

UM SISTEMA PROGRAMÁVEL DE AQUISIÇÃO DE
DADOS METEOROLÓGICOS

Itana Maria de Souza Gímenes



UNICAMP

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E CIÊNCIA DA COMPUTAÇÃO

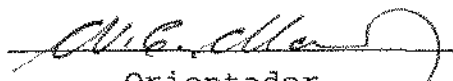
CAMPINAS - SÃO PAULO
BRASIL

UM SISTEMA PROGRAMÁVEL DE AQUISIÇÃO
DE DADOS METEOROLÓGICOS

Este exemplar corresponde a redação final da tese devidamente corrigida e defendida pela Sra. ITANA MARIA DE SOUZA GIMENES e aprovada pela Comissão Julgadora.

Campinas, *9* de *abril* de 1988.

Prof. Dr.


Orientador
Nelson Castro Machado

Dissertação apresentada ao Instituto de Matemática, Estatística e Ciência da Computação, UNICAMP, como requisito parcial para obtenção do Título de Mestre em Ciência da Computação.

**UM SISTEMA PROGRAMÁVEL DE AQUISIÇÃO
DE DADOS METEOROLÓGICOS**

Itana Maria de Souza Gimenes

Orientador : Nelson Castro Machado

Dezembro 1987

*Ao meu esposo Marcelino e
aos meus Pais e Irmãos*

AGRADECIMENTOS

Ao professor Nelson Castro Machado pela orientação deste trabalho e Hilton Silveira Pinto que forneceu as informações necessárias na parte de meteorologia.

Ao CNPq e CAPES pelo apoio financeiro.

Ao Departamento de Ciência da Computação da Unicamp, sede deste trabalho.

À Universidade Estadual de Maringá que possibilitou o desenvolvimento deste trabalho como pesquisa da instituição.

Durante o desenvolvimento de um trabalho como este é inestimável a contribuição que as demonstrações de amizade podem dar, uma palavra de incentivo, uma conversa amigável, a disponibilidade de tornar os caminhos mais simples, faz o difícil parecer mais fácil e fortalece a decisão de prosseguir. Gostaria, portanto, de agradecer a todos que deram a sua parcela de contribuição. Especialmente, gostaria de agradecer ao meu esposo Marcelino pela sua paciência, carinho e dedicação, a Neusa pela sua boa vontade em ajudar, a Paulo Centoducate, companheiro de trabalho, pela sua amizade e colaboração e a Sílvia Chibeni pela sua valiosa amizade.

RESUMO

O presente trabalho descreve o projeto e implementação do software de um sistema programável para aquisição de dados meteorológicos. O sistema consiste de uma estação base e uma estação de campo. Na estação base o meteorologista utiliza um microcomputador convencional e dispõe de uma linguagem para aquisição de dados meteorológicos (LADAM), através da qual pode programar seus experimentos. A interface da estação base com a estação de campo é feita através de um cartucho de memória RAM CMOS com um sistema de "back-up" de bateria, formando um sistema não volátil. O conteúdo inicial do cartucho é o código intermediário correspondente ao experimento programado. A estação de campo possui um interpretador que é capaz de executar os experimentos programados na estação base de acordo com o conteúdo do cartucho, gravando os dados coletados no próprio cartucho. Após encerrada a coleta de dados, o cartucho é novamente levado à estação base onde os dados são transferidos para arquivos em discos, a partir dos quais o usuário poderá analisá-los utilizando pacotes comerciais ou programas de aplicação em qualquer linguagem de alto nível.

ABSTRACT

This work describes the software design and implementation of a Meteorological Data Acquisition System. The system consists of a base station and a remote station. In the base station, the meteorologist uses a conventional microcomputer and is provided with a Meteorological Data Acquisition language (LADAM), which enables him to program the desired experiments. The interface between the base station and the remote station is accomplished with a RAM CMOS cartridge with battery back-up, which constitutes a non-volatile system. The initial cartridge contents is an intermediate code corresponding to the programmed experiment. The remote station has an interpreter which is able to execute the experiments programmed in the base station, according to the contents of the cartridge, recording the collected data in the same cartridge. When the data collection is completed, the cartridge is brought back to the base station, where the data is transferred to disk files. These serve as input to program packages or custom written programs in any high level language which performs the data analysis.

INDICE

Capítulo I - INTRODUÇÃO	001
Capítulo II - ALGUNS SISTEMAS EXISTENTES	006
2.1. Um Sistema Manual	007
2.2. Sistemas Automatizados	009
Capítulo III - O SISTEMA PROPOSTO	016
3.1. Descrição Sumária do Hardware	018
3.2. O Sistema Residente na Estação Base	021
3.2.1. A Linguagem para Programação da Aquisição de Dados	022
3.2.1.1. Estrutura de um Experimento Programado em LADAM	023
3.2.1.2. Objetos da Linguagem	025
3.2.1.2.1. Variáveis e Tipos	027
3.2.1.2.2. Tarefas e Eventos	029
3.2.1.3. Expressões	031
3.2.1.4. Comandos	034
3.2.1.4.1. Comandos Sequenciais	034
3.2.1.4.2. Comandos para Ativação de Tarefas	035
3.2.2. O Tradutor da LADAM	036
3.2.2.1. Código Gerado	036
3.2.2.1.1. As Tabelas de Controle	036
3.2.2.1.2. O Código das Tarefas	041
3.2.2.1.3. A Lista de Eventos	047
3.2.2.2. O Código de Detecção de Erros	049
3.2.2.3. Implementação do Tradutor	050
3.2.3. A Recepção dos Dados Adquiridos	052
3.3. O Sistema Residente na Estação de Campo	053
3.3.1. O Interpretador	054
3.3.1.1. As Rotinas de Atendimento ao Console	054
3.3.1.2. Núcleo Básico do Interpretador	059
3.3.1.2.1. Rotina de Inicialização do Experimento	059
3.3.1.2.2. Rotina de Controle do Experimento	064
3.3.1.2.3. Rotina para Ativação dos Eventos	066
3.3.1.2.4. Rotina para Execução das Tarefas	068
3.3.1.2.5. Rotina de Finalização do Experimento	070
3.3.1.3. A Implementação do Interpretador	072

Capítulo IV - A SIMULAÇÃO DO SISTEMA	076
4.1. O Ambiente Simulado	076
4.2. O Experimento Padrão Mínimo	078
4.3. O Experimento Padrão	081
4.4. O Experimento Micrometeorológico	085
4.5. Resultados Obtidos	090
 Capítulo V - CONCLUSÕES	 094
 BIBLIOGRAFIA	 099
 APENDICE A - Descrição da Sintaxe da LADAM	 101
A.1. - Elementos da LADAM	102
A.2. - Sintaxe da LADAM	102
 APENDICE B - Programas e Relatórios dos Experimentos Simulados	 107
B.1. Programação e Relatórios para o Experimento Mínimo	108
B.1.1. Programação do Experimento Mínimo em LADAM	108
B.1.2. Relatório do Código Intermediário para o Experimento Mínimo	110
B.1.3. Relatório dos Dados Adquiridos para o Experimento Mínimo	114
B.2. Programação e Relatórios para o Experimento Padrão	116
B.2.1. Programação do Experimento Padrão em LADAM	116
B.2.2. Relatório do Código Intermediário para o Experimento Padrão	119
B.2.3. Relatório dos Dados Adquiridos para o Experimento Padrão	128
B.3. Programação e Relatórios para o Experimento Micrometeorológico	131
B.3.1. Programação do Experimento Micrometeorológico em LADAM	131
B.3.2. Relatório do Código Intermediário para o Experimento Micrometeorológico	134
B.3.3. Relatório dos Dados Adquiridos para o Experimento Micrometeorológico	143
 APENDICE C - Arquivo de Sensores e Mensagens de Erro	 147
C.1. Arquivo de Sensores	148
C.2. Mensagens de Erro do Tradutor da LADAM	157

Capítulo I

INTRODUÇÃO

1. INTRODUÇÃO

O clima pode ser considerado como a maior fonte de recursos naturais existente na terra, portanto o seu estudo e conhecimento é extremamente relevante pois leva a uma melhor adequação de diversas atividades dependentes deste, uma vez que se pode dizer que é impossível alterá-lo.

Meteorologia é a ciência que se dedica ao estudo dos fenômenos atmosféricos [01]. Entende-se por aquisição de dados meteorológicos o processo de medida e armazenamento, através de aparelhos adequados, de uma série de parâmetros relativos à meteorologia, tais como: temperatura do ar, índice pluviométrico, radiação solar, velocidade e direção do vento, entre outros.

Estes aparelhos são instalados em estações meteorológicas localizadas nas mais diferentes regiões, formando as redes meteorológicas nacionais ou internacionais. Os países desenvolvidos publicam anualmente os relatórios de seus dados meteorológicos.

A importância da obtenção e armazenamento dos referidos dados é que estes servem de fonte para estudos climatológicos. A climatologia busca discutir o clima, seus componentes e os fatores que determinam e controlam sua distribuição [01], incluindo seu comportamento no passado, valores extremos e períodos de variações. Este conhecimento é indispensável no planejamento de muitas atividades da agricultura, indústria e engenharia.

Exemplificando [02]: Nas atividades comunitárias : O controle de poluição, planejamento do suprimento de água,

projeto e "layout" de cidades, projeto de pontes e estradas; Na indústria : Transporte e armazenagem de materiais, operação de plantas e dispersão da poluição; Nas atividades humanas: Respostas psicológicas, saúde, morbidez, aclimatação e especificação de roupas adequadas; Na agricultura: planejamento do uso da terra, controle de pestes, probabilidade de geadas, frequência de chuvas e necessidade de irrigação.

As primeiras estações meteorológicas, ainda hoje utilizadas devido a seu baixo custo, são aquelas nas quais uma pessoa, "o observador meteorológico", faz as leituras dos aparelhos em horários predeterminados e as anota em planilhas. Este tipo de sistema está sujeito a falhas humanas, além de tornar inviável uma alta frequência de leituras, desejável em certas aplicações.

Um primeiro passo foi dado com a introdução dos computadores digitais, que permitiram armazenar, analisar e apresentar de forma tabular ou gráfica as vastas quantidades de dados climáticos, estimulando assim uma maior utilização destes em problemas práticos como os já citados [02].

Com o desenvolvimento e a redução do custo dos microprocessadores foi possível a introdução de estações automáticas para a coleta de dados meteorológicos, chegando-se até modelos sofisticados nos quais os dados são coletados e transmitidos para uma estação base, onde são armazenados em um Banco de Dados.

Grande parte dos sistemas de custo médio utiliza programação prefixada e gravador com fita cassete para armazenamento dos dados. Esta solução, entretanto, não é adequada em microclimatologia, parte da meteorologia que estuda "detalhada-

mente os aspectos do clima relativos a uma área reduzida, como, por exemplo, na camada de ar junto ao solo [03]. Nestas aplicações são necessários experimentos diferentes dependendo da investigação desejada, utilizando diversos sensores cujas leituras devem obedecer a frequências variadas. Devido ao grande número de dados coletados torna-se inviável armazenar os valores da simples varredura de todos os sensores, sendo pois necessário o cálculo de somatórios, médias, máximos e mínimos, durante o experimento para compressão dos dados. Outra característica desejável seria a possibilidade de interação do meteorologista com a estação durante o experimento, tornando possível, por exemplo, a mudança de uma frequência ou verificação do estado dos sensores.

Visando atender estas necessidades foi projetado e implementado o protótipo do sistema descrito neste trabalho. Cuidou-se também da substituição do gravador cassete por um meio de armazenamento mais rápido e que consumisse menos energia.

O sistema desenvolvido (Sistema de Aquisição de Dados Meteorológicos: SADAM), oferece como ferramenta básica de software uma linguagem para programação dos experimentos de acordo com a necessidade do meteorologista. O sistema consiste de duas estações: a estação base, localizada no próprio ambiente de trabalho do meteorologista, onde os experimentos são programados e gravados em um cartucho de memória RAM CMOS com "back-up" de bateria formando um sistema não volátil; e a estação de campo, localizada na área onde se deseja obter os dados, que recebe a programação do experimento através do cartucho e comanda a coleta de dados, que são armazenados no próprio cartucho.

Este trabalho se refere ao software desenvolvido para a estação base e estação de campo. O hardware e as rotinas de serviço da estação de campo são descritos em [04]. A estação base utiliza um microcomputador convencional [05].

No capítulo 2 são apresentadas algumas estações automáticas semelhantes à desenvolvida. Este capítulo não pretende ser um relatório completo do instrumental para aquisição de dados meteorológicos, mas sim dar idéia de como o SADAM aparece no contexto destes sistemas.

O capítulo 3 descreve o software desenvolvido para as estações base e de campo.

O capítulo 4 apresenta a simulação do sistema como um todo através de três experimentos que utilizam vários recursos da linguagem desenvolvida.

O capítulo 5 contém as conclusões e sugestões finais.

Capítulo II

ALGUNS SISTEMAS EXISTENTES

2. ALGUNS SISTEMAS EXISTENTES

Descreve-se a seguir, sucintamente, alguns sistemas para aquisição de dados meteorológicos.

2.1. Um Sistema Manual

Neste item serão descritas as atividades da estação climatológica da Universidade Estadual de Maringá, com o objetivo de mostrar um sistema de aquisição de dados não automatizado.

A estação coleta dados para meteorologia e climatologia, além de atender às pesquisas da universidade coletando dados de experimentos específicos.

De posse destes dados, que atualmente são armazenados em arquivos tradicionais, a estação fornece informações para a comunidade universitária e comunidade externa de Maringá e região sendo, inclusive, muito alta a frequência das requisições destas informações.

As medições são feitas através de observadores meteorológicos que anotam os valores obtidos em planilhas.

As medidas relativas à meteorologia são feitas às 9:00, 15:00 e 21:00 horas. Os dados obtidos são transcritos num código internacional. Os parâmetros medidos são:

1. Pressão Atmosférica - Barômetro e Barógrafo.
2. Temperatura do Ar:
 - Sêco - Termômetro de bulbo sêco.
 - úmido - Termômetro de bulbo úmido.

Destes dados é obtida a umidade relativa do ar através da depressão psicrométrica (diferença entre as duas leituras).

3. Vento (direção e velocidade) - Anemômetro de placa que fornece a força do vento para posterior obtenção da velocidade através de tabela.
4. Precipitação - Pluviógrafo e Pluviômetro.
5. Observação direta de nuvens e visibilidade.

As medidas relativas à climatologia são as seguintes:

1. Os parâmetros já citados para meteorologia devem ser lidos a cada hora;
2. Evaporação:
 - Superfícies líquidas expostas - Tanque de evaporação;
 - Superfícies úmidas - Evaporímetro de Piché;
 - Leituras às 9:00, 15:00 e 21:00;
3. Temperatura do solo em condições de solo e profundidade variadas, que são:
 - Condições de solo: Desnudo e cobertura vegetal (viva e morta);
 - Profundidade: 2, 5, 10, e 30 cm da superfície;
 - Leituras às 9:00, 15:00 e 21:00 horas;
4. Temperatura mínima da relva no período noturno (21:00 às 7:00);
5. Insolação - Heliógrafo;
6. Radiação Solar - Piranógrafo;
7. Evapotranspiração (Quantifica o balanço hídrico no solo) - Evapotranspirômetro;

8. Orvalho (quantidade de orvalho formado) - orvalhógrafo;

2.2. Sistemas Automatizados

Inicialmente pode-se citar, como exemplo, um dos sistemas típicos que fazem a varredura dos sensores conforme memória pré-programada.

- O "CR21 micrologger" [06], contém um microprocessador e um circuito condicionador de sinais para 9 entradas. Sob um programa de controle fixo, o "micrologger" varre os sensores a cada minuto, manipula os dados obtidos de acordo com certos procedimentos e armazena os sumários das informações resultantes em intervalos selecionados pelo usuário.

O conjunto de procedimentos que podem ser selecionados pelo usuário está pré-programado e armazenado em ROM. Exemplos destes procedimentos são médias, máximas, mínimas, desvio padrão, amostragem simples e condicional e histogramas.

Este "Data logger" pode ser utilizado em redes conectado a um computador central. A recuperação dos dados pode ser feita via linha telefônica, gravador cassete, impressora e console. O sistema é alimentado por baterias do tipo pilha D com duração de até 5000 horas.

- O sistema de aquisição de dados pluviométricos apresentado por Haro [07], enfatiza o interesse em equipamentos específicos e de baixo custo para levantamento de dados meteorológicos aplicados à hidrologia.

O equipamento é composto de um sensor de precipitação, um sistema de aquisição de dados baseado em microprocessador e um programa decodificador dos dados coletados.

O sensor coleta a água proveniente da chuva e a converte em gotas de tamanho aproximadamente constante. As gotas são detectadas quando fecham um circuito elétrico. O sistema conta estas gotas armazenando as contagens em intervalos de tempo programáveis. Em seguida armazena os dados em um gravador cassete. As fitas são removidas e transportadas para uma estação base. Nesta o programa decodificador lê e verifica as informações adquiridas, transmitindo-as, em seguida, para um computador central.

O microcomputador tem 64 bytes de RAM, dos quais 40 estão disponíveis para dados e podem armazenar até 20 pares destes (horário, contador de gotas) e o restante está reservado para execução do programa.

Para viabilizar a alocação de um byte para horário e outro para contador, o horário foi codificado em blocos de 4 horas (240 minutos) e o contador de gotas é sempre dividido por dois. A cada informação armazenada na fita é acrescido um "start bit", um "stop bit" e um bit de paridade, que serão usados como controle no estágio de recuperação dos dados.

Haro menciona dois aspectos importantes que também, foram considerados no projeto do SADAM. Um deles é a necessidade de redução dos custos dos sistemas de aquisição de dados. O outro é o consumo relativamente alto de energia desses sistemas quando se utilizam circuitos do tipo então disponíveis (N-MOS). Sugere então que os circuitos sejam substituídos por suas versões CMOS.

- A "SYSTRON DONNER Weathermeasure" [08] apresenta um conjunto modular (série M733) de quatro sistemas de aquisição de dados meteorológicos de custo e complexidade crescentes. Os sistemas utilizam como unidade de saída gravador casete, impressora ou ambos.

Os dois primeiros, M733D e E, são as versões mais simples e procedem a aquisição dos dados através de três programas padrões: MET-1, AQM-1 e APS-1. A diferença entre os dois modelos é que o M733E possui painel para controle e o M733D é controlado via terminal.

O programa MET-1, programa meteorológico padrão, tem por objetivo fazer a varredura de múltiplos canais, processar, formatar e gravar os dados relativos a estes. Entre os parâmetros passíveis de escolha pelo usuário (via painel) estão a seleção da ordem de varredura, do número de canais, das unidades de medidas e dos tipos de cálculos, assim como o ajuste dos limites de alarme e calibração dos canais. O sistema executa automaticamente varredura dos sensores, calibração diária dos canais para corrigir distorções, formatação dos dados para saída e emissão de relatórios cuja frequência pode ser horária, a cada três horas, ou diária. Os cálculos disponíveis são: valor instantâneo, acumulado, médio, máximo e mínimo com os respectivos horários, tempos de duração e especificação de alarmes.

Os programas AQM-1 e APS-1 são similares, sendo que o primeiro se aplica à monitoração e controle da qualidade do ar, e o segundo aplica-se à aquisição dos parâmetros necessários para controle de aeroportos.

O modelo programável M733-F é o intermediário; a unidade possui um interpretador a partir do qual se podem construir programas de aplicação para coleta, formatação e gravação dos dados, assim como para o controle do processo.

A unidade pode operar com ou sem painel de controle, sendo que a última opção fornece a possibilidade de exibição dos dados e horários, um conjunto de funções para entrada de dados, controle do programa e sinais luminosos para indicação do estado e acompanhamento do programa. O sistema inclui um programa teste para facilitar a programação por parte do usuário.

O monitor possui um programa de recuperação para o caso de falta de energia, tornando possível a reativação da unidade através de uma memória RAM CMOS com "back-up" de bateria, que armazena os dados críticos.

A unidade M733-G utiliza um microcomputador completo sob controle do sistema operacional CP/M, onde as aplicações podem ser programadas em linguagens de alto nível, tais como Basic, Fortran, Pascal, etc.

A nível nacional podem ser citados dois projetos, desenvolvidos pelo Departamento de Meteorologia e Oceanografia do INPE.

- O primeiro deles [09] é um sistema automático portátil e de baixo custo para aquisição de dados meteorológicos com microcomputador. O objetivo deste é a obtenção de dados relativos à temperatura úmida, seca e pressão a bordo de pequenas aeronaves.

Os dados são coletados por um microcomputador de 16k (TK85) que também controla as taxas de amostragem e o armazenamento em um gravador cassete portátil.

O sistema é alimentado por baterias de 24 volts, permite 16 canais de entradas para um conversor A/D de 8 bits e seu software é programado em Basic e linguagem de montagem.

- O segundo, a estação meteorológica automática (EMA) [10], tem o objetivo de coletar e armazenar dados de uma estação agrometeorológica padrão.

O sistema completo compõe-se de: microcomputador da linha Apple-Compatível; monitor de vídeo; interface para controle de disco, interface serial protocolo RS-232C, unidades de disco de 5 1/4; a estação automática EMA; interface condicionadora de sensores e sistema "nobreak".

O protótipo desenvolvido tem capacidade de leitura para 16 sensores analógicos e 10 digitais; memória de 8 Kbytes, relógio digital com amostragem para dia, hora e minuto e um sistema ininterrupto de energia "nobreak".

Os dados são lidos de sensores condicionados ao padrão de entrada da estação. Este padrão consiste em considerar que os dados digitais devem estar no formato hexadecimal e os analógicos em tensão na faixa de zero a cinco volts.

As leituras são feitas em intervalos preestabelecidos e os dados são armazenados na própria memória da estação que tem capacidade para 8.192 dados. Posteriormente estes dados são transmitidos a microcomputadores da linha "Apple".

O software do sistema foi programado em Basic.

- Por último será citado um sistema interessante de aquisição de dados meteorológicos, MDAS [11], que consiste de uma rede local interconectada por fibras ópticas, que suporta até 16 estações remotas (torres).

O sistema foi projetado para atender à necessidade de se coletar os parâmetros micrometeorológicos necessários para a correlação da propagação de ondas com o índice de refração da atmosfera.

Um computador central, PDP-11/24, é responsável pelo controle da rede, armazenamento dos dados provenientes das estações remotas (microprocessadores intel 8031), interações com o usuário e seleção de programas executáveis. Para auxiliar no controle da rede existe uma interface inteligente entre o computador central e as torres.

O software do sistema utiliza seis das sete camadas do protocolo ISO-OSI (exceto o nível de transporte). O nível de aplicação é responsável pela obtenção e transferência dos arquivos objetos necessários para comandar a rede e coletar os dados necessários em tempo real. Este nível é o ponto de entrada dos programas que definem os experimentos a serem executados.

O nível de apresentação implementa as rotinas frequentemente chamadas pelo nível de aplicação, tais como diagnósticos das torres, verificação dos sensores, reconfiguração lógica das ligações, envio e execução de programas prédefinidos do usuário nas estações remotas e um comando para tomar o controle das torres.

O nível de sessão traduz os comandos vindos do nível de apresentação em comandos apropriados para o nível de rede local e mantém o computador central informado do estado do sistema. O nível de rede local é responsável pela decodificação dos comandos provenientes do nível de sessão em comandos que podem ser enviados para as torres e o nível de rede remoto é responsável pela execução dos comandos recebidos.

Merat [11] cita que com um software simples, rodando em cada torre, o sistema é capaz de adquirir dados de 128 canais (16 torres x 8 canais) e transmitir para o computador central a uma taxa de 100 Hz. Programas mais complexos podem reduzir esta taxa para 1 Hz.

Capítulo III

O SISTEMA PROPOSTO

3. O SISTEMA PROPOSTO

O sistema proposto consiste de dois equipamentos independentes que foram chamados de estação base e estação de campo e sua utilização se dá em três fases distintas. A primeira consiste na programação da coleta de dados e ocorre na estação base, localizada no próprio ambiente de trabalho do meteorologista.

O principal recurso desenvolvido para esta fase foi uma linguagem que se chamou de Linguagem de Aquisição de Dados Meteorológicos, abreviada por LADAM.

A LADAM contém um conjunto de primitivas indetificadas como necessárias para a programação de experimentos, em particular de experimentos meteorológicos. Cada experimento é composto de um conjunto de tarefas a serem executadas em determinados horários ou obedecendo a uma certa frequência, portanto foram definidos, além de comandos usuais nas linguagens de alto nível, comandos para ativação programada de tarefas.

A segunda fase é realizada no campo, na estação de Campo. Nesta fase a coleta de dados programada na estação base é efetivada.

A interface entre as duas estações é feita através de um cartucho de memória RAM CMOS com "back-up" de bateria, constituindo um sistema não volátil (os detalhes do hardware estão descritos em [4]). No cartucho é gravado o código intermediário correspondente ao experimento programado, gerado pelo tradutor da LADAM na estação base.

O software da estação de campo é constituído por um interpretador que executa os experimentos programados na esta-

ção base de acordo com o conteúdo do cartucho, gravando os dados coletados no próprio cartucho e por um monitor que é responsável pelas operações com os sensores, console, relógio de tempo real e ativação da estação.

A estação de campo permanece ativa apenas durante a execução de tarefas. Nos intervalos em que está a espera de uma ativação de tarefas ela é quase que totalmente desativada, sendo despertada pelo relógio na ocorrência de um evento, ou pelo usuário através de requisições do console. Isto permite aumentar consideravelmente a autonomia da estação de campo.

Após a coleta de dados ser completada o cartucho é novamente levado à estação base onde tem lugar a fase três, nesta fase os dados são copiados para um arquivo em disquete e um relatório de aceitação é gerado. A partir do arquivo em disquete o usuário pode proceder à análise dos dados coletados, utilizando pacotes estatísticos comerciais ou desenvolvendo programas de aplicação em qualquer linguagem de alto nível.

A figura 3.1 mostra o fluxo do sistema proposto.

3.1. Descrição Sumária do Hardware

O hardware da estação de campo [04] foi projetado para minimizar o consumo de energia possibilitando assim o uso de baterias de pequeno porte, pois a estação pode ser instalada em lugares onde não se dispõe de energia elétrica e os experimentos podem durar até um mês. Para tanto os componentes utilizados são, na medida do possível, em tecnologia CMOS e foram implementados dois modos de operação. No primeiro, Modo

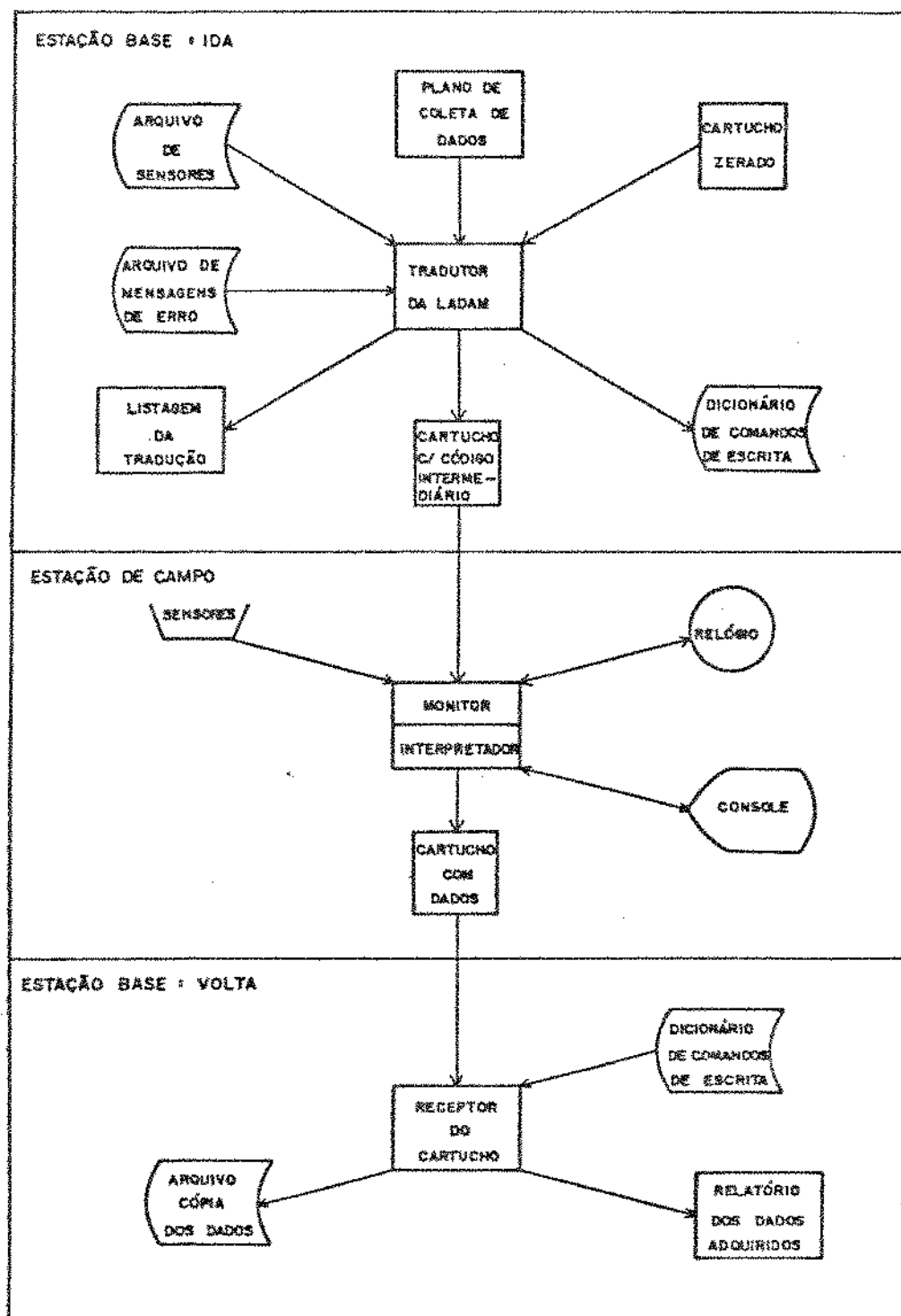


Fig. 3.1. - Fluxo do Sistema de Aquisição de Dados

normal, os componentes estão energizados; é utilizado quando o sistema está executando um ciclo de aquisição de dados ou interagindo com o operador. No segundo, Modo stand-by, a maioria dos componentes estão desenergizados; só permanecem energizados o relógio de tempo real, a CPU, e os componentes das interfaces dos sensores sem os quais perde-se-iam os dados (caso de alguns sensores digitais).

Obedecendo ao princípio de minimização de consumo foi projetado um cartucho com memória RAM CMOS associada a um sistema de "back-up" de bateria (pilhas comuns), tornando-o, no conjunto, uma memória não volátil. Outra opção para o cartucho seria a utilização de memória EPROM; esta foi descartada por apresentar o inconveniente de necessitar de uma tensão da ordem de +25 volts para a gravação e de ser muito mais lenta que a RAM, o que não apresenta problema para a presente aplicação, mas que poderia inviabilizar o sistema para aplicações em outras áreas, onde a velocidade de gravação é um requisito importante.

Os principais componentes da estação de campo são:

CPU - Microprocessador, NSC-800, de propósito geral. Este processador reúne características do 8085 (hardware simples, mesmo conjunto de interrupções, etc.) e do Z80 (mesmo conjunto de instruções).

Relógio de tempo real - MM58167 da National, entre outras facilidades este relógio possui dois tipos de interrupções programáveis, que podem atuar em uma frequência predeterminada e ou em horários fixos.

Memória RAM - 8 Kbytes utilizada para implementação da pilha do sistema e área de comunicação entre o monitor e interpretador.

EPROM - 24 Kbytes, onde deve residir o monitor e o interpretador.

Conversor A/D - Placa de conversão A/D da DICON com as seguintes características: barramento tipo STD, conversor A/D de 12 bits, tempo de conversão igual a 25 useg, 16 canais de entrada, amplificador instrumental com ganho programável por "jumper" de 1,10,100 ou 1000.

Teclado - 40 teclas implementando os caracteres alfanuméricos e alguns especiais, bem como funções pré-programadas.

Display - Alfanumérico mais caracteres especiais, de cristal líquido, com duas linhas de 16 caracteres cada.

Interfaces - São personalizadas, isto é, uma para cada tipo de sensor, compatibilizando os sinais dos sensores à entrada do conversor (no caso de sensores analógicos), ou ao microprocessador (no caso de sensores digitais). Porém todos possuem um sistema de identificação automática que é utilizado pelo monitor para verificar a configuração real do sistema.

3.2. O Sistema Residente na Estação Base

O sistema residente na estação base foi definido para ser instalado em microcomputadores Itautec I-7000, sob o sistema operacional SIM/M [05]. A utilização do I-7000 facilitou a comunicação com o cartucho que pode ser instalado neste, sem alteração no hardware do microcomputador. Outros equipa-

mentos (compatíveis com CP/M) também podem ser utilizados desde que seja adaptada uma interface para o cartucho.

O sistema é composto de um programa tradutor da LADAM e um programa receptor dos dados adquiridos na estação de campo.

O tradutor é responsável pela análise léxica, sintática e semântica de experimentos escritos em LADAM, gerando um código intermediário que é gravado no cartucho.

O receptor de dados, atua sobre o cartucho no retorno da estação de campo para gerar um arquivo em disquete contendo dados de forma mais acessível para que o usuário possa proceder à análise dos dados.

3.2.1. A Linguagem para Programação da Aquisição de Dados

A LADAM foi definida com o propósito específico de oferecer ao metereologista primitivas através das quais ele pode especificar o seu plano de coleta de dados de maneira simples e compacta.

No projeto da LADAM, devido à natureza da linguagem, procurou-se acompanhar as características básicas das linguagens de programação para tempo real, descritas por Young [13]. Estas características se referem à estruturação dos programas e parte sequencial. A parte concorrente não se aplica pois no sistema não ocorre ativação de tarefas concorrentes.

A LADAM é uma linguagem interpretada de maneira combinada [14]. Na primeira instância, o experimento, que é o programa fonte, é traduzido da sua forma original para um código intermediário, tentando-se tornar este código o mais fá-

cil possível de ser executado e livre de erros, restando apenas aqueles erros que podem ocorrer durante a execução. Nesta fase também são geradas estruturas de dados auxiliares, que tornarão o trabalho do interpretador mais simples. A segunda instância vai ocorrer na estação de campo, como será descrito em 3.3, onde o código intermediário é diretamente interpretado.

As primitivas da LADAM estão divididas em dois grupos. O primeiro grupo é formado pelos comandos para programação sequencial comumente utilizadas em linguagens de alto nível e adota o estilo léxico da linguagem Pascal. O segundo oferece comandos para ativação programada de tarefas, semelhantes às "scheduled clauses" da linguagem Pearl [15].

Inicialmente, foi implementado um subconjunto que satisfaz ao propósito, porém alguns recursos como tipos estruturados e procedimentos não foram implementados. Estes recursos podem ser adicionados à linguagem sem prejuízo de sua estrutura.

3.2.1.1. Estrutura de um Experimento Programado em LADAM

A estrutura de um experimento escrito em LADAM procura seguir uma ordem inicial de definição dos objetos principais da linguagem: tarefas e eventos, e está formada como segue:

- seção 1 : Declaração dos sensores
- seção 2 : Declaração das variáveis
- seção 3 : Declaração das tarefas
- seção 4 : Declaração dos eventos

A seção 1 começa com a palavra reservada `assign` e nesta devem ser definidas as relações entre os sensores e as portas lógicas.

Para cada sensor é declarado um código através do qual se tem acesso ao registro correspondente no arquivo de sensores (ver apêndice C.1), que contém as características básicas de todos os sensores que podem ser utilizados em um experimento. Este arquivo deve estar disponível para acesso no momento da tradução.

Cada sensor (dependendo do seu tipo) pode fornecer, como saída, vários valores tais como valor instantâneo, valor acumulado, identificação, etc. Portanto para cada um deles são declaradas as opções de leituras desejadas naquele experimento. Para receber os valores lidos são declaradas, nesta seção, variáveis que foram chamadas de variáveis externas. A cada opção de leitura de cada sensor corresponde uma variável externa.

As variáveis externas são implementadas através de cadeias de bytes cujas informações sobre tamanho e possíveis conversões aplicáveis, são obtidas do arquivo de sensores.

De posse destas informações o tradutor deve montar um descritor de sensores e um descritor do ambiente em que ocorrerá o experimento. Estes descritores serão estruturas auxiliares na interpretação e no confronto do ambiente instalado contra o programado.

A seção 2 começa com a palavra reservada `var` e nesta são declaradas as variáveis necessárias no decorrer da coleta de dados. Todas as variáveis declaradas são globais. Esta seção dá origem à área de dados.

A seção 3 é formada por um conjunto de definições de tarefas, cada uma começando com a palavra reservada `task`, seguida do nome da tarefa e encerrada com `endtk`. O corpo contém os comandos de programação sequencial associados às tarefas. Cada tarefa é uma unidade individual, cujos comandos são traduzidos para o código intermediário correspondente, que é gravado em cartucho. Os endereços são deixados em tabela para futuras referências na seção de declaração de eventos.

A seção 4 começa com as palavras reservadas `event section` e termina com `endevt`. Nesta seção são declarados os eventos associados ao experimento. Os eventos são descritos através de um rótulo opcional, estado inicial e de um comando para ativação programada de tarefas. A partir desta seção é gerada a lista de eventos que é utilizada como "time-table" pelo interpretador na estação de campo durante a execução de um experimento.

A figura 3.2. mostra um plano de coleta de dados escrito em LADAM. Exemplos mais completos, que tentam mostrar a potencialidade da linguagem são apresentados no capítulo 4 que descreve a simulação do sistema.

Os elementos e a sintaxe da LADAM na forma normal de Bakus são apresentados no apêndice A.

3.2.1.2. Objetos da Linguagem

Os objetos da LADAM são: variáveis, tarefas e eventos.

```

program coletex;
/* exemplo da programação de uma coleta de dados */
assign
  01S port 1 0:tempari;
/* opção de leitura 0 associada a variável externa tempari */
  02A port 2 0:radsoliq;
/* 02A é o código do tipo de sensor no arquivo de sensores */
var
  radsoliqt : real;
  horat     : time;
  datat     : date;
  tempmax   : integer;
/*-----*/
task header
  read(ck,datat,horat);
  write(ch,'inicio de experimento',datat,horat);
  tempmax := minvalue;
endtk;
/*-----*/
task t1
  read (sn,tempari);
  write(ch,tempari);
  if integer (tempari) > tempmax
  then
    tempmax:= integer(tempari);
    /* conversão da variável externa tempari para inteiro */
  endif;
endtk;
/*-----*/
task t2
  read(sn,radsoliq);
  radsoliqt := radsoliqt + real(radsoliq);
  /* conversão de variável externa radsoliq para real */
endtk;
/*-----*/
task t3
  write(ch,tempmax,radsoliqt);
  if dataref > 12/07/87 then trailler endif;
  /* trailler - pedido de encerramento do experimento */
endtk;
/*-----*/
event section
  ev1 : at 07:00,18:00 do t3 endo;
  every 01 hs do t1,t2 endo;
endevt;

```

Fig. 3.2. - Exemplo de uma coleta de Dados em LADAN

3.2.1.2.1. Variáveis e Tipos

As variáveis aparecem na seção de declaração das variáveis e são consideradas globais com duração até o final do experimento podendo, portanto, serem acessadas a partir de qualquer tarefa.

Para cada uma das variáveis é declarado um tipo, dentre os disponíveis na linguagem e, opcionalmente, um valor inicial.

Estão predefinidas as variáveis: maxvalue, minvalue (maior e menor inteiro possível), dataref e horaref (respectivamente do tipo date e time). As duas últimas são reservadas para armazenar a data e hora programada para ativação da tarefa corrente, e foram chamadas de data e hora de referência da tarefa. Com este recurso o usuário pode ter o controle do atraso entre o instante de ativação e a efetiva execução das tarefas, através da comparação da data e hora de referência com a data e hora corrente. Está predefinida também a variável memavail cujo valor num dado instante é o número de bytes ainda disponíveis na área de dados do cartucho.

Os tipos disponíveis na linguagem são:

INTEGER:

Valores inteiros no intervalo de -32768 a 32767. São alocados dois bytes na forma de complemento de dois.

REAL:

Os valores reais são armazenados em 4 bytes, com 23 bits de mantissa sem sinal alocados nos três primeiros bytes,

começando pelo byte de mais baixa ordem e no quarto byte está armazenado o expoente em potência de dois, conforme mostra a figura 3.3. O bit de mais alta ordem de mantissa é sempre 1 e não é armazenado. O bit de mais alta ordem do terceiro byte é o bit de sinal. O expoente é armazenado em notação "excesso de 128".

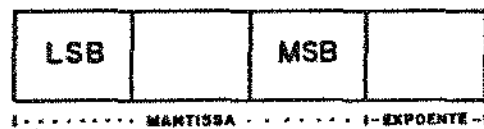


Fig. 3.3 - Representação de uma variável real

TIME e DATE :

Os valores são armazenados em BCD, para facilitar comparações foi utilizado o mesmo sistema do relógio de tempo real da estação de campo.

Time representa um horário composto de hora, minuto e segundo, ocupando 3 bytes.

Date representa uma data composta de dia, mês e ano, ocupando 3 bytes.

BYTSTRING

é um "string" de bytes cujo tamanho é definido na declaração da variável.

Além das variáveis já descritas, existem as variáveis externas. Diferentemente das demais variáveis, cujo espaço de armazenamento está na área de dados, as variáveis externas são armazenadas no descritor do ambiente, sob a forma de

bystrings com tamanho definido em função do tipo de sensor. A cada variável externa corresponde ainda um apontador para o tipo de sensor no descritor de sensores onde se encontram todas as informações sobre conversões aplicáveis a estas variáveis.

3.2.1.2.2. Tarefas e Eventos

As tarefas são objetos independentes definidos na seção de declaração de tarefas e referenciadas na seção de declaração dos eventos. Nesta seção os comandos de ativação programada de tarefas, determinam os instantes (eventos) nos quais as tarefas devem ser ativadas durante a execução de um experimento.

O conteúdo de uma tarefa é um conjunto de comandos sequenciais que, traduzidos para uma linguagem intermediária são executáveis pelo interpretador na estação de campo.

Existem duas tarefas especiais : "header" e "trailer" que podem ser declaradas ou não. A tarefa "header", quando declarada, é automaticamente executada pelo interpretador no início de cada experimento. Seu objetivo é fornecer ao usuário um meio de executar procedimentos particulares ao início de cada experimento sem ter a preocupação de declarar condicionais para detectar o início do experimento. A tarefa "trailer" é executada como parte dos procedimentos de encerramento programado de um experimento. "trailer" pode ser ativada mesmo sem ter sido declarada; neste caso o sistema provê um "default" que escreve no cartucho uma mensagem de término programado. A declaração destas tarefas não é obrigatória, mas

é no entanto recomendada sempre que for necessário tornar bem definidos os procedimentos iniciais e ou finais de um experimento.

Os eventos aparecem na seção de declaração de eventos. Um evento é considerado como uma ocorrência dependente do tempo e é ativado assincronamente na medida em que a hora corrente coincide com a hora programada. Foi utilizado um conceito de evento semelhante ao descrito para a linguagem Pearl [15].

Conforme descrito no apêndice A.2, a declaração de um evento consiste de um rótulo e estado inicial (opcionais) mais um comando de ativação programada de tarefas que referencia uma ou mais tarefas previamente declaradas.

O estado inicial de um evento pode ser ativo ou inativo; este estado pode ser modificado durante a execução de uma tarefa, geralmente quando uma condição programada é satisfeita. Este tipo de referência aos eventos é feita no corpo das tarefas através das instruções: `activate <nome do evento>` e `terminate <nome do evento>`.

A manipulação de estados dos eventos descrita acima provê o usuário com o recurso de mudar a sistemática de coleta de dados caso seja detectada alguma situação específica (tal como temperatura muito alta ou muito baixa, chuvas excessivas, etc) durante a execução do experimento.

Durante a tradução são construídos os seguintes atributos para cada evento:

- Horário
- Número do evento
- Tipo (Frequência ou Fixo)

- Estado de Ativação
- Número de tarefas ativadas pelo evento
- Endereço das tarefas ativadas pelo evento

Se evento baseado em frequência:

- Frequência
- Intervalos de ativação

Os eventos com seus respectivos atributos constituem as células da lista de eventos. Os eventos baseados em horário fixo podem gerar várias células com o mesmo número de evento. A lista de eventos (ver figura 3.4), ordenada cronologicamente, é passada para o interpretador através do cartucho. A estrutura interna da lista será descrita em 3.2.2.1.3. Note-se a ocorrência de dois eventos predeclarados: "Início" e "Término", associados automaticamente às tarefas "header" e "trailer" respectivamente.

3.2.1.3. Expressões

Uma expressão em LADAM é formada por uma sequência de operandos e operadores dispostos como descrito na sintaxe da linguagem (apêndice A.2). As expressões devem ser homogêneas e as conversões devem aparecer de maneira explícita, mesmo no caso de real para inteiro e vice-versa; isto previne erros de execução desde que nenhuma conversão "default" é adotada.

Na primeira fase, as expressões são analisadas e traduzidas para forma pós-fixada, o que facilita a obtenção do resultado final quando interpretada na estação de campo.

Os operadores disponíveis são:

ARITMÉTICOS:

+	soma	reais inteiros
-	subtração	reais inteiros
/	divisão	reais inteiros
*	multiplicação	reais inteiros

RELACIONAIS:

<	compara se menor	reais inteiros times dates
<=	compara se menor ou igual	reais inteiros times dates
>	compara se maior	reais inteiros times dates
>=	compara se maior ou igual	reais inteiros times dates
=	compara se igual	reais inteiros times dates
<>	compara se diferente	reais inteiros times dates

Os operandos podem ser variáveis, constantes ou chamadas de funções de conversão. Estas funções são aplicáveis a variáveis internas, constantes e a variáveis externas, para transformá-las no real ou inteiro correspondente e aparecem da seguinte forma:

Real | Integer (<variável>|<constante>)

As funções de conversão de variáveis internas e constantes para real ou inteiro estão automaticamente disponíveis no sistema residente na estação de campo. Além destas estão disponíveis também rotinas para conversão linear ($y = ax$

+ b) para real ou inteiro, a partir de variáveis externas, entretanto para utilizá-las é necessário que no registro do sensor correspondente estejam especificados a possibilidade de conversão e os parâmetros necessários: fator de conversão (a) e o deslocamento (b).

Para estender as possibilidades de conversão para além das acima descritas, o usuário pode especificar e escrever rotinas de conversão especiais, que devem estar disponíveis no momento da tradução. Durante a análise o tradutor detecta a necessidade de tais rotinas através de verificação no registro do sensor correspondente, e ao final da tradução requisita e copia tais rotinas em cartucho.

Para escrever as rotinas de conversão especiais o usuário pode utilizar os recursos de operações aritméticas já disponíveis no sistema residente na estação de campo. Isto é feito através da chamada de uma rotina de transferência residente, especificando o número da rotina desejada e os parâmetros necessários para a operação requisitada, de acordo com a tabela de rotinas do sistema. Durante a interpretação a rotina de transferência fará o desvio para a rotina desejada tendo no topo da pilha o endereço de retorno. Cabe ao usuário executar o retorno.

Vale observar que o cartucho deve ser poupado para armazenar dados, portanto essas rotinas especiais devem utilizar o mínimo de espaço, de preferência devem ser escritas em linguagem de montagem, procurando utilizar os recursos já disponíveis no "firmware" da estação.

3.2.1.4. Comandos

Os comandos da LADAM estão divididos em sequenciais e de ativação programada de tarefas.

3.2.1.4.1. Comandos Sequenciais

O comando `read` admite como dispositivo de entrada, `sn` (sensores), `ck` (relógio) e `cs` (console). No caso da não especificação o dispositivo assumido é o `sn`.

Para os sensores devem ser definidas variáveis externas para receberem os valores lidos. O relógio utiliza variáveis do tipo "date" e ou "time". A console aceita todos os tipos de dados disponíveis na LADAM.

Para o comando `write`, podem ser especificados os seguintes dispositivos de saída: `ch` (cartucho), que é assumido no caso de não especificação, e `cs` (console). Qualquer tipo de variável pode ser escrita nestes dispositivos.

Os comandos de atribuição e condicional funcionam de maneira semelhante aos comumente utilizados em linguagens de alto nível.

Os comandos de ativação (`activate`), e desativação (`terminate`), tornam disponíveis na linguagem recursos para alterar o "time-table" da estação de acordo com situações detectadas durante a execução de uma tarefa. Eventos declarados inativos podem ser ativados e quando desejado serem desativados, assim como eventos inicialmente ativos podem se tornar desnecessários durante a execução e serem desativados.

3.2.1.4.2. Comandos para Ativação Programada de Tarefas

Estes comandos concentram os principais recursos da LADAM e diferem fundamentalmente dos precedentes, sendo os únicos comandos que podem aparecer nas declarações de eventos e não podem ser utilizados nas declarações de tarefas. O código gerado por estes comandos é a lista de eventos ordenada cronologicamente, incorporando os atributos necessários para execução da coleta de dados na estação de campo.

Utilizaram-se dois comandos distintos (*every* e *at*) para diferenciar coletas baseadas em frequência de coletas em horários fixados. Poderia ser adotado apenas um comando como em Pearl [15], que englobaria toda a definição, porém ficaria extenso e mais difícil de ser absorvido pelo usuário.

No primeiro, comando *every*, o usuário define a frequência de coleta, que pode ser dada em hora, minuto, ou segundo e, se desejar, uma sequência de intervalos nos quais a aquisição deve ser efetuada. No caso de não especificação dos intervalos, a coleta é contínua durante todo experimento. A cada ativação de um evento deste tipo, é somado ao horário anterior o período, sendo obtido o novo horário de ativação do evento.

O segundo, comando *at*, é adequado para coletas que devem ser feitas em horários fixados, cuja frequência não é uniforme.

A opção de se ativar várias tarefas nestes comandos, torna modular a declaração das tarefas, dando a possibilidade de uma tarefa ser utilizada por mais de um evento.

3.2.2. O Tradutor da LADAM

3.2.2.1. Código Gerado

O código intermediário é gerado pelo tradutor da LADAM e armazenado em cartucho. Foi projetado para conter todas as informações necessárias para que o interpretador da estação de campo possa comandar um experimento, permitindo também que o trabalho do interpretador seja facilitado ao máximo. Além disso, o código intermediário deve ocupar o mínimo de espaço em cartucho, de modo a não prejudicar o espaço para armazenar dados, que é sua função principal.

Para atender a tais propósitos o código intermediário foi dividido em três partes: a primeira sendo um conjunto de tabelas de controle, a segunda corresponde ao código das tarefas a serem executadas e a terceira é a lista de eventos.

3.2.2.1.1. As Tabelas de Controle

As tabelas de controle que compõe a primeira parte do código intermediário são as seguintes:

1. TABELA DE ENDEREÇOS BASE
2. DESCRITOR DOS SENSORES
3. DESCRITOR DO AMBIENTE DO EXPERIMENTO
4. AREA DE DADOS
5. TABELA DE ROTINAS DE CONVERSÕES ESPECIAIS

TABELA DE ENDEREÇOS BASE

A tabela de endereços base serve como ponto inicial para acesso às demais tabelas. É composta de pares contendo o endereço e tamanho de cada tabela citada acima, do código das tarefas, da lista de eventos e mais dois pares para controle do espaço de dados. O primeiro destes indica onde começa o espaço livre para dados, e qual o tamanho do mesmo. O segundo indica o endereço final do espaço ocupado com dados após a coleta e o respectivo tamanho. Esta entrada estará inicialmente nula, sendo preenchida pelo interpretador ao final da coleta de dados e será um parâmetro importante na recepção dos dados coletados no retorno à estação base. Os campos que compõe a tabela de endereços base podem ser resumidos como se segue:

Informação	nº de bytes
I. Endereço e tamanho do descritor de sensores	4
II. Endereço e tamanho do descritor do ambiente	4
III. Endereço e tamanho da área de dados	4
IV. Endereço e tamanho do código das tarefas	4
V. Endereço e tamanho da tabela de rotinas de conversões especiais	4
VI. Endereço e tamanho da lista de eventos	4
VII. Endereço e tamanho da primeira posição livre para dados e tamanho da área livre	4
VIII. Endereço final do espaço ocupado com dados coletados e tamanho da área ocupada	4

Total de bytes ocupados pela tabela	32 bytes

DESCRITOR DOS SENSORES

O descritor dos sensores é formado a partir de informações lidas do arquivo de sensores, conforme descrito na seção 3.2.1.1. Os campos que compõe cada entrada do descritor dos sensores estão dispostos como se segue:

Informação	nr de bytes [min,max]
I. Código do sensor(número,tipo)	2
II. Tipo do sinal	1
III. Medida Máxima Permitida	4
IV. Medida Mínima Permitida	4
V. Precisão	4

VI. Registro de Opções de leitura	[4,50]
VII. Número da Opção	1
VIII. Endereço da próxima de Opção	1

IX. Registro de conversão para inteiro	[1,24]
IX.1 Conversão para inteiro	1
IX.2 Número da rotina de conversão	1
IX.3 Número de parâmetros	1
IX.4 Tamanho dos parâmetros	1
IX.5 Valores dos parâmetros	[1,20]
X. Registro de conversão para real	[1,24]
mesmos campos definidos para inteiro	

[1..5] registros possíveis	[4,250]
Total de bytes Ocupados por sensor	[19,265]

DESCRITOR DO AMBIENTE DO EXPERIMENTO

O descritor do ambiente também é construído durante a tradução da seção de declaração de sensores e tem o objetivo de servir como base de informações sobre o ambiente do experimento. Este descritor foi organizado por variável externa e armazena em cada uma de suas entradas o nome da variável externa, a porta lógica à qual está associada, o deslocamento em relação ao descritor de sensores (indicando o sensor associado), a opção de leitura desejada e o comprimento da variável externa em bytes, reservando também o espaço correspondente para o próprio valor da variável.

Além destes dados, que são preenchidos na base, está reservado um byte para que seja armazenado na estação de campo, o número do sensor que efetivamente foi utilizado, de modo que fique registrado com qual versão do sensor a coleta de dados foi executada. Os campos que compõe cada entrada do descritor do ambiente estão dispostos como se segue:

Informação	nº de bytes
I. Nome da variável externa associada	10
II. Deslocamento do sensor associado em relação ao descritor de sensores	2
III. Número do sensor utilizado	1
IV. Número da porta lógica associada	1
V. Opção de leitura	1
VI. Comprimento da variável	1
VII. Valor	x
Total de bytes ocupados por variável	16+x

AREA DE DADOS

Nesta área está reservado espaço para as variáveis declaradas na seção de declaração de variáveis. Para cada uma delas está especificado o tipo e é reservado o espaço necessário. O valor inicial das variáveis armazenado em cartucho é aquele especificado pelo usuário, durante a declaração, ou nulo se não foi especificado valor inicial.

As referências a esta área são feitas através de um deslocamento em relação a sua base, que está definida na tabela de endereços base. O conteúdo de cada entrada da área de dados é o seguinte:

Informação	ns de bytes
I. Tipo da variável	1
0 - inteiro	
1 - time	
2 - real	
3 - bytstring	
4 - date	
[II]. Tamanho se bytstring	1
III. Valor da variável	[2,x]
Total de bytes ocupados por variável	[3,2+x]

TABELAS DAS ROTINAS DE CONVERSÕES ESPECIAIS

Conforme descrito em 3.2.1.3. o usuário pode especificar rotinas de conversão especiais, caso as armazenadas no "firmware" da estação não sejam suficientes para o processamento desejado.

Estas rotinas são copiadas para o cartucho após a tradução do plano de coleta de dados e para cada uma das rotinas transferidas é deixado nesta tabela o endereço em que foi armazenada e o tamanho da rotina.

Quando é detectada a necessidade de uma conversão especial, durante a execução de um experimento, a estação passa a executar o código armazenado em cartucho buscando o endereço desta na tabela aqui descrita.

O conteúdo de cada entrada na tabela de rotinas de conversão especiais pode ser resumido como segue:

Informação	nº de bytes
I. número de rotina	1
II. endereço	2
Total	3

3.2.2.1.2. O Código das Tarefas

Após traduzidas, as tarefas continuam a ser unidades individuais cujas instruções obedecem a uma linguagem intermediária, que pode ser executada pelo interpretador da estação de campo.

Caracteres de controle foram incluídos na linguagem intermediária para auxiliar na estruturação das primitivas. Estes caracteres ocupam um byte cada e são os seguintes:

- CTL 0 (0) - Fim de instrução
- CTL A (1) - Fim de expressão
- CTL B (2) - Início de Tarefa
- CTL C (3) - Fim de Tarefa
- CTL D (4) - Deslocamento de variável interna
- CTL E (5) - Deslocamento de variável externa
- CTL F (6) - Chamada de função de conversão
- CTL G (7) - Constante

As tarefas foram estruturadas como segue:

CTL B end.fim.....instruções.....CTL C

Note-se que, aos caracteres de controle B, C, D, E, F e G seguem-se parâmetros ocupando dois ou mais bytes.

As primitivas da linguagem intermediária são as seguintes:

INSTRUÇÃO DE LEITURA

Objetivo: Leitura de variáveis dado o dispositivo de entrada.

Código: RDEEdid2.....dnCTL0 n > 0

Onde:

EE = Dispositivo de entrada (en, cs, ck);

d_i = Deslocamento da i ésima variável em relação à área de dados (se cs ou ck) ou do descritor do ambiente (se sn);

Tamanho(bytes) = $5 + 2n$ (número de variáveis)

INSTRUÇÃO DE ESCRITA

Objetivo: Escrita de variáveis no dispositivo de saída especificado

Código: $WRmmSSD1/c_1...D_i/c_i...D_n/ckCTL@$ $n, k > 0$

Onde:

mm = Número do comando de escrita;

SS = Dispositivo de saída (ch, cs);

D_i = Deslocamento da i ésima variável;

c_i = Uma cadeia de caracteres delimitada por (e);

Tamanho(bytes) = $7 + 3n + \text{somatório}(1, k)$ dos tamanhos das cadeias

Cada D_i corresponde a um byte referente ao código do deslocamento (CTL D ou E) mais dois bytes referentes ao valor do deslocamento. O número do comando de escrita, atribuído pelo tradutor é usado para indicar no cartucho o comando responsável pelo aparecimento da mensagem e permite a interpretação dos dados na fase 3 juntamente com o dicionário de comandos de escrita. Os números de comandos de escrita 1 e 2 são reservados para o sistema e corresponde aos registros inicial e final automaticamente escritos no cartucho. Aos comandos "write" que ocorrem na programação são atribuídos números sequenciais a partir de 3.

INSTRUÇÃO DE ATRIBUIÇÃO

Objetivos: Calcular o valor da expressão e atribuir o resultado à variável destino.

Código: $ABdtexpressãoCTL\emptyset$

Onde:

dt = Deslocamento da variável destino em relação à área de dados.

Tamanho(bytes) = 5 + Tamanho da expressão

INSTRUÇÃO CONDICIONAL

Objetivo: Executar determinadas instruções de acordo com o valor de uma expressão.

Código: $IFexpressãoDFeie1\dots$ código do then... $DSe2e2\dots$ código do else...CTL \emptyset

Onde:

$DFeie1$ = Desvie se falso para o endereço $eie1$

$DSe2e2$ = Desvie sempre para $e2e2$

Tamanho(bytes) = 7 + 4(se IF completo) + tamanho da expressão + tamanho dos códigos

INSTRUÇÃO PARA ATIVAÇÃO DE EVENTOS

Objetivo: Ativar um evento que se encontra desativado na lista de eventos.

Código: ATnumevtCTL@

Onde:

numevt = número do evento a ser ativado

Tamanho(bytes) = 4

INSTRUÇÃO PARA DESATIVAR EVENTOS

Objetivo: Desativar um evento que se encontra ativado na lista de eventos.

Código: DTnumevtCTL@

Onde:

numevt = número do evento a ser desativado

Tamanho(bytes) = 4

INSTRUÇÃO PARA ENCERRAR EXPERIMENTO

Objetivo: Encerrar um experimento em execução.

Código: EDCTL@

Tamanho(bytes) = 3

As expressões na linguagem intermediária estão traduzidas para a forma pós-fixa. Para indicar o fim de uma expressão foi utilizado CTL A, e os elementos que fazem parte das expressões foram codificados como segue:

1) CONSTANTES

Código: CTLG, tipo, valor da constante

Onde:

Tipo = nº do tipo correspondente;

As constantes que podem aparecer são inteiras, reais, time, date e bytstring e estão codificadas como segue:

Constante	Código	Total (bytes)
Inteira	CTLG 0 msb lsb	4
Real	CTLG 2 lsb..msb exp	6
Time	CTLG 1 hh mm ss	5
Date	CTLG 4 dd mm aa	5
Bytstring	CTLG 3 n x...x	3 + n

2) VARIÁVEIS

Na linguagem intermediária as variáveis são denotadas por seu deslocamento. No caso de variáveis externas, estes deslocamentos são em relação ao descritor do ambiente e no caso de variáveis internas em relação à área de dados.

As variáveis são representadas da seguinte forma:

CRTLD deslocamento - variável interna

CRTLE deslocamento - variável externa

Cada uma delas ocupa 3 bytes.

3) FUNÇÕES DE CONVERSÃO

As funções de conversão aparecem na linguagem intermediária da seguinte forma: CTLF tipo constante/variável; o total de bytes é $2 + x$, onde x é o número de bytes necessários para representar a constante ou variável.

3.2.2.1.3. A Lista de Eventos

A lista de eventos, construída durante a tradução da seção de declaração de eventos, é o guia do interpretador da estação de campo, pois é baseado nela que este vai programar o relógio de tempo real para despertar nos instantes de ativação das tarefas.

A lista de eventos está estruturada como uma lista ligada circular, organizada sequencialmente de acordo com a ordem cronológica dos horários das primeiras ocorrências dos eventos. Cada célula desta lista é formada pelo conjunto de atributos dos eventos mais os parâmetros para controle da lista.

Dois eventos especiais fazem parte da lista, o evento "Início" e o evento "Término", responsáveis pela ativação das tarefas "header" e "trailer", já descritas na seção 3.2.1.2.2. Estes eventos estão alocados fisicamente no início da lista e estarão ativos se as tarefas correspondentes forem declaradas. Assim podemos dizer que são quatro os tipos de células que fazem parte da lista de eventos:

- Célula de evento "Início"

- Célula de evento "Término"
- Célula de evento baseado em frequência
- Célula de evento baseado em horário fixo

A figura 3.4 mostra resumidamente a estrutura da lista de evento:

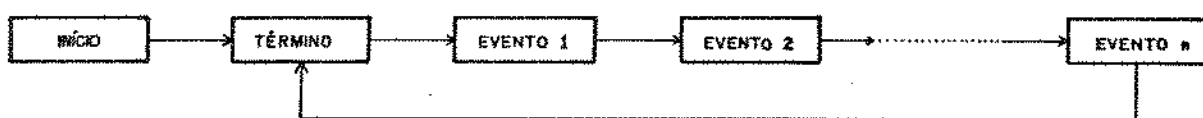


Fig 3.4 - Estrutura da lista de eventos

Um dos parâmetros que aparecem na lista é o modo do evento e neste será armazenado a informação sobre o tipo do evento e o estado que pode ser ativo ou inativo, conforme segue:

20H - "Início"

EOH - "Término"

A1H - Frequência ativo com horário não ajustado

AOH - Frequência ativo

80H - Frequência inativo

60H - Fixo ativo

40H - Fixo inativo

O bit 5, do byte de modo, indica se o evento está ativo (1) ou inativo (0).

Cada entrada da lista de eventos pode conter, de acordo com o modo, os seguintes campos:

Informação	nº de bytes [min,max]
I. Horário	3
II. número do evento (ordem física na lista)	1
III. modo	1
IV. Endereço do próximo evento na lista ligada	2
VI. Frequência (opcional)	2
VI. Intervalos (opcional)	
VI.1 número de intervalos	1
VI.2 Intervalo corrente	1
VI.3 Valores dos intervalos	[6,36]
VII. Endereço dos códigos associados	
VII.1 número de tarefas associadas	1
VII.2 Valores dos endereços absolutos das tarefas	[2,10]
Total de bytes ocupados por evento	[10,58]

3.2.2.2. O Código de Detecção de Erros

Para possibilitar à estação de campo o reconhecimento de defeitos na gravação do código no cartucho, bem como para que a estação base possa reconhecer erros introduzidos nos registros de dados foi adotado um código de detecção de erros.

Este código foi o LRC (Longitudinal Redundancy Code), que consiste em adicionar um caracter ao final de um bloco,

no qual cada bit é resultado da aplicação da função OR-EXCLUSIVO sobre os bits de cada posição correspondente dos caracteres do bloco [16].

O LRC foi adotado por ser melhor que a paridade simples e, também, pelo fato de não ser necessário um código de detecção de erros mais sofisticado, pois é pequena a probabilidade de erros no cartucho.

O LRC foi incluído ao fim dos seguintes módulos do código intermediário e área de dados compreendendo todos os bytes do respectivo módulo:

- Descritor de sensores;
- Descritor de ambiente;
- Código da tarefas: Um LRC ao fim de cada tarefa e um LRC global das tarefas, constituído pelo OR-EXCLUSIVO dos LRC's de cada tarefa;
- Tabela de rotinas de conversão;
- Lista de eventos;
- Registro de escrita (um para cada registro).

O LRC é verificado pelo interpretador da estação de campo no momento de ativação do experimento e por demanda (via console), bem como pelo receptor dos dados adquiridos no retorno do cartucho à estação base.

3.2.2.3. Implementação do Tradutor

O tradutor da LADAM foi implementado na linguagem Pascal em ambiente SIM/M - Itautec. A tradução foi implementada em um passo constituído de análise léxica, sintática, semântica e geração do código intermediário.

O tradutor recebe como entrada um plano de coleta de dados (i.e. o programa fonte escrito em LADAM) e um arquivo de sensores contendo a descrição de todos os sensores disponíveis para utilização e um arquivo de mensagens de erro, gerando o código intermediário em cartucho (ver figura 3.1). Além deste, é gerada uma listagem e um arquivo contendo o "Dicionário de comandos de escrita" que é utilizado, quando da recepção dos dados, para possibilitar a decodificação da área de dados do cartucho.

O analisador sintático é descendente recursivo. A tabela de símbolos foi implementada inicialmente como uma lista ligada simples. Não foram implementadas, nesta primeira versão, rotinas de tratamento de erro portanto o tradutor pára ao encontrar o primeiro erro.

A geração de código foi implementada através de rotinas em linguagem de montagem acopladas ao tradutor, facilitando e tornando mais seguro o acesso ao cartucho. Para tal foi definida uma área de interface para armazenar as variáveis acessadas durante a transferência do código para o cartucho. As rotinas de gravação de código em cartucho foram alocadas nas primeiras posições disponíveis na memória, para garantir que ocupem área de memória independente daquela em que será mapeado o cartucho no momento em que seja necessário acessá-lo [05]. Pelo mesmo motivo foram utilizadas variáveis absolutas, pois o compilador Pascal aloca suas variáveis em memória baixa. A figura 3.5 mostra a alocação de memória utilizada.

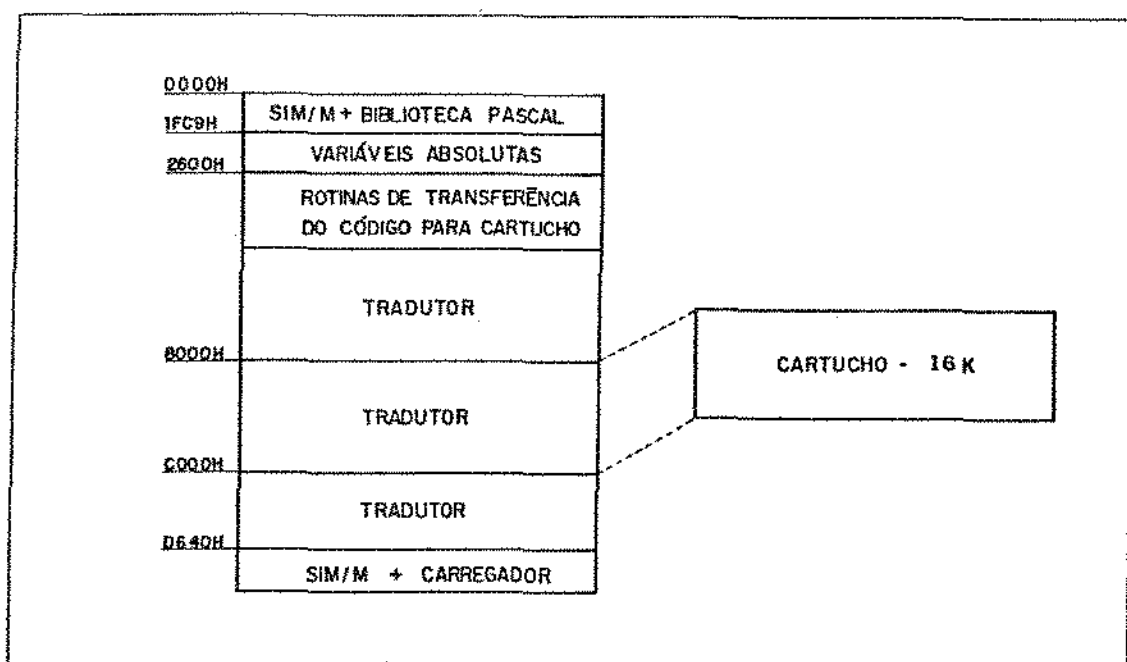


Fig. 3.5 - Alocação de memória utilizada pelo Tradutor

3.2.3. A Recepção dos Dados Adquiridos

Um programa utilitário foi desenvolvido para interpretar os dados adquiridos, gerar um arquivo em disquete com o conteúdo da área de dados e listar o código intermediário gravado em cartucho.

A primeira opção, que lista o código intermediário gravado em cartucho, pode ser utilizada pelo meteorologista para verificar a correção do conteúdo da programação feita na estação base.

A segunda opção lista todos os registros da área de dados de acordo com o dicionário gerado na criação do código intermediário (ver figura 3.1). Se algum registro não está de acordo com as especificações do dicionário, é listado em hexadecimal cimal.

Além disto o programa verifica os LRC's dos registros de dados emitindo mensagem caso seja detectado erro. Nesta opção também é feita a gravação de um arquivo em disquete contendo os dados adquiridos para posterior manipulação por pacotes comerciais ou programas que possam vir a ser construídos pelo usuário para interpretação dos dados.

No apêndice B podem ser observados os relatórios emitidos pelo receptor de dados para cada experimento simulado.

3.3. O Sistema Residente na Estação de Campo

O sistema residente na estação de campo foi projetado para ser instalado no protótipo descrito na seção 3.1. Este sistema foi dividido em dois programas básicos que controlam o funcionamento da estação. Um deles é o interpretador, capaz de executar a programação de experimentos, armazenada em cartucho na estação base na forma já descrita em 3.2.2.1. O outro é o monitor, responsável pela verificação do funcionamento do hardware e por rotinas de serviços para interações com o console, sensores e relógio de tempo real; este será descrito em [04].

A estação de campo ao ser ativada passa o controle para o monitor que faz a verificação inicial do hardware instalado, detectando inclusive a configuração dos sensores atualmente conectados e suas respectivas portas. A partir destas informações o monitor constroi uma tabela em memória RAM que será utilizada pelo interpretador; em seguida entrega o controle para o interpretador que fica disponível para requi-

sições do usuário. Entre estas requisições está o pedido de ativação do experimento.

A figura 3.6 a seguir mostra o relacionamento do monitor com o interpretador.

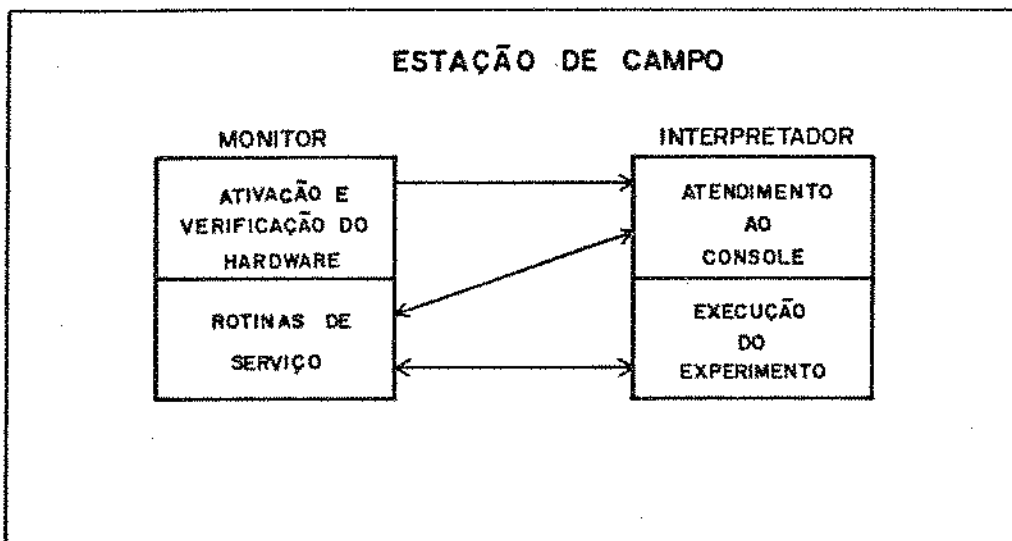


Fig. 3.6 - Relacionamento do monitor com o interpretador

3.3.1. O Interpretador

O interpretador também pode ser dividido em duas partes interligadas. A primeira compreende as rotinas de atendimento ao console e a segunda o núcleo básico responsável pela execução dos experimentos.

3.3.1.1. As Rotinas de Atendimento ao Console

O objetivo das rotinas de atendimento ao console (ATCS na Figura 3.7) é dar ao experimentador a oportunidade de interagir com a estação de campo. Estas interações podem se dar antes, depois ou durante a execução de um experimento; pa-

ra tal foi definido um conjunto de comandos através dos quais o experimentador pode requisitar os serviços do interpretador. Estes comandos podem ser simples consultas, como também podem causar alterações ou mesmo interromper um experimento em andamento.

Durante a execução de um experimento as rotinas de atendimento ao console tem baixa prioridade, sendo atendidas apenas nos intervalos entre tarefas.

Quando o console é atendido, através de interrupção gerada pelo teclado, as rotinas requisitadas são colocadas na fila de tarefas prontas, esperando assim a sua execução. Durante a execução de tais tarefas as interfaces com o console estão sujeitas a um "time-out" para evitar que haja atraso na execução das tarefas do experimento devido a interferências externas. Este "time-out" será controlado pela rotina de serviço do monitor que atende ao console.

Os comandos implementados são :

CONSULTAS:

D - Data do dia

Requisita data e hora corrente.

S - "Status" do experimento

Requisita informações sobre o andamento do experimento corrente. O sistema responde:

Data do início do experimento

Local de instalação

Situação do cartucho quanto ao espaço alocado

Código: xxxx bytes

Dados: xxxx bytes

Livre: xxxx bytes

C - Configuração dos sensores

Requisita ao sistema a configuração dos sensores para o experimento corrente.

O sistema mostra para cada porta lógica:

- Código do sensor instalado
- Número do sensor instalado.

L - Listagem dos eventos

Requisita ao sistema a listagem resumida da lista de eventos.

O sistema mostra para cada evento:

- Número do evento
- Modo
- Horário
- Frequência e intervalos, se o evento é de frequência
- Número e endereço das tarefas associadas.

M - Mensagens de erro

Requisita ao sistema a listagem das mensagens de erro acumuladas no arquivo de ocorrência do experimento. Este arquivo consiste em uma área reservada da RAM da estação de campo, em que são acumulados dados das mensagens de erro ocorridas durante o experimento, bem como sua situação atual. Ao

término do experimento, se houver espaço disponível no cartucho, será gravado o conteúdo deste arquivo de ocorrências.

O sistema mostra para cada mensagem o número, tamanho e conteúdo.

R - Leitura dos valores instantâneos dos sensores.

Requisita ao sistema a leitura do valor corrente de uma determinada porta lógica.

O sistema pede o número da porta lógica e a opção de leitura, respondendo o valor lido.

V - Verifica o LRC do cartucho

Requisita a verificação dos códigos de detecção de erro do cartucho.

O sistema verifica os LRCs calculados para o cartucho e havendo algum erro informa o número da tabela de controle, ou tarefa onde ocorreu.

INTERVENÇÕES

I - Inicie experimento

Requisita ao sistema a ativação do experimento instalado.

O sistema pede confirmação e, sendo esta positiva, o interpretador passa, através do seu núcleo básico, a executar o experimento instalado.

E - Encerrar experimento

Requisita ao sistema o encerramento do experimento corrente.

O sistema pede confirmação e, sendo esta positiva, o interpretador encerra o experimento corrente e volta ao modo de atendimento puro ao console.

F - Alteração da frequência de um evento

Requista ao sistema a alteração da frequência de um evento controlado por frequência.

O sistema pede que seja informado o número do evento a ser alterado, lista o evento correspondente e pede a confirmação; sendo esta positiva, o sistema requisita a nova frequência e, não havendo erros, altera a frequência do evento.

T - Reinicia o experimento

Requisita ao sistema o reinício do experimento instalado desprezando a área de dados até então construída, que será reutilizada (a ser implementada).

G - Grava mensagem no cartucho

Requisita ao sistema a gravação de uma mensagem em cartucho.

O sistema pede o número e conteúdo da mensagem, com tamanho máximo de 256 bytes, e grava em cartucho da seguinte forma :

```
| U | tamanho | número | <-- conteúdo --> | U | irc |
```

O tamanho total da mensagem será o tamanho do conteúdo mais 5 devido às marcas incluídas pelo sistema.

3.3.1.2. Núcleo Básico do Interpretador

O núcleo básico do interpretador será inicialmente ativado quando o usuário requisitar, através do console, a ativação do experimento instalado.

Este núcleo é responsável pelo controle da lista de eventos e execução das tarefas a estes associadas, controlando também o atendimento ao console através da colocação das tarefas requeridas na fila de tarefas prontas, ou chamando o monitor quando a rotina é concernente a este. O núcleo é capaz de interpretar todas as informações sobre a execução do experimento gravadas, na estação base, pelo tradutor da LADAM.

Para apresentação do núcleo básico do interpretador, este será dividido em rotinas da seguinte forma:

- ROTINA DE INICIALIZAÇÃO DO EXPERIMENTO
- ROTINA DE CONTROLE DO EXPERIMENTO
- ROTINA PARA ATIVAÇÃO DOS EVENTOS
- ROTINA PARA EXECUÇÃO DAS TAREFAS
- ROTINA DE FINALIZAÇÃO DO EXPERIMENTO

A figura 3.7 mostra o esquema de funcionamento do núcleo básico do interpretador.

3.3.1.2.1. Rotina de Inicialização do Experimento

Esta rotina é responsável pelos procedimentos iniciais necessários para ativação do experimento e é colocada na fila de tarefas prontas como resultado da execução do comando de console "I".

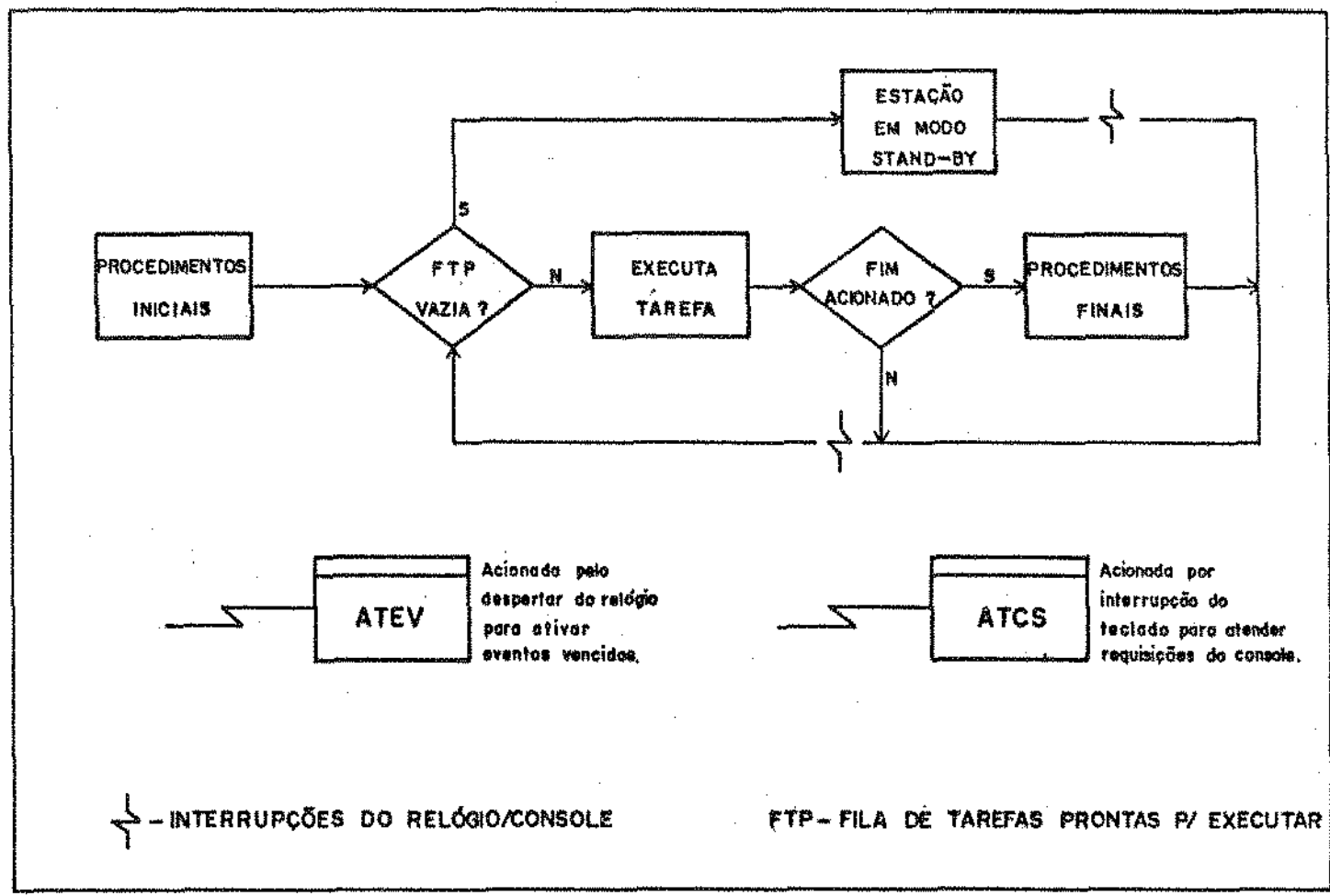


Fig. 3.7. - Fluxo do núcleo básico do Interpretador

O primeiro destes procedimentos faz a verificação do conteúdo do cartucho através do LRC, utilizado para detecção de erro conforme descrito em 3.2.2.2.

Em seguida é feito o confronto da configuração instalada e a programada; o monitor ao fazer a verificação do hardware cria uma tabela na área de interface com o interpretador, localizada na RAM(E000H-FFFFH), contendo para cada uma das portas lógicas o código do sensor instalado e o número do sensor utilizado. Caso não haja sensor instalado em alguma porta a entrada correspondente fica zerada.

Com base nesta tabela, no descritor do ambiente e no descritor dos sensores o interpretador procede à verificação. Para cada porta verificada o interpretador mostra no console o código do sensor programado, do sensor instalado e uma mensagem de aceitação ou erro na configuração. Em caso de erro é requerido ao usuário a decisão de continuar o experimento ou cancelar para que a configuração seja corrigida ou o cartucho trocado.

Durante a execução desta rotina é copiado para o descritor do ambiente o número do sensor instalado, documentando assim com qual versão do sensor o experimento foi executado.

Em seguida o sistema requisita ao usuário o local de instalação e o nome do responsável bem como, a data e hora para início da aquisição de dados. Se o usuário não desejar especificar esta data e hora, o sistema assumirá a data e horário corrente, ou seja determina o início imediato da aquisição de dados. A estas informações fornecidas pelo usuário deu-se o nome de data e hora de partida do experimento.

Em seguida, é gravada uma mensagem do sistema em cartucho e no arquivo de ocorrências explicitando:

- Data e horário da ativação do experimento (1)
- Data e horário de Partida do experimento (2)
- Local de instalação (x)
- Responsável pela instalação (y)

O formato da mensagem será o seguinte:

```
S 01 t d m a h m s d m a h m s xxxxxxxxxxxx yyyyyyyyyy S lrc
```

O tamanho da mensagem com as marcas, de início e fim "S", número da mensagem(01) e tamanho(t), é de 36 bytes.

Após vencida esta etapa o interpretador passa a trabalhar sobre a lista de eventos, conforme os seguintes passos:

1. Se existe evento "Início" definido pelo usuário colocá-lo na fila de tarefas prontas.
2. Ajusta o horário de início dos eventos de frequência de acordo com o horário de partida do experimento.
3. Busca na lista ligada de eventos o primeiro evento que satisfaça à seguinte condição:
 horário do evento \geq horário de partida do experimento.
 O endereço deste evento é armazenado como apontador para o evento corrente.
4. Coloca o relógio para despertar no horário do evento encontrado.

A data de partida será a data de referência inicial para o experimento; a data de referência mudará cada vez que o ponteiro da lista de eventos completar um ciclo, ou seja apontar para o evento "Término" que, por sua vez, serve também como terminal da lista circular de eventos. A figura 3.8 abaixo ilustra o ciclo dos eventos.

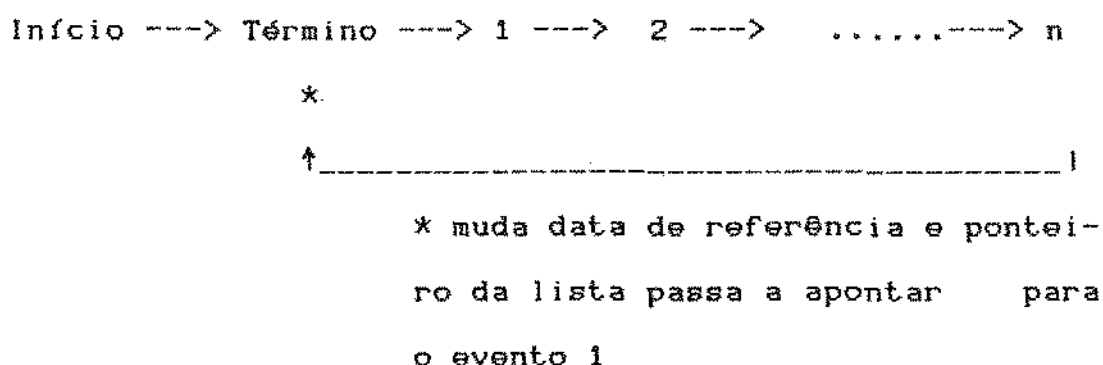


Fig. 3.8 - Ciclo dos eventos

O segundo passo (na lista acima) consiste em ajustar os horários dos eventos baseados em frequência. O ajuste do horário de início dos eventos é necessário porque um evento de frequência possui intervalos nos quais deve ser ativado, ao invés de horários fixos. Portanto a partir do conhecimento do horário de partida determina-se o horário de início destes eventos. No caso de eventos para os quais não foram definidos intervalos, o horário de início do evento será o próprio horário de partida. Para os demais será verificado se o horário de partida está dentro de algum dos intervalos, caso em que o horário de início do evento também coincide com o horário de partida. Caso contrário o horário de início do evento será

igual ao limite inferior do intervalo imediatamente posterior ao horário de partida.

Após decidido qual o horário de início para um evento este é, então, realocado na lista de modo manter a lista de eventos ordenada. Vale lembrar que o horário dos eventos de frequência, originalmente criados pelo tradutor é zero, logo eles estão inicialmente alocados (alocação física) como os primeiros elementos da lista de eventos, logo após os eventos "Início" e "Término".

O terceiro passo consiste numa simples busca do primeiro evento, cujo horário de ativação programado é maior ou igual ao horário de partida do experimento.

Em seguida, no quarto passo, coloca-se o relógio de tempo real para despertar no horário de ativação do evento encontrado no terceiro passo.

3.3.1.2.2. Rotina de Controle do Experimento

Após concluída a fase de inicialização (comando "I"), o interpretador se concentra na ativação dos eventos e execução de suas respectivas tarefas, podendo eventualmente atender a requisições do console (ver figura 3.7).

A rotina de controle tem a função de detectar a presença de tarefas na fila de tarefas prontas para executar e comandar a chamada da rotina que efetua a execução destas podendo, no entanto, ser interrompida pelo relógio de tempo real para ativação de eventos e pelo usuário através de requisições do console.

A fila de tarefas prontas para executar é circular e cada célula contém:

- Data e horário de referência do evento associado à tarefa;
- Tipo da tarefa (ver abaixo);
- Endereço da tarefa.

A data e horário de referência acima citados representam a data e horário programados para ativação do evento associado à tarefa e servirá para atualizar as variáveis "dataref" e "horaref" da área de dados ao se iniciar a execução das tarefas, conforme descrito em 3.2.1.2.1.

O tipo da tarefa distingue tarefas programadas a serem interpretadas de tarefas embutidas no "firmware", para atendimento ao console, por exemplo; estas já em linguagem de máquina. Quando a estação é despertada pelo relógio ou por uma intervenção via console, é acionada a rotina de ativação de eventos, a quem compete colocar as tarefas associadas ao evento ativado na fila de tarefas prontas para executar.

Ao detectar a presença de tarefas nesta fila, a rotina de controle chama a rotina para execução de tarefas. Caso a fila esteja vazia o interpretador comunicará ao monitor que este deve colocar a estação em modo "stand-by". A estação sai do modo "stand-by" pelo despertar do relógio ou através de requisição do console.

Após a execução de cada tarefa a rotina de controle verifica se o encerramento do experimento foi acionado; em caso positivo é iniciada a execução dos procedimentos finais,

caso contrário a rotina deve continuar no seu laço de controle.

3.3.1.2.3. Rotina para Ativação dos Eventos

A rotina para ativação dos eventos (ATEV na Figura 3.7) é acionada pelo despertar do relógio de tempo real, através de interrupção. Pode-se descrever o comportamento geral da rotina da seguinte forma:

- I. Coloca todas as tarefas associadas aos eventos vencidos na fila de tarefas prontas para executar;
 1. Coloca tarefas associadas ao evento ativado, que é o evento corrente, na fila de tarefas prontas;
 2. Se o evento é de frequência então
 - 2.1. Determina próximo horário do evento;
 - 2.2. Ajusta evento com novo horário na lista de eventos;
 3. Busca próximo evento ativo;
 4. Faz evento corrente = próximo evento ativo; se horário do evento corrente \leq horário corrente vai para 1 senão prossegue;
- II. Programa relógio para despertar no horário do evento corrente e retorna da interrupção.

Assim sendo, todas as tarefas associadas aos eventos cujo horário programado para ativação são maiores ou iguais ao horário corrente são colocadas na fila de tarefas prontas para executar. O acesso ao evento ativado se dá através de um apontador (aqui chamado evento corrente), para a lista de

eventos, que é mantido sempre atualizado indicando o evento que está para ser ativado pelo despertar do relógio. A partir deste o acesso aos próximos eventos é feito percorrendo a lista ligada de eventos (ver figura 3.8).

Cada evento contém, conforme descrito em 3.2.2.1.3, o endereço das tarefas a serem executadas quando ocorre sua ativação. A partir destas informações as tarefas são inseridas na citada fila e o apontador para último da fila é atualizado.

A inserção de tarefas na fila é feita por uma rotina específica para este propósito, a qual pode ser chamada tanto pela rotina aqui descrita como também pela rotina de atendimento ao console. Tal rotina pode, eventualmente, descobrir que a fila de tarefas prontas está cheia, não suportando mais inserções.

Este fato é extremamente indesejável e pode ocorrer devido a uma programação incoerente com os recursos da estação. Ou seja, o programador da coleta de dados estimou uma frequência de execução de tarefas muito alta, de modo que a rotina de ativação de eventos não consegue sair do laço principal, portanto não consegue esvaziar a lista de tarefas prontas.

Neste caso, para evitar um possível colapso do sistema, é adotado um procedimento sumário: A rotina responsável pela inserção de tarefas na fila ao detectar que esta se encontra cheia, interrompe temporariamente as inserções e em seguida grava uma mensagem de alerta em cartucho e no arquivo de ocorrências do sistema, comandando logo após a execução das tarefas atualmente presentes na fila.

Este fato certamente provocará um deslocamento nos horários em que devem ser feitas as aquisições de dados.

No segundo passo, o modo do evento é examinado e caso seja um evento de frequência (modo = AOH) o próximo horário do evento é determinado somando-se o período ao horário anterior e verificando se o horário obtido pertence aos intervalos definidos para o evento. Caso não pertença, o novo horário será igual ao limite inferior do intervalo imediatamente seguinte ao horário obtido.

Após definido o novo horário do evento deve-se proceder ao ajuste da lista de eventos, do mesmo modo que foi feito na rotina de inicialização.

Em seguida, os próximos eventos da lista são acessados. Se o evento encontrado estiver inativo, deve-se, continuar a varredura na lista de eventos até que se encontre o primeiro evento ativo, este então será o evento considerado como o novo evento corrente. Note que os eventos inativos são mantidos na própria lista de eventos devido ao fato de que o número destes, em geral, é pequeno em relação aos ativos. Caso a situação não fosse esta seria melhor manter estes eventos numa lista separada, inserindo-os na lista de eventos quando ativados.

O último passo da rotina é a chamada do monitor para programação do relógio, passando como parâmetro o horário obtido nos passos anteriores, retornando em seguida da interrupção.

3.3.1.2.4. Rotina para Execução das Tarefas

A rotina para execução das tarefas é chamada pela rotina de controle ao detectar a presença de tarefas na fila de tarefas prontas para executar.

A tarefa a ser executada é determinada através de um apontador para o primeiro da fila de tarefas prontas para executar.

Os passos seguidos pela rotina são :

1. Copia a data e hora da fila para a data e hora de referência da área de dados;
2. Se tipo da tarefa = sistema residente então
 - 2.1. Cede o controle do processador para a tarefa senão
 - 2.2. Executa tarefa através do interpretador de instruções;
3. Atualiza apontador para primeiro da fila.

No primeiro passo a rotina copia a data e hora programada para ativação do evento associado a tarefa para o endereço das variáveis "dataref" e "horaref" da área de dados, mantendo-as atualizada.

No segundo passo a rotina verifica o tipo da tarefa a ser executada. As tarefas podem ser de dois tipos :

1. Tarefas do sistema residente, que é o caso das tarefas geradas pela rotina de atendimento ao console, as quais residem no "firmware" da estação.
2. Tarefas escritas em linguagem intermediária que são aquelas programadas pelo experimentador na estação base, as quais residem em cartucho, e são especialmente interpretadas.

Desse modo, no caso de tarefas do tipo um a rotina para execução de tarefas cede o controle do processador para a execução da tarefa.

No caso de rotinas do tipo dois a rotina para execução das tarefas controla a interpretação de cada instrução chamando o interpretador de instruções.

O interpretador de instruções, juntamente com o avaliador de expressões é capaz de executar cada uma das instruções descritas em 3.2.2.1.2.

Após concluída a execução da tarefa em questão, a rotina atualiza o apontador para primeiro da fila e retorna.

3.3.1.2.5. Rotina de Finalização do Experimento

A rotina de finalização do experimento pode ser ativada conforme programação feita na estação base, ou através de requisição do console, ou ainda quando atingido o fim da memória de dados em cartucho.

No momento em que a rotina é iniciada as interrupções tanto da console quanto do relógio são inibidas.

Os passos seguidos pela rotina são :

1. Exibe no "display" a mensagem "EXPERIMENTO EM ENCERRAMENTO" e faz anotações sobre o motivo do encerramento;
2. Se existe evento "Término" definido pelo usuário, colocam-se as tarefas associadas a este na fila de tarefas prontas;

3. Coloca tarefa responsável pelas mensagens de encerramento e mudança do estado da estação na fila de tarefas prontas.

No passo um a rotina exibe no "display" uma mensagem deixando explícito que o experimento iniciou seus procedimentos finais e deixa marcado qual o motivo de encerramento, que pode ser:

- 1 - Fim normal programado
- 2 - Encerramento forçado pelo usuário na estação.
- 3 - Fim da memória de dados em cartucho.

Os casos dois e três são utilizados para impedir imediatamente a continuação do experimento, dado que no caso três fica impossibilitada qualquer gravação futura em cartucho. O caso dois é um recurso deixado ao usuário para ser utilizado quando este decide encerrar o experimento antes da hora programada. Nestes casos, a rotina de encerramento cancela a execução de tarefas da fila e segue diretamente para o passo três.

No terceiro passo é colocada na fila de tarefas prontas a tarefa do sistema responsável pela alteração do estado da estação de experimento em andamento para experimento encerrado. Esta tarefa também grava uma mensagem final do sistema em cartucho e no arquivo de ocorrência com o seguinte conteúdo:

- Data e horário de encerramento do experimento (dmahms);
- Espaço total ocupado com dados (YY YY)
- Motivo de encerramento (M)

O formato da mensagem é o seguinte :

```
| S | 02 | t | d | m | a | h | m | s | YY | YY | M | S | LRC |
```

O tamanho da mensagem com as marcas de início e fim (S), número da mensagem (02) e tamanho (t) é de 14 bytes.

No final desta tarefa as interrupções são liberadas e o sistema volta ao ciclo da rotina de controle.

Eventualmente, pode ter restado alguma programação do relógio ainda não atendida, que aparece após as interrupções terem sido liberadas; esta é completamente descartada devido à mudança de estado da estação.

3.3.1.3. A IMPLEMENTAÇÃO DO INTERPRETADOR

O interpretador da estação de campo foi implementado em linguagem de montagem Z-80 compatível com o microprocessador NSC-800 da estação de campo.

O programa foi dividido nos seguintes módulos:

- 1 - Rotina de controle, inicialização e ativação de eventos e tarefas;
- 2 - Rotinas de atendimento ao console;
- 3 - Rotinas de ajuste de eventos, finalização e gravação de mensagem do sistema;

- 4 - Rotinas auxiliares;
- 5 - Avaliador de expressões;
- 6 - Interpretador de instruções;
- 7 - Área de dados;

Estes módulos foram montados individualmente e ligados para formar o código objeto.

O código final foi de aproximadamente 14 Kbytes (excluídas as rotinas de simulação e avaliando a memória a ser utilizada para interface com o monitor), que serão alocados a partir da posição 2000H, conforme figura 3.9, que mostra a proposta inicial de divisão de espaços na memória da estação de campo.

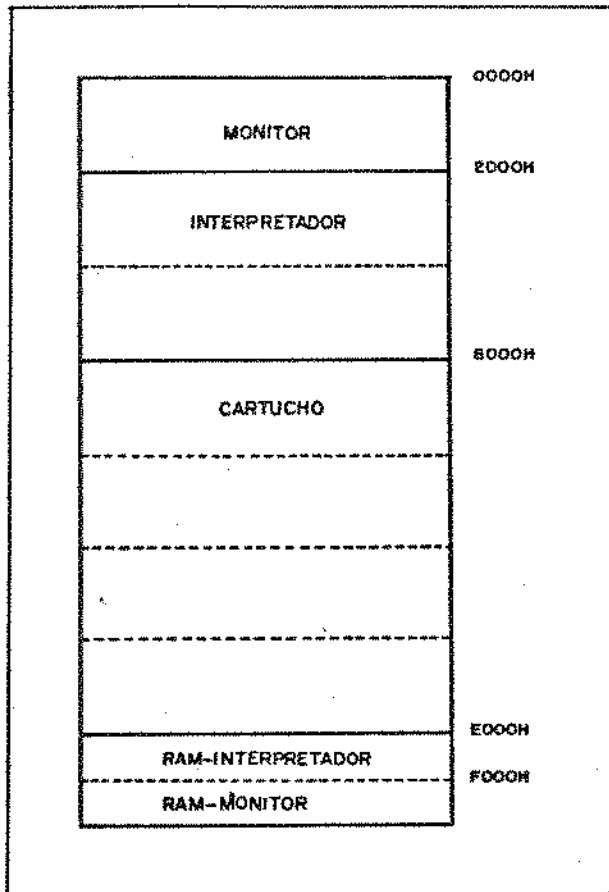


Fig. 3.9. - Alocação da memória na Estação de Campo

Capitulo IV

A SIMULAÇÃO DO SISTEMA

4. A SIMULAÇÃO DO SISTEMA

Conforme descrito no capítulo 3, o sistema apresentado compreende duas fases distintas que operam em locais diferentes. A primeira fase, residente na estação base, foi simulada no seu próprio ambiente. Já a segunda que, na prática, ocorreria na estação de campo foi simulada no I7000 sob o sistema operacional CP/M.

Neste capítulo são apresentadas as simulações efetuadas bem como os resultados obtidos.

Como objeto da simulação foram escolhidos três experimentos de níveis crescentes de complexidade. O primeiro deles corresponde ao experimento aplicado em estações termo-pluviométricas convencionais, cuja programação é simples e exige pouco da linguagem. O segundo é o experimento padrão aplicado em postos meteorológicos completos; este exige um código maior e utiliza mais recursos da linguagem. O terceiro é um exemplo de experimento micrometeorológico utilizado para verificar efeito de baixas temperaturas e perda de água em culturas de café. Este é o caso que mais utiliza as facilidades de especificação de eventos da linguagem, assim como é o representante da classe de experimentos microclimatológicos, já citada como a classe para a qual o SADAM encontra sua maior aplicabilidade.

4.1. O Ambiente Simulado

Para a simulação da estação de campo, as rotinas de interface com o console que seriam parte integrante do monitor

foram programadas para o I7000 utilizando as rotinas padrão do BIOS/CPM.

Os sensores foram simulados através de arquivos de dados, um para cada porta lógica especificada para o experimento. Tomou-se o cuidado de construir tais arquivos utilizando para dados valores típicos que os sensores gerariam no caso de um experimento real. Para tanto foram utilizados registros climatológicos reais obtidos da estação climatológica da Universidade Estadual de Maringá.

O fluxo seguido pelo sistema foi o mesmo descrito no capítulo 3, ver figura 3.7, excetuando as interrupções. Deste modo, o sistema sempre executa todas as tarefas da fila e entra em modo "stand-by"; o despertar do relógio foi simulado igualando-se a hora corrente à hora programada. Isto faz com que a estação saia de "stand-by" seguindo para a execução da rotina ATEV.

A execução da rotina ATCS foi simulada através da ativação de pedidos da console entre uma tarefa e outra.

O relógio de tempo real foi iniciado com zero horas e incrementado de um segundo a cada leitura sendo alterado, conforme descrito acima, a cada passagem pelo modo "stand-by" quando assume o horário do evento ativado.

Para cada experimento foi simulado um período de 24 horas, uma vez que neste período se percorre toda a programação. Os outros dias seriam apenas repetições do simulado com dados diferentes.

4.2. O Experimento Padrão Mínimo

O experimento padrão mínimo é o mais primário; seu objetivo é simular uma estação termo-pluviométrica convencional, armazenando valores extremos da temperatura máxima e mínima, bem como o total pluviométrico diário.

As observações feitas neste experimento são as seguintes:

1. Em cada horário padrão, gravação da temperatura (lida naquele momento) e da quantidade de chuva medida desde o horário padrão anterior até o presente.
2. Leitura da temperatura a cada hora, avaliando-se para o período a máxima e mínima das 24 leituras efetuadas.
3. Gravação dos valores da temperatura máxima e mínima com os respectivos horários de ocorrência e total pluviométrico diário no final do período (i.e. às 7:00).

O período considerado vai das 07:00 às 07:00 horas do dia seguinte. Os horários padrão considerados para leitura dos sensores são os seguintes: 7:00, 9:00, 14:00, 15:00 e 21:00 horas.

Os sensores utilizados são : semicondutor (temperatura) e pluviômetro (chuva).

A duração do experimento é indeterminada.

A programação do experimento mínimo, apresentada no programa B.1.1, é simples; segue-se uma discussão de diversas partes do código para facilitar seu entedimento.

Na seção de declaração de sensores aparecem o semicondutor (código O1A) e o pluviômetro (código OBA). Note-se que os parâmetros destes sensores estão disponíveis no cartucho porque ali foram colocados pelo tradutor, que os obteve do arquivo de sensores (ver apêndice B.1.2. e C.1) na estação base.

As variáveis externas "tempar" e "qtdchuva" armazenam as leituras instantâneas do semicondutor e pluviômetro respectivamente e formam o descritor do ambiente.

O conteúdo do descritor dos sensores e do descritor do ambiente pode ser observado no apêndice B.1.2, relativo ao código gerado para o experimento.

Para atender às especificações do experimento foram programadas as tarefas: "header", "varredura", "avaltemp", e "gravperiod".

A tarefa "header" tem sempre o objetivo de fazer as inicializações dos experimentos; no caso apresentado: atribuir valores limites à temperatura máxima e mínima e contador de chuva, assim como iniciar o valor de "chuvant" que representa o primeiro valor lido do pluviômetro. Conforme descrito na seção 3.2.1.2.2, esta tarefa é executada apenas no momento de ativação do experimento.

A tarefa "varredura" é executada a partir da ativação do evento "evt1" e atende à primeira especificação do experimento, que é amostrar os valores de temperatura do ar e

quantidade de chuva nos horários padrão (7:00, 9:00, 14:00, 15:00 e 21:00 horas). Observe-se que, neste caso, os rótulos "evt1", "evt2", "evt3" são dispensáveis porque não há necessidade fazer referência a estes eventos. Os rótulos foram colocados para facilitar esta descrição do programa.

A obtenção da quantidade de chuva desde o horário anterior é feita através da diferença entre o valor lido e valor anterior. A cada leitura é verificado se o sensor ultrapassou a sua capacidade máxima, no caso 4096 (12 bits), caso em que é necessário levar em conta o "overflow" como se pode ver na programação da tarefa (linha 250-300). Obviamente, a capacidade do pluviômetro é tal que nunca ocorrem mais que 4096 contagens em um intervalo.

A tarefa "avaltemp" é executada a partir da ativação do evento "evt2". Sua função é registrar as temperaturas máxima e mínima do período, bem como os horários em que ocorreram.

A tarefa "gravperiod" é executada a partir da ativação do evento "evt3" e tem o objetivo de gravar os valores obtidos para temperatura máxima, mínima e total pluviométrico referentes ao período anterior. Uma vez que esta tarefa é executada no limiar entre um período e o seguinte, nesta é analisado o espaço disponível em cartucho, através da função "memavall", para decidir se o cartucho suporta gravações por mais um período ou o experimento deve ser encerrado.

O total de espaço necessário para um período é obtido examinando-se o espaço necessário para cada "write" de acordo com o número de bytes ocupados pelas variáveis gravadas mais 5 bytes referentes às marcas incluídas em cada registro gravado pelo sistema.

No programa aqui analisado (programa B.1.1):

linha 330 2 bytes (tempar) + 2 bytes (qtdchuvai) +
5 bytes (marcas) = 9 bytes.

linha 520 3 bytes (datmin) + 3 bytes (hormin) + 4
bytes (tempmin) + 3 bytes (datmax) + 3 bytes (hor-
max) + 4 bytes (tempmax) + 4 bytes (totpluv) + 5 by-
tes (marcas) = 19 bytes.

Um período consumirá 5 (número de horários) * 9 +
19, isto é, 64 bytes.

Note que neste caso a duração do experimento é inde-
terminada portanto a programação do encerramento é uma precau-
ção para que o experimento não avance, dado que não há mais
espaço para um período completo de gravação. Se esta programa-
ção não fosse feita o sistema encerraria o experimento, no
momento apropriado, pelo seu próprio controle de espaço dispo-
nível.

Uma outra opção de encerramento seria o experimenta-
dor requisitá-lo através do console, quando achar conveniente.

A ordem de execução dos eventos é : "evt1", "evt2",
"evt3", exceto quando os três coincidem e a ordem passa a ser:
"evt1", "evt3", "evt2" devido à prioridade do evento fixo so-
bre o de frequência. Vale lembrar que eventos de mesma catego-
ria (fixo ou frequência) seguem a ordem da declaração.

4.3. O Experimento Padrão

O experimento padrão refere-se às medições feitas
em postos meteorológicos completos.

As observações feitas neste experimento são:

1. Leitura e gravação dos valores instantâneos de todos os sensores utilizados nos horários padrões para fins de climatologia.
2. Temperatura do ar seco - Leitura da temperatura a cada hora, avaliando-se para o período a temperatura máxima e mínima das 24 leituras efetuadas e gravação dos valores obtidos com os respectivos horários de ocorrência no final do período (i.e. às 7:00). Gravação da temperatura média do período de acordo com a seguinte fórmula:

$$TM = (T7 + T14 + 2 \times T21) / 4$$

onde:

TM = temperatura média diária.

T7 = temperatura obtida às 7:00 horas.

T14 = temperatura obtida às 14:00 horas.

T21 = temperatura obtida às 21:00 horas.

3. Temperatura da Relva - Avaliação da temperatura mínima registrada no período noturno (i.e. das 21:00 às 7:00 horas), durante o inverno, a partir de amostragens a cada minuto e gravação do valor obtido mais o respectivo horário em que ocorreu, no final do período.

4. Radiação Solar Global - Integrar os valores instantâneos lidos, no período, a partir de amostragens a cada minuto e gravar o valor obtido no final do período.

5. Insolação - Contagem do número de horas de sol por período, gravando o valor obtido no final deste. Considerar valores significativos de insolação aqueles que causam leitura maior ou igual a 0,7 mv no sensor de insolação. A amostragem deve ser feita a cada minuto.

6. Vento (velocidade) - Amostrar os valores instantâneos, a cada minuto, no intervalo de mais e menos 5 minutos em torno dos horários padrão, isto é, considerar os seguintes intervalos:
[6:55 , 7:05]; [8:55 , 9:05]; [13:55 , 14:05];
[14:55 , 15:05] e [20:55 , 21:05].
Calcular a velocidade média em cada intervalo e gravar o valor obtido no final deste.

7. Vento (direção) - Amostrar direção instantânea nos horários padrão.

8. Chuva - Acumular total pluviométrico diário gravando o valor obtido no final do período.

Os sensores utilizados no experimento são os seguintes:

- Semicondutor - Temperatura do Ar Sêco,
Temperatura da Relva.
- Higrômetro - Umidade Relativa do Ar.
- Piranômetro - Radiação Solar Global.
- Heliógrafo - Insolação
- Anemômetro - Velocidade e Direção do Vento.
- Pluviômetro - Quantidade de Chuva.

A duração do experimento é indeterminada.

Os horários padrão considerados são : 7:00, 9:00, 14:00, 15:00 e 21:00 horas. O período considerado para o experimento vai das 7:00 às 7:00 horas do dia seguinte.

O experimento padrão, apresentado no programa B.2.1, requer uma programação mais extensa e a utilização maiores recursos da linguagem.

Para esta coleta de dados foram especificados 8 sensores, 10 tarefas e 7 eventos, o que gera uma área de código intermediário de 1704 bytes.

Neste experimento foi incluída a tarefa "trailer" com o objetivo de mostrar como o experimentador pode especificar em seu programa as finalizações particulares a cada coleta de dados.

Pode-se observar uma maior utilização de cálculos durante a coleta, como em "gravperiod" e "calcvento". Note-se que estes cálculos proporcionam uma grande economia de espaço em cartucho, pois se este recurso não estivesse disponível ter

-se-ia que armazenar os valores instantâneos em cada período para um futuro cálculo na estação base.

Note que nesta tarefa também é mostrada uma outra maneira de se encerrar experimentos através do teste da data e hora de ativação das tarefas (linhas 1050-1080).

Na seção de declaração de eventos, além de um maior número destes pode-se destacar "evt4", "evt5" e "evt6" que utilizam o recurso de limitar as observações de eventos de frequência a um ou mais intervalos específicos.

O "evt4" atende ao requisito 3 do experimento, avaliando a temperatura apenas no período noturno.

O "evt6" torna possível o cálculo da velocidade média do vento correspondente a cada horário padrão, o que não seria viável se amostrássemos apenas o valor instantâneo num determinado horário padrão. A partir do sensor é obtido o espaço percorrido pelo vento, que amostrado num dado intervalo de tempo, possibilita o cálculo da velocidade média.

O diagrama da figura 4.1 mostra o fluxo de ativação dos eventos com as respectivas tarefas associadas a estes.

4.4. O Experimento Micrometeorológico

Este experimento é feito com o objetivo de verificar os efeitos de baixas temperaturas (geadas) e perda de água em culturas de café.

O experimento é feito em dez dias consecutivos nos meses de junho e julho (referentes ao inverno), e nos meses de novembro e dezembro (referentes ao verão).

As observações feitas são :

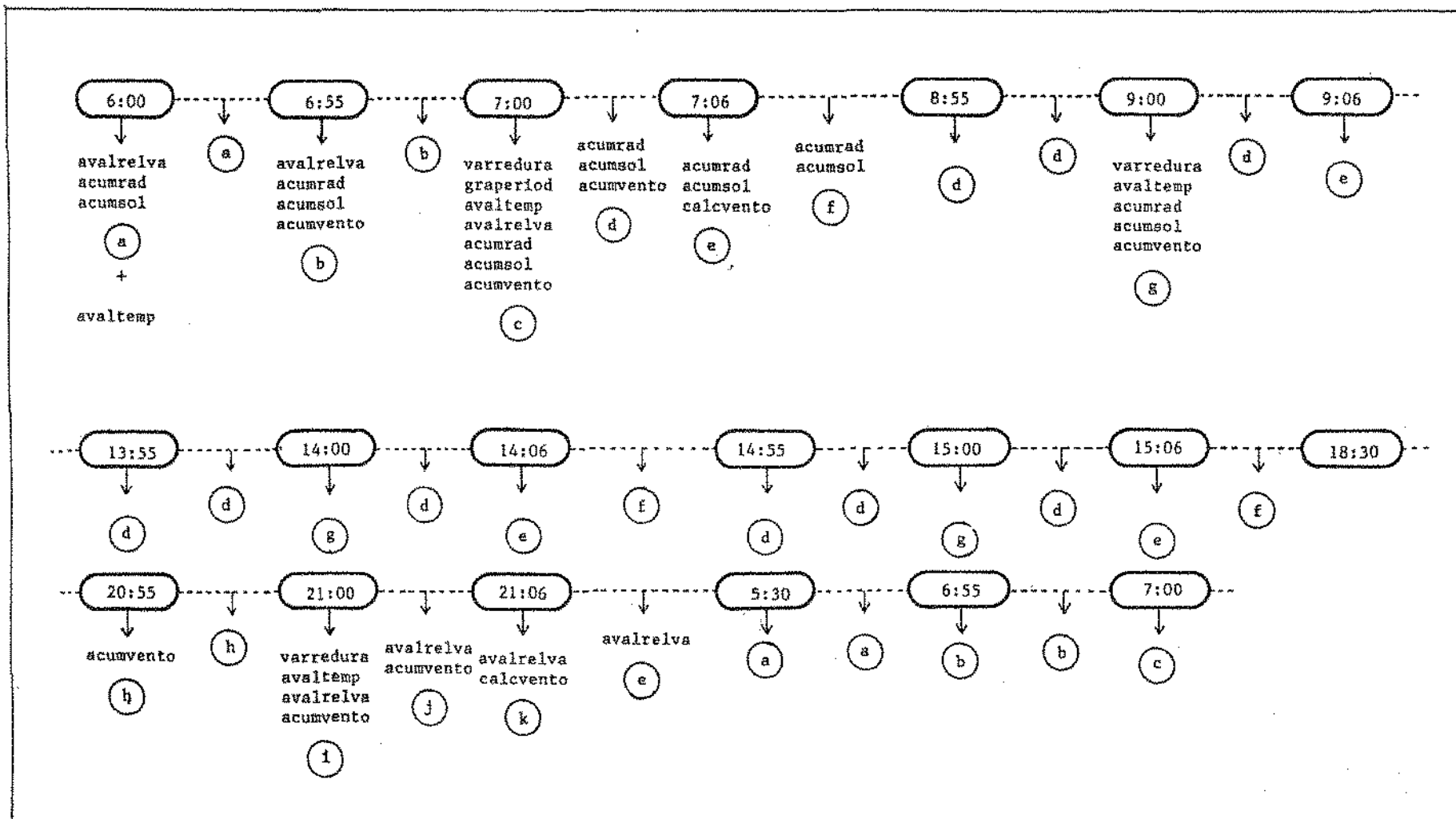


Fig. 4.1. - Diagrama de execução das tarefas do Experimento Padrão

1. Temperatura do ar sêco e úmido a 1,70 m no abrigo meteorológico - O registro deve ser feito de 15 em 15 minutos tomando-se como valor a temperatura média obtida de 6 amostragens feitas de 10 em 10 segundos durante o minuto precedente. O mesmo procedimento deve ser feito para as temperaturas 2 e 3 abaixo. Estas medidas ocupam duas portas com semicondutores.

2. Temperatura do solo para medição do fluxo térmico no solo. Ocupa 4 portas com termopares, localizados na superfície e a 5 cm, 10 cm e 25 cm da superfície.

3. Temperatura das folhas - Ocupa 4 portas com termopares localizados em folhas a 5, 20 cm e 40 cm do solo e o último nas folhas expostas.

4. Energia:
 - a) Radiação solar "PAR" - Radiação fotossinteticamente ativa. Obter somatório dos valores lidos no período, amostrando-os a cada minuto. Gravar valor obtido no final do período (i.e. às 7:00). Além disso, ler e gravar os valores instantâneos a cada 15 minutos. Ocupa uma porta com quantômetro.

- b) Radiação líquida - idem à anterior, acumulando o total positivo, negativo e a diferença entre ambos. Ocupa uma porta com pirgeômetro.
5. Chuva - Obter o valor a cada intervalo de 15 minutos e o total pluviométrico diário. Ocupa uma porta com pluviômetro.
6. Vento (velocidade) - Acumular valores instantâneos lidos a cada minuto, obtendo a velocidade média no final de cada 15 minutos. Obter também o total acumulado diário. Ocupa uma porta com anemômetro.
7. Vento (direção) - Obter direção instantânea do vento a cada 15 minutos. Ocupa uma porta com anemômetro.

O período considerado para este experimento vai das 7:00 às 7:00 horas do dia seguinte.

A duração do experimento é de dez dias.

Os sensores utilizados são:

Semicondutor - Temperatura do ar seco e úmido.

Termopar - Temperatura do solo e folhas.

Pirgeômetro - Radiação líquida.

Quantômetro - Radiação PAR

Anemômetro - Velocidade e direção do vento.

Pluviômetro - Chuva.

Os experimentos micrometeorológicos utilizam um grande número de sensores e a frequência de amostragem geral-

mente é bem maior que em outros experimentos, de modo que pode haver a necessidade da estação de campo permanecer ligada durante toda a coleta de dados.

Para o experimento micrometeorológico apresentado aqui, programa B.3.1, foram especificados 15 sensores, 7 tarefas e 6 eventos.

é interessante notar a programação de "evt1" e "evt2", pois estes mostram o recurso disponível na LADAM que possibilita a ativação e desativação de eventos. Conforme descrito no ítem um da especificação do experimento, as temperaturas devem ser calculadas pela média de seis amostragens de 10 em 10 segundos durante o minuto precedente. Para tal a cada 14 minutos é disparada uma tarefa que ativa "evt2", que é responsável pela ativação da tarefa "calctemp". Esta tarefa acumula as temperaturas das seis amostragens e no final calcula a média e desativa "evt2".

Outra aplicação deste recurso seria em situações de erro ou advertência nas quais, por exemplo, uma temperatura ou quantidade de chuva atinge valores extremos de modo que é necessário aumentar-se a frequência de amostragens para um melhor registro dos dados sobre a ocorrência.

Para simulação foi utilizado um cartucho com apenas 8 Kbytes portanto este experimento foi encerrado por falta de espaço em cartucho, como pode ser observado no último registro (motivo de encerramento 3) da seção B.3.3, que apresenta o relatório dos dados coletados para o experimento.

Note que a finalização do experimento, para efeitos de simulação, está programada para 1 dia após o início, conforme linha 1160 do programa B.3.1. Na realidade seriam 10 dias.

No apêndice B.3.3 são apresentados somente alguns trechos do relatório dos dados adquiridos para o experimento micrometeorológico pois este é muito extenso.

4.5. Os Resultados Obtidos

No apêndice B são apresentados os programas e relatórios correspondentes aos dados obtidos em um dia de simulação para os experimentos mínimo, padrão e micrometeorológico.

Pode-se observar que cada registro de escrita corresponde a um comando "write" da programação inicial e foi decodificado pelo programa utilitário com a ajuda do dicionário de comandos de escrita gerados na primeira fase.

Na tabela 4.1 são apresentados os dados referentes aos tamanhos dos códigos intermediários e áreas de dados gerados para os experimentos. Pode-se notar que o esquema de tradução para linguagem intermediária na estação base proporcionou um código intermediário bastante compacto, possibilitando seu armazenamento no cartucho e ocupando apenas uma pequena parte de sua capacidade. Note-se que o tamanho do código intermediário poderia ainda ser consideravelmente reduzido, em uma nova versão que utilizasse um menor número de caracteres para denotar os comandos para o interpretador. Isto não foi feito na versão atual para possibilitar a interpretação humana do código intermediário, facilitando assim a depuração do sistema.

Considerando-se que tipicamente o tamanho de um código objeto executável é diversas vezes maior que o de um código intermediário para interpretação, se fosse adotado um

procedimento de compilação ao invés de tradução, o código objeto ocuparia, nos casos mais complexos, grande parte do cartucho ou nem caberia neste.

Experimento	Código Intermediário	Área de dados
Mínimo	661 bytes	162 bytes
Padrão	1736 bytes	350 bytes
Micrometeorológico	1923 bytes	6261 bytes

Tabela 4.1 - Tamanhos dos Códigos Intermediários e Dados Adquiridos

De acordo com os testes de desempenho da estação remota pode-se dizer que o tempo entre a ativação de um evento e término da execução da tarefa a este associada é de aproximadamente 0.3 segundos, considerando-se uma tarefa de porte médio, por exemplo, a tarefa "calctemp" do experimento micrometeorológico. Isto nos leva a crer que frequências de ativação de eventos da ordem de 1 Hz serão aceitáveis para tarefas deste porte.

Os testes de desempenho da estação remota foram feitos com auxílio do temporizador 8253 do I7000 [05], pois este não possui relógio de tempo real. O experimento consistia de um evento cuja última instrução da tarefa associada requiritava o término do experimento. No início da rotina ATEV o temporizador foi disparado para interromper após 1, 0.8, 0.5, 0.3,

0.1 e 0.03 segundos. A rotina de atendimento de interrupção dava um alarme assim que alcançada. Ao mesmo tempo o término do experimento, no final da tarefa provocava uma mensagem na tela. Comparando-se quais dos sinais eram observados em primeiro lugar chegou-se ao valor (0.3) acima citado.

Capítulo V

CONCLUSÕES

5. CONCLUSÕES

O trabalho desenvolvido fornece à área de meteorologia, especialmente micrometeorologia, uma maneira simples, de custo intermediário e potencialmente efetiva para a programação de experimentos e coleta de dados.

Simple porque o sistema foi desenvolvido com a preocupação de proporcionar fácil entedimento. Para tal a LADAM foi projetada para ser semelhante a uma linguagem de programação existente, bem sucedida e de uso amplo, que é o Pascal, facilitando assim a assimilação por parte dos meteorologistas. Esta foi, também, a razão para não se ter definido as primitivas da LADAM em português.

Já os comandos interativos da estação de campo foram definidos em português simples, para facilitar a interação com os observadores meteorológicos.

Apesar do sistema ainda não ter sido testado na estação remota real (que se encontra em desenvolvimento), as simulações efetuadas mostraram amplamente seu bom funcionamento, visto que experimentos reais foram programados e os dados relativos a estes foram coletados com sucesso. Note-se ainda que a depuração e simulação do módulo remoto utilizou um sistema e arquitetura semelhante à da estação remota, incluindo precisamente o mesmo microprocessador operando à mesma velocidade.

Assim foi possível inclusive avaliar o desempenho do software da estação remota. Concluiu-se que os tempos de processamento, no caso dos experimentos simulados, são perfeitamente compatíveis com as necessidades de operação em tempo

real, relativamente modestas, do campo de aplicação pretendido (frequências de amostragens máxima da ordem de 1 Hz). Obviamente, o módulo da estação base foi totalmente depurado e testado na máquina proposta para esta finalidade.

Pode-se notar que, apesar do sistema ter sido desenvolvido para meteorologia, ele é genérico para a área de aquisição de dados, pois basta que as entradas sejam alteradas para que o sistema possa processar, por exemplo, dados de processos industriais e laboratoriais. Naturalmente seria necessário avaliar a adequação da velocidade de operação do hardware e software remoto face às exigências, possivelmente mais elevadas, destas outras áreas.

Foram utilizadas as linguagens Pascal e "assembler/Z80" para programação do sistema da estação base e de campo respectivamente, mais alguns recursos da linguagem C, no que diz respeito às rotinas de ponto flutuante. Pode-se observar que o tradutor, programado em Pascal, gerou um código intermediário muito extenso (36K). Isto tornou o tempo de carga do programa muito grande, assim como deixou pouco espaço para as estruturas dinâmicas como a tabela de símbolos.

Um passo seguinte a este trabalho seria uma otimização do tradutor de modo a tentar diminuir o tamanho do código no que diz respeito à programação. Poder-se-ia ainda, programá-lo numa linguagem que ofereça a possibilidade de geração de código mais eficiente, no caso a linguagem C.

A simples tradução de Pascal para C não seria aconselhável, dado que toda linguagem de programação tem suas particularidades e para se obter um código ótimo é necessário que se aproveite o máximo de recursos adicionais que esta lingua-

gem possa vir a oferecer. Acreditamos que um estudo da eficiência das linguagens Pascal, C e "assembler", nas suas respectivas implementações disponíveis, para o desenvolvimento de software básico poderia oferecer valiosa contribuição para as pessoas que desenvolvem este tipo de projeto.

O código do tradutor, escrito em linguagem de montagem, apesar de mais trabalhoso, gerou um código (14K) considerado ótimo dado ao número de recursos introduzidos na estação de campo. O comprimento do código ainda poderia ser melhorado se as bibliotecas de ponto flutuante fossem específicas para o projeto. Outra opção seria, mantendo as bibliotecas utilizadas, incluir no código novas primitivas para tirar proveito de módulos disponíveis, atualmente não utilizados (logaritmos, funções trigonométricas, etc)

Uma facilidade que não foi implementada mas torna-se necessária são as estruturas do tipo "array", assim como funções predefinidas que facilitem a programação por parte do usuário. Outros melhoramentos seriam a implementação da área de dados do experimento na RAM local da estação de campo (ao invés de ocupar espaço no cartucho) e a compactação de zeros ou outros códigos repetitivos na área de dados coletados no cartucho.

No código intermediário, a lista de eventos poderia ser duplamente encadeada gastando mais espaço em cartucho, porém facilitando o percurso na lista pois as alterações são muito frequentes devido à manutenção da ordem cronológica dos eventos.

A situação atual do mercado de microcomputadores, onde se evidencia a substituição dos microcomputadores de oito

bits pelos da linha PC, torna aconselhável o desenvolvimento de uma versão que utilize o PC como estação base. Teria que ser desenvolvido para esta nova estação base, um acessório que permitisse a leitura e gravação do cartucho no PC. Este projeto não apresenta maiores dificuldades do ponto de vista de hardware. O software da estação base, tendo sido desenvolvido em uma linguagem de alto nível, seria facilmente portado para o PC.

O sistema desenvolvido pode também servir como gerador de programas em cartucho de EPROM para estações que trabalham com coleta de dados através de programação prefixada e utilizam outra forma de gravação dos dados (envio por teleprocessamento, por exemplo). Obviamente, para esta finalidade seria utilizado um acessório gravador de EPROM na estação base.

Enxergando a LADAM no sistema MDAS [11], apresentado no capítulo 2, esta poderia ser a ferramenta para programação de experimentos no nível de aplicação.

BIBLIOGRAFIA

BIBLIOGRAFIA

- [01] Koeppe, C. E. & Long, G. C.
Weather and Climate
McGraw-Hill Book Company - 1958.
- [02] Mather, J. R.
Climatology Fundamentals and Applications
McGraw-Hill, Inc. 1974
- [03] Geiger, R.
Manual de Microclimatologia.
Friedr. Vieweg & Sohn Verlag - 1961.
- [04] Centoducate, P. C., Um Sistema de Aquisição de Dados Meteorológicos - Hardware. Tese de Mestrado em preparação.
Departamento de Ciência da Computação IMECC - UNICAMP
- [05] Manual de Assembler I-7904
Itaú Tecnologia S.A. - 1983
- [06] Campbell Scientific, Inc.
Introducing to CR21 Micrologger.
July, 1983
- [07] Haro, H.
Precipitation Instrumentation Package for Improved Spatial and Temporal Sampling of Rainfall.
IEEE Transactions on Instrumentation and Measurement.
vol. IM-32, Nº 3. September 1983, pp. 423-429.
- [08] SYSTRON DONNER. Weathermeasure Division.
1981 Catalog. Sacramento - USA.
- [09] Setzer, A. W. & Carvalho, P. R.
Sistema Automático de Aquisição de Dados Meteorológicos Portátil e de Baixo Custo com Microcomputador.
Catálogo de Protótipos da FAPESP - 1986

- [10] Arlino, P. R. A. & Nogueira, J. L. M. & Inoue, J. K.
Sistema Automático de Aquisição de Dados com utilização
de microcomputadores: Estação Automática (EMA)
INPE - Departamento de Meteorologia - 1987
- [11] Merat, F. L. & Gibbons, J. C. & Simons R. & Podany M.
A Meteorological Data Acquisition System
Cleveland Electrical/Electronics Conference and Exposition
OH, USA. 4-5 October 1983
New York, USA : IEEE november 1983, pp. 73-78
- [12] Gimenes, I. M. S. & Centoducate, P. C.
Um Sistema de Aquisição de Dados Baseado em Microproces-
sador
Anais da Primeira Semana de Informática da UFBA.
Março 1986, pp. 97-101.
- [13] Young, S. J.
Real Time Language. Design and Development.
Ellis Horwood Series in Computers and their Applications,
West Sussex - England, 1983
- [14] Pratt, T. W.
Programming Languages. Design and Implementation.
Prentice-Hall, 1975
- [15] Digital Equipment Corporation
PDP-11 PEARL. Language and Reference Manual.
Version 1.0, September 1980.
- [16] Tarouco, L. M. R.
Redes de Comunicação de Dados
Livros Técnicos e Científicos editora
Rio de Janeiro - RJ, 1977
- [17] Kowaltowski, T.
Implementação de linguagens de Programação.
Editora Guanabara Dois S.A.
Rio de Janeiro - RJ, 1983.

Apêndice A

DESCRIÇÃO DA SINTAXE DA LADAM

A.1. Elementos da LADAM

Os elementos que constituem a LADAM podem ser descritos como se segue:

1. Letras : a, b, c, z
2. Dígitos: 0, 1, 2, 9
3. Símbolos especiais : ., ,, ;, [,], :, =, <, >, +, -, *,
(,)
4. Símbolos especiais compostos : :=, /*, */, <=, >=, <>
5. Palavras reservadas :
 program, assign, var, task, endtk, if, then, else,
 endif, do, endo, terminate, at, every, within, read,
 write, port, event section, endevt, hs, min, seg, or.
6. Comentários : entre /* e */

A.2. Síntaxe da LADAM

1. <program> ::= program <identificador>;<bloco>.
2. <bloco> ::= <parte de declaração do ambiente>
 [<parte de declaração das variáveis>]
 <parte de declaração das tarefas>
 <parte de declaração dos eventos>

PARTE DE DECLARAÇÕES

3. <parte de declaração do ambiente> ::=
 assign <declaração do sensor> (;<declaração do sensor>);
4. <declaração do sensor> ::= <código do sensor> port
 <número da porta> <variável externa> (;<variável externa>)

5. <código do sensor> ::= <número>[<letra>]
6. <variável externa> ::= <opção de leitura>;<identificador>
7. <parte da declaração das variáveis> ::=
 var <declaração das variáveis> {,<declaração das variáveis>}
8. <declaração das variáveis> ::= <lista de identificadores>
 ; <tipo> [(<valor inicial>)]
9. <tipo> ::= integer | real | bytetring [<número>] | time |
 date
10. <valor inicial> ::= <constante numérica>
11. <parte de declaração das tarefas > ::=
 <declaração de tarefa> { ; <declaração de tarefa> }
12. <declaração de tarefa> ::= task <nome de tarefa>
 <comandos sequenciais> { ; <comandos sequenciais> }
 endtk
13. <parte de declaração dos eventos> ::=
 event section <declaração de eventos> { ; <declaração de
 eventos> } endevt
14. <declaração de eventos> ::=
 [<nome de evento> [(0 | 1)]] ;]
 <Comando de Ativação Programada de Tarefas>

COMANDOS SEQUENCIAIS

15. <comando sequenciais> ::= <comando de entrada> |
 <comando de atribuição> | <comando de saída> |
 <comando condicional> |
 <comando para ativação de evento> |
 <comando para desativação de evento> |
 <comando de encerramento do experimento>

16. <comando de entrada> ::= read([<dispositivo de entrada>],
 <lista de identificadores>)
17. <comando de escrita> ::= write([<dispositivo de saída>],
 <sequência de escrita> (,<sequência de escrita>))
18. <sequência de escrita> ::=
 <string de caracteres> | <lista de identificadores>
19. <comando de atribuição> ::= <identificador> := <expressão>
20. <comando condicional> ::= if <expressão> then
 <comandos sequenciais> (;<comandos sequenciais>) else
 <comandos sequenciais> (;<comandos sequenciais>) endif
21. <comando para ativação de evento> ::=
 activate <nome de evento>
22. <comando para desativação de evento> ::= terminate
 <nome de evento>
23. <comando de encerramento do experimento> ::= trailer

COMANDOS PARA ATIVAÇÃO PROGRAMADA DE TAREFAS

24. <CAPT> ::= <CAPT por horário fixo> |
 <CAPT por frequência >
25. <CAPT por horário fixo> ::= at <horário> (,<horário>)
 do <nome de tarefa> (,<nome de tarefa>) endo
26. <CAPT por frequência> ::= every <frequência>
 [within <intervalo> (or <intervalo>)]
 do <nome de tarefa> (,<nome de tarefa>) endo

EXPRESSÕES

27. <expressão> ::= <expressão simples> [<relação>
 <expressão simples>]
28. <relação> ::= <> | = | < | > | <= | >=
29. <expressão simples> ::= [-] <termo> ((+|-) <termo>)
30. <termo> ::= <fator> ((*|/) <fator>)
31. <fator> ::= <identificador> |
 <constante numérica> | <string de caracteres>
 <chamada de conversão> |
 (<expressão>)

OUTROS

32. <constante numérica> ::= <constante rel> | <número> |
 <horário> | <data>
33. <chamada de conversão> ::= <função de conversão>
 (<identificador> | <constante numérica>)
34. <função de conversão> ::= integer | real
35. <constante real> ::= <número>, <número>
36. <data> ::= <dia> / <mês> / <ano>
37. <dia> ::= 1 | 2 | 3 | | 31
38. <mês> ::= 1 | 2 | 3 | 4 | | 12
39. <ano> ::= <dígito> <dígito>
40. <horário> ::= <hora>; <minuto>[:<segundo>]
41. <hora> ::= 00 | 01 | 02 | | 23
42. <minuto> ::= 00 | 01 | 02 | | 59
43. <segundo> ::= 00 | 01 | 02 | | 59
44. <frequência> ::= <hora> h_h | <minuto> min | <segundo> seg

45. <intervalo> ::= [<horário>,<horário>]
46. <lista de identificadores> ::= <identificador>
{,<identificador>}
47. <nome de tarefa> ::= <identificador>
48. <nome de evento> ::= <identificador>
49. <opção de leitura> ::= <número>
50. <identificador> ::= <letra> (<letra> | <dígito>)
51. <letra> ::= a | b | c | | z
52. <número> ::= <dígito> (<dígito>)
53. <dígito> ::= 0 | 1 | 2 | | 9
54. <dispositivo de entrada> ::= en | es | ck
55. <dispositivo de saída> ::= ch | es
56. <string de caracteres> ::= '(character)' max = 40

Apêndice B

**PROGRAMAS E RELATÓRIOS DOS
EXPERIMENTOS SIMULADOS**

B.1. Programação e Relatórios para o Experimento Mínimo

B.1.1 Programação do Experimento Mínimo em LADAN

```

10 program expmin;
20 /* programacao do posto meteorologico padrao minimo */
30 assign
40     01A     port     1     0:tempar: /* semicondutor */
50     08A     port     2     0:qtdchuva: /* pluviometro */
60 /* ..... declaracao das variaveis ..... */
70 var
80     tempmin, tempmax : real;
90     datatual, datmin, datmax : date;
100    horatual, hormin, hormax : time;
110    chuvamax, chuvant, qtdchuvai : integer;
120    totpluv : real;
130 /* tarefa para inicializacao do experimento */
140 task header
150     read(ck,datatual,horatual);
160     tempmax := real(minvalue);
170     tempmin := real(maxvalue);
180     chuvamax := 4096; /* maximo valor obtido do pluviometro */
190     read(sn,qtdchuva);
200     chuvant := integer(qtdchuva);
210 endtk;
220 /* varredura dos sensores nos horarios padroes 7,9,14,15 e 21hs */
230 task varredura
240     read(sn,tempar,qtdchuva);
250     if integer(qtdchuva) - chuvant > 0
260     then
270         qtdchuvai := integer(qtdchuva) - chuvant
280     else
290         qtdchuvai := chuvamax - chuvant + integer(qtdchuva);
300     endif;
310     chuvant := qtdchuvai;
320     totpluv := totpluv + real(qtdchuvai);
330     write(ch,tempar,qtdchuvai); /* registro de escrita 3 */
340 endtk;
350 /* avaliacao horaria da temperatura maxima e minima */
360 task avaltemp
370     read(sn,tempar);
380     if real(tempar) < tempmin
390     then
400         read(ck,datmin,hormin);
410         tempmin := real(tempar);
420     endif;
430     if real(tempar) > tempmax
440     then
450         read(ck,datmax,hormax);
460         tempmax := real(tempar);
470     endif;
480 endtk;
490 /* gravacao da temperatura maxima e minima referente ao periodo */
500 task gravperiod
510 /* registro de escrita 4 */
520     write(ch,datmin,hormin,tempmin,datmax,hormax,tempmax,totpluv);
530     tempmin := real(maxvalue);
540     tempmax := real(minvalue);

```

```
550 /* testa final de espaco em cartucho */
560 if memavail < 64 then trailer endif;
570 endtk;
580 /* ..... declaracao dos eventos ..... */
590 event section
600 evt1: at 7:00, 9:00, 14:00,15:00, 21:00 do varredura endo;
610 evt2: every 01 hs do avaltemp endo;
620 evt3: at 7:00 do gravperiod endo;
630 endevt.
    ZERO ERROS DETECTADOS
    FIM DA COMPILACAO
```

B.1.2. Relatório do Código Intermediário para o Experimento Mínimo

1. TABELA DE ENDEREÇOS BASE

OBJETO	ENDEREÇO	TAMANHO
DESCRITOR DOS SENSORES	8020	82
DESCRITOR DO AMBIENTE	8073	36
AREA DE DADOS	8098	70
CODIGO DAS TAREFAS	80DF	347
ENDEREÇO DAS CONVERSOES	823B	0
LISTA DE EVENTOS	823B	94
POSICAO LIVRE NO CARTUCHO	829A	15718
ULTIMA POSICAO GRV. DADOS	0000	0

2. DESCRITOR DOS SENSORES

```
=====
CODIGO = 1A      SINAL = A      MED. MINIMA = -20.0000
MED. MAXIMA = 160.0000      PRECISAO = 0.2000
-----
```

OPCOES DE LEITURA

```
-----
OPCAO = 0      PROX. OPCAO = 0
```

1. CONVERSAO PARA INTEIRO

```
ROTINA = 2      NUM. PARMS = 2      COMPRIMENTO = 8
VALORES = 0000008000000000
```

2. CONVERSAO PARA REAL

```
ROTINA = 1      NUM. PARMS = 2      COMPRIMENTO = 8
VALORES = 0000008000000000
```

```
=====
CODIGO = 8A      SINAL = D      MED. MINIMA = 0.0000
MED. MAXIMA = 100.0000      PRECISAO = 1.0000
-----
```

OPCOES DE LEITURA

```
-----
OPCAO = 0      PROX. OPCAO = 0
```

1. CONVERSAO PARA INTEIRO

```
ROTINA = 2      NUM. PARMS = 2      COMPRIMENTO = 8
VALORES = 0000008100000000
```

2. CONVERSAO PARA REAL

```
ROTINA = 1      NUM. PARMS = 2      COMPRIMENTO = 8
VALORES = 0000008100000000
```

LRC = FD

3. DESCRITOR DO AMBIENTE

 VARIABEL EXTERNA = tempar DESL. SENSOR = 0 NUMERO SENSOR = 0

PORTA LOGICA = 1 OPCAO DE LEITURA = 0 COMPR. VARIABEL = 2
 VALOR = 0000

 VARIABEL EXTERNA = qtdchuva DESL. SENSOR = 41 NUMERO SENSOR = 0

PORTA LOGICA = 2 OPCAO DE LEITURA = 0 COMPR. VARIABEL = 2
 VALOR = 0000
 LRC = 61

4. AREA DE DADOS

I2=32767 I2=-32767 T=23:59:59 T=00:00:00 D=00/00/00 T=00:00:00
 R4=0.0000000 R4=0.0000000 D=00/00/00 D=00/00/00 D=00/00/00
 T=00:00:00 T=00:00:00 T=00:00:00 I2=0 I2=0 I2=0
 R4=0.0000000

7. CODIGO DAS TAREFAS

TAREFA 1(80DF-811F)

RD ck 43 55

AB 38 CVRV14

AB 33 CVRV11

AB 67 CI4096

RD sn 29

AB 70 CVIX29 LRC = 85

TAREFA 2(8121-8194)

RD sn 11 29

IF CVIX29 V 70 - C10 >
 THEN

AB 73 CVIX29 V 70 -
 ELSE

AB 73 V 67 V 70 - CVIX29 +

AB 70 V 73

AB 76 V 76 CVRV73 +

WR ch

RELACAO DOS PARAMETROS

3 X 11 V 73

LRC = 20

TAREFA 3(8196-81EF)

RD sn 11

IF CVRX11 V 33 <
THEN

RD ck 47 59

AB 33 CVRX11

IF CVRX11 V 38 >
THEN

RD ck 51 63

AB 38 CVRX11

LRC = 5C

TAREFA 4(81F1-8238)

WR ch

RELACAO DOS PARAMETROS

4 V 47 V 59 V 33 V 51 V 63 V 38 V 76

AB 33 CVRV11

AB 38 CVRV14

IF FPD 1 C164 <
THEN

ED FIM DO EXPERIMENTO

LRC = 2E

LRC GLOBAL DAS TAREFAS = D7

6. ENDERECO DAS CONVERSOES

TABELA DE ROTINAS DE CONVERSAO VAZIA

8. LISTA DE EVENTOS

EVENTO 1

HH= 00:00:00 MD= 20 PRX= 8245 NCDGS= 1 CDG 1=80DF

EVENTO 2

HH= 00:00:00 MD= 00 PRX= 824F

EVENTO 4

HH= 00:00:00 MD= A1 PRX= 825D FQ= 1H
 NINT= 0 INTCORR=0
 NCDGS= 1 CDG 1=8196

EVENTO 3

HH= 07:00:00 MD= 60 PRX= 8267 NCDGS= 1 CDG 1=8121

EVENTO 5

HH= 07:00:00 MD= 60 PRX= 8271 NCDGS= 1 CDG 1=81F1

EVENTO 3

HH= 09:00:00 MD= 60 PRX= 827B NCDGS= 1 CDG 1=8121

EVENTO 3

HH= 14:00:00 MD= 60 PRX= 8285 NCDGS= 1 CDG 1=8121

EVENTO 3

HH= 15:00:00 MD= 60 PRX= 828F NCDGS= 1 CDG 1=8121

EVENTO 3

HH= 21:00:00 MD= 60 PRX= 8245 NCDGS= 1 CDG 1=8121

LRC= 8E

8. POSICAO LIVRE NO CARTUCHO

INICIO= 829A TAMANHO= 15718

B.1.3. Relatório dos Dados Adquiridos para o Experimento Mínimo

RELATORIO DOS DADOS ADQUIRIDOS NA ESTACAO DE CAMPO

1. REGISTRO DE ESCRITA : 1 TAMANHO = 37

D=14/12/86 T=00:00:02 D=14/12/86 T=06:00:00 S[10]=Maringa
S[10]=Itana
LRC= 2C

2. REGISTRO DE ESCRITA : 3 TAMANHO = 9

BI 2]=002A I2=0
LRC= 20

3. REGISTRO DE ESCRITA : 4 TAMANHO = 29

D=14/12/86 T=06:00:02 R4=20.5000000 D=14/12/86 T=06:00:03
R4=20.5000000 R4=0.0000000
LRC= 1B

4. REGISTRO DE ESCRITA : 3 TAMANHO = 9

BI 2]=002B I2=0
LRC= 21

5. REGISTRO DE ESCRITA : 3 TAMANHO = 9

BI 2]=0035 I2=0
LRC= 3F

6. REGISTRO DE ESCRITA : 3 TAMANHO = 9

BI 2]=0034 I2=0
LRC= 3E

7. REGISTRO DE ESCRITA : 3 TAMANHO = 9

BI 2]=002E I2=3
LRC= 27

8. REGISTRO DE ESCRITA : 3 TAMANHO = 9

BI 2]=002A I2=3
LRC= 23

9. REGISTRO DE ESCRITA : 4 TAMANHO = 29

D=15/12/86 T=06:00:02 R4=20.5000000 D=14/12/86 T=13:00:02
R4=27.0000000 R4=6.0000000

LRC= B2

10. REGISTRO DE ESCRITA : 2 TAMANHO = 14

D=15/12/86 T=07:00:07 I2=162 B1 11=01

LRC= 2E

B.2. Programação e Relatórios para o Experimento Padrão

B.2.1 Programação do Experimento Padrão em LADAM

```

10 program expmed;
20 /* programacao da coleta de dados para o posto meteorológico padrao */
30 assign
40     01A     port     1     0:tempars; /* semicondutor */
50     01B     port     2     0:temprelv; /* semicondutor */
60     03A     port     3     0:umrel; /* higrometro */
70     04A     port     4     0:radglob; /* piranometro */
80     07A     port     5     0:insol; /* heliografo */
90     08A     port     6     0:qtdchuva; /* pluviometro */
100    10A     port     7     0:vento; /* anemometro - velocidade */
110    10B     port     8     0:dirvento; /* anemometro - direcao */
120 /* ..... declaracao das variaveis ..... */
130 var
140    datatual, datmin, datmax, ddrelmin : date;
150    horatual, hormin, hormax, hhrelmin : time;
160    tempmax, tempmin, relvmin, t7, t14, t21, tm : real;
170    radglobt, minsol, hhsol, velvento, ventot, totpluv : real;
180    chuvamax, chuvant, qtdchuvai : integer;
190 /* tarefa de inicializacao do experimento */
200 task header
210    tempmax := real(minvalue);
220    tempmin := real(maxvalue);
230    relvmin := real(maxvalue);
240    read(ck, datatual, horatual);
250    chuvamax := 4096; /* maximo valor obtido do pluviometro */
260    read(sn, qtdchuva);
270    chuvant := integer(qtdchuva);
280 endtk;
290 /* tarefa de finalizacao do experimento */
300 task trailer
310    read(ck, datatual, horatual);
320 /* registro de escrita 3 */
330    write(ch, 'fim do experimento', datatual, horatual);
340 endtk;
350 /* varredura completa dos sensores nos horarios padroes */
360 task varredura
370    read(sn, tempars, umrel, radglob, insol, qtdchuva, vento, dirvento);
380    if integer(qtdchuva) - chuvant > 0
390    then
400        qtdchuvai := integer(qtdchuva) - chuvant
410    else
420        qtdchuvai := chuvamax - chuvant + integer(qtdchuva);
430    endif;
440    chuvant := qtdchuvai;
450    totpluv := totpluv + real(qtdchuvai);
460 /* guarda temperaturas para calculo da media */
470    if horaref = 7:00
480    then
490        t7 := real(tempars)
500    else
510        if horaref = 14:00
520        then
530            t14 := real(tempars)
540        else

```

```

550     if horaref = 21:00
560     then
570         t21 := real(tempars);
580     endif;
590     endif;
600 endif;
610 /* registro de escrita 4 */
620 write(ch,tempars,umrel,radglob,insol,qtchuva,vento,dirvento);
630 endtk;
640 /* avalia temperatura maxima e minima do ar seco */
650 task avaltemp
660 read(sn,tempars);
670 if real(tempars) < tempmin
680 then
690     read(ck,datmin,hormin);
700     tempmin := real(tempars);
710 endif;
720 if real(tempars) > tempmax
730 then
740     read(ck,datmax,hormax);
750     tempmax := real(tempars);
760 endif;
770 endtk;
780 /* avalia temp. minima da relva no periodo de 21:00 as 7:00 horas */
790 task avalrelva
800 read(sn,temprelv);
810 if real(temprelv) < relvmin
820 then
830     read(ck,ddrelmin,hhrelmin);
840     relvmin := real(temprelv);
850 endif;
860 endtk;
870 /* acumula valores instantaneos da radiacao solar */
880 task acumrad
890 read(sn,radglob);
900 radglobt := radglobt + real(radglob);
910 endtk;
920 /* gravacao de valores acumulados */
930 task gravperiod
940 tm := (t7 + t14 + 2.0*t21)/4.0;
950 /* registro de escrita 5 */
960 write(ch,datmin,hormin,tempmin,datmax,hormax,tempmax,tm)
970 hhsol := minsol/60.0
980 /* registro de escrita 6 */
990 write(ch,totpluv,radglobt,hhsol,ddrelmin,hhrelmin,relvmin);
1000 radglobt := 0.0;
1010 minsol := 0.0;
1020 relvmin := real(maxvalue);
1030 tempmin := real(maxvalue);
1040 tempmax := real(minvalue);
1050 if dataref = 25/12/86
1060 then
1070     if horaref = 7:00 then trailler endif;
1080 endif;
1090 endtk;
1100 /* acumula momentos de sol com frequencia de um minuto */
1110 task acumsol
1120 read(sn,insol);

```

```

1130  if real(insol) > 0.1
1140  then
1150      minsol := minsol + 1.0
1160  endif;
1170 endtk;
1180 /* acumula voltas do sensor de vento para calculo da velocidade */
1190 task acumvento
1200  read(sn,vento);
1210  ventot := ventot + real(vento);
1220 endtk;
1230 /* calcula velocidade do vento */
1240 task calcvento
1250  velvento := (ventot/11.0)*0.06;
1260  ventot := 0.0;
1270  /* registro de escrita 7 */
1280  write(ch,velvento);
1290 endtk;
1300 /* ..... declaracao do eventos ..... */
1310 event section
1320  evt1 : at 07:00, 9:00, 14:00, 15:00, 21:00 do varredura endo;
1330  evt2 : every 01 hs do avaltemp endo;
1340  evt3 : at 7:00 do gravperiod endo;
1350  evt4 : every 01 min within [21:00,23:59:59] or [00:00,7:00] do
avalrelva endo;
1360  evt5 : every 01 min within [5:30,18:30] do acumrad, acumsol endo;
1370  evt6 : every 01 min within [6:55,7:05] or
1380          [8:55,9:05] or
1390          [13:55,14:05] or
1400          [14:55,15:05] or
1410          [20:55,21:05] do acumvento endo;
1420  evt10 : at 7:06, 9:06, 14:06, 15:06, 21:06 do calcvento endo;
1430 endevt.
      ZERO ERROS DETECTADOS
      FIN DA COMPILACAO

```

B.2.2. Relatório do Código Intermediário para o Experimento Padrão

1. TABELA DE ENDEREÇOS BASE

OBJETO	ENDEREÇO	TAMANHO
DESCRIPTOR DOS SENSORES	8020	328
DESCRIPTOR DO AMBIENTE	8169	144
AREA DE DADOS	81FA	128
CODIGO DAS TAREFAS	827B	868
ENDEREÇO DAS CONVERSOES	85E0	0
LISTA DE EVENTOS	85E0	236
POSICAO LIVRE NO CARTUCHO	86CD	14643
ULTIMA POSICAO GRV. DADOS	0000	0

2. DESCRIPTOR DOS SENSORES

```
=====
CODIGO = 1A      SINAL = A      MED. MINIMA = -20.0000
MED. MAXIMA = 160.0000      PRECISAO = 0.2000
-----
```

OPCOES DE LEITURA

```
-----
OPCAO = 0      PROX. OPCAO = 0
-----
```

```
1. CONVERSAO PARA INTEIRO
ROTINA = 2      NUM. PARMS = 2      COMPRIMENTO = 8
VALORES = 0000008000000000
```

```
2. CONVERSAO PARA REAL
ROTINA = 1      NUM. PARMS = 2      COMPRIMENTO = 8
VALORES = 0000008000000000
```

```
=====
CODIGO = 1B      SINAL = A      MED. MINIMA = -30.0000
MED. MAXIMA = 160.0000      PRECISAO = 0.2000
-----
```

OPCOES DE LEITURA

```
-----
OPCAO = 0      PROX. OPCAO = 0
-----
```

```
1. CONVERSAO PARA INTEIRO
ROTINA = 2      NUM. PARMS = 2      COMPRIMENTO = 8
VALORES = 0000008000000000
```

```
2. CONVERSAO PARA REAL
ROTINA = 1      NUM. PARMS = 2      COMPRIMENTO = 8
VALORES = 0000008000000000
```

```
=====
CODIGO = 3A      SINAL = A      MED. MINIMA = 0.0000
MED. MAXIMA = 100.0000      PRECISAO = 1.0000
-----
```

OPCOES DE LEITURA

```
-----
OPCAO = 0      PROX. OPCAO = 0
-----
```


1. CONVERSAO PARA INTEIRO
 ROTINA = 2 NUM. PARMS = 2 COMPRIMENTO = 8
 VALORES = 0000008100000000

2. CONVERSAO PARA REAL
 ROTINA = 1 NUM. PARMS = 2 COMPRIMENTO = 8
 VALORES = 0000008100000000

=====

CODIGO = 4A SINAL = A MED. MINIMA = 0.0000
 MED. MAXIMA = 100.0000 PRECISAO = 1.0000

OPCOES DE LEITURA

OPCAO = 0 PROX. OPCAO = 0

1. CONVERSAO PARA INTEIRO
 ROTINA = 2 NUM. PARMS = 2 COMPRIMENTO = 8
 VALORES = DE02097E00000000

2. CONVERSAO PARA REAL
 ROTINA = 1 NUM. PARMS = 2 COMPRIMENTO = 8
 VALORES = DE02097E00000000

=====

CODIGO = 7A SINAL = A MED. MINIMA = 0.0000
 MED. MAXIMA = 100.0000 PRECISAO = 1.0000

OPCOES DE LEITURA

OPCAO = 0 PROX. OPCAO = 0

1. CONVERSAO PARA INTEIRO
 ROTINA = 2 NUM. PARMS = 2 COMPRIMENTO = 8
 VALORES = 0000608400000000

2. CONVERSAO PARA REAL
 ROTINA = 1 NUM. PARMS = 2 COMPRIMENTO = 8
 VALORES = 0000608400000000

=====

CODIGO = 8A SINAL = D MED. MINIMA = 0.0000
 MED. MAXIMA = 100.0000 PRECISAO = 1.0000

OPCOES DE LEITURA

OPCAO = 0 PROX. OPCAO = 0

1. CONVERSAO PARA INTEIRO
 ROTINA = 2 NUM. PARMS = 2 COMPRIMENTO = 8
 VALORES = 0000008100000000

2. CONVERSAO PARA REAL
 ROTINA = 1 NUM. PARMS = 2 COMPRIMENTO = 8
 VALORES = 0000008100000000

```
=====
CODIGO = 10A      SINAL = A      MED. MINIMA = 0.0000
MED. MAXIMA = 120000.0000      PRECISAO = 1.0000
-----
```

OPCOES DE LEITURA

```
-----
OPCAO = 0      PROX. OPCAO = 0
```

```
1. CONVERSÃO PARA INTEIRO
ROTINA = 2      NUM. PARMS = 2      COMPRIMENTO = 8
VALORES = 0000008100000000
```

```
2. CONVERSÃO PARA REAL
ROTINA = 1      NUM. PARMS = 2      COMPRIMENTO = 8
VALORES = 0000008100000000
```

```
=====
CODIGO = 10B      SINAL = A      MED. MINIMA = 1.0000
MED. MAXIMA = 8.0000      PRECISAO = 1.0000
-----
```

OPCOES DE LEITURA

```
-----
OPCAO = 0      PROX. OPCAO = 0
```

```
1. CONVERSÃO PARA INTEIRO
ROTINA = 2      NUM. PARMS = 2      COMPRIMENTO = 8
VALORES = 0000008100000000
```

```
2. CONVERSÃO PARA REAL
ROTINA = 1      NUM. PARMS = 2      COMPRIMENTO = 8
VALORES = 0000008100000000
```

LRC = C3

3. DESCRITOR DO AMBIENTE

```
-----
VARIÁVEL EXTERNA = tempars      DESL. SENSOR = 0      NUMERO SENSOR = 0
```

```
PORTA LÓGICA = 1      OPÇÃO DE LEITURA = 0      COMPR. VARIÁVEL = 2
VALOR = 0000
```

```
-----
VARIÁVEL EXTERNA = temprelv      DESL. SENSOR = 41      NUMERO SENSOR = 0
```

```
PORTA LÓGICA = 2      OPÇÃO DE LEITURA = 0      COMPR. VARIÁVEL = 2
VALOR = 0000
```

```
-----
VARIÁVEL EXTERNA = umrei      DESL. SENSOR = 82      NUMERO SENSOR = 0
```

```
PORTA LÓGICA = 3      OPÇÃO DE LEITURA = 0      COMPR. VARIÁVEL = 2
VALOR = 0000
```

```
-----
VARIÁVEL EXTERNA = radglob      DESL. SENSOR = 123      NUMERO SENSOR = 0
```

```
PORTA LÓGICA = 4      OPÇÃO DE LEITURA = 0      COMPR. VARIÁVEL = 2
VALOR = 0000
```

 VARIABEL EXTERNA = insol DESL. SENSOR = 164 NUMERO SENSOR = 0

PORTA LOGICA = 5 OPCAO DE LEITURA = 0 COMPR. VARIABEL = 2
 VALOR = 0000

 VARIABEL EXTERNA = qtdchuva DESL. SENSOR = 205 NUMERO SENSOR = 0

PORTA LOGICA = 6 OPCAO DE LEITURA = 0 COMPR. VARIABEL = 2
 VALOR = 0000

 VARIABEL EXTERNA = vento DESL. SENSOR = 246 NUMERO SENSOR = 0

PORTA LOGICA = 7 OPCAO DE LEITURA = 0 COMPR. VARIABEL = 2
 VALOR = 0000

 VARIABEL EXTERNA = dirvento DESL. SENSOR = 287 NUMERO SENSOR = 0

PORTA LOGICA = 8 OPCAO DE LEITURA = 0 COMPR. VARIABEL = 2
 VALOR = 0000
 LRC = 214

4. AREA DE DADOS

I2=32767 I2=-32767 T=23:59:59 T=00:00:00 D=00/00/00 T=00:00:00
 D=00/00/00 D=00/00/00 D=00/00/00 D=00/00/00 T=00:00:00 T=00:00:00
 T=00:00:00 T=00:00:00 R4=0.0000000 R4=0.0000000
 R4=0.0000000 R4=0.0000000 R4=0.0000000
 R4=0.0000000 R4=0.0000000 R4=0.0000000
 R4=0.0000000 R4=0.0000000 R4=0.0000000
 R4=0.0000000 R4=0.0000000 I2=0 I2=0 I2=0

7. CODIGO DAS TAREFAS

TAREFA 1(827B-82C6)

AB 65 CVRV14

AB 70 CVRV11

AB 75 CVRV11

RD ck 33 49

AB 130 CI4096

RD sn 101

AB 133 CVIX101

LRC = 13

TAREFA 2(82C8-82F5)

RD ck 33 49

UR ch

RELACAO DOS PARAMETROS

3 (fim do experimento) V 33 V 49

LRC = 64

TAREFA 3(B2F7-83E2)

RD sn 11 47 65 83 101 119 137

IF CVIX101 V133 - C10 >

THEN

AB 136 CVIX101 V133 -

ELSE

AB 136 V130 V133 - CVIX101 +

AB 133 V136

AB 125 V125 CVRV136 +

IF V 29 CT07:00:00 =

THEN

AB 80 CVRX11

ELSE

IF V 29 CT14:00:00 =

THEN

AB 85 CVRX11

ELSE

IF V 29 CT21:00:00 =

THEN

AB 90 CVRX11

UR ch

RELACAO DOS PARAMETROS

4 X 11 X 47 X 65 X 83 X101 X119 X137

LRC = D4

TAREFA 4(83E4-843D)

RD sn 11

IF CVRX11 V 70 <
THEN

RD ck 37 53

AB 70 CVRX11

IF CVRX11 V 65 >
THEN

RD ck 41 57

AB 65 CVRX11 LRC = 93

TAREFA 5(843F-8471)

RD sn 29

IF CVRX29 V 75 <
THEN

RD ck 45 61

AB 75 CVRX29 LRC = 2C

TAREFA 6(8473-848E)

RD sn 65

AB 100 V100 CVRX65 + LRC = 2E

TAREFA 7(8490-855A)

AB 95 V 80 V 85 + CR 2.00000 V 90 * + CR 4.00000 /

WR ch

RELACAO DOS PARAMETROS

5 V 37 V 53 V 70 V 41 V 57 V 65 V 95

AB 110 V105 CR 60.00000 /

WR ch

RELACAO DOS PARAMETROS

6 V125 V100 V110 V 45 V 61 V 75

AB 100 CR 0.00000

AB 105 CR 0.00000

AB 75 CVRV11

AB 70 CVRV11

AB 65 CVRV14

IF V 25 CD25/12/86 =
THEN

IF V 29 CT07:00:00 =
THEN

ED FIM DO EXPERIMENTO LRC = 8F

TAREFA 8(855C-858D)

RD sn 83

IF CVRX83 CR 0.10000 >
THEN

AB 105 V105 CR 1.00000 + LRC = A5

TAREFA 9(858F-85AA)

RD sn 119

AB 120 V120 CVRX119 + LRC = 0B

TAREFA 10(85AC-85DD)

AB 115 V120 CR 11.00000 / CR 0.06000 *

AB 120 CR 0.00000

WR ch
RELACAO DOS PARAMETROS
7 V115 LRC = AO

LRC GLOBAL DAS TAREFAS = B3

6. ENDERECO DAS CONVERSOES

TABELA DE ROTINAS DE CONVERSAO VAZIA

8. LISTA DE EVENTOS

EVENTO 1

HH= 00:00:00 MD= 20 PRX= 85EA NCDGS= 1 CDG 1=827B

EVENTO 2

HH= 00:00:00 MD= E0 PRX= 85F4 NCDGS= 1 CDG 1=82C8

EVENTO 4

HH= 00:00:00 MD= A1 PRX= 8602 FQ= 1H
 NINT= 0 INTCORR=0
 NCDGS= 1 CDG 1=83E4

EVENTO 6

HH= 00:00:00 MD= A1 PRX= 861C FQ= 1M
 NINT= 2 INTCORR=1
 HH1=21:00:00 HH2=23:59:59
 HH1=00:00:00 HH2=07:00:00
 NCDGS= 1 CDG 1=843F

EVENTO 7

HH= 00:00:00 MD= A1 PRX= 8632 FQ= 1M
 NINT= 1 INTCORR=1
 HH1=05:30:00 HH2=18:30:00
 NCDGS= 2 CDG 1=8473CDG 2=855C

EVENTO 8

HH= 00:00:00 MD= A1 PRX= 865E FQ= 1M
 NINT= 5 INTCORR=1
 HH1=06:55:00 HH2=07:05:00
 HH1=08:55:00 HH2=09:05:00
 HH1=13:55:00 HH2=14:05:00
 HH1=14:55:00 HH2=15:05:00
 HH1=20:55:00 HH2=21:05:00
 NCDGS= 1 CDG 1=858F

EVENTO 3

HH= 07:00:00 MD= 60 PRX= 8668 NCDGS= 1 CDG 1=82F7

EVENTO 5

HH= 07:00:00 MD= 60 PRX= 8672 NCDGS= 1 CDG 1=8490

EVENTO 9

HH= 07:06:00 MD= 60 PRX= 867C NCDGS= 1 CDG 1=85AC

EVENTO 3

HH= 09:00:00 MD= 60 PRX= 8686 NCDGS= 1 CDG 1=82F7

EVENTO 9

HH= 09:06:00 MD= 60 PRX= 8690 NCDGS= 1 CDG 1=85AC

EVENTO 3

HH= 14:00:00 MD= 60 PRX= 869A NCDGS= 1 CDG 1=82F7

EVENTO 9

HH= 14:06:00 MD= 60 PRX= 86A4 NCDGS= 1 CDG 1=85AC

EVENTO 3

HH= 15:00:00 MD= 60 PRX= 86AE NCDGS= 1 CDG 1=82F7

EVENTO 9

HH= 15:06:00 MD= 60 PRX= 86B8 NCDGS= 1 CDG 1=85AC

EVENTO 3

HH= 21:00:00 MD= 60 PRX= 86C2 NCDGS= 1 CDG 1=82F7

EVENTO 9

HH= 21:06:00 MD= 60 PRX= 85EA NCDGS= 1 CDG 1=85AC

LRC= 75

8. POSICAO LIVRE NO CARTUCHO

INICIO= 86CD TAMANHO= 14643

B.2.3. Relatório dos Dados Adquiridos para o Experimento Padrão

RELATORIO DOS DADOS ADQUIRIDOS NA ESTACAO DE CAMPO

1. REGISTRO DE ESCRITA : 1 TAMANHO = 37

D=24/12/86 T=00:00:02 D=24/12/86 T=06:00:00 S[10]=Maringa
S[10]=Itana
LRC= 2C

2. REGISTRO DE ESCRITA : 4 TAMANHO = 19

BI 2]=002C BI 2]=0053 BI 2]=0003 BI 2]=0003 BI 2]=0000
BI 2]=003C BI 2]=0004
LRC= 50

3. REGISTRO DE ESCRITA : 5 TAMANHO = 29

D=24/12/86 T=06:00:03 R4=21.0000000 D=24/12/86 T=06:00:04
R4=21.0000000 R4=5.5000000
LRC= AC

4. REGISTRO DE ESCRITA : 6 TAMANHO = 27

R4=0.0000000 R4=13.1124001 R4=1.0000000 D=00/00/00
T=00:00:00 R4=32767.0000000
LRC= EF

5. REGISTRO DE ESCRITA : 7 TAMANHO = 9

R4=3.4854543
LRC= 6D

6. REGISTRO DE ESCRITA : 4 TAMANHO = 19

BI 2]=0031 BI 2]=0046 BI 2]=0008 BI 2]=0008 BI 2]=0000
BI 2]=0000 BI 2]=0004
LRC= 64

7. REGISTRO DE ESCRITA : 7 TAMANHO = 9

R4=0.1145454
LRC= 41

8. REGISTRO DE ESCRITA : 4 TAMANHO = 19

BI 2]=003A BI 2]=003B BI 2]=0009 BI 2]=0009 BI 2]=0000
BI 2]=003C BI 2]=0008

LRC= 22

9. REGISTRO DE ESCRITA : 7 TAMANHO = 9

R4=3.5999999

LRC= EA

10. REGISTRO DE ESCRITA : 4 TAMANHO = 19

BI 2]=003B BI 2]=003A BI 2]=0008 BI 2]=0008 BI 2]=0000 BI
2]=003C BI 2]=0008

LRC= 22

11. REGISTRO DE ESCRITA : 7 TAMANHO = 9

R4=3.5999999

LRC= EA

12. REGISTRO DE ESCRITA : 4 TAMANHO = 19

BI 2]=0034 BI 2]=0046 BI 2]=0000 BI 2]=0000 BI 2]=0000
BI 2]=0000 BI 2]=0006

LRC= 63

13. REGISTRO DE ESCRITA : 7 TAMANHO = 9

R4=0.0000000

LRC= 36

14. REGISTRO DE ESCRITA : 4 TAMANHO = 19

BI 2]=002C BI 2]=0053 BI 2]=0003 BI 2]=0003 BI 2]=0000
BI 2]=003C BI 2]=0004

LRC= 50

15. REGISTRO DE ESCRITA : 5 TAMANHO = 29

D=25/12/86 T=05:00:03 R4=19.5000000 D=24/12/86 T=15:00:05

R4=29.5000000 R4=25.7500000

LRC= B4

16. REGISTRO DE ESCRITA : 6 TAMANHO = 27

R4=0.0000000 R4=66B.0618896 R4=12.5500002 D=25/12/86

T=04:58:02 R4=20.0000000

LRC= C2

17. REGISTRO DE ESCRITA : 3 TAMANHO = 29

S[18]=fim do experimento D=25/12/86 T=07:00:12
LRC= BD

18. REGISTRO DE ESCRITA : 2 TAMANHO = 14

D=25/12/86 T=07:00:13 I2=350 BI I1=01
LRC= F7

B.3. Programação e Relatórios para o Experimento Micrometeorológico

B.3.1. Programação do Experimento Micrometeorológico em LADAN

```

10 program microclima;
20 /* experimento microclima em culturas de cafe */
30 assign
40 01A port 1 0:tempar1; /* semicondutor */
50 01A port 2 0:tempar2; /* semicondutor */
60 02J port 3 0:tsolosup; /* termopar */
70 02J port 4 0:tsolo5; /* termopar */
80 02J port 5 0:tsolo10; /* termopar */
90 02J port 6 0:tsolo25; /* termopar */
100 02J port 7 0:tfolha5; /* termopar */
110 02J port 8 0:tfolha20; /* termopar */
120 02J port 9 0:tfolha40; /* termopar */
130 02J port 10 0:tfolhexp; /* termopar */
140 05A port 11 0:radpar; /* quantometro */
150 05A port 12 0:radliq; /* pirgeometro */
160 08A port 13 0:qtdchuva; /* pluviometro */
170 10A port 14 0:vento; /* anemometro - velocidade */
180 10B port 15 0:dirvento; /* anemometro - direcao */
190 var
200 temparm1, temparm2, tsolomsup, tsolom5, tsolom10 : real;
210 tsolom25, tfolhexpm, tfolham20, tfolham40, tfolham5 : real;
220 radpart, radliqp, radliqn, radliqt : real;
230 totpluv, ventot, ventinst, ventant, radliqr : real;
240 qtdchuvai, chuvant, cont : integer;
250 chuvamax : integer(4096);
260 datain : date;
270 /* inicializa data de inicio do experimento */
280 task header
290 read(ck,datain);
300 read(sn,qtdchuva);
310 chuvant := integer(qtdchuva);
320 endtk;
330 /* ativa calculo da temperatura media */
340 task ativtemp
350 activate evt2;
360 endtk;
370 /* calculo da temperatura media */
380 task calctemp
390 cont := cont + 1;
400 read(sn,tempar1,tempar2,tsolosup,tsolo5,tsolo10,tsolo25);
410 read(sn,tfolha5,tfolha20,tfolha40,tfolhexp);
420 temparm1 := temparm1 + real(tempar1);
430 temparm2 := temparm2 + real(tempar2);
440 tsolomsup := tsolomsup + real(tsolosup);
450 tsolom5 := tsolom5 + real(tsolo5);
460 tsolom10 := tsolom10 + real(tsolo10);
470 tsolom25 := tsolom25 + real(tsolo25);
480 tfolhexpm := tfolhexpm + real(tfolhexp);
490 tfolham20 := tfolham20 + real(tfolha20);
500 tfolham40 := tfolham40 + real(tfolha40);
510 tfolham5 := tfolham5 + real(tfolha5);
520 if cont = 6 then
530 temparm1 := temparm1/6.;
540 temparm2 := temparm2/6.;

```

```

550     tsolomsup := tsolomsup/6.;
560     tsolom5 := tsolom5/6.;
570     tsolom10 := tsolom10/6.;
580     tsolom25 := tsolom25/6.;
590     tfolhexpm := tfolhexpm/6.;
600     tfolham20 := tfolham20/6.;
610     tfolham40 := tfolham40/6.;
620     tfolham5 := tfolham5/6.;
630     /* registro de escrita 3 */
640     write(ch,temparm1,temparm2,tsolomsup,tsolom5,tsolom10,tsolom25);
650     /* registro de escrita 4 */
660     write(ch,tfolhexpm,tfolham20,tfolham40,tfolham5,horaref);
670     temparm1 := 0.; temparm2 := 0.;
680     tsolomsup := 0.; tsolom5 := 0.;
690     tsolom10 := 0.; tsolom25 := 0.;
700     tfolhexpm := 0.; tfolham20 := 0.;
710     tfolham40 := 0.; tfolham5 := 0.; cont := 0;
720     terminate evt2;
730 endif;
740 endtk;
750 /* acumula radiacao solar e vento */
760 task acumrad
770   read(sn,radpar,radliq);
780   radpart := radpart + real(radpar);
790   if real(radliq) > 0.
800   then
810     radliqp := radliqp + real(radliq)
820   else
830     radliqn := radliqn + real(radliq);
840   endif;
850 endtk;
860 /* acumula vento para calcular velocidade */
870 task acumvento
880   read(sn,vento);
890   ventot := ventot + real(vento);
900 endtk;
910 /* amostragem dos valores instantaneos */
920 task valinst
930   read(sn,radpar,radliq,qudchuva,dirvento);
940   if integer(qudchuva) - chuvant > 0
950   then
960     qudchuvai := integer(qudchuva) - chuvant
970   else
980     qudchuvai := chuvamax - chuvant + integer(qudchuva);
990   endif;
1000   chuvant := qudchuvai;
1010   totpluv := totpluv + real(qudchuvai);
1020   ventinst := ((ventot - ventant)/15.)*0.06;
1030   ventant := ventot;
1040   /* registro de escrita 5 */
1050   radliqr := real(radliq);
1060   write(ch,radpar,radliq,radliqr,qudchuvai,ventinst,dirvento);
1070 endtk;
1080 /* grava valores acumulados no periodo */
1090 task gravperiod
1100   radliqt := radliqp + radliqn;
1110   /* registro de escrita 6 */
1120   write(ch,radpart,radliqp,radliqn,radliqt,ventot,totpluv);

```

```
1130 radpart := 0.;
1140 radliqt := 0.; radliqp := 0.; radliqn := 0.;
1150 ventot := 0.; ventant := 0.; totpluv := 0.;
1160 if real(dateref) - real(datain) = 1.0
1170 then
1180     if horaref = 7:00 then trailler endif;
1190 endif;
1200 endtk;
1210 /* secao para declaracao dos eventos */
1220 event section
1230 evt1 : every 14 min do ativtemp endo;
1240 evt2[0] : every 10 seg do calctemp endo;
1250 evt3 : every 01 min within [5:30,18:30] do acumrad endo;
1260 evt4 : every 01 min do acumvento endo;
1270 evt5 : every 15 min do valinst endo;
1280 evt6 : at 07:00 do gravperiod endo;
1290 endevt.
      ZERO ERROS DETECTADOS
      FIM DA COMPILACAO
```

B.3.2. Relatório do Código Intermediário para o Experimento Micrometeorológico

1. TABELA DE ENDERECOS BASE

OBJETO	ENDERECO	TAMANHO
DESCRITOR DOS SENSORES	8020	287
DESCRITOR DO AMBIENTE	8140	270
AREA DE DADOS	824F	133
CODIGO DAS TAREFAS	82D5	1095
ENDERECO DAS CONVERSOES	871D	0
LISTA DE EVENTOS	871D	106
POSICAO LIVRE NO CARTUCHO	8788	14456
ULTIMA POSICAO GRV. DADOS	0000	0

2. DESCRITOR DOS SENSORES

```
=====
CODIGO = 1A      SINAL = A      MED. MINIMA = -20.0000
MED. MAXIMA = 160.0000      PRECISAO = 0.2000
-----
```

OPCOES DE LEITURA

```
-----
OPCAO = 0      PROX. OPCAO = 0
```

1. CONVERSAO PARA INTEIRO

```
ROTINA = 2      NUM. PARMS = 2      COMPRIMENTO = 8
VALORES = 0000008000000000
```

2. CONVERSAO PARA REAL

```
ROTINA = 1      NUM. PARMS = 2      COMPRIMENTO = 8
VALORES = 0000008000000000
```

```
=====
CODIGO = 2J      SINAL = A      MED. MINIMA = -0.4000
MED. MAXIMA = 3.2000      PRECISAO = 0.0040
-----
```

OPCOES DE LEITURA

```
-----
OPCAO = 0      PROX. OPCAO = 0
```

1. CONVERSAO PARA INTEIRO

```
ROTINA = 2      NUM. PARMS = 2      COMPRIMENTO = 8
VALORES = 0000008000000000
```

2. CONVERSAO PARA REAL

```
ROTINA = 1      NUM. PARMS = 2      COMPRIMENTO = 8
VALORES = 0000008000000000
```

```
=====
CODIGO = 6A      SINAL = A      MED. MINIMA = 0.0000
MED. MAXIMA = 100.0000      PRECISAO = 1.0000
-----
```

OPCOES DE LEITURA

```
-----
OPCAO = 0      PROX. OPCAO = 0
```

1. CONVERSÃO PARA INTEIRO
 ROTINA = 2 NUM. PARMS = 2 COMPRIMENTO = 8
 VALORES = DE02097E000000000

2. CONVERSÃO PARA REAL
 ROTINA = 1 NUM. PARMS = 2 COMPRIMENTO = 8
 VALORES = DE02097E000000000

=====

CODIGO = 5A	SINAL = A	MED. MINIMA = -10.0000
MED. MAXIMA = 100.0000	PRECISAO = 1.0000	

OPCOES DE LEITURA

OPCAO = 0 PROX. OPCAO = 0

1. CONVERSÃO PARA INTEIRO
 ROTINA = 2 NUM. PARMS = 2 COMPRIMENTO = 8
 VALORES = DE02097E000000000

2. CONVERSÃO PARA REAL
 ROTINA = 1 NUM. PARMS = 2 COMPRIMENTO = 8
 VALORES = DE02097E000000000

=====

CODIGO = 8A	SINAL = D	MED. MINIMA = 0.0000
MED. MAXIMA = 100.0000	PRECISAO = 1.0000	

OPCOES DE LEITURA

OPCAO = 0 PROX. OPCAO = 0

1. CONVERSÃO PARA INTEIRO
 ROTINA = 2 NUM. PARMS = 2 COMPRIMENTO = 8
 VALORES = 0000008100000000

2. CONVERSÃO PARA REAL
 ROTINA = 1 NUM. PARMS = 2 COMPRIMENTO = 8
 VALORES = 0000008100000000

=====

CODIGO = 10A	SINAL = A	MED. MINIMA = 0.0000
MED. MAXIMA = 120000.0000	PRECISAO = 1.0000	

OPCOES DE LEITURA

OPCAO = 0 PROX. OPCAO = 0

1. CONVERSÃO PARA INTEIRO
 ROTINA = 2 NUM. PARMS = 2 COMPRIMENTO = 8
 VALORES = 0000008100000000

2. CONVERSÃO PARA REAL
 ROTINA = 1 NUM. PARMS = 2 COMPRIMENTO = 8
 VALORES = 0000008100000000


```
=====
CODIGO = 10B      SINAL = A      MED. MINIMA = 1.0000
MED. MAXIMA = 8.0000      PRECISAO = 1.0000
-----
```

OPCOES DE LEITURA

```
-----
OPCAO = 0      PROX. OPCAO = 0
```

```
1. CONVERSÃO PARA INTEIRO
ROTINA = 2      NUM. PARMS = 2      COMPRIMENTO = 8
VALORES = 000000B100000000
```

```
2. CONVERSÃO PARA REAL
ROTINA = 1      NUM. PARMS = 2      COMPRIMENTO = 8
VALORES = 000000B100000000
```

LRC = 36

3. DESCRITOR DO AMBIENTE

```
-----
VARIÁVEL EXTERNA = tempa1      DESL. SENSOR = 0      NUMERO SENSOR = 0
```

```
PORTA LÓGICA = 1      OPÇÃO DE LEITURA = 0      COMPR. VARIÁVEL = 2
VALOR = 0000
```

```
-----
VARIÁVEL EXTERNA = tempa2      DESL. SENSOR = 0      NUMERO SENSOR = 0
```

```
PORTA LÓGICA = 2      OPÇÃO DE LEITURA = 0      COMPR. VARIÁVEL = 2
VALOR = 0000
```

```
-----
VARIÁVEL EXTERNA = tsolosup      DESL. SENSOR = 41      NUMERO SENSOR = 0
```

```
PORTA LÓGICA = 3      OPÇÃO DE LEITURA = 0      COMPR. VARIÁVEL = 2
VALOR = 0000
```

```
-----
VARIÁVEL EXTERNA = tsolo5      DESL. SENSOR = 41      NUMERO SENSOR = 0
```

```
PORTA LÓGICA = 4      OPÇÃO DE LEITURA = 0      COMPR. VARIÁVEL = 2
VALOR = 0000
```

```
-----
VARIÁVEL EXTERNA = tsolo10      DESL. SENSOR = 41      NUMERO SENSOR = 0
```

```
PORTA LÓGICA = 5      OPÇÃO DE LEITURA = 0      COMPR. VARIÁVEL = 2
VALOR = 0000
```

```
-----
VARIÁVEL EXTERNA = tsolo25      DESL. SENSOR = 41      NUMERO SENSOR = 0
```

```
PORTA LÓGICA = 6      OPÇÃO DE LEITURA = 0      COMPR. VARIÁVEL = 2
VALOR = 0000
```

```
-----
VARIÁVEL EXTERNA = tfolha5      DESL. SENSOR = 41      NUMERO SENSOR = 0
```

```
PORTA LÓGICA = 7      OPÇÃO DE LEITURA = 0      COMPR. VARIÁVEL = 2
VALOR = 0000
```

 VARIABEL EXTERNA = tfolha20 DESL. SENSOR = 41 NUMERO SENSOR = 0

PORTA LOGICA = 8 OPCAO DE LEITURA = 0 COMPR. VARIABEL = 2
 VALOR = 0000

 VARIABEL EXTERNA = tfolha40 DESL. SENSOR = 41 NUMERO SENSOR = 0

PORTA LOGICA = 9 OPCAO DE LEITURA = 0 COMPR. VARIABEL = 2
 VALOR = 0000

 VARIABEL EXTERNA = tfolhexp DESL. SENSOR = 41 NUMERO SENSOR = 0

PORTA LOGICA = 10 OPCAO DE LEITURA = 0 COMPR. VARIABEL = 2
 VALOR = 0000

 VARIABEL EXTERNA = radpar DESL. SENSOR = 82 NUMERO SENSOR = 0

PORTA LOGICA = 11 OPCAO DE LEITURA = 0 COMPR. VARIABEL = 2
 VALOR = 0000

 VARIABEL EXTERNA = radiq DESL. SENSOR = 123 NUMERO SENSOR = 0

PORTA LOGICA = 12 OPCAO DE LEITURA = 0 COMPR. VARIABEL = 2
 VALOR = 0000

 VARIABEL EXTERNA = qtdchuva DESL. SENSOR = 164 NUMERO SENSOR = 0

PORTA LOGICA = 13 OPCAO DE LEITURA = 0 COMPR. VARIABEL = 2
 VALOR = 0000

 VARIABEL EXTERNA = vento DESL. SENSOR = 205 NUMERO SENSOR = 0

PORTA LOGICA = 14 OPCAO DE LEITURA = 0 COMPR. VARIABEL = 2
 VALOR = 0000

 VARIABEL EXTERNA = dirvento DESL. SENSOR = 246 NUMERO SENSOR = 0

PORTA LOGICA = 15 OPCAO DE LEITURA = 0 COMPR. VARIABEL = 2
 VALOR = 0000
 LRC = 173

4. AREA DE DADOS

12=32767 12=-32767 T=23:59:59 T=00:00:00 D=00/00/00 T=00:00:00

R4=0.0000000 R4=0.0000000 R4=0.0000000

R4=0.0000000 R4=0.0000000 R4=0.0000000

R4=0.0000000 R4=0.0000000 R4=0.0000000

R4=0.0000000 R4=0.0000000 R4=0.0000000

R4=0.0000000 R4=0.0000000 R4=0.0000000

R4=0.0000000 R4=0.0000000 R4=0.0000000

R4=0.0000000 12=0 12=0 12=0 12=4096 D=00/00/00

7.CODIGO DAS TAREFAS

TAREFA 1(82D5-82F3)

RD ck 140

RD sn 227

AB 131 CVIX227 LRC = 6B

TAREFA 2(82F5-82FC)

AT 4 LRC = 6E

TAREFA 3(82FE-853E)

AB 134 V134 C11 +

RD sn 11 29 47 65 83 101

RD sn 119 137 155 173

AB 33 V 33 CVRX11 +

AB 38 V 38 CVRX29 +

AB 43 V 43 CVRX47 +

AB 48 V 48 CVRX65 +

AB 53 V 53 CVRX83 +

AB 58 V 58 CVRX101 +

AB 63 V 63 CVRX173 +

AB 68 V 68 CVRX137 +

AB 73 V 73 CVRX155 +

AB 78 V 78 CVRX119 +

IF V134 C16 =

THEN

AB 33 V 33 CR 6.00000 /

AB 38 V 38 CR 6.00000 /

AB 43 V 43 CR 6.00000 /

AB 48 V 48 CR 6.00000 /
 AB 53 V 53 CR 6.00000 /
 AB 58 V 58 CR 6.00000 /
 AB 63 V 63 CR 6.00000 /
 AB 68 V 68 CR 6.00000 /
 AB 73 V 73 CR 6.00000 /
 AB 78 V 78 CR 6.00000 /

WR ch
 RELACAO DOS PARAMETROS
 3 V 33 V 38 V 43 V 48 V 53 V 58

WR ch
 RELACAO DOS PARAMETROS
 4 V 63 V 68 V 73 V 78 V 29

AB 33 CR 0.00000
 AB 38 CR 0.00000
 AB 43 CR 0.00000
 AB 48 CR 0.00000
 AB 53 CR 0.00000
 AB 58 CR 0.00000
 AB 63 CR 0.00000
 AB 68 CR 0.00000
 AB 73 CR 0.00000
 AB 78 CR 0.00000

AB 134 C10

DT 4 LRC = 12

TAREFA 4(8540-8596)

RD sn 191 209

AB 83 V 83 CVRX191 +

IF CVRX209 CR 0.00000 >
 THEN

AB 88 V 88 CVRX209 +
ELSE

AB 93 V 93 CVRX209 + LRC = 04

TAREFA 5(8598-85B3)

RD sn 245

AB 108 V108 CVRX245 + LRC = 12

TAREFA 6(85B5-866A)

RD sn 191 209 227 263

IF CVIX227 V131 - CIO >
THEN

AB 128 CVIX227 V131 -
ELSE

AB 128 V137 V131 - CVIX227 +

AB 131 V128

AB 103 V103 CVRV128 +

AB 113 V108 V118 - CR 15.00000 / CR 0.06000 *

AB 118 V108

AB 123 CVRX209

WR ch

RELACAO DOS PARAMETROS

5 X191 X209 V123 V128 V113 X263 LRC = 53

TAREFA 7(866C-871A)

AB 98 V 88 V 93 +

WR ch

RELACAO DOS PARAMETROS

6 V 83 V 88 V 93 V 98 V108 V103

AB 83 CR 0.00000

AB 98 CR 0.00000

AB 88 CR 0.00000

AB 93 CR 0.00000

AB 108 CR 0.00000

AB 118 CR 0.00000

AB 103 CR 0.00000

IF CVRV25 CVRV140 - CR 1.00000 =
THEN

IF V 29 CT07:00:00 =
THEN

ED FIM DO EXPERIMENTO LRC = 8C

LRC GLOBAL DAS TAREFAS = DE

6. ENDERECO DAS CONVERSOES

TABELA DE ROTINAS DE CONVERSAO VAZIA

8. LISTA DE EVENTOS

EVENTO 1

HH= 00:00:00 MD= 20 PRX= 8727 NCDGS= 1 CDG 1=82D5

EVENTO 2

HH= 00:00:00 MD= 00 PRX= 8731

EVENTO 3

HH= 00:00:00 MD= A1 PRX= 873F FQ= 20M
NINT= 0 INTCORR=0
NCDGS= 1 CDG 1=82F5

EVENTO 4

HH= 00:00:00 MD= 80 PRX= 874D FQ= 16S
NINT= 0 INTCORR=0
NCDGS= 1 CDG 1=82FE

EVENTO 5

HH= 00:00:00 MD= A1 PRX= 8761 FQ= 1M
NINT= 1 INTCORR=1
HH1=05:30:00 HH2=18:30:00
NCDGS= 1 CDG 1=8540

EVENTO 6

HH= 00:00:00 MD= A1 PRX= 876F FQ= 1M
NINT= 0 INTCORR=0
NCDGS= 1 CDG 1=8598

EVENTO 7

HH= 00:00:00 MD= A1 PRX= 877D FQ= 21M
NINT= 0 INTCORR=0
NCDGS= 1 CDG 1=8585

EVENTO 8

HH= 07:00:00 MD= 60 PRX= 8727 NCDGS= 1 CDG 1=866C

LRC= FE

8. POSICAO LIVRE NO CARTUCHO

INICIO= 8788 TAMANHO= 14456

B.3.3. Relatório dos Dados Adquiridos para o Experimento Micrometeorológico

RELATORIO DOS DADOS ADQUIRIDOS NA ESTACAO DE CAMPO

1. REGISTRO DE ESCRITA : 1 TAMANHO = 37

D=24/12/86 T=00:00:02 D=24/12/86 T=06:00:00 S[10]=Maringa
S[10]=Itana
LRC= 2C

2. REGISTRO DE ESCRITA : 5 TAMANHO = 21

BI 2]=0001 BI 2]=FFFF R4=-0.1338000 I2=0 R4=0.2400000
BI 2]=0004
LRC= 7B

3. REGISTRO DE ESCRITA : 3 TAMANHO = 29

R4=20.5000000 R4=18.0000000 R4=34.5000000
R4=30.5000000 R4=26.0000000 R4=25.0000000
LRC= 4F

4. REGISTRO DE ESCRITA : 4 TAMANHO = 24

R4=21.5000000 R4=19.5000000 R4=20.0000000
R4=19.0000000 T=06:01:05
LRC= 16

5. REGISTRO DE ESCRITA : 5 TAMANHO = 21

BI 2]=0001 BI 2]=FFFF R4=-0.1338000 I2=0 R4=3.5999999
BI 2]=0004
LRC= DA

6. REGISTRO DE ESCRITA : 3 TAMANHO = 29

R4=20.0000000 R4=18.0000000 R4=34.5000000
R4=30.5000000 R4=26.0000000 R4=25.0000000
LRC= 4B

7. REGISTRO DE ESCRITA : 4 TAMANHO = 24

R4=21.0000000 R4=19.0000000 R4=19.5000000
R4=18.5000000 T=06:15:04
LRC= 33

8. REGISTRO DE ESCRITA : 3 TAMANHO = 29

R4=20.0000000 R4=18.0000000 R4=34.5000000
R4=30.5000000 R4=26.0000000 R4=25.0000000
LRC= 4B

R4=28.5000000 R4=23.0000000 R4=36.5000000
 R4=32.5000000 R4=26.5000000 R4=25.5000000
 LRC= 4A

84. REGISTRO DE ESCRITA : 4 TAMANHO = 24

R4=29.5000000 R4=27.5000000 R4=28.0000000
 R4=27.0000000 T=12:19:04
 LRC= 1B

85. REGISTRO DE ESCRITA : 5 TAMANHO = 21

BI 21=000B BI 21=0007 R4=0.9366000 I2=0 R4=3.5999999
 BI 21=0001
 LRC= D6

86. REGISTRO DE ESCRITA : 3 TAMANHO = 29

R4=28.5000000 R4=23.0000000 R4=36.5000000
 R4=32.5000000 R4=26.5000000 R4=25.5000000
 LRC= 4A

87. REGISTRO DE ESCRITA : 4 TAMANHO = 24

R4=29.5000000 R4=27.5000000 R4=28.0000000
 R4=27.0000000 T=12:33:04
 LRC= 31

88. REGISTRO DE ESCRITA : 5 TAMANHO = 21

BI 21=000A BI 21=0007 R4=0.9366000 I2=0 R4=3.5999999
 BI 21=0001
 LRC= D7

89. REGISTRO DE ESCRITA : 3 TAMANHO = 29

R4=28.5000000 R4=23.0000000 R4=36.5000000
 R4=32.5000000 R4=26.5000000 R4=25.5000000
 LRC= 4A

90. REGISTRO DE ESCRITA : 4 TAMANHO = 24

R4=29.5000000 R4=27.5000000 R4=28.0000000
 R4=27.0000000 T=12:47:04
 LRC= 45

91. REGISTRO DE ESCRITA : 5 TAMANHO = 21

BI 21=000A BI 21=0006 R4=0.8028000 I2=0 R4=3.5999999
 BI 21=000B

LRC= 20

244. REGISTRO DE ESCRITA : 3 TAMANHO = 29

R4=23.0000000 R4=19.0000000 R4=35.0000000
 R4=31.0000000 R4=26.5000000 R4=25.5000000
 LRC= 51

245. REGISTRO DE ESCRITA : 4 TAMANHO = 24

R4=24.0000000 R4=22.0000000 R4=22.5000000
 R4=21.5000000 T=01:09:03
 LRC= 7F

246. REGISTRO DE ESCRITA : 5 TAMANHO = 21

BI 21=0001 BI 21=FFFF R4=-0.1338000 I2=0 R4=3.4199998
 BI 21=0004
 LRC= 40

247. REGISTRO DE ESCRITA : 3 TAMANHO = 29

R4=23.0000000 R4=19.0000000 R4=35.0000000
 R4=31.0000000 R4=26.5000000 R4=25.5000000
 LRC= 51

248. REGISTRO DE ESCRITA : 4 TAMANHO = 24

R4=24.0000000 R4=22.0000000 R4=22.5000000
 R4=21.5000000 T=01:23:03
 LRC= 55

249. REGISTRO DE ESCRITA : 5 TAMANHO = 21

BI 21=0001 BI 21=FFFF R4=-0.1338000 I2=0 R4=3.5999999
 BI 21=0004
 LRC= DA

250. REGISTRO DE ESCRITA : 3 TAMANHO = 29

R4=22.5000000 R4=19.0000000 R4=35.0000000
 R4=31.0000000 R4=26.5000000 R4=25.5000000
 LRC= 5D

251. REGISTRO DE ESCRITA : 4 TAMANHO = 24

R4=23.5000000 R4=21.5000000 R4=22.0000000
 R4=21.0000000 T=01:37:03
 LRC= 21

252. REGISTRO DE ESCRITA ; 5 TAMANHO = 0

5700050001FFFFDE02897E00006666668200

252. REGISTRO DE ESCRITA ; 2 TAMANHO = 14

D=25/12/86 T=01:45:04 I2=6261 B[1]=03

LRC= 93

Apêndice C

ARQUIVO DE SENSORES
E MENSAGENS DE ERRO

C.1. Arquivo de Sensores

codigo : 1A

tipo do sinal : A

unidade de medida : mv

classe dos sensores : semicondutor

medida minima : -20.000 medida maxima : 160.000

precisao : 0.200

----- OPCAO DE LEITURA : 0

comprimento da variavel : 2

....Conversao para Inteiro

n0. rotina de conversao : 2

Parametro passados

tipo do parametro : 2 valor : 0.500

tipo do parametro : 2 valor : 0.000

....Conversao para real

n0. rotina de conversao : 1

Parametro passados

tipo do parametro : 2 valor : 0.500

tipo do parametro : 2 valor : 0.000

codigo : 1B

tipo do sinal : A

unidade de medida : mv

classe dos sensores : semicondutor

medida minima : -30.000 medida maxima : 160.000

precisao : 0.200

----- OPCAO DE LEITURA : 0

comprimento da variavel : 2

....Conversao para Inteiro

n0. rotina de conversao : 2

Parametro passados

tipo do parametro : 2 valor : 0.500

tipo do parametro : 2 valor : 0.000

....Conversao para real

n0. rotina de conversao : 1

Parametro passados

tipo do parametro : 2 valor : 0.500

tipo do parametro : 2 valor : 0.000

codigo : 2J

tipo do sinal : A

unidade de medida : mv

classe dos sensores : termopar cobre constantan

medida minima : -0.400 medida maxima : 3.200

precisao : 0.004

----- OPCAO DE LEITURA : 0

comprimento da variavel : 2

....Conversao para Inteiro

n0. rotina de conversao : 2

Parametro passados

tipo do parametro : 2 valor : 0.500

tipo do parametro : 2 valor : 0.000

....Conversao para real

n0. rotina de conversao : 1

Parametro passados

tipo do parametro : 2 valor : 0.500

tipo do parametro : 2 valor : 0.000

codigo : 3A

tipo do sinal : A

unidade de medida : mv

classe dos sensores : higrometro

medida minima : 0.000 medida maxima : 100.000

precisao : 1.000

----- OPCAO DE LEITURA : 0

comprimento da variavel : 2

....Conversao para Inteiro

n0. rotina de conversao : 2

Parametro passados

tipo do parametro : 2 valor : 1.000

tipo do parametro : 2 valor : 0.000

....Conversao para real

n0. rotina de conversao : 1

Parametro passados

tipo do parametro : 2 valor : 1.000

tipo do parametro : 2 valor : 0.000

codigo : 4A

tipo do sinal : A

unidade de medida : mv

classe dos sensores : piranometro

medida minima : 0.000 medida maxima : 100.000

precisao : 1.000

----- OPCAO DE LEITURA : 0

comprimento da variavel : 2

....Conversao para inteiro

n0. rotina de conversao : 2

Parametro passados

tipo do parametro : 2 valor : 0.134

tipo do parametro : 2 valor : 0.000

....Conversao para real

n0. rotina de conversao : 1

Parametro passados

tipo do parametro : 2 valor : 0.134

tipo do parametro : 2 valor : 0.000

codigo : 5A

tipo do sinal : A

unidade de medida : mv

classe dos sensores : pirgeometro

medida minima : -10.000 medida maxima : 100.000

precisao : 1.000

----- OPCAO DE LEITURA : 0

comprimento da variavel : 2

....Conversao para Inteiro

n0. rotina de conversao : 2

Parametro passados

tipo do parametro : 2 valor : 0.134

tipo do parametro : 2 valor : 0.000

....Conversao para real

n0. rotina de conversao : 1

Parametro passados

tipo do parametro : 2 valor : 0.134

tipo do parametro : 2 valor : 0.000

codigo : 6A

tipo do sinal : A

unidade de medida : mv

classe dos sensores : quantometro

medida minima : 0.000 medida maxima : 100.000

precisao : 1.000

----- OPCAO DE LEITURA : 0

comprimento da variavel : 2

....Conversao para Inteiro

n0. rotina de conversao : 2

Parametro passados

tipo do parametro : 2 valor : 0.134

tipo do parametro : 2 valor : 0.000

....Conversao para real

n0. rotina de conversao : 1

Parametro passados

tipo do parametro : 2 valor : 0.134

tipo do parametro : 2 valor : 0.000

codigo : 7A

tipo do sinal : A

unidade de medida : mv

classe dos sensores : heliografo

medida minima : 0.000 medida maxima : 100.000

precisao : 1.000

----- OPCAO DE LEITURA : 0

comprimento da variavel : 2

....Conversao para inteiro

n0. rotina de conversao : 2

Parametro passados

tipo do parametro : 2 valor : 14.000

tipo do parametro : 2 valor : 0.000

....Conversao para real

n0. rotina de conversao : 1

Parametro passados

tipo do parametro : 2 valor : 14.000

tipo do parametro : 2 valor : 0.000

codigo : 8A

tipo do sinal : D

unidade de medida : cn

classe dos sensores : pluviometro

medida minima : 0.000 medida maxima : 100.000

precisao : 1.000

----- OPCAO DE LEITURA : 0

comprimento da variavel : 2

....Conversao para Inteiro

n0. rotina de conversao : 2

Parametro passados

tipo do parametro : 2 valor : 1.000

tipo do parametro : 2 valor : 0.000

....Conversao para real

n0. rotina de conversao : 1

Parametro passados

tipo do parametro : 2 valor : 1.000

tipo do parametro : 2 valor : 0.000

codigo : 9A

tipo do sinal : A

unidade de medida : mv

classe dos sensores : orvalhometro

medida minima : 0.000 medida maxima : 1.000

precisao : 1.000

----- OPCAO DE LEITURA : 0

comprimento da variavel : 2

....Conversao para Inteiro

n0. rotina de conversao : 2

Parametro passados

tipo do parametro : 2 valor : 1.000

tipo do parametro : 2 valor : 0.000

....Conversao para real

n0. rotina de conversao : 1

Parametro passados

tipo do parametro : 2 valor : 1.000

tipo do parametro : 2 valor : 0.000

codigo : 10A

tipo do sinal : A

unidade de medida : mv

classe dos sensores : anemometro velocidade

medida minima : 0.000 medida maxima : 120000.000

precisao : 1.000

----- OPCAO DE LEITURA : 0

comprimento da variavel : 2

....Conversao para Inteiro

n0. rotina de conversao : 2

Parametro passados

tipo do parametro : 2 valor : 1.000

tipo do parametro : 2 valor : 0.000

....Conversao para real

n0. rotina de conversao : 1

Parametro passados

tipo do parametro : 2 valor : 1.000

tipo do parametro : 2 valor : 0.000

codigo : 10B

tipo do sinal : A

unidade de medida : mv

classe dos sensores : anemometro direcao

medida minima : 1.000 medida maxima : 8.000

precisao : 1.000

----- OPCAO DE LEITURA : 0

comprimento da variavel : 2

....Conversao para Inteiro

nO. rotina de conversao : 2

Parametro passados

tipo do parametro : 2 valor : 1.000

tipo do parametro : 2 valor : 0.000

....Conversao para real

nO. rotina de conversao : 1

Parametro passados

tipo do parametro : 2 valor : 1.000

tipo do parametro : 2 valor : 0.000

C.2. Mensagens de Erro do Tradutor da LADAM

ERRO	MENSAGEM
1	"program" esperado
2	identificador esperado
3	":" esperado
4	"." esperado
5	fim de arquivo fonte foi encontrado
6	"assign" esperado
7	"procedure" esperado
8	sensor inexistente
9	"port" esperado
10	numero de porta logica invalido
11	"," esperado
12	fim de arquivo antecipado
13	valor inicial indefinido
14	identif. predefinido ou numerico esperado
15	"(" esperado
16	")" esperado
17	tipo indefinido
18	caracter invalido
19	identificador escrito incorretamente
20	string maior que o permitido ou fim de arquivo
21	tipo declarado na concorda com constante
22	":" esperado
23	constante numerica esperada
24	horario invalido ou repetido
25	comando invalido
26	identificador indefinido

- 27 variavel externa esperada
- 28 identificador ou string esperado
- 29 "!=" esperado
- 30 expressao invalida
- 31 frequencia invalida
- 32 "[" esperado
- 33 "]" esperado
- 34 "do" esperado
- 35 "then" esperado
- 36 excesso de intervalos definido
- 37 nome de tarefa esperado
- 38 nome de tarefa duplicado
- 39 intervalos nao estao ordenados
- 40 precisao excede permitida
- 41 conversao incompativel
- 42 identificador duplicado
- 43 mensagem na utilizada
- 44 funcao de conversao invalida
- 45 categoria do identificador invalida
- 46 tipo incompativel
- 47 mensagem nao utilizada
- 48 nome de evento esperado
- 49 "within" esperado
- 50 variavel externa esperada
- 51 data invalida
- 52 evento nao declarado
- 53 codigo de sensor esperado
- 54 opcao de leitura invalida
- 55 bytstring muito longo