioteca com rotinas de manipulação do hardware, que são incorporadas diretamente ao aplicativo.

2.6.1 Ferramentas de software

Em um projeto com microcontroladores, o software geralmente representa a maior parte do custo de desenvolvimento do projeto [70], obrigando o projetista a se preocupar com ferramentas que auxiliem o desenvolvimento do software e possam permitir uma redução deste custo. As ferramentas auxiliares para o desenvolvimento de software podem ser manuais do fabricante, exemplos de programas, simuladores, montadores e compiladores eficientes.

Atualmente as ferramentas CASE (*Computer Aided Software Enginnering*) podem ser usadas no trabalho de software com microcontroladores, auxiliando no projeto, implementação, consistência, testes, organização e documentação, tornando-se um eficiente recurso para o gerenciamento de projeto de software [35, 70].

Outro tipo de ferramenta auxiliar são os simuladores, que permitem executar, num processador, os programas desenvolvidos para outro tipo de processador diferente, possibilitando ainda que sejam analisados os registradores, áreas da memória e portas de entrada/saída, além de permitir a simulação da chegada de sinais em portas lógicas ou pinos de interrupção.

O software básico final é resultado de muitos testes a fim de se conseguir fazer uma perfeita interação entre o hardware e o software. Até que esta interação seja conseguida, podem ser detectadas falhas como ruídos, temporizações, lógica combinacional errada ou níveis de alimentação, que exijam alterações no projeto ou montagem do hardware.

Este trabalho é facilitado pelos simuladores, onde o software pode ser experimentado em um ambiente livre de problemas de hardware [01, 35] e pelos

circuitos emuladores, que permitem depurar o desenvolvimento do hardware, facilitando os testes e interação software-hardware [09, 13].

2.7 Memórias

As memórias são dispositivos que armazenam informações, sendo normalmente implementadas através de dispositivos semicondutores ou magnéticos. Atualmente há milhares de diferentes tipos de memórias [54].

Basicamente, as memórias podem ser classificadas em **voláteis e não voláteis**. As primeiras perdem o seu conteúdo na falta de alimentação, enquanto as não voláteis retêm o conteúdo.

As memórias voláteis podem ser subdivididas em *dinâmicas e estáticas*. As primeiras são formadas internamente por capacitores, cuja carga determina o valor do bit armazenado. Estas memórias apresentam um baixo consumo de energia, porém, exigem que os capacitores internos sejam constantemente recarregados, para que não percam a informação armazenada. Isto requer um circuito adicional especial, que se encarrega de recarregar a memória constantemente. O segundo tipo de memórias voláteis são as memórias estáticas, que permitem rápida manipulação das informações, fazendo o armazenamento interno através de dispositivos do tipo *flip-flop* [54].

As memórias não voláteis podem ser encontradas na forma de memórias somente para leitura, que vêm gravadas pelo fabricante e não podem ter o seu conteúdo alterado pelo usuário, no entanto, há memórias não voláteis especialmente projetadas para serem gravadas e apagadas pelo usuário, permitindo novas gravações. O processo de se apagar uma memória não volátil é através da exposição da mesma à luz ultra-violeta ou, em alguns tipos especiais, este apagamento pode ser feito eletricamente, sob controle do processador [54].

A utilização de uma memória exige a análise de certos fatores que permitem determinar sua adequação ao projeto, tais como a velocidade de acesso, a capacidade de armazenamento, o consumo de energia e o custo [54].

2.7.1 Memória flash

Uma significativa evolução no desenvolvimento das memórias não voláteis, eletricamente apagáveis, foi dado com a construção das memórias do tipo **flash**, que possuem vantagens sobre as tradicionais não voláteis do tipo EEPROM, como menor consumo de energia, maior densidade e maior velocidade de escrita, chegando a ser 15 vezes mais rápidas [56].

Muitas aplicações, principalmente na área de coleta de dados, necessitam que os dados sejam armazenados, por longos períodos de tempo. Este armazenamento poderia ser feito por um sistema, que seria composto por uma memória volátil e uma fonte auxiliar de alimentação, que garantiria a alimentação da memória, em caso de falhas da fonte principal de alimentação. Este sistema, além da bateria auxiliar, que também pode falhar, requer um chaveamento para controlar o fornecimento de energia. Outro processo, que poderia ser usado, seria o uso de memórias magnéticas, compostas por discos ou fitas, porém, este sistema exigiria um controle eletromecânico, que dificultaria o hardware e aumentaria o custo, além de ser mais suscetível a falhas. Um terceiro e eficiente método é o uso de memórias flash, que são facilmente controladas pelo processador e não perdem o conteúdo com a falta de alimentação [54].

As memórias flash vêm ganhando grande espaço no mercado, em função de suas características [33]. Como elas podem ser facilmente programadas e reprogramadas, já são usadas como unidades de "disco de estado sólido", em computadores portáteis (*laptops*) como o modelo MC400 da Psion Inc. [56], ou como unidades armazenadoras do sistema básico de entrada e saída (BIOS) e até do próprio sistema operacional, em microcomputadores.

Usando-se uma flash, o BIOS pode ser facilmente substituído por uma versão mais recente, sem necessidade de alterações no hardware. O próprio sistema operacional, quando totalmente armazenado em uma flash, não precisaria ser carregado do disco, tornando mais rápida a inicialização do sistema e permitindo a instalação de outra versão, através da simples reprogramação da memória [34]. Este novo processo de armazenamento tem exigido esforços no desenvolvimento de produtos confiáveis e padronizados. Assim as indústrias que atuam nesta área se uniram na PCMIA (*Personal Computer Memory Card International Association*), com a finalidade de criar padrões específicos de operacionalidade [54].

Estas memórias, além de versáteis, apresentam uma alta confiabilidade, ou seja, podem reter suas características inalteradas por longos períodos de uso contínuo. Os fabricantes garantem a realização confiável de pelo menos 10.000 ciclos de programação/deleção, porém, na prática, este valor pode atingir mais de 100.000 ciclos [54, 56].

2.8 Interfaceamento

Uma UCP necessita de um circuito mínimo para funcionar e frequentemente existe a necessidade de serem ligados vários dispositivos que atendam às exigências requeridas do hardware, como interfaces seriais, paralelas, teclados ou visores. A conexão das partes do sistema é chamada interfaceamento [31], e exige que sejam observados itens como a sincronização, sentido da transmissão de dados ou ajuste dos níveis do sinal.

Um dos dispositivos mais comuns de entrada e saída são as interfaces paralelas, que normalmente permitem o fluxo de dados de forma bidirecional, em grupos de 8 ou mais bits. Existem dispositivos programáveis específicos para esta função, porém os microcontroladores já vêm com este dispositivo embutido, permitindo inclusive a manipulação direta de bits isolados [57]. Uma porta de entrada e saída bidirecional de um bit pode ser esquematizada conforme a figura 2.6 [25].

Fig. 2.6 Diagrama de um pino de E/S

Os dispositivos interfaceados normalmente podem ser controlados através de um determinado endereço associado ao mesmo, sendo que este endereço é chamado de porta de entrada ou saída, dependendo do sentido do fluxo dos dados.

Uma técnica de se associar um dispositivo de E/S a um determinado endereço é com um processo chamado de *entrada/saída isolada*, onde um pino especial da UCP, permite ativar o dispositivo relativo ao endereço, enquanto a memória permanece desabilitada. Este pino é automaticamente acionado em função de instruções específicas para manipulação de dispositivos. Isto permite que um mesmo endereço seja usado para endereçar, de forma não simultânea, a memória e a porta de E/S [31].

Outra maneira de se habilitar dispositivos de E/S é através de uma técnica chamada de *entrada e saída mapeada em memória* [25, 31]. Neste caso, o processador não possui o pino especial para ativação de portas de E/S, e todo o controle é feito por um circuito combinacional, que auxiliado pelos pinos de leitura e escrita da UCP, decodifica um determinado endereço, habilitando a memória ou uma determinada porta de entrada e saída.

2.9 Interrupções

A interrupção é um recurso disponível nas unidades processadoras, permitindo que a mesma atenda, através de uma rotina especial, a ocorrência de

determinados eventos. A unidade processadora pode ter diversas fontes de interrupções, que, quando devidamente acionadas, permitem que a UCP inicie o processamento da rotina de tratamento.

Após o acionamento de uma interrupção, o processo de atendimento da mesma é constituído por uma sequência de passos [31, 40], onde a UCP genericamente realiza as seguintes atividades:

- 1 - Termina a execução da instrução corrente.
- 2 - Coloca na pilha o conteúdo do contador de programas (PC) e de alguns registradores.
- 3 - Faz um desvio para determinada posição da memória, onde está armazenado o início de uma rotina que deve ser executada quando houver o pedido de interrupção.
- 4 - Ao terminar a rotina de tratamento da interrupção, os dados, inicialmente empilhados, são recuperados.
- 5 - Retorna da rotina de interrupção, continuando o processamento a partir do ponto inicialmente interrompido.

A maioria das UCPs possui interrupções chamadas de **mascaradas**, cujo atendimento pode ser desabilitado por software, permitindo maior controle sobre as mesmas [25]. Nos sistemas com várias fontes de interrupções, normalmente, existem registradores especiais que permitem determinar quais interrupções serão ativadas, ou que podem permitir a desabilitação simultânea de todas as interrupções [57].

Além das interrupções geradas por sinais externos, existem interrupções que podem ser geradas por software ou por dispositivos internos. Os sinais que acionam as interrupções externas devem ter certas características, como, por exemplo, nível ou variações de rampa.

Como duas ou mais interrupções podem ser acionadas simultaneamente, a UCP deve ter algum critério para determinar qual será atendida. A UCP normalmente faz um "polling" (verificação), analisando quais pinos foram ativados, segundo uma ordem pré definida, atendendo segundo esta ordem. Alguns sistemas permitem que se usem níveis diferentes de prioridades, fazendo a verificação dos pedidos de interrupções com maior prioridade e depois nas interrupções de menor prioridade. Como esta verificação é feita de modo seqüencial, o sinal externo de interrupção deve ter um tempo mínimo de duração, para que não corra o risco de ser ignorado.

Em alguns sistemas, quando o processador estiver atendendo a uma interrupção, todos os outros pedidos de interrupções, deste período, serão ignorados. Outros sistemas possuem um processo "nesting" [25], onde uma interrupção de maior prioridade pode interromper o processamento já iniciado por uma interrupção de menor prioridade. Neste caso, após atender a interrupção de maior prioridade, o sistema retorna à execução da rotina que estava sendo atendida anteriormente.

Existe um intervalo de tempo mínimo, até a solicitação de uma interrupção e o início da rotina associada. Este tempo mínimo depende de como a UCP faz a verificação de interrupções, da sua velocidade de processamento e do número de ciclos de máquina requeridos pela instrução corrente no momento da ativação da interrupção. Nos microcontroladores da linha MCS-51, por exemplo, ao fim de cada ciclo de máquina, o controlador atualiza os flags de interrupção, e, no ciclo de máquina seguinte, executa verificação dos flags anteriormente ativados. Esta verificação é baseada nos níveis de prioridades e, dependendo da instrução em execução, o atendimento à interrupção pode demorar de 3 a 9 ciclos de máquina, onde cada ciclo exige 12 períodos do oscilador. Isto equivale a um tempo de 3 a 9 microsegundos, usando um cristal de 12 MHz [57].

2.10 Dispositivos de temporização

O computador pode ser usado para manipular eventos em função do tempo, como sistemas de temporização, geração de pulsos em determinada frequência, medição de sinais externos, geração de taxa de transmissão em sistemas de comunicação, etc [25].

Embora existam dispositivos programáveis de temporização, os microcontroladores possuem um ou mais temporizadores internos, que podem ser configurados, por software, para funcionarem como temporizador interno, trabalhando em função da frequência do cristal oscilador, ou como contadores de eventos externos, conforme figura 2.7 [57, 69].

Fig. 2.7 Esquema de um dispositivo temporizador/contador

Na função de temporização há um registrador, que é incrementado em função de um certo número de períodos do oscilador, permitindo uma contagem precisa de tempo. Quando houver estouro na contagem do registrador, será automaticamente gerado um pedido de interrupção, associado ao temporizador. Normalmente o processador, através do software, possui pleno controle sobre o timer, porém existem alguns timers com "free-running" [11], que, quando acionados, não podem ser manipulados por software. Os temporizadores de microcontroladores

podem ser configurados de várias formas, em função do software, como, por exemplo, com diferentes tamanhos de registradores de contagem, permissão de recarga automática, divisão automática da contagem ou ainda geração de "baud-rate". Quando configurado para atuar como contador externo, o temporizador terá seu registrador incrementado em função de cada sinal ativado no pino externo de contagem [57].

Alguns sistemas de temporização também podem ser configurados para atuarem com operações do tipo captura, onde o valor de contagem do temporizador é armazenado em um registrador dedicado, no momento em que houver a ativação de um sinal no pino externo [11].

2.11 Comunicação de dados

Um recurso computacional importante é a possibilidade de um ou mais computadores trocarem informações entre si, via interface serial. Esta forma de comunicação é muito eficiente, com possibilidade de comunicação em alta velocidade e entre computadores diferentes, desde que operem com interfaces padronizadas [51].

A interconexão de diversos equipamentos permite a formação de uma rede de comunicação, onde os mesmos podem compartilhar recursos de hardware e software [58]. O meio físico de ligação entre os computadores é normalmente feito através de cabos diretos, rede telefônica, rádio transmissão, fibras óticas e até mesmo transmissão via satélite [61]. As comunicações podem apresentar falhas em função do meio em que são transmitidas. Assim, conforme a velocidade de transmissão aumenta, o comprimento do cabo de ligação direta deve diminuir para que evite erros [60], pois, quanto maior a taxa de transmissão, menor será o comprimento do pulso elétrico que representa a informação e, portanto, poderá sofrer mais interferências [51].

Como a comunicação pode envolver diferentes equipamentos, de muitos fabricantes, é fundamental que existam parâmetros de padronização, que permitam estas ligações. Algumas organizações, como o IEEE o EIA (*Electronics Industries Association*) ou a ISO (*International Standards Organization*), procuram estabelecer padrões que disciplinem todo o contexto relativo à comunicação de dados.

Embora algumas destas organizações sejam internas dos Estados Unidos, elas possuem grande importância e influência internacional. A padronização é vantajosa para o fabricante, que torna seus produtos mais comercializados, e para o usuário, que ganha em confiabilidade e possibilidade de expansões.

A ISO criou um modelo padrão para comunicação de dados, chamado OSI (*open systems interconnection*), que foi dividido em sete camadas ou níveis, abrangendo as ligações físicas de hardware, protocolos, roteamento, níveis aceitáveis de erros, início ou fim de conexões, padrões de manipulação de dados como criptografia ou compressão de dados e finalmente os aplicativos usados no sistema de comunicação [51, 61].

2.11.1 Modos de transmissão

A transmissão de dados pode ser feita de modo paralelo ou serial. No modo paralelo, cada bit da informação é transmitido através de um cabo próprio e de modo simultâneo aos demais. Assim, a transmissão paralela de um byte exige oito cabos paralelos. Na transmissão serial é exigido apenas um cabo de transmissão, pois os bits do dado são transmitidos sequencialmente, um a um. Como na transmissão paralela os bits são transmitidos simultaneamente, existe uma maior velocidade de transmissão, porém, também existe a exigência de mais cabos, que aumentam o custo, dificultam a manipulação física e podem tornar a transmissão mais sujeita a falhas.

Nas partes internas do computador, os dados são trocados de forma paralela, através do barramento de dados, porém, externamente, a nível de comunicação, os dados são transmitidos bit a bit, de forma serial, para se evitar os problemas mencionados no parágrafo acima. O processo serial, além de mais econômico, pode ser conectado à linha telefônica, através de modems [51].

A forma de ligação física dos cabos, entre dois computadores, na comunicação serial, pode ser feita através de três métodos diferentes, que são: o método simplex, onde só existe um transmissor e um receptor, o método half-duplex, onde os dois dispositivos podem transmitir e receber, porém, nunca simultaneamente

e, finalmente, o método full-duplex, onde os dois dispositivos podem transmitir e receber, inclusive simultaneamente [25, 51].

A transmissão de dados pode ser feita no modo síncrono ou assíncrono. Na transmissão síncrona deve haver um pulso (*clock*) especial para sincronização, que coordene o fluxo de comunicação. Na transmissão assíncrona, existem bits de controle para cada dado transmitido, dispensando assim os pulsos de sincronismo. Desta forma, cada dado transmitido é precedido e finalizado por bits especiais chamados respectivamente de "start bit" e "stop bit" [61].

O modo síncrono de comunicação oferece algumas vantagens, como a eliminação de bits extras de controle e possibilidade de ligação de micros com "mainframes" que operem com comunicação síncrona. A grande vantagem da comunicação assíncrona é o custo normalmente mais reduzido de software e hardware, em relação aos sistemas síncronos [51].

2.11.2 A interface serial

A idéia de uma interface serial é a de um dispositivo que converta os sinais da forma paralela para serial e vice-versa. Uma interface popular é a do tipo UART (*universal asynchronous receiver/transmitter*), que, além da conversão paralelo/serial, faz, por exemplo, o monitoramento de ligações com modems ou armazenamento do dado até que haja possibilidade de comunicação [61].

Um dos padrões para comunicação serial é o RS-232, publicado em 1969, pela EIA, que definiu normas como a formas de conectores físicos, função dos pinos de conexão e níveis de sinais. Este padrão, além dos pinos de transmissão e recepção, possui alguns outros pinos que auxiliam o controle, verificando, por exemplo, a possibilidade de comunicação [29, 60].

Muitos microcontroladores possuem interface serial interna, que podem operar de modo síncrono ou assíncrono, em diversas velocidades de transmissão. Nestes casos, a serial possui apenas dois pinos, que são destinados à

transmissão e recepção de dados, sendo que os sinais de controle, exigidos pelo padrão RS-232, podem ser diretamente manipulados por software, através de pinos comuns de entrada e saída. No caso dos dispositivos computacionais que operam com lógica TTL, como os microcontroladores, deve haver uma conversão, no nível do sinal, para o padrão RS-232, que pode ser feita por circuitos integrados especialmente projetados para este fim.

Capítulo III

O kernel programável de controle

3.1 Introdução

Neste capítulo será descrito o kernel programável de controle desenvolvido como parte do projeto de mestrado. Serão especificadas as suas características de hardware e software, bem como as rotinas existentes no sistema operacional desenvolvido.

3.2 O hardware do kernel

O hardware do kernel, cujo circuito eletrônico é mostrado no apêndice I, é baseado em um microcontrolador programável 80N535, que possibilitou o desenvolvimento de um circuito computacional com reduzido número de componentes. A organização do kernel é mostrada na figura 3.1 e suas características são as seguintes:

- 01 - Processamento com microcontrolador
- 02 - Memória de programa (EPROM) com 16 Kbytes
- 03 - Memória de dados, não volátil, com 64 Kbytes
- 04 - Oito entradas para conversão A/D
- 05 - Interface serial no padrão RS-232
- 06 - Duas entradas de interrupções
- 07 - Dois temporizadores/contadores de 16 bits
- 08 - Um temporizador de 16 bits
- 09 - Teclado matricial de 16 teclas

10 - Visor de cristal líquido

11 - *Clock* de 12 MHz

Fig. 3.1 Diagrama lógico do kernel

3.2.1 A arquitetura de Harvard

A organização do kernel foi desenvolvida em um sistema baseado na arquitetura de Harvard. Nesta arquitetura existem barramentos de endereços diferenciados para a memória de programa e memória de dados, permitindo um aumento da capacidade total de armazenamento do sistema. Assim, um sistema com 16 linhas de endereçamento, por exemplo, poderia endereçar 64 Kbytes de memória de dados e 64 Kbytes de memória de programa.

Na arquitetura de von Neumann existe apenas um barramento de endereços, que é utilizado pela memória de programa e pela memória de dados. Assim, por exemplo, um sistema com barramento de endereços de 16 linhas poderia endereçar, no máximo, um total de 64 Kbytes de memórias. Ou seja, neste caso, a capacidade de armazenamento da memória de programa, somada à capacidade de armazenamento da memória de dados, seria de no máximo 64 Kbytes [31].

O microcontrolador usado kernel, possui apenas um barramento de endereços, porém, possui um pino especial de controle de memórias, que é automaticamente ativado em função do software. Este pino permite que o sistema diferencie quando um endereço é relativo à memória de dados ou à memória de programas, assim, com 16 linhas de endereços, o kernel pode endereçar até 64 Kbytes de memória de programa e 64 Kbytes de memória de dados. Embora exista somente um barramento de endereços, este sistema se assemelha à arquitetura de Harvard [05], pois existe um aumento da capacidade de armazenamento e as memórias podem possuir endereços iguais.

3.2.2 As memórias do kernel

Fig. 3.2 (A) memória interna de dados
(B) memória externa de programas
(C) memória externa de dados

O kernel possui uma memória de programa do tipo EPROM, modelo 27128A, com 16 Kbytes. A memória externa de dados é uma memória não volátil, do tipo *Flash*, modelo 28F512, com 64 Kbytes. Este dispositivo foi escolhido pelo fato de ser não volátil e pela alta densidade, que permite o armazenamento de 64 Kbytes

em um único componente, pelo baixo consumo de energia e pelo fato de ter todas as suas operações diretamente controladas pelo microcontrolador.

O sistema também conta com uma segunda memória de dados com 256 bytes interna do microcontrolador. A figura 3.2 mostra as três memórias existentes no sistema.

O acesso às memórias interna ou externa de dados, pode ser feito através de instruções com modos diferentes de endereçamento, não exigindo nenhum esquema especial de hardware.

O acesso à memória externa de dados ou à memória externa de programas é feito através de instruções diferentes. As instruções de acesso à memória externa de dados ativam os pinos RD e WR do microcontrolador, enquanto que uma instrução de acesso à memória externa de programa, ativa o pino PSEN (*Program Store Enable*), conforme tabela 3.1. Isto permite que um mesmo endereço seja utilizado pelas memórias de dados e de programas, possibilitando o uso do microcontrolador no desenvolvimento de um circuito computacional baseado na arquitetura de Harvard.

Tabela 3.1 Pinos de controle das memórias externas

<i>PSEN</i>	<i>RD</i>	<i>WR</i>	<i>OPERAÇÃO</i>
1	1	0	Escrita na memória de dados
1	0	1	Leitura na memória de dados
0	1	1	Leitura na memória de programas
1	1	1	Memórias externas desativadas

3.2.3 Velocidade de processamento

O kernel usa um cristal de 12 MHz, garantindo que a maioria das instruções do processador seja executada em 1 microsegundo. Como o manual do microcontrolador informa o número de ciclos de máquina necessários para a execução de cada instrução, podemos facilmente calcular o tempo gasto no processamento de um programa. Os demais critérios de velocidade, como atendimento a uma interrupção ou conversão analógico/digital, serão discutidos nas respectivas seções.

3.2.4 O teclado

O teclado utilizado é um modelo matricial de 16 teclas (4x4), controlado diretamente através de alguns pinos de entrada/saída do microcomputador. Estes pinos são especificados pela notação Px.y, onde x é o número da porta paralela e y é o número de um bit da porta. Os pinos de P5.4 a P5.7 foram configurados como linhas de varredura e os pinos de P4.0 a P4.3 foram configurados como linhas de retorno. As entradas dos pinos de retorno foram colocadas em nível alto por hardware (*pull-up*), enquanto que as saídas dos pinos de varredura foram colocadas, por software, em nível baixo.

Quando uma tecla é pressionada, é feito um contato entre uma linha de varredura e uma linha de retorno, aterrando-se o sinal originariamente alto. Na rotina de manipulação do teclado, as linhas de retorno são analisadas para verificar qual delas teve seu sinal aterrado. Logo em seguida, os pinos de varredura têm o seu sinal colocado em nível alto, um a um, e é feita uma verificação se o sinal de retorno voltou ao nível alto. Isto permite que se verifique qual a linha e a coluna do teclado que foram pressionadas.

No projeto de hardware não foi implementada uma interrupção especial para teclado, assim a rotina para leitura do teclado fica em laço (*loop*) até que seja pressionada alguma tecla. Isto deixa uma entrada a mais de interrupção disponível para aquisição de dados externos. Caso haja a necessidade de uma interrupção

específica para teclado, pode-se usar um dos três temporizadores disponíveis para fazer uma verificação no teclado, de tempos em tempos.

3.2.5 O visor

O visor utilizado é um visor de cristal líquido, com 2 linhas e 16 colunas, compatível com linha LTN da Philips. Este visor é um módulo programável, possuindo uma pinagem padronizada, com oito pinos de dados e três de controle. Assim, o kernel poderá ser dotado de qualquer um dos diversos modelos comerciais, que seguem este padrão.

Um visor de cristal líquido apresenta as vantagens de um baixo consumo de energia e possibilidade de ser diretamente controlado pelo microcontrolador, sem necessidade de componentes auxiliares. Estes visores são uma importante interface homem/máquina, sendo utilizados inclusive como terminais gráficos coloridos [36].

Este tipo de visor é formado por um material que apresenta características intermediárias entre os estados sólido e líquido. Este estado intermediário da matéria foi descoberto em 1888 pelo botânico austríaco Friedrich Reinitzer [37, 38, 41]. Porém, somente em 1968 começaram a surgir os primeiros ensaios de aplicações desta descoberta, quando foram divulgadas pesquisas sobre a influência de um campo elétrico sobre os cristais líquidos [41].

No estado sólido as moléculas têm um posicionamento fixo e rígido, enquanto que no estado líquido as moléculas têm uma direção aleatória de alinhamento. Em um cristal líquido, a direção de alinhamento das moléculas pode ser alterada e pode ser diretamente influenciada por um campo elétrico. Esta direção de alinhamento determinará o índice de refração da luz que se propaga pelo cristal líquido, dando ao mesmo uma aparência clara ou escura [36, 38] e permitindo a formação de caracteres.

No kernel, o visor é diretamente controlado por pinos de E/S. Neste caso, as informações são enviadas diretamente pela porta paralela P0. O pino P5.0 habilita o visor, o pino P5.2 determina se o mesmo executará uma operação de escrita de informação ou se executará a leitura de um flag especial e o pino P5.1 determina se a informação, presente na porta P0, é um dado a ser impresso ou se é uma informação de controle do visor. A tabela 3.2 mostra os pinos de E/S usados no controle do visor.

Tabela 3.2 Pinos de controle do visor

<i>P0</i>	<i>P5.0</i>	<i>P5.1</i>	<i>P5.2</i>	<i>OPERAÇÃO</i>
X	0	X	X	Visor desabilitado
Flag	1	X	1	Leitura do flag
Controle	1	0	0	Controle do visor
ASCII	1	1	0	Imprime caractere no visor

3.2.6 Entradas de sinais digitais

O kernel possui duas entradas de interrupções e dois dispositivos contadores. As interrupções podem ser programadas para serem ativadas por nível ou transição do sinal e poderão ser atendidas, segundo uma ordem de prioridades, em um tempo mínimo que varia de 3 a 9 microsegundos. O pino de entrada da interrupção zero (INT0) foi ligado a um acoplador ótico TIL113, que garante uma proteção contra sobrecargas. O pino de interrupção um (INT1) não está acoplado em nenhum dispositivo especial, ficando disponível para outras implementações.

Além das duas entradas de interrupções, o kernel dispõe de dois dispositivos do tipo temporizador/contador, que podem ser configurados, por

software, para operarem como contadores de sinais externos, com precisão de 16 bits, operando com uma frequência de até 500 KHz.

3.2.7 Entradas analógicas

No kernel existem oito entradas multiplexadas, para conversão analógico/digital, de 8 bits, com um tempo de conversão de 15 microsegundos. O conversor A/D está embutido no microcontrolador e pode manipular sinais em uma faixa que varia de 0 a 5 volts, com exatidão de 5 a 20 mV. O conversor analógico/digital do microcontrolador é explicado de modo mais detalhado na seção 3.2.10.

3.2.8 Temporizadores

Existem três temporizadores de 16 bits disponíveis no kernel. Os temporizadores T0 e T1 fazem parte dos dispositivos do tipo temporizador/contador, já mencionados no item 3.6. O temporizador T2 também faz parte de um dispositivo temporizador/contador, mas teve seu pino de contagem usado como linha de dado do visor de cristal líquido, podendo ser programado somente para operar como temporizador.

Estes temporizadores, que trabalham com uma frequência de 1 MHz, podem operar em diferentes modos, usando, por exemplo, recarga automática ou divisão da contagem. Os temporizadores T0 e T1 também podem ter seu funcionamento associados aos pinos de interrupção INT0 e INT1, respectivamente. Neste caso, o temporizador só funciona se o pulso presente no pino de interrupção for alto, facilitando a medição da largura de pulsos. Os temporizadores T1 e T2 podem ser configurados para gerar a taxa de transmissão do sistema de comunicação serial.

3.2.9 A interface serial

O kernel possui uma interface serial que opera no padrão RS-232, em modo síncrono e assíncrono. O microcontrolador possui dois pinos que realizam a saída e entrada de dados na comunicação serial, operando em padrão TTL. Assim, usamos um dispositivo 1488 que faz a conversão TTL/RS e um dispositivo 1489 que faz a conversão RS/TTL, para padronização dos sinais em nível RS-232.

O padrão RS-232, além dos pinos de transmissão e recepção, exige alguns pinos adicionais de controle. Por isso, alguns pinos de E/S do microcontrolador foram conectados aos conversores de níveis TTL/RS e RS/TTL, para serem configurados, por software, como pinos de controle da interface, permitindo a implantação do padrão mínimo exigido. A tabela 3.3 mostra a função dos pinos de E/S do 80N535 e sua ligação a um conector DB25, para implementação de uma interface RS-232.

Tabela 3.3 configuração da RS-232 no kernel

<i>DB25</i>	<i>FUNÇÃO</i>	<i>PINO DE CONTROLE DO MICROCONTROLADOR</i>
02	TX (transmite dado)	P3.1
03	RX (Recebe dado)	P3.0
04	RTS (Solicita envio)	P5.3
05	CTS (Livre para enviar)	P4.4
06	DSR (Modem pronto)	P4.7
07	TERRA	Vss
08	DCD (Detetor de portadora)	P4.5
20	DTR (Terminal de dados pronto)	P4.6

3.2.10 O microcontrolador usado no kernel

O microcontrolador utilizado no desenvolvimento do kernel foi o modelo 80N535, fabricado pela empresa AMD, esquematizado na figura 3.3. Esta escolha foi feita segundo os critérios sugeridos por Melear [66] e Vaglica [67], discutidos na seção 2.5.1. Estes critérios levaram em consideração os sinais a serem manipulados, capacidade de endereçamento, volume do processamento a ser realizado e ferramentas de trabalho.

Fig. 3.3 Configuração lógica do microcontrolador 80N535

Entre as características do microcontrolador programável 80N535, podemos destacar [69]:

- 1 - 256 bytes de memória RAM interna
- 2 - Seis portas paralelas de 8 bits
- 3 - Três temporizadores/contadores
- 4 - Saída serial (full-duplex)
- 5 - Doze fontes de interrupções (internas e externas)

- 6 - Oito entradas de conversão A/D
- 7 - Software compatível com a linha MCS-51
- 8 - endereçamento para 64 kbytes de memória de dados e 64 Kbytes para memória de programas

Além do endereçamento externo e dos 256 bytes de memória RAM interna, endereçada de 00 a 0FFh, o microcontrolador possui uma segunda RAM interna com 128 bytes, endereçada de 80h a 0FFh, chamada de SFR (*special function register*). O SFR armazena alguns registros especiais de trabalho. Embora haja uma sobreposição de endereços, existem modos diferentes de endereçamento para a memória interna e o SFR.

A SFR contém 41 registradores especiais que incluem registros associados a interrupções, acesso a portas lógicas, transferência serial de dados, manipulação do conversor A/D, ponteiros, registradores de dados, etc.

O 80N535 possui 6 portas paralelas (48 linhas) de entrada e saída. As portas P0 e P2 são usadas como linhas de endereçamento, sendo que a porta P0 é usada para multiplexar a parte menos significativa do barramento de endereços e o barramento de dados, assim o endereçamento deve ser demultiplexado através de um dispositivo do tipo latch, controlado pelo pino ALE (*Address Latch Enable*) do microcontrolador.

O pino PSEN, já descrito na seção 3.3, tem a finalidade de habilitar a memória externa de programas.

Este microcontrolador possui uma versão, chamada 80N515, que contém uma memória ROM interna. Assim, se o pino EA (*External Access Enable*) estiver em nível 1, o processador ativará os primeiros 8 kbytes de endereços de programa na ROM interna e o restante na ROM externa. Se este pino estiver em nível baixo, como é sempre feito no 80N535, todo acesso será feito exclusivamente em uma memória externa de programa. Em um sistema que use somente as memórias internas de programa e dados do microcontrolador, as portas paralelas P0 e P2 poderão ser usadas para entrada/saída.

As portas paralelas P1 e P3 têm uma dupla configuração, e sua utilização deve ser especificada pelo usuário, pois podem funcionar como portas de E/S ou terem seus pinos destinados para fins específicos de interrupções, saída serial, contadores ou RD/WR.

As portas P4 e P5 são usadas exclusivamente como portas paralelas de entrada/saída, podendo alimentar ou drenar até quatro cargas TTL-LS.

O microcontrolador 80N535 possui 8 entradas multiplexadas para **conversão analógico/digital** (AN0-AN7). A conversão é feita pelo método de aproximação sucessiva requerendo um tempo de 15 microsegundos para uma conversão, quando se usa um oscilador de 12 MHz. Existem tensões de referência interna, que determinam a exatidão da conversão e estabelecem os valores mínimo e máximo do sinal de entrada. Esta tensão pode ser ajustada por software, em função da tensão nos pinos externos VAREF (*Reference voltage*) e VAGND (*Ground Reference*).

Os limites mínimo e máximo, das tensões de referência externa são 0 e 5 volts respectivamente, sendo que as tensões de referência interna podem ser ajustadas, por software, em passos de 1/16 relativos a tensão de referência externa e permitem que o A/D seja ajustado para uma exatidão na faixa de aproximadamente 5 a 20 mV. Deve existir um mínimo de 1 volt entre as tensões de referência interna para que o A/D funcione convenientemente.

O conversor também pode ser ajustado por software para realizar a conversão de modo contínuo ou realizar apenas uma única conversão do sinal de entrada. O monitoramento da conversão pode ser feito através de um flag especial, chamado *BSY* que é pertencente ao registrador *ADCON*, que indica se a conversão está sendo executada ou se o seu resultado pode ser manipulado. O A/D também pode ser programado para gerar um pedido interrupção após cada conversão.

O Pino VPD (*Power-Down Supply*) é usado no caso de queda na entrada de alimentação (VCC). Quando for detectada uma falha no sistema de alimentação, alguns dados relevantes poderão ser salvos nos endereços de 88 a 127 da RAM interna. Enquanto a alimentação normal não se restabelece, o conteúdo destas

posições será garantido pela alimentação do pino VPD. Quando a alimentação se restabelecer, os dados poderão ser recuperados nestes endereços.

O pino VBB (*Substrate Pin*) deve ser conectado ao terra, através de um capacitor, para funcionamento do conversor A/D.

Os pinos XTAL1 e XTAL2 são usados para conexão de um cristal com frequência que pode variar de 1,2 a 12 MHz, para funcionamento do oscilador interno do microcontrolador.

O pino RESET deve ser mantido em nível baixo por dois ciclos de máquina, para que haja uma reinicialização, após o que, o microcontrolador armazena valores específicos nos registradores.

3.3 O software do kernel

O software do kernel é desenvolvido em função do microcontrolador programável 80N535, que é compatível com a linha MCS-51, podendo ser desenvolvido em linguagens de alto ou baixo nível. Inicialmente foi desenvolvido um conjunto de sub-rotinas para controle direto do hardware do kernel. Isto facilita a implementação de novos programas, permitindo que o programador concentre seu trabalho basicamente no aplicativo a ser desenvolvido.

Estas sub-rotinas estão gravadas em um arquivo chamado **monitor.asm**, que pode ser diretamente incluído nos aplicativos. As rotinas existentes no programa monitor não destroem áreas da memória ou registradores, podendo ser diretamente ativadas através de chamadas de sub-rotinas. As rotinas disponíveis são as seguintes:

01 - **SETLCD**: Inicializa o visor de cristal líquido, com 8 bits de dados, cursor piscante e sem rotação.

Exemplo: lcall setlcd

02 - **COMLCD**: O conteúdo do acumulador é enviado ao visor e encarado como um comando.

Exemplo: mov a,#0f2h

lcall comlcd

03 - **STRING**: Imprime uma string no visor. O endereço do primeiro caractere da string deve estar no apontador de endereços e a mesma deve terminar com um cifrão ('\$').

Exemplo: str db "mensagem para o visor \$"

mov dptr,#str

lcall string

04 - **CHARLCD**: Imprime um caractere no visor. O código ASCII do caractere deve estar no acumulador.

Exemplo: mov a,#35h

lcall charlcd

05 - **READKEY**: Faz uma leitura do teclado e retorna, no acumulador, o valor digitado, entre 0 e 15. Esta rotina trava o processamento até que seja digitada alguma tecla.

06 - **RED2D**: Faz a leitura de um número decimal de dois dígitos no teclado, imprimindo o valor digitado no visor e retornando o decimal digitado (00 ... 99) no acumulador.

07 - **WRITE2D**: Imprime no visor o conteúdo do acumulador, usando dois dígitos decimais. O conteúdo do acumulador deve estar entre 00 e 99.

08 - **WRITE3D**: Imprime no visor o conteúdo do acumulador, usando três dígitos decimais.

09 - **DELFLASH**: Realiza a operação programada de apagamento da memória não volátil de dados, do tipo flash.

10 - **GRAFLASH**: Realiza a operação programada de gravação na flash. O valor gravado será o conteúdo do acumulador, no endereço contido no apontador de endereços.

11 - **PROGIO**: Programa e habilita a interrupção externa zero (INT0) para operar com rampa descendente.

12 - **PROGT0**: Programa o temporizador zero (TC0) para gerar vinte interrupções por segundo.

13 - **PROGT1**: Programa o temporizador um (TC1) para gerar uma taxa de transmissão de 1200 BPS para a interface serial de comunicação.

14 - **SERIAL**: Programa a interface serial do microcontrolador para comunicação assíncrona, sem paridade, com 1200 BPS, 8 bits por dado, um start bit e um stop bit.

15 - **ADCONV**: Faz uma conversão do sinal de entrada analógica, com sensibilidade de 5mV. O número da entrada deve estar no acumulador e a tensão de entrada deve estar entre 0 e 1.25V. O valor convertido será retornado no acumulador.

16 - **ADCONV5**: É semelhante a rotina ADCONV, porém, a tensão de entrada deve estar entre 0 e 5V, permitindo uma sensibilidade de 20mV.

Capítulo IV

Aplicações em pluviometria e PCR

4.1 Introdução

Este capítulo descreve dois aplicativos desenvolvidos para o kernel programável de controle. O primeiro aplicativo é uma configuração para automação da coleta de dados em pluviômetros, onde o kernel registra o volume de chuvas ocorrido em uma região. A segunda aplicação desenvolvida é uma configuração onde o kernel atua no controle de um processo para automação de experimentos com PCR. Esta técnica, conforme já mencionado na introdução, é usada na área biomédica e permite a reprodução de DNA *in vitro*.

4.2 Pluviometria

Pluviometria é a medição do volume de chuvas. Esta grandeza é chamada índice pluviométrico e é medida através de instrumentos chamados pluviômetros.

Os pluviômetros mais antigos apresentam o problema de necessitarem de um observador humano para registrar as medidas. Alguns modelos, também antigos, fazem este registro através de gráficos, por um processo mecânico, que pode se tornar muito sensível a falhas. Estes fatores tornam difíceis as medições em tempo real ou em locais remotos, mas podem ser contornados através de um sistema para automação de coleta de dados em pluviômetros.

4.2.1 Automação da pluviometria

O kernel foi configurado para atuar como um sistema que permite a coleta automática das medições feitas pelo pluviômetro e a sua transmissão, em tempo real, para os centros de pesquisas meteorológicas. Para automação, o pluviômetro é dotado de um dispositivo elétrico que fornece um sinal em função dos índices pluviométricos. Este sinal é enviado ao kernel que registra a ocorrência do evento.

Como o kernel é um dispositivo programável, podemos implementar uma coleta inteligente de dados, estabelecendo critérios de horários, análise de memória disponível ou pré processamentos gerais. Além disto o kernel possui uma interface serial no padrão RS-232, podendo ser conectado à rede telefônica, via modem. Em regiões sem disponibilidade de rede telefônica e onde as pesquisas exigissem uma coleta de dados em tempo real, a transmissão poderia ser feita através de rádio transmissores ou mesmo com transmissão de dados via satélite.

Alguns tipos de pesquisas meteorológicas não necessitam da coleta dos índices pluviométricos em tempo real. Neste caso, os dados são coletados durante determinados períodos de tempo e somente usados após a coleta. Este tipo de tratamento de dados é chamada de *pos-factum* e dispensa a transmissão de dados via linha telefônica.

Na coleta *pos-factum*, sem uso de transmissão via rede telefônica, o kernel seria retirado do campo e levado aos laboratórios, onde seria feita a transferência de dados a um computador central. Outra possibilidade seria que um técnico levasse um computador portátil (*laptop*) e fizesse esta coleta diretamente no campo ou simplesmente trocasse a memória de dados não volátil.

4.2.2 Tipos de pluviômetros

O primeiro passo para desenvolver um sistema de aquisição de dados em pluviômetros é uma análise dos tipos de pluviômetros existentes. Os tipos de pluviômetros estudados foram o pluviômetro do tipo caçamba, o basculante e o pluviógrafo.

O pluviômetro do tipo **caçamba** é formado por um recipiente, similar a um balde, que coleta e armazena a água da chuva. Verificando o nível de água dentro do recipiente, pode-se determinar o índice pluviométrico ocorrido. Após a realização da verificação, a caçamba pode ser esvaziada através de uma torneira própria. Este pluviômetro possui uma medição limitada pelo tamanho da caçamba e as medições não devem ser realizadas muito tempo após a ocorrência das chuvas, uma vez que a amostra coletada poderia evaporar, causando uma medição errada.

O pluviômetro do tipo **basculante** é composto por uma espécie de eixo com pequenas cubas nas extremidades, semelhante a uma "gangorra". Quando uma das cubas é enchida, ocorre uma alternância na posição da gangorra, indicando a ocorrência de um determinado volume de chuvas. Para que ocorra uma alternância do sistema, deve haver a ocorrência de um volume mínimo de chuvas, proporcional ao tamanho das cubas e inércia do eixo. Volumes de chuva menores que este mínimo podem induzir a uma margem de erros. Este tipo de pluviômetro, além de menor margem de erros, apresenta maior precisão.

O **pluviógrafo** é um tipo especial de pluviômetro, que permite o registro gráfico do volume de chuvas. Neste sistema existe um pluviômetro semelhante ao tipo caçamba, que contém uma bóia e um sistema de vazão. A bóia é associada a uma espécie de caneta, assim, à medida que a bóia se movimenta, em função do aumento do volume de água na caçamba, a caneta também se movimenta e registra a movimentação em um cilindro giratório coberto por um papel. O pluviógrafo é um instrumento totalmente mecânico e que trabalha exposto a diferentes variações ambientais, sendo, portanto, mais sujeito a falhas, além de apresentar uma difícil manutenção.

4.2.3 Interfaceamento entre o kernel e o pluviômetro

Para cada tipo de pluviômetro, poderíamos desenvolver uma interface especial que fornecesse um sinal elétrico em função de determinado volume de chuvas. Um método é o uso de pluviômetros do tipo basculante, onde um ímã é colocado no eixo das cubas. Quando ocorre uma alternância na posição das cubas, o ímã passa em frente a um relé magnético, fechando momentaneamente seu contato e fornecendo o sinal elétrico que indicará a ocorrência de determinado índice pluviométrico. Este tipo de sistema é fabricado no Brasil e recebe o nome de "sensor para telepluviometria". Outro modelo disponível é um pluviômetro basculante contendo um tubo com mercúrio, paralelo ao centro do seu eixo. Neste caso, quando ocorre uma alternância, o mercúrio se movimenta dentro do tubo e faz o contato para fornecimento de um sinal elétrico.

O pluviômetro adotado para estudos foi o modelo "SPP-1", da empresa Hidrologia SA, que possui uma precisão de leitura de 0,1 mm de chuva, com faixa de medição de até 4,2 mm/minuto. Para cada 0,1 mm de chuva ocorre o fechamento momentâneo do relé magnético, que permite o envio de um sinal elétrico ao kernel de aquisição de dados.

O sinal oriundo do pluviômetro aciona uma rotina de interrupção externa do microcontrolador. Esta rotina de interrupção registra na memória de dados o horário em que foi acionada, assim, cada registro da memória representa o horário de ocorrência de 0,1 mm de chuva.

A rotina de tratamento da interrupção externa tem um tempo de execução máximo de 3,5 ms. Este tempo é variável, porque a operação de gravação na memória de dados, que é do tipo flash, pode exigir mais de uma tentativa até a obtenção de sucesso. Isto permite que o kernel realize até 285 registros por segundo.

O pluviômetro utilizado no sistema tem precisão de 0,1 mm de chuvas e medição de até 4,2 mm/minuto, totalizando no máximo 42 registros por minuto. Considerando-se o pior caso com um tempo gasto na gravação de um registro, o kernel seria capaz de atender até 285 eventos por segundo ou 17.142 eventos por

minuto, que é uma taxa muito maior que os 42 registros por minuto, que o pluviômetro SPP-1 pode exigir.

4.2.4 Funcionamento do sistema

O temporizador do kernel trabalha com precisão de segundos. Ao ligar o kernel, o operador faz a inicialização do mesmo, digitando a data e horário da inicialização, sendo que estas informações são transferidas para a memória de dados. O temporizador é automaticamente ligado e inicia uma contagem de segundos, a partir do zero. Desta forma, cada valor armazenado na memória representa o número de segundos em que ocorreu a interrupção após a inicialização.

Supondo que na memória exista um registro com valor 15.840. Isto significa que ocorreu um evento 15.840 segundos após a inicialização do sistema. Sabendo-se o horário e a data de inicialização, pode-se fazer diretamente a conversão de um registro, para se determinar o instante da ocorrência do volume de chuvas.

Caso o sistema opere em condições que possibilitem a ocorrência de falhas momentâneas na alimentação, poderemos dotá-lo de um relógio externo de referência, alimentado com bateria própria, ou mesmo de um controle por software que permitisse uma reinicialização automática, sem interferência do operador.

A estrutura de dados utilizada neste sistema, representada na figura 4.1, armazena inicialmente a data e o horário de inicialização do sistema, usando 6 bytes para armazenar estas duas informações. Em seguida são usados registros de 3 bytes (24 bits) para representar a contagem de segundos. Assim, cada registro pode armazenar a contagem de até 2^{24} , ou 16.777.215 segundos. Considerando que cada período de 30 dias tem 2.592.000 segundos, um registro de 3 bytes poderia armazenar um número de segundos relativos ao período de 6,47 meses ou 194 dias.

Fig. 4.1 Armazenamento de dados no sistema de pluviometria

Como o relógio interno usa um registro de 3 bytes para contagem de segundos, após o período de 194 dias, o registro sofreria um estouro de sua capacidade de contagem e seria automaticamente zerado, reiniciando a contagem e continuando a funcionar normalmente. Desta forma, se for encontrado, na memória de dados, um registro numericamente menor que o registro imediatamente anterior, é porque se passaram mais de 194 dias e o contador do relógio foi automaticamente zerado.

Usando uma memória de dados de 64 Kbytes no kernel, pode-se armazenar até 21.843 registros. Como cada registro representa 0,1 mm de chuva, no pluviômetro SPP-1, então poderíamos registrar a data e o horário da ocorrência de um volume de até 2.184 mm de chuva. O volume de chuvas depende da região e da época do ano, sendo que, no Brasil, este volume pode atingir taxas superiores a 2.500 mm por ano nas regiões de clima equatorial, como na amazônia, e também pode atingir médias inferiores a 800 mm nas regiões semi-áridas do nordeste.

As características do sistema de pluviometria são:

- 01 - 64 Kbytes de memória não volátil de dados (flash)
- 02 - Capacidade de pré-processamento

- 03 - Interação com o usuário, via teclado e visor de cristal líquido
- 04 - Registro de horários com precisão de segundos
- 05 - Armazenamento de até 21.843 registros
- 06 - Reconhecimento de até 285 registros por segundo
- 07 - Transmissão de dados com interface serial no padrão RS-232
- 08 - Precisão de 0,1 mm de chuva e faixa máxima de 4,2 mm/minuto (dependendo do pluviômetro e não do kernel)

4.3 O processo de experimentos com PCR

Uma segunda aplicação do kernel de aquisição de dados é no controle térmico para automatização de experimentos com PCR. A técnica de **PCR (Polimerase Chain Reaction)** permite a sintetização de milhões de cópias de um fragmento específico de DNA, a partir de moléculas iniciadoras, chamadas *primers*. O termo PCR significa uma "reação de polimerização em cadeia", ou seja, é uma reação que permite a combinação de várias moléculas que formam um composto mais complexo, que no caso é o DNA.

O DNA, em termos físicos, é uma molécula de grandes dimensões e de grande peso molecular. Por volta de 1870 foi descoberta a existência de uma substância contendo fósforo e nitrogênio, no núcleo das células. Esta substância foi chamada de ácido nucleico e, nos seres vivos, foram encontrados dois tipos diferentes de ácidos nucleicos que foram chamados de DNA (*Ácido Desoxirribonucleico*) e RNA (*Ácido Ribonucleico*) [75]. A molécula de DNA contém toda carga genética de um indivíduo, que é um grande número de informações [50]. Além disto, ela permite a *replicação*, que é a produção de cópias idênticas a si mesma [75, 76, 55].

A proposta de um modelo da estrutura da molécula de DNA, apresentando a forma de uma gigantesca dupla hélice, foi publicada por Watson e Crick em 1953 e revolucionou o estudo da genética [50, 63]. Neste modelo, a molécula de DNA é composta por dois filamentos de *polinucleotídeos* (associação de inúmeros *nucleotídeos*), dispostos helicoidalmente em torno de um mesmo eixo. Um nucleotídeo, por sua vez, é uma molécula orgânica formada pela reunião de outras

três moléculas mais simples, que são: um radical fosfato, uma pentose e uma base nitrogenada, conforme figura 4.2 [75].

Fig 4.2 Composição química do nucleotídeo

As *bases nitrogenadas*, existentes no nucleotídeo, são moléculas que apresentam uma estrutura em anel de carbono e nitrogênio. No nucleotídeo do DNA encontramos quatro tipos diferentes de bases nitrogenadas, que são: a *adenina* (*A*), a *guanina* (*G*), a *timina* (*T*) e a *citossina* (*C*). Cada base nitrogenada de um filamento da molécula de DNA se une a uma base nitrogenada do outro filamento através de uma *ponte de hidrogênio*, permitindo a formação da estrutura em dupla hélice. Esta ligação de bases forma pares definidos de bases nitrogenadas do tipo: adenina/timina e guanina/citossina (*A/T* e *G/C*) [75, 55].

Quando as ligações de hidrogênio da molécula de DNA se rompem, os dois filamentos se separam e se recompõem formando novas cadeias complementares ligadas pelos pares bases *A/T* e *G/C*, dando origem a duas novas moléculas de DNA, idênticas à original. A figura 4.3 exemplifica o rompimento e recomposição dos filamentos [75].

Esta replicação ocorre na presença de uma *enzima* especial denominada *DNA-polimerase*, que exige uma cadeia de DNA como matriz, além de outras matérias primas [75].

Fig. 4.3 O processo replicação do DNA

A enzima é uma proteína capaz de acelerar a velocidade de uma reação química, permitindo uma diminuição do nível de energia necessário para a realização da reação. Reações químicas que exigiriam altas temperaturas podem ocorrer na temperatura ambiente, quando na presença de enzimas apropriadas. A própria ação de uma enzima depende diretamente da temperatura, sendo que a ação duplica ou triplica a cada 10°C de incremento na temperatura, até atingir um ponto ótimo, a partir do qual diminui até a inativação, conforme figura 4.4 [75, 55].

Figura 4.4: Variação da Atividade Enzimática.

Bioquímicos e geneticistas sempre pesquisaram formas de sintetizar material genético em tubos de ensaio. Tal objetivo começou a ser vislumbrado quando se isolou e purificou a enzima DNA-polimerase [76]. A técnica de PCR, que permite a replicação do DNA em tubos de ensaio, foi lançada em 1985 e rapidamente se tornou uma importante ferramenta da biologia molecular e engenharia genética [73]. Esta técnica consiste em expor uma amostra de DNA a um ciclo repetitivo e controlado de temperaturas pré-determinadas, que permitem a reprodução da mesma, de forma exponencial.

A técnica de PCR também é útil em aplicações de diagnósticos clínicos [74], permitindo a detecção de anti-corpos e análise do comportamento reativo a antibióticos [73]. Outra aplicação da técnica de PCR é na área de medicina legal, através da chamada "impressão digital genética". Nesta técnica, vestígios biológicos do criminoso podem ser analisados para levantamento do seu genótipo, que poderá ser posteriormente comparado com o genótipo de um suspeito do crime. Esta aplicação foi usada pela primeira vez nos Estados Unidos, em 1986, e, desde então, foi usada centenas de vezes, com valor legal [72].

A técnica de PCR depende diretamente do controle térmico, que pode ser dividido em três passos. Estes passos, que foram um ciclo completo de reprodução exponencial, são constituídos pela divisão da amostra, a ligação dos primers e a síntese do DNA. Para a execução de cada um dos três passos do ciclo térmico a amostra deve ser submetida a uma temperatura específica por certo período de tempo, como mostra a figura 4.5.

Fig. 4.5 Ciclo térmico do processo de PCR

4.3.1 O instrumento de automação de PCR

Um instrumento para automatização de experimentos com PCR pode ser formado, conforme a figura 4.6, por quatro partes básicas, que são:

- 01 - Um suporte para colocar os tubos com DNA
- 02 - Um sensor de temperatura.
- 03 - Um módulo computacional de controle.
- 04 - Uma fonte de calor.

Um sensor de temperatura é colocado junto das amostras de DNA e envia, ao módulo computacional de controle, um sinal proporcional à temperatura das amostras. Este sinal é recebido e analisado pelo módulo que envia um sinal externo que controla a fonte de calor. Nos modelos comerciais a transmissão de calor, entre a fonte e as amostras de DNA, é feita através do fluxo de água, que exige bombas injetoras para este fim.

O kernel desenvolvido, quando configurado para este fim, pode sensoriar até oito pontos diferentes de temperatura, permitindo uma verificação da homogeneidade térmica em diversos pontos do suporte.

Fig. 4.6 Sistema para automação de PCR

O kernel programável também possui um teclado e um visor, possibilitando uma interatividade direta com o usuário. Na sua inicialização, o usuário digita o número de passos do ciclo, o número de ciclos e finalmente a temperatura e tempo de cada passo do ciclo. A partir daí, o kernel passa a monitorar a temperatura e controlar a fonte de calor conforme especificações do usuário.

As características do sistema, desenvolvido com o kernel, para sensoriamento e controle térmico na automação de experimentos com PCR, são:

- 01 - Sensoriamento térmico de até 8 pontos diferentes
- 02 - Interatividade com o usuário por teclado e visor de cristal líquido
- 03 - Medição de temperatura na faixa de 0 a 127°C
- 04 - Precisão de 0,5°C
- 05 - Até 99 repetições do ciclo térmico
- 06 - Até 99 passos por ciclo térmico
- 07 - Até 99 segundos por passo
- 08 - Monitoramento térmico no visor em tempo-real
- 09 - Interface de comunicação serial RS-232

4.3.2 Medição térmica

A temperatura é uma grandeza fundamental nos experimentos com PCR, uma vez que todo o processo está baseado em um ciclo térmico. Esta grandeza é tão importante e usada que, no Dicionário de Física [44], de Horácio Macedo, há 31 extensões diferentes para o conceito de temperatura, como temperatura Fermi, temperatura absoluta, temperatura de cor, etc.

O conceito de temperatura pode ser dado através do exemplo de dois corpos que são colocados em contato e ocorre um fluxo de calor do mais quente para o mais frio, até que ambos atinjam um equilíbrio térmico. A termodinâmica define formalmente o conceito de temperatura, como a propriedade que representa o estado de agitação das partículas, medindo a agitação média dos átomos e moléculas, e não

deve ser confundida com a quantidade de calor, que é a média da energia total e depende da massa do corpo [45, 71].

Um dos primeiros registros de medição da temperatura foi feito por Galileo, em 1592, porém, somente em 1726 é que Fahrenheit criou uma escala fácil de ser reproduzida, utilizando um termômetro construído a base de mercúrio. Em 1742, Anders Celsius propôs uma escala para medida de temperaturas que fosse feita em função da temperatura para o ponto de solidificação e evaporação da água, atribuindo-lhes os valores zero e cem, respectivamente, e que, por isto, foi chamada de centígrada. Esta escala se tornou muito popular e, em 1948, teve seu nome mudado para *Celsius*, em homenagem ao seu criador [46].

Atualmente são exigidos padrões cada vez mais precisos para a calibragem e medição de temperatura. Estes padrões são determinados nas Conferências Internacionais de Pesos e Medidas, cuja primeira reunião foi em 1889 e a décima oitava em 1990. Nestas conferências, foram criadas normas gerais de pesos e medidas, dentre os quais encontramos a ITS (*The International Temperature Scale*) que regulamenta padrões de medição da temperatura [46].

4.3.3 Controle térmico

No processo de automação de experimentos com PCR, além da medição da temperatura, deve existir um controle sobre uma fonte de calor. Um sistema de controle térmico pode, em muitos casos, ser complicado e exigir uma modelagem matemática, normalmente realizada por equações diferenciais [40, 80]. Depois de entendido o comportamento do sistema, pode-se determinar qual a técnica de controle mais viável para atuar sobre o mesmo.

Existem várias técnicas diferentes de controle, como a liga/desliga, a proporcional, a integral, a derivativa ou a PID, todas com características próprias de estabilização do controle [43, 80]. No controle liga/desliga ("*on/off*"), o sistema fica medindo a temperatura e ligando ou desligando a fonte de calor para manter a temperatura em torno de um ponto desejado.

Os demais sistemas, mencionados acima, fazem o controle através da corrente elétrica que circula pela fonte térmica. Quando estes controles são implementados com sistemas computacionais, normalmente existe um conversor digital/analógico que fornece uma referência para controle do valor da corrente.

O controle liga/desliga apresenta o inconveniente de manter a temperatura oscilando em torno de um determinado valor, sem uma estabilização fixa. Além disto, a técnica de ligar e desligar constantemente a fonte de calor, pode causar um maior desgaste físico no dispositivo de chaveamento do controle, principalmente quando o mesmo for eletromecânico, como no caso dos relês magnéticos. Apesar dos inconvenientes, este controle apresenta algumas vantagens, como a simplicidade de operação e facilidade de implementação, que normalmente podem permitir uma redução de custo.

No sistema de controle de experimentos com PCR, foi utilizado um controle liga/desliga, que apresentou satisfatórios resultados nos testes realizados e que serão descritos na próxima seção.

4.3.4 Sensoriamento e controle térmico em PCR

Para desenvolvimento do sistema de sensoriamento e controle térmico em automação de experimentos com PCR, foi desenvolvido um protótipo de uma fonte de calor, composto por um resistor de níquel-cromo, com resistência de 17 Ohms, que, quando submetido a uma tensão de 5 Volts, apresentou as variações de temperatura mostradas no gráfico da figura 4.7. Assim, esta tensão foi utilizada, pois apresentava variações de temperatura em tempo satisfatório para os testes.

Para que o kernel pudesse controlar a fonte de calor, foi desenvolvido um circuito de chaveamento, conforme figura 4.8. Neste caso, um pino de E/S do kernel envia um sinal de controle a um circuito eletrônico, que permite a passagem de corrente para a fonte de calor. Como o pino de E/S assume somente valores 0 ou 1, o controle será exclusivamente liga/desliga.

Figura 4.7: Variação da Temperatura (escala: 10seg/cm).

O sensor de temperatura utilizado foi um modelo LM35, que permite a manipulação de temperaturas até um valor de 150°C e fornece uma saída linear de 10mV/°C. Desta forma o mesmo foi inicialmente ligado de modo direto ao conversor A/D do kernel, que permite a manipulação de sinais na ordem mínima de 4mV. Inicialmente, os testes práticos mostraram um comportamento indesejável do conversor A/D para sinais próximos a este valor. Isto exigiu que fossem realizados alguns testes de confiabilidade do conversor, cujos resultados estão exemplificados na tabela 4.1.

Fig. 4.8 Controle da fonte de calor

Tabela 4.1 Medidas do A/D com precisão de 5mV

<i>SINAL DE ENTRADA (mV)</i>	<i>VALOR DO A/D</i>	<i>VALOR ESPERADO</i>
0	2	0
10	6	2
20	8	4
30	10	6
40	11	8
50	14	10
100	24	20
150	34	30
200	46	40
500	108	102
1000	214	204

Para testes do A/D, foi utilizada uma fonte estável e com precisão 10mV e o conversor foi programado para operar com uma amplitude de 5mV, medindo sinais na faixa de 0 a 1.27 V. Foram feitas várias medidas, diversas vezes e nas mesmas condições, sendo que os dados estão na tabela 4.1. Analisando-se esta tabela, pode se verificar que o conversor A/D apresentou valores diferentes dos esperados. Embora tenha havido um problema de exatidão, o conversor se mostrou preciso, pois repetiu os mesmos valores em várias medições.

Para se confirmar a exatidão do conversor A/D, foi feita a sua reprogramação, para operar com sinais de 19,6 mV e foram realizadas outras medições, cujos valores estão na tabela 4.2. Operando nesta faixa de valores, pode-se observar um comportamento exato e preciso do conversor. O valor 19,6mV (5V/255) é a maior precisão permitida para o A/D de 8 bits, existente no kernel

Tabela 4.2 medidas do A/D com precisão de 19,6 mV

<i>SINAL DA FONTE (mV)</i>	<i>VALOR DO A/D</i>	<i>VALOR ESPERADO</i>
0	0	0
20	1	1
100	5	5
200	10	10
500	25	25
1000	51	51
2000	102	102
3000	153	153
4000	204	204
5000	255	255

Como o kernel irá manipular sinais com 5mV, que representam 0.5 °C, torna-se necessário algum ajuste do A/D para que estes níveis de sinais sejam manipulados corretamente. A primeira possibilidade é programar o A/D para manipular sinais na ordem de 5mV e usar uma tabela de correção que corrigiria o valor retornado pelo conversor. Este processo tornaria a conversão mais lenta e teria alguns pontos impossíveis de serem corrigidos, pois representam uma zona-morta para o conversor. A segunda possibilidade é a programação do A/D para operar com sinais na ordem de 19.6 mV e usar um amplificador operacional, conforme figura 4.9, para aumentar o sinal oriundo do sensor com um ganho de aproximadamente 3,92. Isto daria a relação direta de um incremento do A/D por 0,5 °C.

A segunda possibilidade foi adotada pelo fato de não exigir tabela de correção, permitir exatidão e precisão do conversor e manipular sinais com maior amplitude e, portanto, menos sujeitos a ruídos.

Fig. 4.9 Amplificador de sinais

Outra vantagem deste processo é que o amplificador utilizado possui um ganho que é ajustado por um resistor variável de precisão, permitindo uma calibração do sensor, caso necessário. A calibração pode ser feita colocando-se o sistema em uma temperatura de 25°C e ajustando-se o ganho do amplificador até que o sinal entregue ao A/D seja de 980mV.

Tabela 4.3 Relação entre precisão e faixa de operação térmica

<i>PRECISÃO DA MEDIÇÃO TÉRMICA (°C)</i>	<i>FAIXA MÁXIMA DE OPERAÇÃO (°C)</i>
0,1	0 a 25,5
0,2	0 a 51
0,3	0 a 76,5
0,4	0 a 102
0,5	0 a 127,5
0,6	0 a 153
0,7	0 a 178,5
0,8	0 a 204
0,9	0 a 229,5
1,0	0 a 255

Como o A/D opera de modo preciso e exato, com sinais na ordem de 19,6mV, o amplificador operacional foi ajustado para um ganho de 3.92 vezes, dando uma relação de 19,6mV/0,5°C e permitindo medições em uma faixa de 0 a 127°C. Todavia, o ganho do amplificador pode ser ajustado para aumentar a sensibilidade do sistema, dando uma relação de, por exemplo, 20mV/0,1°C, o que causaria uma diminuição na faixa de operação. A tabela 4.3 mostra a relação entre sensibilidade e faixa de operação.

4.3.5 Testes do sistema desenvolvido

Depois de feitos todos os ajustes necessários para o monitoramento térmico do processo, foi implementado um controle do tipo liga/desliga. Inicialmente foi feito um teste para controlar a temperatura em 30°C. O resultado deste teste pode ser observado na figura 4.10, onde se observa que as oscilações causadas neste controle, tiveram uma amplitude máxima de 0.5°C, que é exatamente a precisão do conversor A/D.

Figura 4.10: Controle em 30°C (escala: 1seg/cm).

Para uma confirmação deste resultado, a amplitude das oscilações foi analisada em uma escala de 10 segundos por centímetro, conforme figura 4.11. Isto permite uma análise mais detalhada das variações de temperatura, onde pode-se confirmar que a amplitude das oscilações se mantém em 0,5 °C.

Figura 4.11: Controle em 30°C (escala: 10seg/cm).

Também foi feita uma análise a respeito do *overshoot*, que é um valor de pico máximo que a temperatura atinge antes de iniciar o processo de estabilização. Este problema normalmente existe quando usamos o controle liga/desliga. Para esta análise foi feito o levantamento do gráfico da figura 4.12, onde se pode notar que o valor extremo apresenta uma diferença de aproximadamente 0,7°C acima do valor médio das oscilações, que pode ser considerada uma resposta satisfatória do sistema.

Também foi realizado um teste para verificação do comportamento do sistema na realização de um ciclo térmico completo, exigido pelo experimento com PCR. O resultado deste teste pode ser analisado através do gráfico da figura 4.13.

Figura 4.12: Análise de *overshoot* (escala: 10seg/cm).

Figura 4.13: Ciclo Térmico de PCR (escala: 10seg/cm).

A amplitude das oscilações pode ser analisada na figura 4.14. Os valores medidos se enquadram dentro dos resultados esperados, que são oscilações em torno de 0.5°C , que é exatamente a precisão do conversor A/D.

Figura 4.14: Erro do Ciclo Térmico de PCR (escala: 10seg/cm).

Capítulo V

Conclusões e trabalho futuro

5.1 Conclusões

O objetivo inicial deste trabalho era o desenvolvimento de um kernel computacional que pudesse ser facilmente configurado para diferentes aplicações. Inicialmente foram implementadas aplicações na automação da coleta de dados em pluviômetros e no controle de experimentos com PCR.

Os resultados obtidos nas aplicações, que são completamente diferentes, comprovaram a versatilidade do kernel e permitem concluir que o objetivo inicial foi atingido.

O aplicativo para aquisição automática de dados em pluviômetros apresentou resultados bastante satisfatórios e será utilizado pelo Instituto de Pesquisas Meteorológicas (IPMET) da UNESP, localizado em Bauru-SP. A idéia inicial é o desenvolvimento de diversas unidades que serão espalhadas por vários pontos do estado, permitindo um levantamento do índice pluviométrico nestas regiões.

O aplicativo desenvolvido para controle de experimentos com PCR permitiu o estudo e desenvolvimento de um protótipo do princípio de sensoramento e controle térmico, usados neste processo, comprovando a viabilidade de se desenvolverem pesquisas com este tipo de automação.

5.2 Avaliação crítica

O desenvolvimento deste trabalho proporcionou um entendimento aprofundado de diversos tópicos relativos à instrumentação, aquisição de dados e do desenvolvimento de dispositivos computacionais com microcontroladores programáveis. Além dos estudos teóricos, houve o desenvolvimento de uma importante experiência prática, que permitiu o contato com sofisticados instrumentos de medição e um aprendizado de técnicas utilizadas na implementação e testes de protótipos em laboratório.

Um projeto computacional, que proponha um produto final de baixo custo, deve levar em consideração o número de componentes utilizados, uma vez que este fator afeta diretamente o custo, além do consumo de energia e dimensões físicas. Assim, o projetista deve levar em consideração a possibilidade de utilizar um microcontrolador programável, que permite o desenvolvimento de um dispositivo computacional com reduzido número de componentes auxiliares.

No projeto do kernel optamos pelo uso do microcontrolador programável 80N535, que é compatível com a linha MCS-51 e possui características que o destacam entre os microcontroladores de 8 bits, como o número de portas de entrada/saída ou o conversor analógico/digital. O kernel também foi o primeiro projeto, com este microcontrolador, desenvolvido no Laboratório de Instrumentação Eletrônica (LIE) do IFQSC. Desta forma, o projeto também serviu para testes do próprio microcontrolador.

Embora o controlador tenha diversos recursos, foram notados problemas relativos a determinadas faixas de conversão A/D. Também foi observada uma sensibilidade durante os testes efetuados em bancada de laboratório, tendo inclusive ocorrido danos em componentes.

Outro problema encontrado foi quanto ao suporte fornecido pelo representante oficial do fabricante do microcontrolador, uma vez que o mesmo trabalha com vendas em grandes quantidades, exigindo importações diretas ou compras de revendedores não oficiais.

Embora tenham surgido problemas de implementação, é conveniente ressaltar que todos foram solucionados através de software ou complementações de hardware, conforme proposta original.

5.3 Trabalho futuro

Além do monitoramento e controle térmico, já estudados, o instrumento completo para automação de experimentos com PCR necessita de um sistema para aquecimento das amostras de DNA. Atualmente este sistema é comercialmente feito com fluxo de água, controlados por bombas injetoras. Futuramente poderá ser estudado um sistema que use ar para transmissão de calor. Este novo sistema, envolveria estudos de transmissão de calor entre a fonte de calor e as amostras de DNA, e estudos de homogeneidade térmica, que garante a mesma temperatura em toda a superfície onde são colocados os recipientes com DNA. Este novo processo poderá permitir o desenvolvimento de um instrumento com custo menor e que poderia ser mais facilmente instalado em regiões com menor infraestrutura.

O aplicativo de pluviometria pode ter o seu software ampliado para um sistema que permita a formação de uma rede de telepluviometria. Isto exigiria que fossem feitos estudos a respeito de ligações com modems, técnicas de roteamento e segurança de dados, que iriam viabilizar a implementação dos níveis do padrão ISO/OSI. Além disto, o kernel pode ser acoplado a diversos outros sensores meteorológicos, permitindo a automatização de uma estação meteorológica completa, ou formação de uma rede de medição meteorológica.

Além disto, como foi comprovada a versatilidade do kernel, pode-se vislumbrar a implementação de diversos outros aplicativos, como na automação de edificações (*domótica*), veículos automotores ou controle de processos industriais.

Bibliografia

- [01] COOK, Rick. Embedded Systems in Control. Byte, v.16, n.6, p. 153-160, Junho 1991.
- [02] COLE, Bernard C.. The Embedded Processor Breaks Out of Its Niche. Eletronics, v.61, n.6, p. 61-63, Março, 1988.
- [03] COLE, Bernard C. Designer's Dream Machine. Electronics, v.60, n.5, Março, 1987.
- [04] COLE, Bernard C. Ferment in Microcontrollers - New Players Stir the Competition. Electronics, v.60, p. 5-62, Outubro, 1987.
- [05] KYTE, David. The 8031/8731 Microcontroller. Elektor Electronics, v.16, n.180, p. 36-37, Julho, 1990.
- [06] YOUNG, Simon. ASIC Microcontrollers. Elektor Electronics, v.15, n.170, p. 55-57, Setembro, 1989.
- [07] SOKOLOWSKI, Steve. 8051 Single Board Computer. Elektor Eletronics, v.18, n.204, p. 40-43, Outubro, 1992.
- [08] WEISS, Ray. 8 bit Microcontroller Evaluation Boards. EDN, v.37, n.2, Janeiro, 1992.
- [09] DAVID, Kyte. 8031 In Circuit Emulator. Elektor Electronics, v.17, n.185, Janeiro, 1991
- [10] FLETCHER, Peter. 32-Bit Embedded Processor is INMOSS Entry in an Increasingly Tough Market - Ready for a Fight. Electronics, v.64, n.5, p. 33-36, Maio, 1991.

- [11] WRIGHT, Jeff. An Introduction to Single Chip-Microcomputers/Controllers. Electronics - Maplin Magazine, n.41, p. 17-23, dezembro 1990 - Janeiro, 1991.
- [12] WILSON, Ron. Embedded Systems Manipulate Distributed Tasks. Computer Design, p. 49-61, Setembro 1987.
- [13] BAILLEUX, O. 8751 Emulator. Elektor Electronics, v.18, n.198, p. 53-57, Março, 1992.
- [14] KESKAR, p. Y.. Retrofitting a Low Cost Data Logger to Improve Boiler Performance For an Industrial cogenerations Plant. ISA, v.45, parte 1, p. 179-190, 1990.
- [15] PHILBIN, Brian. Event History in Process Control. ISA, v.45, parte 1, p. 25-30, 1990.
- [16] TOYONOR, Y.; YUHARA, H.; TAKAHASHI, H.; MICALLEF, C.. Advanced Data Logger With High-Speed Trend Recording. ISA, v.42, parte 3, p. 1675-1680, 1987.
- [17] LEAL, Aquilino R.. Microcontroladores: O Futuro em Nossas Mãos em uma Unica "pastilha". Anntena - Eletronica Popular, v.100, n.1, p. 31-32, 1990.
- [18] BERLIN, Michael A.. 960 Magahertz Radio as a Telemetry Transmission Medium. ISA, v.38, parte 1, p. 311-315, 1983.
- [19] DeWITT, Steven A. Evaluating Telemetry Alternatives in Water Distribution Systens. ISA, v.40, parte 1, p. 255-261, 1985.
- [20] ROSS, J.; GUILLEMETTE, R.. Smart Telemetry Transmitter and Recorder. ISA, v.38, parte 1, p. 1323-1335, 1983.
- [21] DENNING, Peter J.. Saving All the Bits. American Scientist, v.78, p. 402-405, Setembro, 1990.

- [22] HICKEY, J. Thomas. The Special Problems of Instrumentation Specifications. ISA, v.40, parte 1, p. 231-239, 1990.
- [23] ABELSON, Philip H.. Instrumentation and Computers. American Scientist, v.74, p. 182-192, Março, 1986.
- [24] RESNICK, Robert; HALLIDAY, David. Física 1. Tradução de Antônio Maximo Luz *et al.* 3ª edição, LTC, Rio de Janeiro, 1981.
- [25] SILVA JUNIOR, Vidal Pereira da. Microcontrolador 8051. 1ª edição, Erica, Sao Paulo, 1990.
- [26] GRODZIK, R. 8031 Single Board Computer. Elektor Electronics, v.17, n.185, p. 54-57, Janeiro 1991.
- [27] REELSEN, H.. 8032/8052 Single-Board Computer. Elektor Electronics, v.17, n.198, p. 17-23, Maio, 1991.
- [28] WALD, J.M.. 8098 Evaluations Board. Elektor Electronics, v.15, n.172, p. 20-25, Novembro, 1989.
- [29] BERRY, John. Programando em C++. Tradução de Mário M. Fecchio. 1ª edição, Campus, Rio de Janeiro, 1983.
- [30] DAHMKE, Mark. Sistemas Operacionais Para Microcomputadores. Tradução de Rui Soares Bigio. 1ª edição, Campus, Rio de Janeiro, 1983.
- [31] TOKHEIM, Roger L.. Introdução aos Microprocessadores. Tradução de Anatolio Laschuk. 1ª edição, MacGraw-Hill, Sao Paulo, 1985.
- [32] WILSON, Dave. Embedded 68030 Takes on Real-Time Video-Effects Processing. Computer Design, v.30, n.9, p. 54, Junho, 1991.

- [33] SHERER, T. Flash Eproms. Elektor Electronics, v.18, n.204, p. 28-31, Outubro, 1992.
- [34] ZALES, Saul. Flash Memory Outshines Rom and Eprom. E.E. Times, Maio, 1990.
- [35] WILLIAMS, Tom. New Tools Bring CASE Power to Embedded Systems. Computer Design. Computer Design, v.30, n.10, p. 92, Julho 1991.
- [36] ONDRIS-CRAWFORD, Renate; GRAWFORD, Gregory p. ; DOANE, J. William. The Phase of the Future. The Physics Teacher, v. 30, n.6, p. 332-339, Setembro, 1992.
- [37] HAYN, Carl H. Liquid Crystal Displays. The Physics Teacher, v. 19, n.4, p. 25-257, Abril, 1981.
- [38] NICASTRO, Anthony J. Demonstrations of Some Optical properties of Liquid Crystals. The Physics Teacher, v. 21, n.3, p. 18-182, Março, 1983.
- [39] TOMPKINS, Willis J.; WEBSTER, John G. Interfacing Sensors to The IBM PC, Prentice Hall, Englewood Cliffs, New Jersey, 1988.
- [40] RANGAN, C.S.; SARMA, G.R.; MANI, v.S.v. Instrumentation - Devices and Systems, 2 edição, McGraw-Hill, New Delhi, 1985.
- [41] CLARKSON, Douglas. All About Liquid Crystals. ETI. p. 25-30, Abril, 1993.
- [42] FRAIGI, Liliana B.; ANGEL, Patricia M. Introducion a la Domótica - Cap. 3, Tomo 1, VI EBAI, Embalse-Argentina, Julho, 1993.
- [43] NIGIM, K.A. PWM Temperature Controller, Elektor Electronics, p. 50-51, Fevereiro, 1992
- [44] MACEDO, Horácio. Dicionário de Física, Rio de Janeiro/RJ, Nova Fronteira, 1976

- [45] CALLEN, Herbert B. Thermodynamics and an Introduction to Thermostatistics, 2 edição, John Wiley & Sons, 1985.
- [46] LEEWIS W. The International Temperature Scale ITS-90. ISA, v. 46, parte 1, p. 31-42, 1991.
- [47] WEST, Bruce J.; SHLESINGER, Michael. The Noise in Natural Phenomena. American Scientist, v. 78, p. 40-45, 1990.
- [48] JENSEN, Roderick v. Classical Chaos. American Scientist, v.75, p. 168-181, 1987.
- [49] GLEICK, James. Caos - A Criação de Uma Nova Ciência. Campus, Rio de Janeiro, 1990.
- [50] WATSON, James D. Initial Implications of The Double Helix. Structure, v.1, n.1, Abril, 1993.
- [51] JORDAN, Larry E.; Churchill, Bruce. Communications and Networking for the IBM PC. Robert Brady Company - Prentice-Hall, Bowie-Maryland - USA, 1983.
- [52] AGUILAR, M.; Pascual, p. J.; Santisteban, A. Scanning Tunneling Microscope Automation - IBM Journal of Research Developer, v.30, n.5, p. 525-532, Setembro, 1986.
- [53] BINNING, G.; ROHRER, H. Scanning Tunneling Microscopy. IBM Journal of Research Developer, v.30, n.4, p. 355-368, Julho, 1986.
- [54] INTEL Corporation, Memory Products. order number 210830-010, USA, 1991.
- [55] SOARES, José Luis. Biologia Básica. v.1, Scipione, São Paulo, 1988.
- [56] PASKLEY, Richard; Stefan, K. L. Flash Memories: the Best of Two Worlds. IEEE Spectrum, Dezembro, 1989.

- [57] INTEL Corporation. 8-bit embedded controllers, order number 270645-003, USA, 1991.
- [58] DAVIS, S. William. Sistemas Operacionais - Uma visão sistemática. Tradução de Dalton Alencar. Campus, São Paulo, 1990.
- [59] MONTEIRO, Mário A. Introdução a Organização de Computadores. 1ª edição, Rio de Janeiro, LTC, 1992.
- [60] CAMPBELL, Joe. RS-232 Técnicas de Interface. Tradução de Niuton Braga, 1ª edição, EBRAS, 1986
- [61] JÚNIOR, José Zariq. Redes Locais - O Estudo de seus Elementos, 1ª edição, LTC, Rio de Janeiro, 1988.
- [62] CIARCIA, Steve. Electromagnetic Interference. Byte, p. 48, Janeiro, 1981.
- [63] HENDRICKSON, Wayne; BRANDEN, Carl-Ivar. Introduction. Structure, v.1, n.1, Abril 1993.
- [64] SMIT, Jaroslav, Rádio Propagação. 1ª edição, Erica, São Paulo, 1986.
- [65] PIÓRICHKINE, A. v.; RÓDINA, n. A. Física 1, Mir, Moscou, 1986.
- [66] MELEAR, Charles. Taking the Mystery out of Microcontroller selection. I&CS, v.64, p. 19-21, Agosto, 1991.
- [67] VAGLICA, John J.; GILMOUR, Peter S. How to Select a Microcontroller. IEEE Spectrum, v.27, p. 106-109, Novembro, 1990.
- [68] MOSLEY, J.D. 8 and 16 bit microcontrollers. EDN Magazine. v.34, n.20, p. 108-121. Setembro, 1988

[69] AMD (ADVANCED MICRO DEVICES), Microcontrollers - Data Book/Handbook, USA, 1988

[70] GILMOUR, Peter S. How to Select Tools for Nicrocontroller software. IEEE Spectrum. v.28, p. 37-39. Fevereiro, 1991.

[71] ISAACS, Alan; PITT, Valeric. FÍSICA. Tradução de Maria p. B. M. Charlier e René F. J. Charlier. Edições Melhoramentos - USP, São Paulo, 1979.

[72] ERLICH, Henry A.; GELFAND, David; SNINSKY, John J. Recent Advances in the Polymerase Chain Reaction. Science. v. 252, p. 1643-1651, 21 Junho 1991.

[73] SANO, Takeshi; SMITH, Cassandra L.; CANTOR, Charles R. Immuno-PCR: Very Sensitive Antigen Detction by Means of Specific Antibody-DNA Conjugates. Science. v. 258, p. 120-122. 2 Outubro 1992.

[74] SAIKI, Randall K.; *et al.* Enzymatic Amplification of B-Globin Genomic Sequences and Restriction Site Analysis for Diagnosis of Sickle Cell Anemia. Science. v. 230, Pag. 1350 - 1354, Dezembro, 1985.

[75] MARCONDES, Ayrton Cesar; LAMMOGLIA, Domingos A. Aulas de Biologia - Citologia e Embriologia. v.1, 2ª edição, Atual, São Paulo, 1981.

[76] DeBUSK, Gib A. Genética Molecular. Tradução de José T. Gurgel e João L. Azevedo. USP - Polígono - São Paulo, 1971.

[77] Modulo Consultoria e Informática. Linguagem C - Programação e Aplicações. Livros Técnicos e Científicos S.A., Rio de Janeiro, 1986.

[78] RYAN, Bob. Built For Speed. byte. v.17, n.2, p. 123-135, Fevereiro, 1992.

[79] BURGER, Robert M.; HOLTON William C. Reshaping The Microchip. byte, v. 17, n.2, p. 137-148, Fevereiro, 1992.

[80] MIYAGI, Paulo Eigi; BARRETO, Marcos P. R.; SILVA, José R. Domótica: Controle e Automação, Tomo II, VI EBAI, Embalse-Argentina, Julho, 1993.

[81] CHUTATAPE O.; TEO, C.Y. Design of a Flexible Distributed Data Acquisition Controller With The 80C51FA Microcontroller. Microprocessors and Microsystems, v. 15, p. 417-425, 1991.