**" INCUBADORA MICROPROCESSADA COM CONTROLE
TÉRMICO ADAPTATIVO POR FLUXO CONTÍNUO DE AR
FILTRADO "**

GIUSEPPE MAGDALENA STEPHAN

Dissertação apresentada à Faculdade de Filosofia,
Ciências e Letras de Ribeirão Preto para obtenção do
título de Mestre em Ciências-Área: Física Aplicada à
Medicina e Biologia.

DEDALUS - Acervo - FFCLRP

20800020987

RIBEIRÃO PRETO-SP

Março de 1996

UNIVERSIDADE DE SÃO PAULO
FACULDADE DE FILOSOFIA, CIÊNCIAS E LETRAS DE RIBEIRÃO PRETO
DEPARTAMENTO DE GEOLOGIA, FÍSICA E MATEMÁTICA

INCUBADORA MICROPROCESSADA COM CONTROLE TÉRMICO ADAPTATIVO
POR FLUXO CONTÍNUO DE AR FILTRADO

Aluno: Giuseppe Magdalena Stephan
Orientador: Prof. Dr. Carlos Alberto Pelá

MEMBROS DA COMISSÃO JULGADORA DA DISSERTAÇÃO DE MESTRADO
DE GIUSEPPE MAGDALENA STEPHAN, APRESENTADA A FACULDADE DE
FILOSOFIA, CIÊNCIAS E LETRAS DE RIBEIRÃO PRETO DA UNIVERSIDADE
DE SÃO PAULO, NO DIA 14 DE MARÇO DE 1996.

COMISSÃO JULGADORA:

Prof. Dr. CARLOS ALBERTO PELÁ

Prof. Dr. JURANDIR NADAL

Prof. Dr. SALIM MOYSES JORGE

Dedico este trabalho aos patriotas que se empenham na busca do verdadeiro desenvolvimento científico, tecnológico e social do nosso Brasil.

Dedico ainda à esposa Renata, à filha Giovana e à todos os parentes que sempre compreenderam os momentos de ausência.

AGRADECIMENTOS

Equipamentos eletromédicos de suporte a vida atingem o grau de confiabilidade ideal após inúmeros testes e aprimoramentos que envolvem multi-profissionais.

No esforço de não esquecer nada e ninguém, o autor pede aos que não estão explicitamente citados que considerem a concretização do seu auxílio na realização deste trabalho como o mais valioso agradecimento aos que honrosamente preferem ficar no anonimato.

Ao professor Dr. Carlos Alberto Pelá, por sua enorme compreensão, superando as distâncias na realização deste trabalho.

Ao professor Dr. Jurandir Nadal, pelo companheirismo, incentivo e dedicação, com sacrifícios até dos seus momentos de descanso.

Aos Professores Dr. Richard Magdalena Stephan, Dr. Fernando Soares Schlindwein, Dr. Leão e demais docentes da COPPE, que ajudaram na minha formação.

Aos professores Dr. Thomaz Ghilardi Netto, Dr. Oswaldo Baffa Filho e todos os outros professores do departamento, pela paciência de me ensinar Física como se Físico eu fosse.

Ao Sr. Diretor do CENTRO DE ELETRÔNICA DA MARINHA, CMG Carlos Afonso Pierantoni Gamboa, por me conceder os dias de licença que necessitei.

Aos professores Dr. Salim Moyses Jorge, Dr. Arthur L. Gonçalves e Equipe de Enfermagem do Departamento de Pediatria do Hospital das Clínicas pela dedicação na fase de avaliações em pacientes

Aos técnicos e amigos Bruçó, Eldereis e Élcio, pelo seus serviços profissionais e pelo companheirismo no trabalho.

Ao CIDRA pela utilização de sua estrutura.

Ao CNPq, pelo apoio financeiro.

À Indústria Olidef cz, e seus funcionários, por terem colocado sua estrutura a favor do desenvolvimento deste trabalho.

FICHA CATALOGRÁFICA

Stephan, Giuseppe Magdalena

INCUBADORA MICROPROCESSADA COM CONTROLE TÉRMICO
ADAPTATIVO POR FLUXO CONTÍNUO DE AR FILTRADO / Giuseppe
Magdalena Stephan. -- Ribeirão Preto 1996.

Tese (Mestrado) -- FACULDADE DE FILOSOFIA CIÊNCIAS E LETRAS DO
CAMPUS DA USP DE RIBEIRÃO PRETO - CURSO DE PÓS GRADUAÇÃO EM
FÍSICA APLICADA _ MEDICINA E BIOLOGIA.

Áreas: Instrumentação, Controle adaptativo, Neonatologia.

Orientador: Prof. Dr. Carlos Alberto Pellá.

Descritores: 1. Conforto Térmico
2. Temperatura Ambiental Termoneutra
3. Incubadoras.

RESUMO

O presente trabalho é resultado da cooperação entre o Departamento de Pericultura e Pediatria do Hospital das Clínicas e o Departamento de Física Aplicada à Medicina e Biologia da Universidade de São Paulo em Ribeirão Preto, dando suporte para a indústria Brasileira OLIDEF cz no projeto, construção e teste de uma incubadora aperfeiçoada para o cuidado de recém-nascidos de risco.

A incubadora desenvolvida apresenta uma série de características inovadoras em relação as existentes.

Um sistema direcionador do ar continuamente renovado mantém baixa concentração do dióxido de carbono. Com uma velocidade do ar sobre o colchão de cerca de 0,01 metros por segundo, são obtidas baixas perdas por evaporação e baixos níveis de ruído com uniformidade de temperatura.

O sistema de abertura da cúpula dá acesso à ambos os lados e à cabeça do recém-nascido (RN). Quando o compartimento do paciente é aberto, um fluxo de ar mais quente e intenso mantém o calor do RN reduzindo as perdas para o ambiente.

Um sistema fácil e rápido de posicionamento de chapas de raio X torna possível este exame sem abertura do compartimento do RN.

Um sistema microprocessado de monitoração e memorização contínua do peso do recém-nascido também reduz a necessidade de manipulação do paciente. Dispositivos eletromecânicos permitem inclinar o corpo do recém-nascido. Também por dispositivo eletromecânico, a cúpula pode ser aberta.

As partes principais da incubadora são feitas por "vacuum-forming". Este processo produz superfícies sem rugosidades, reduzindo assim o risco de contaminação.

O "hardware" consiste basicamente de um módulo intercambiável composto por um sistema de aquisição de dados, baseado em uma CPU 8088, 32K EPROM, 8K RAM e 8 canais analógicos multiplexados para um conversor Analógico Digital de 12 "bits".

Sensores de temperatura extremamente rápidos (27.0 °C até 37.0°C em menos de 1 segundo), baseados na mesma técnica de fabricação de extensômetros, foram usados.

Superadas as dificuldades inerentes à introdução de novas tecnologias no processo de fabricação de incubadoras, foram fabricados dois protótipos.

Após os testes de laboratório duas incubadoras foram enviadas para unidades de cuidado intensivo para um longo período de testes "in vivo". A segurança e operacionalidade dos equipamentos foram comprovadas em uso diário de rotina.

Tendo sido os protótipos aprovados, foram produzidas duas incubadoras de cada modelo totalizando um lote de seis cabeça-de-série.

ABSTRACT

MICROPROCESSOR ADAPTIVE THERMAL CONTROL FOR AN INFANT INCUBATOR WITH CONTINUOUS RENOVATION OF FILTERED AND HEATED AIR.

The present thesis is the result of cooperation between the Clinical Hospital Neonatal Department and the Department of Physics in Medicine of São Paulo University at Ribeirão Preto that supported the Brazilian Industry OLIDEF to design, construct and test an improved incubator for premature infants.

The developed incubator presents innovative features compared with current models.

The system for continuous air renovation avoids carbon dioxide concentration without increased air velocity. With air velocity over the mattress of 0,01 m/sec, there is low evaporative water loss, low infant compartment sound level and more temperature uniformity.

The aperture system gives access to both sides and the head of the baby. When the compartment is opened, a hotter and stronger air flux maintains the infant heat, reducing loss to the environment.

A quick and easily positioned X-ray chassis makes possible examination without opening the compartment.

A continuous digital infant weight monitoring and recording system also permits less infant manipulation.

There are electromechanical devices to change infant position and to open the compartment.

The main parts of incubator are made by vacuum-forming. This process produces surfaces without rugosity, reducing contamination risk.

The "hardware" of the controller is basically an interchangeable module consisting of a data acquisition system based on a 8088 CPU, 32K EPROM, 8K RAM and 8 channel 12 bit analog to digital converter.

Fast temperature sensors (27.0°C to 37.0°C in less

ÍNDICE

CAPITULO 1 - MOTIVAÇÃO DO TRABALHO	2
OBJETIVOS DO TRABALHO	2
JUSTIFICATIVA DO TRABALHO	2
REVISÃO DO MERCADO NACIONAL DE INCUBADORAS.	3
INTERESSE DA INDÚSTRIA NACIONAL.	6
IMPORTÂNCIA DA CONFORMIDADE DO PROJETO COM NORMAS	6
CAPÍTULO 2 - ANÁLISE GERAL DO PROJETO	8
IMPORTÂNCIA DA ASSEPSIA NO DESENHO E ESCOLHA DOS MATERIAIS E TÉCNICAS DE INDUSTRIALIZAÇÃO DA INCUBADORA PROPOSTA	8
IMPORTÂNCIA DO LIVRE ACESSO AO PACIENTE POR TRÊS LADOS DO LEITO	9
IMPORTÂNCIA DA MINIMIZAÇÃO DAS INTERVENÇÕES PARA EXAMES DE DIAGNÓSTICO.	11
MONITORAÇÃO CONTÍNUA DO PESO DO PACIENTE.	11
SISTEMA DE ALOCAÇÃO DO CHASSIS DE RAIO X.	12
SISTEMA DE INCLINAÇÃO DO PACIENTE.	12
IMPORTÂNCIA DA RENOVAÇÃO CONTÍNUA DE AR FILTRADO PARA O PACIENTE E SUAS IMPLICAÇÕES	13
CAPITULO 3- ASPECTOS FÍSICOS DA INCUBADORA .	15
A ENERGIA NECESSÁRIA PARA MANTER O CONFORTO TÉRMICO.	15
TIPOS DE TRANSFERÊNCIA DE CALOR	15
CONDUÇÃO	16
CONVECÇÃO	17
RADIAÇÃO	19
EVAPORAÇÃO	20
TROCAS DE CALOR COM O AMBIENTE	21
OS SISTEMAS DE UMIDIFICAÇÃO E OXIGENAÇÃO E RENOVAÇÃO DO AR.	23
OS SISTEMAS DE ALARME.	24
CAPÍTULO 4 - OS SISTEMAS DE CONTROLE .	25
A FILOSOFIA DO SISTEMA DE CONTROLE.	25
CARACTERÍSTICAS FÍSICAS DOS SENSORES	25
TEMPO DE RESPOSTA	25
IMUNIDADE AS VARIAÇÕES DE TEMPERATURA DO AR EXTERNO A PAREDE DA CÚPULA	26
O EQUILÍBRIO DAS TEMPERATURAS NO INTERIOR DA CÚPULA.	27
TIPOS DE CONTROLE EMPREGADOS.	28
CONTROLE LIGA/DESLIGA	28
CONTROLE PROPORCIONAL	29
CONTROLE PROPORCIONAL INTEGRAL	31
CONTROLE ADAPTATIVO	32
CONCLUSÃO	33

CAPÍTULO 5 - O HARDWARE. **34**

DIAGRAMA DE BLOCOS	34
DESCRIÇÃO DA PLACA DE CPU E SUAS CALIBRAÇÕES.	35
DECODIFICAÇÃO DOS ENDEREÇOS:	35
CALIBRAÇÃO:	35
DESCRIÇÃO DA PLACA DE FONTE.	37
TENSÃO DE + 12 VCC.	37
TENSÃO DE - 12 VCC.	37
TENSÃO DE + 5 VCC.	37
ALIMENTAÇÃO DO “BACKLIGHT” E SINAL DE FREQUÊNCIA DA REDE DOBRADA.	38
TENSÃO DA BATERIA.	38
TENSÃO DE ALIMENTAÇÃO DA RESISTÊNCIA DE AQUECIMENTO:	39
DESCRIÇÃO DA PLACA DE SENSORES DE TEMPERATURA.	39
DESCRIÇÃO DA PLACA DE BALANÇA E SUA CALIBRAÇÃO.	39
DESCRIÇÃO DA PLACA DE ACIONAMENTO DOS MOTORES.	41

CONCLUSÃO **42**

RESULTADOS OBTIDOS	42
TRABALHOS FUTUROS	43

REFERÊNCIAS BIBLIOGRÁFICAS **44**

A PÊNDICE A - TRABALHO APRESENTADO NO CONGRESSO MUNDIAL DE 94

APÊNDICE B - DIAGRAMAS DO HARDWARE

APÊNDICE C - LISTAGEM DO SOFTWARE

1. CAPITULO 1 - MOTIVAÇÃO DO TRABALHO

1.1. *Objetivos do trabalho*

Os principais objetivos do trabalho são:

1 -Projetar, construir e testar uma família de incubadoras com inovações para recém nascidos de risco, respeitando normas da AAMI e IEC.

2 - Analisar, sob a óptica da Física, os problemas práticos de funcionamento, através de testes efetuados "in vitro" e "in vivo".

1.2. *Justificativa do Trabalho*

Desde o advento dos microprocessadores, na década dos 60, toda a instrumentação e controle analógicos, vêm sendo gradualmente informatizados. Esta tendência mundial tem chegado ao Brasil na grande maioria das vezes em pacotes fechados, dificultando as adaptações e manutenções que naturalmente se tornam necessárias para satisfazer plenamente as exigências do mercado local.

O bom atendimento a recém nascidos de risco depende primordialmente de uma equipe de profissionais bem treinados e dotados de equipamentos adequados às suas necessidades.

O presente trabalho se justifica na medida em que se engaja nas constantes pesquisas desenvolvidas na Faculdade de Filosofia Ciências e Letras da USP de Ribeirão Preto - curso de pós-graduação em Física Aplicada à Medicina e Biologia em conjunto com o Dep. de Puericultura do Hospital das Clínicas da Faculdade de Medicina da USP de Ribeirão Preto - curso de

1. CAPITULO 1 - MOTIVAÇÃO DO TRABALHO

1.1. *Objetivos do trabalho*

Os principais objetivos do trabalho são:

1 - Projetar, construir e testar uma família de incubadoras com inovações para recém nascidos de risco, respeitando normas da ANVI e IEC.

2 - Analisar, sob a óptica da Física, os problemas práticos de funcionamento, através de testes efetuados "in vitro" e "in vivo".

1.2. *Justificativa do Trabalho*

Desde o advento dos microprocessadores, na década dos 60, toda a instrumentação e controle analógicos, vêm sendo gradualmente informatizados. Esta tendência mundial tem chegado ao Brasil na grande maioria das vezes em pacotes fechados, dificultando as adaptações e manutenções que naturalmente se tornam necessárias para satisfazer plenamente as exigências do mercado local.

O bom atendimento a recém nascidos de risco depende primordialmente de uma equipe de profissionais bem treinados e dotados de equipamentos adequados às suas necessidades.

O presente trabalho se justifica na medida em que se engaja nas constantes pesquisas desenvolvidas na Faculdade de Filosofia Ciências e Letras da USP de Ribeirão Preto - curso de pós-graduação em Física Aplicada à Medicina e Biologia em conjunto com o Dep. de Puericultura do Hospital das Clínicas da Faculdade de Medicina da USP de Ribeirão Preto - curso de

pós-graduação em Pediatria e a empresa Olidef cz ind. com. de Aparelhos Hospitalares Ltda., objetivando atender adequadamente aos recém-nascidos de risco.

1.2.1. Revisão do mercado nacional de incubadoras.

O Brasil apresenta alguns fabricantes que atendem ao mercado do terceiro mundo. A tabela 1 relaciona os fabricantes conhecidos no Brasil e o país sede da indústria.

FABRICANTE	PAIS
Air Shields	USA
Atom	JAPÃO
Fanen	BRASIL
Gigante Recém nascido	BRASIL
KLS	BRASIL
Médipérma	FRANÇA
Medix	ARGENTINA
Ohmeda	USA
Olidef cz	BRASIL
Vemex	ARGENTINA
Vickers	INGLATERRA
Dräger	ALEMANHA
Deltronix	BRASIL

Tabela 1 : Fabricantes conhecidos no Brasil.

Os hospitais brasileiros são supridos, na sua grande maioria, por incubadoras nacionais, principalmente por questões de custo. Os impostos de importação inviabilizam as incubadoras estrangeiras, apesar de sua qualidade superior.

No intuito de atender ao mercado atual, nos seus mais variados níveis de exigência, foram criados três modelos de incubadoras. As Figuras 1, 2 e 3 apresentam o painel de cada modelo. A tabela 2 correlaciona as características de cada modelo.

CARACTERÍSTICA	MODELO		
	S	SC	SCL
ACIONAMENTO ELÉTRICO PARA POSICIONAR O COLCHÃO DO RN		x	x
MONITORAÇÃO E CONTROLE DA TEMPERATURA DO RN		x	x
GAVETA PARA CHASSIS DE RAIO X		x	x
ACIONAMENTO ELÉTRICO PARA ABRIR A CÚPULA			x
BALANÇA PARA MONITORAR PESO DO RN			x

Tabela 2 - Correlação entre as características de cada modelo

Fig. 1 - Painel do modelo S

Fig. 2 - Painel do modelo SC

Fig. 3 - Painel do modelo SCL

1.2.2. Interesse da Indústria nacional.

Com a abertura do mercado de informática, em Agosto de 1992, a entrada de equipamentos microprocessados no país foi muito facilitada. A barreira protecionista que acomodava a indústria nacional deixou de existir.

Os fabricantes de equipamentos médicos estão sendo obrigados a microprocessar seus produtos para poderem competir, em termos de qualidade, com os importados. Existem dois caminhos para esta tarefa: ou comprar o projeto pronto ou desenvolver a própria tecnologia. A segunda opção gera mão de obra mais especializada e possibilita o projeto de um produto mais adequado ao mercado local.

1.2.3. Importância da conformidade do projeto com normas

O crescente aumento da complexidade tecnológica dos produtos médico-hospitalares nas últimas décadas, aliado à necessidade de garantir a segurança dos usuários destes produtos, vem incrementando as atividades normativas nacionais e internacionais.

As indústrias têm procurado a certificação de conformidade de seus produtos, na maior parte das vezes, por exigências compulsórias das políticas de Saúde dos Governos. Em alguns casos, a exigência parte do próprio mercado, a cada dia mais consciente da importância da qualidade.

O Ministério da Saúde tem as Incubadoras

Neonatais como prioritárias na exigência de certificado de conformidade. Esta exigência apresenta força de lei pela Portaria Conjunta das Secretarias de Vigilância Sanitária e de Assistência à Saúde Número 1, de 17 de maio de 1993.

O grupo reconhecido pela ABNT para elaboração das normas brasileiras sobre incubadoras, está adotando as normas IEC 601-1 [17], IEC 601-2-19 [16] e AAMI Association for the Advancement of Medical Instrumentation II36 - 1991 [15], como base de seus trabalhos.

Sempre que pertinente, esta tese fará referência a estas normas.

2. CAPÍTULO 2 - ANÁLISE GERAL DO PROJETO

O levantamento de dados para estudos da ergonomia, dos materiais e dos sistemas da incubadora, resultaram no seguinte:

2.1. *Importância da assepsia no desenho e escolha dos materiais e técnicas de industrialização da incubadora proposta*

Na rotina dos berçários, a assepsia das incubadoras é freqüente. Portanto, devemos facilitar ao máximo esta tarefa. No modelo proposto foram eliminados todos os cantos vivos. Desta forma além de simplificarmos a limpeza, reduzimos o risco de contaminações por cultura de microorganismos no interior da incubadora [1][2][3][4][5][6][7]. A figura 4 abaixo ilustra as diferenças:

Fig. 4 - Acabamentos atuais (a) e o proposto (b)

Para atingir este tipo de acabamento foi utilizada a técnica de "vacum-forming" repuxando chapas do polímero ABS auto-extingüível para a base e acrílico para a cúpula.

A localização do módulo de comando separado do corpo da incubadora, além de facilitar a manutenção por permitir o intercâmbio para efeito de manutenção, reduz o risco de penetração de líquidos e respingos na parte eletrônica

durante a assepsia, conforme requerido no item 44.3 da Norma IEC 601-2-19 [16] e item 3.2.7.6 da ANSI/AAMI II36-1991 [15].

Além disto o processo de "vacuum-forming", largamente utilizado no acabamento interno de geladeiras, permite a redução de custo em material e mão de obra, quando comparado aos processos convencionais de corte, dobra e solda de chapas de aço.

2.2. *Importância do livre acesso ao paciente por três lados do leito*

Em muitas intervenções de rotina ou urgência, o RN precisa ser assistido por dois ou mais profissionais. Um esquema simplificado do posicionamento destes em relação ao RN foi elaborado a partir de filmagens em fitas de vídeo, sendo apresentado na fig. 5, onde as posições relativas dos Médicos e Enfermeiras são representadas pelas letras **M** e **E** respectivamente.

ASPIRAÇÃO DE SECREÇÕES DE CÂNULA ENDO TRAQUEAL.

INTUBAÇÃO TRAQUEAL PARA VENTILAÇÃO MECÂNICA.

CATETERIZAÇÃO UMBILICAL

DRENAGEM DE SECREÇÕES (PNEUMOTÓRAX)

EXSANGÜÍNIO TRANSFUSÃO

DISSECÇÃO DE VEIA

E = Enfermeiro(a).
M = Médico (a).

Fig. 5 - Posicionamento de Médicos e Enfermeiras em intervenções invasivas.

O sistema de abertura da cúpula dá acesso a ambos os lados e à cabeça do R.N. Quando a cúpula é aberta, um fluxo de ar mais aquecido e mais intenso reduz as perdas de calor do R.N. para o ambiente, por impedir a penetração do ar externo no ambiente de conforto do paciente.

2.3. Importância da minimização das intervenções para exames de diagnóstico.

O projeto procurou fornecer aos profissionais de saúde condições de examinar o RN sem necessidade de removê-lo do seu ambiente de conforto. Desta forma foram idealizados os seguintes sistemas:

2.3.1. Monitoração contínua do peso do paciente.

A bandeja que suporta o colchão e o RN, quando na posição horizontal baixa, fica apoiada por apenas quatro pinos que se interligam por uma estrutura em alumínio fundido a uma célula de carga. Desta forma, o peso suportado pelos quatro pinos é totalmente transmitido para a célula de carga.

A célula de carga na realidade mede força. Estamos interessados em medir massa. Em equipotenciais do campo gravitacional, a aceleração da gravidade funciona como uma constante de proporcionalidade entre a massa de um corpo e sua força peso.

Sabemos que a aceleração da gravidade em cm/s, ao nível do mar, varia de acordo com a latitude segundo a formula:

$$g = (980,632272 - 2,586145 \cos 2.B + 0,002878 \cos 4.B - 0,000004 \cdot \cos 6.B)$$

onde B é a latitude em graus.

Adotando os valores tabelados da aceleração da gravidade na referência [8] página 251, chegamos à conclusão de que tomando o menor valor de aceleração possível, ou seja latitude zero e ao nível do mar, e dividindo pelo maior (na latitude 90°) chegamos ao valor $983,2213 / 978,0490 = 1,0063$ que é superior à (0,6%).

Tendo em vista que a precisão da balança é de 2 g em 8192 g (0,2%), ou seja, mais de três vezes inferior à variação devida a latitude, notamos ser necessária a calibração da balança no local onde a mesma será usada (ver capítulo 5 item 5.5).

2.3.2. Sistema de alocação do chassis de raio X.

O dispositivo para radiografias localizado na face frontal da base, é constituído por uma gaveta, confeccionada em material plástico resistente, que desliza sobre trilhos telescópicos, com limitador de abertura, para evitar quedas acidentais. Projetado para a instalação de chassis de dimensões 13x18, 18x24 e 24x30, cuja localização, após o fechamento da gaveta, é facilitada por uma escala impressa na parte externa da mesma, coincidente com outra interna impressa em réguas móveis na bandeja.

Com este dispositivo, é possível radiografar qualquer área do corpo do recém-nascido sem necessidade de abrir a cúpula.

2.3.3. Sistema de inclinação do paciente.

Consiste de dois conjuntos eletromecânicos compostos por haste com cremalheira e motor de 12 Vcc. Permite inclinações da bandeja na faixa de - 10 graus até +10 graus.

O sistema foi projetado levando-se em conta o item 49.3 da norma IEC 601 - 01 [17] , traduzido a seguir:

"Tem que existir meio que permita retirar restrições mecânicas ao paciente quando faltar energia".

Caso seja necessária a alteração da posição do paciente, de forma urgente, durante a falta de energia, a incubadora prevê o acionamento manual da cúpula.

2.4. Importância da renovação contínua de ar filtrado para o paciente e suas implicações

Um sistema inédito de direcionamento e renovação contínua do ar consegue manter a uniformidade de temperatura e a baixa concentração de dióxido de carbono sem a necessidade de aumentarmos a velocidade do ar, constituindo um aperfeiçoamento em relação as incubadoras convencionais. Com uma média de velocidades do ar sobre o colchão de 0,01m/s conseguimos baixas taxas de perdas de água por evaporação, além de obtermos baixos níveis de ruído.

O sistema de aquecimento e circulação consiste de um modulo de chapa de aço cortada, dobrada e soldada. Tal chapa forma um duto de ar que permite a sustentação do miniventilador padrão de 3 polegadas, e da resistência de aquecimento.

O mini-ventilador, de tensão nominal 12 Vcc, enquanto a cúpula estiver fechada, é alimentado com 10 Vcc. No momento em que a cúpula inicia seu movimento de abertura, uma chave fim de curso passa a alimentar o mini-ventilador com 12 Vcc aumentando assim o fluxo de ar aquecido sobre o paciente.

A corrente que alimenta o ventilador é monitorada através do canal 3 do sistema de aquisição, que a amostra 120

vezes por segundo. Existe um ajuste (TP1 placa de CPU) para compatibilizar todos os tipos de mini-ventiladores encontrados em nosso mercado. Caso a corrente saia de sua faixa de normalidade, imediatamente o alarme de falta de circulação de ar é acionado.

Quanto ao alarme de falta de circulação de ar, implementado no protótipo testado, podemos afirmar que está satisfazendo plenamente a norma ANSI/AAMI II36-1991 [15]. O item 101.1 da norma IEC 601-2-19 [16], para ser cumprido, exige um "software" que leva em conta sutis variações na corrente de alimentação provocadas por obstruções na entrada ou saída do ar na cúpula, justificando assim a necessidade do conversor de 12 "bits" mesmo nos modelos sem balança.

3. CAPITULO 3- ASPECTOS FÍSICOS DA INCUBADORA .

3.1. *A energia necessária para manter o conforto térmico.*

Durante a existência de qualquer ser vivo, seu corpo está em constante adaptação contra o meio ambiente. Podemos dizer que um ser só está vivo se existem forças mantendo um desequilíbrio entre seu corpo e o meio ambiente.

Enquanto um ser humano encontra-se no útero materno, o meio que o circunda não é tão agressivo. Mas ao perder este abrigo natural, o recém nascido passa a encontrar uma série de dificuldades com as quais não estava habituado.

O RN necessita de energia para o seu crescimento e para manter sua temperatura. Esta energia ele obtém dos alimentos que recebe e do oxigênio que respira. Através do consumo de oxigênio podemos estimar, indiretamente, a energia requerida pelo RN para manter a própria temperatura.

Correlacionando a temperatura do ambiente do RN com o consumo de oxigênio, observa-se que existe uma faixa de temperatura onde o RN consome o mínimo deste gás. É nesta faixa de temperaturas que o RN encontra-se em conforto térmico.

3.2. *Tipos de transferência de calor*

Consideremos apenas as trocas de calor ocorridas no corpo do RN. Sabemos que ocorrem trocas de calor por condução, convecção, radiação e evaporação. O sentido e

intensidade da transferencia do calor, nos três primeiros mecanismos, depende do gradiente de temperatura existente entre o RN e o corpo com o qual o calor é trocado. Podemos então simplificar as equações que modelam as trocas de calor da seguinte forma:

$$\dot{Q} = K \cdot \Delta T$$

onde:

$\dot{Q}$ = calor trocado por unidade de tempo.

K = coeficiente de transferência de calor.

ΔT = gradiente de temperatura.

Independente do tipo de troca do calor, a equação é sempre a mesma.

Consideremos agora o coeficiente de cada tipo de troca.

3.2.1. Condução

Este tipo de troca ocorre entre sólidos em contato direto com o corpo do RN. Normalmente os lençóis, fraldas, sensores, eletrodos, sondas e catéteres são os sólidos em contato com o RN. Ocorre também pelos tecidos internos do RN, levando o calor dos órgãos que o produzem até a superfície da pele.

Podemos representar este tipo de troca pela seguinte expressão:

$$\dot{Q}_{\text{cond.}} = \frac{\lambda \cdot A}{D} \cdot (T_{\text{RN}} - T_{\text{ext.}})$$

Onde: quanto maior a área de contato (A) ou a condutividade térmica do material (λ), mais calor é trocado

por unidade de tempo. Quanto maior a espessura (D) entre a superfície emissora e a receptora do material, menos calor é trocado por unidade de tempo [9].

3.2.2. Convecção

Este tipo de troca ocorre por arraste de fluidos.

Normalmente o RN perde calor por convecção de quatro formas:

- Dos órgãos internos para a superfície da pele, pelo fluxo sangüíneo.
- Pelo movimento do ar que circunda a pele do RN.
- Pela transferência do ar inspirado frio, que após aquecido é expirado.
- Ao excretar o RN perde o calor das fezes e urina.

A convecção é o meio de troca mais direto sobre o qual o sistema de controle da incubadora pode atuar, pois a temperatura do ar e sua velocidade são elementos de atuação.

A equação geral que descreve as trocas por convecção é:

$$\dot{Q}_{\text{conv.}} = M \cdot c \cdot (T_{\text{int.}} - T_{\text{ext.}})$$

onde:

M = massa do veículo de transferência em gramas.

c = calor específico da massa de transferência.

(0,87 cal/g/°C para o sangue e aproximadamente 0,24 cal/g/°C para o ar).

T = Temperaturas interna e externa em °C.

Embora os mesmos princípios gerais de transferência por convecção se apliquem, quando a transferência de calor ocorre pelo movimento do ar que circunda a pele do RN existe uma equação que melhor a descreve, pois leva em conta a velocidade do ar:

$$\dot{Q}_{conv.} = h_c \times (T_{pele} - T_{ar})$$

onde:

h_c = coeficiente de transferência por convecção:

T_{pele} = Temperatura média da pele do recém nascido.

T_{ar} = Temperatura do ar em torno do recém nascido.

$$h_c = d^{0,67} \times (2,3 + 7,5 V^{0,67})$$

onde:

d = densidade do ar e V a velocidade do fluxo de ar.

3.2.3. Radiação

A radiação de calor é na realidade a emissão de ondas eletromagnéticas na faixa do infravermelho. Não é necessário nem mesmo que exista matéria entre os corpos emissor e receptor para que a troca ocorra.

Devemos entender que a origem da radiação está na geração da onda eletromagnética, e não no gradiente de temperatura. Ou seja o grau de emissão de infravermelho depende apenas do grau de agitação das cargas do emissor, isto é, depende apenas de sua temperatura.

Logo, não podemos evitar que o RN emita ondas eletromagnéticas, pois ele possui fontes de radiação de calor. O que podemos fazer é compensar a energia perdida provocando um fluxo líquido resultante de onda eletromagnética entrando no RN.

A equação que rege as trocas de calor por radiação é:

$$Q_{\text{rad.}} = \alpha (T_1^4 - T_2^4)$$

onde:

T_1 é a temperatura do corpo com maior temperatura e

T_2 o de menor.

α coeficiente que depend da área iluminada entre os corpos.

Se fosse possível espelhar a cúpula de tal forma que todo o infravermelho fosse refletido, o RN não apresentaria perdas por radiação tão elevadas. Uma tentativa de implementar este espelhamento na prática foi feita aplicando-se um filme filtro de infravermelho produzido pela 3M, cujo nome comercial é insulfilme. O resultado porém não foi aceito por dificultar muito a visualização do recém-nascido.

3.2.4. Evaporação

Ao nascer, o corpo da criança sai úmido. O procedimento de secar o recém nascido diminui as perdas por evaporação.

Normalmente o RN perde calor por evaporação de três formas:

- evaporação insensível pela superfície da pele.
- evaporação do suor.
- evaporação da mucosa do trato respiratório na expiração.

Cada grama de água perdida por evaporação, arrasta consigo 0,58 kcal de calor. Este valor representa o calor latente de evaporação da água. Por isto é importante controlar a umidade no interior da cúpula [10][11][12].

A equação que descreve a perda de calor por evaporação por unidade de tempo é:

$$\dot{Q}_{\text{evap.}} = 0,58 \text{ (kcal)} \cdot m_a$$

onde :

m_a = massa de água evaporada (g)

3.3. *Trocas de calor com o ambiente*

A importância relativa de cada um dos quatro mecanismos de transferência de calor varia de acordo com as condições ambientais internas e externas. As principais variáveis externas são a diferença entre a temperatura do corpo do RN e do ambiente, a velocidade do fluxo de ar, a umidade relativa e a área exposta para troca de calor (depende por exemplo se o RN está com roupas ou despido dentro da incubadora). Sob condições termoneutras as perdas de calor ocorrem, principalmente, por radiação e convecção.

Pode-se reduzir as perdas por radiação do RN, despido em uma incubadora de parede simples, aquecido por ar quente recirculante, de duas formas: ou aquecemos a parede da cúpula ou vestimos o RN.

A perda de calor por evaporação varia significativamente de acordo com as condições fisiológicas e ambientais. É aumentada quando a temperatura ambiental é alta, causando transpiração e hiperventilação ou qualquer outra condição patológica ou fisiológica que aumente a ventilação. É também influenciada inversamente pela umidade relativa do ambiente e diretamente pela velocidade do fluxo de ar em torno da superfície de evaporação.

O controle total das trocas de calor com o ambiente é uma característica fundamental de todos os animais homeotérmicos e representa a provisão essencial para dissipação do calor metabólico. Em temperaturas ambientes elevadas ou baixas temperaturas do corpo, a direção usual do fluxo de calor para o exterior é invertida e o armazenamento

3.3. *Trocas de calor com o ambiente*

A importância relativa de cada um dos quatro mecanismos de transferência de calor varia de acordo com as condições ambientais internas e externas. As principais variáveis externas são a diferença entre a temperatura do corpo do RN e do ambiente, a velocidade do fluxo de ar, a umidade relativa e a área exposta para troca de calor (depende por exemplo se o RN está com roupas ou despido dentro da incubadora). Sob condições termoneutras as perdas de calor ocorrem, principalmente, por radiação e convecção.

Pode-se reduzir as perdas por radiação do RN, despido em uma incubadora de parede simples, aquecido por ar quente recirculante, de duas formas: ou aquecemos a parede da cúpula ou vestimos o RN.

A perda de calor por evaporação varia significativamente de acordo com as condições fisiológicas e ambientais. É aumentada quando a temperatura ambiental é alta, causando transpiração e hiperventilação ou qualquer outra condição patológica ou fisiológica que aumente a ventilação. É também influenciada inversamente pela umidade relativa do ambiente e diretamente pela velocidade do fluxo de ar em torno da superfície de evaporação.

O controle total das trocas de calor com o ambiente é uma característica fundamental de todos os animais homeotérmicos e representa a provisão essencial para dissipação do calor metabólico. Em temperaturas ambientes elevadas ou baixas temperaturas do corpo, a direção usual do fluxo de calor para o exterior é invertida e o armazenamento

torna-se maior que a perda de calor, provocando aumento da temperatura corpórea.

A equação geral que descreve o equilíbrio térmico do corpo com o ambiente [13] que foi apresentada por Hardy em 1938 é a seguinte:

$$\dot{Q}_{\text{Metabolizado}} = \dot{Q}_{\text{Perdido}} + \dot{Q}_{\text{Armazenado}}$$

No equilíbrio térmico, o armazenamento de calor é zero, e a perda de calor é igual ao calor metabolizado. Esta é a condição fisiológica de equilíbrio. A perda excedendo a produção metabólica resulta em queda da temperatura corporal, enquanto que, por outro lado, o armazenamento de calor se manifesta por aumento da temperatura do corpo.

A alteração no total de calor armazenado no corpo pode ser calculada pela formula:

$$\Delta s = m \cdot c \cdot \frac{(0,6 \Delta T_{\text{int}} + 0,4 \Delta T_{\text{pele}})}{\Delta t}$$

onde:

Δs = Calor armazenado(+)ou perdido(-) [kcal/hora]

m = Massa do RN [kg]

c = Calor específico do corpo=0,84 [kcal/(kg °C)]

ΔT_{int} = Alteração da temperatura do corpo.[°C]

ΔT_{pele} = Variação da temperatura da pele.[°C]

Δt = Intervalo de tempo em horas.

As condições físicas do ambiente que influenciam a direção do fluxo de calor são: a temperatura do ar, a temperatura das superfícies de contato ou de radiação em torno do RN, a umidade relativa e a velocidade do ar.

O ambiente termoneutro é definido pela União Internacional das Sociedades de Fisiologia como:

"A faixa de temperatura ambiente na qual a taxa de metabolismo está em um mínimo e a regulação de temperatura é obtida por evaporação estando o indivíduo em equilíbrio térmico com o ambiente".

3.4. Os sistemas de umidificação e oxigenação e renovação do ar.

A forma de umidificação mais adequada, segundo testes "in vivo" no Hospital das Clínicas de Ribeirão Preto, foi por nebulizador externo de névoa aquecida, sendo esta técnica utilizada até mesmo nos modelos de incubadora que possuem sistema próprio de umidificação.

A oxigenação deve ser administrada por meio de capacete, reduzindo-se assim o consumo de O_2 do hospital.

Pelo fato desta incubadora renovar continuamente o ar, não existe possibilidade de alta concentração de O_2 reduzindo assim os riscos ao paciente e de explosões.

3.5. Os sistemas de alarme.

O conjunto "hardware" e "software" monitora com segurança, possíveis falhas no sistema de controle. São alarmadas as seguintes situações:

- Falta de circulação de ar.
- Ar muito quente.
- Ar muito frio.
- Hipertermia.
- Hipotermia.
- Falta de energia.

Em qualquer uma destas ocorrências, um alarme sonoro é disparado. Segundo a norma IEC 601-2-19 [16], e a ANSI/AAMI II36 item 3.2.3.6 [15] existe uma tecla para silenciar o alarme, mantendo a mensagem no display como alerta visual da situação de perigo.

4. CAPÍTULO 4 - OS SISTEMAS DE CONTROLE .

4.1. *A filosofia do sistema de controle.*

Existem dois modos de operação do sistema de controle do aquecimento, seguindo os itens 2.1.103 e 2.1.104 da norma IEC 601-02-19 e AAMI:

- Modo ATC (Air Temperature Controlled), no qual a temperatura do ar da cúpula é mantida no valor selecionado pelo operador. Neste modo o alarme de "AR QUENTE" é disparado quando a temperatura do ar na cúpula excede $1,0^{\circ}\text{C}$ da desejada e o de "AR FRIO" quando cai mais de $1,0^{\circ}\text{C}$ abaixo da desejada. Neste modo, ainda, a temperatura do recém nascido é apenas monitorada.
- Modo ITC (Infant Temperature Controlled), no qual a temperatura do recém nascido é mantida no valor selecionado pelo operador. Neste modo o alarme de "HIPERTERMIA" é disparado quando a temperatura do recém nascido excede $0,5^{\circ}\text{C}$ da desejada e o de "HIPOTERMIA" quando a temperatura do recém nascido cai $0,5^{\circ}\text{C}$ abaixo da desejada.

4.2. *Características físicas dos sensores*

4.2.1. Tempo de resposta

Sensores de temperatura extremamente rápidos ($27,0^{\circ}\text{C}$ à $37,0^{\circ}\text{C}$ em menos de 1 seg.), fabricados seguindo as mesmas técnicas de construção dos "strain-gauges", foram empregados.

O circuito do sensor de temperatura (apêndice B-5) serve para compatibilizar pequenas diferenças entre os sensores, permitindo assim intercambiabilidade após a calibração.

O tempo de resposta foi medido com o auxílio de dois tanques com água homogeneizada por agitadores, sendo um com temperatura estabilizada em $27,0^{\circ}\text{C}$ e outro em $37,0^{\circ}\text{C}$. Ao retirar o sensor do tanque de temperatura mais baixa, percorrendo 10 cm pelo ar do laboratório mantido em 22°C , e inseri-lo no tanque de $37,0^{\circ}\text{C}$, observa-se a estabilização na tensão de saída em menos de 1 seg.

Estes sensores recebem o nome de PTFR-50.

4.2.2. Imunidade as variações de temperatura do ar externo a parede da cúpula

Apesar da excelente uniformidade de temperatura sobre o colchão (diferença menor que $0,3^{\circ}\text{C}$), existem gradientes de temperatura tanto no lado onde o ar entra na cúpula como no lado oposto. A figura 6 ilustra a distribuição de temperaturas no interior da cúpula. A linha tracejada representa uma situação onde o ar externo está muito mais frio que o ar interno à cúpula.

Fig 5 - Representação dos valores de temperatura ao longo do comprimento da incubadora.

Após exaustivos testes (28 meses) observou-se que a temperatura sobre o colchão correspondia à média aritmética dos valores lidos por dois sensores estrategicamente posicionados; um na entrada e do ar e o outro no lado oposto na cúpula.

A temperatura sobre o colchão corresponde sempre à média aritmética dos valores lidos nos dois sensores e independe da variação da temperatura do ambiente externo. Este fato garante que o valor indicado pela incubadora corresponde à temperatura real no interior da cúpula.

Observou-se também que quanto mais frio estivesse o ar fora da cúpula, maior a diferença entre os valores lidos na entrada e na saída do ar da cúpula.

Apesar de afastarem-se os valores lidos pelos dois sensores, a média dos mesmos continua correspondendo à temperatura real. Assim sendo, quanto maior a diferença entre os valores lidos pelos dois sensores, maior a potência entregue pelo "software" a resistência, compensando desta forma as variações externas de temperatura.

4.2.3. O equilíbrio das temperaturas no interior da cúpula.

A norma AAMI no seu item 4.1.2(g) descreve assim o método de medida da uniformidade de temperatura: "Cinco sensores de temperatura calibrados devem ser colocados em um plano paralelo e a 10 cm acima do colchão no centro de cada uma das quatro áreas definidas por linhas dividindo o compartimento e a largura na metade e no centro do colchão".

Conforme pudemos observar nos gráficos obtidos com um registrador gráfico acoplado a cinco sensores de temperatura,

dispostos conforme determina o item 4.1.2(g) da norma ANSI/AAMI 1136-1991, a posição do colchão afeta a uniformidade de temperaturas em menos de 1°C.

4.3. Tipos de controle empregados.

Definições iniciais:

- **sensor 1** = sensor mais próximo à entrada do ar na cúpula, normalmente está mais quente que o **sensor 2**.
- **sensor 2** = sensor mais próximo a saída do ar da cúpula.
- **diferença** = valor lido pelo **sensor 1** - valor lido pelo **sensor 2**
- **temperatura real** = (valor **sensor 1** + valor **sensor 2**) / 2
- **ERRO** = temperatura desejada - temperatura real.
- **ciclo** = intervalo de tempo entre as interrupções para amostragens = 1/(120 Hz).
- **NL** = Número de ciclos com a resistência Ligada.

A seguir será feita a descrição de cada sistema de controle testado em laboratório.

4.3.1. Controle Liga/Desliga

Neste sistema, usamos a seguinte lógica: SE temperatura real > temperatura desejada ENTÃO LIGA A RESISTÊNCIA. SENÃO DESLIGA A RESISTÊNCIA.

O "ripple" apresentado foi muito alto (2,5°C), chegando a superar o das incubadoras convencionais.

4.3.2. Controle Proporcional

Neste sistema, liberamos uma porcentagem da potência total da resistência PROPORCIONAL à diferença entre a temperatura desejada e a real (definida anteriormente como ERRO).

Para graduar os percentuais de potência foi adotada a seguinte técnica:

Determinamos o número de níveis de potência que desejamos. No sistema implantado, escolhemos 80 níveis. Desta forma podemos graduar o percentual de potência entregue com intervalos de $1/80 = 1,25 \%$.

Para explicar como se processa esta graduação, demonstraremos com 4 níveis o que no sistema implantado ocorre com 80.

No sistema, um contador de ciclos é zerado toda vez que atinge o número de níveis de potência desejado (nesta demonstração = 4):

Fig 4- Conteúdo de variáveis e tensões ao longo do tempo.

A cada passagem do contador por zero, um novo valor de **ERRO** é calculado, o qual determinará o comportamento do sistema até que o contador retorne a zero.

O valor do **ERRO** determina quantos semi-ciclos da rede irão alimentar a resistência:

O sistema de controle opera determinando, a cada grupo de 80 ciclos, qual o número de ciclos que a resistência permanecerá ligada.

O percentual de tempo correspondente à resistência ligada equivale a $NL / 80$.

Ao decorrer mais um conjunto de 80 semi-ciclos, calcula-se o **ERRO** da seguinte forma:

SE $NL = 0$ **ENTÃO** $ERRO = \text{temperatura desejada} - \text{temperatura real}$

O valor do erro determina quantos semi-ciclos serão liberados para a resistência (NL).

Um problema apresentado pelo sistema de controle proporcional é que, para manter a temperatura dentro do compartimento do RN mais alta que a temperatura externa, tal controle tem sempre que liberar alguma potência para a resistência de aquecimento. Por isto, o controlador proporcional mantém sempre um erro necessário a manter o aquecimento. Este erro é conhecido como erro de regime permanente [14].

4.3.3. Controle Proporcional Integral

A fim de evitar o erro de regime, foi acrescentada a parcela erro anterior/2 no cálculo do novo erro. Desta forma estamos integrando o erro.

SE $NL = 0$

ENTÃO $\text{erro} = \text{temperatura desej} - \text{temperatura real} + \text{erro} / 2.$

A integração elimina o erro de regime. Entretanto, a resposta do sistema, medida segundo a normas AAMI e IEC, apresentou um over-shoot" de 3°C e um "ripple" de 0,5°C, os quais ainda podiam ser melhorados.

4.3.4. Controle Adaptativo

Para melhorar o nível de "over-shoot" e "ripple" foi usado um controle que delibera potência para a resistência de acordo com a integral do erro e ainda leva em conta a temperatura externa.

Levando-se em conta a peculiaridade de conseguirmos informações sobre a temperatura externa pela diferença entre os dois sensores, experimentalmente, obtivemos melhor resultado calculando o erro de acordo com a seguinte adaptação:

$$\text{ERRO} (t) = \text{SET POINT} - \text{TEMPERATURA REAL} + \text{ERRO} (t - 1)$$

$$\text{NL} = \text{ERRO} (t) + \text{DIFERENÇA}$$

Deste modo, quando o ar fora do compartimento do RN fica mais frio, automaticamente o percentual da potência total (NL) aumenta.

Isto caracteriza a adaptação rápida do sistema de controle, que responde à variação da temperatura externa antes que esta determine variação na interna.

Desta forma, quanto maior a diferença entre os valores lidos pelos dois sensores, maior a potência fornecida à resistência de aquecimento, compensando desta forma as variações externas de temperatura.

Assim o valor obtido de "over-shoot" caiu para 1,5°C e o "ripple" para 0,1°C medidos de acordo com a norma AAMI.

4.4. Conclusão

Dentre os diversos sistemas de controle experimentados, aquele que apresentou melhor resposta foi o ADAPTATIVO. Vale lembrar que o posicionamento dos sensores influi fortemente na resposta do sistema. Suas localizações definitivas foram resultado de cerca de um ano de experimentações com diversos materiais de sensores e suportes.

5. CAPÍTULO 5 - O HARDWARE.

5.1. Diagrama de blocos

5.2. Descrição da placa de CPU e suas calibrações.

A placa de CPU é composta pelos seguintes circuitos. O esquemático encontra-se no apêndice B-2:

5.2.1. Decodificação dos endereços:

A decodificação de endereços foi implementada com um único circuito integrado 74LS139 (duplo decodificador 2x4). Desta forma foi gerado o seguinte mapeamento de memória e de dispositivos de entrada e/ou saída.

ENDEREÇOS	função	local	componente
0000H a 7FFFH	RAM	IC 115	6264
8000H a FFFFH	EPROM	IC 100	27256
0000H e 0001H	CONVERSOR A/D	IC 20	AD7870
0002H e 0003H	SAÍDAS	IC 12	74LS373
0004H e 0005H	ENTRADAS	IC 15	74LS244
0006H e 0007H	DISPLAY INCUBADORA		
0008H e 0009H	DISPLAY BALANÇA		

5.2.2. Calibração:

Existem apenas 3 (três) ajustes na placa de CPU. Tais ajustes correspondem aos trimpots TP1, TP5, TP6, os quais apresentam as seguintes funções:

TP 1	ajuste do limiar de disparo do alarme de falta de circulação de ar.
TP 2	ajuste do limiar de tensão inferior na entrada do conversor A/D. (zero volts)
TP 3	Ajuste do limiar de tensão superior na entrada do conversor A/D. (5 volts)

A sequência de calibração tem que ser respeitada pois o ajuste do limiar de disparo do alarme de falta de circulação de ar depende diretamente da faixa dinâmica do sistema de aquisição. A calibração da faixa dinâmica do conversor A/D tem que ser executada antes de quaisquer ajustes.

Com o auxílio de um calibrador, que consiste de um regulador de tensão termocompensado, inserimos pelo conector do RN as tensões de 0,000V e 5,000V.

O "software" de calibração é acionado quando as teclas de set-point são pressionadas simultaneamente. Ao identificar esta combinação de teclas, a CPU passa a executar um "loop" onde amostra continuamente o canal 3, correspondente ao sensor do RN, mostrando o resultado da conversão em decimal no "display" I.

A calibração consiste em:

- 1) Pressionar as teclas de "set-point" simultaneamente;
- 2) Inserir o calibrador com a chave comutada na posição 0v.
- 3) Ajustar TP2 até que o valor indicado no "display" esteja no limiar entre 0000 e 0001.
- 4) Comutar a chave do calibrador para a posição 5v.
- 5) Ajustar TP3 até que o valor indicado no "display" esteja no limiar entre 4094 e 4095.

5.3. Descrição da placa de fonte.

A placa de alimentação é composta pelos seguintes circuitos. O esquemático encontra-se no apêndice B-3:

5.3.1. Tensão de + 12 Vcc.

Regulada e protegida contra curto-circuito pelo circuito integrado 7812, alimenta, o conversor A/D, o multiplexador, os sensores de temperatura, o amplificador da balança, o mini-ventilador.

É composto pelos componentes: IC1, D4, D10, C1, C4, C7.

5.3.2. Tensão de - 12 Vcc.

Regulada e protegida contra curto-circuito pelo circuito integrado 7912, alimenta os sensores de temperatura, o amplificador da balança, o conversor A/D e o "buzzer".

É composto pelos componentes: IC3, D5, D9, C2, C5, C8.

5.3.3. Tensão de + 5 Vcc.

Regulada e protegida contra curto-circuito pelo circuito integrado LM 323-05, é dotado de "NO-BREAK". Alimenta os circuitos TTL, a CPU e as memórias.

É composto pelos componentes: IC2, D1, D2, D3, D8, C3, C6, C9, C10.

5.3.4. Alimentação do "Backlight" e Sinal de frequência da rede dobrada.

É composto pelos componentes: D6, D7, R2, R4, R5. Este circuito possui dupla função nos modelos que apresentam "DISPLAY" com "backlight". No modelo **SC** temos apenas um "DISPLAY" com "backlight", portanto limitamos a corrente com três resistores de 5,6 ohms enquanto no modelo **S**, não há "backlight".

Além de alimentar o "backlight", esta parte da fonte, gera uma senóide retificada em onda completa. Na placa de CPU, este sinal passa por um comparador com histerese implementado com os resistores R43 a R46 e 1/4 do IC11 (LM 324), sendo depois transformado em pulsos de interrupção pelo monoestável formado por R17, R22 à R24, C5 e 1/4 do IC11. As interrupções são geradas com o dobro da frequência da rede, permitindo assim a filtragem digital dos ruídos induzidos pela linha.

5.3.5. Tensão da bateria.

Em casos de falta de energia da rede, a bateria mantém alimentados o "buzzer", a CPU em "halt" e a memória por mais de 15 minutos. Desta forma o item 101.3 da norma IEC 601-2-19 de 1990-12 e o item 3.2.3.5 "Power failure" da norma ANSI/AAMI 1136-1991 são plenamente atendidos.

Caso a carga da bateria seja insuficiente, os Relés RL1, 2 e 3 não permite que a incubadora seja ligada. Em caso de emergência, a incubadora pode ser usada mesmo com carga

insuficiente para operação normal, bastando para isto comutar a chave "cargador da bateria" para a posição "ligado".

5.3.6. Tensão de alimentação da resistência de aquecimento:

É composto pelos componentes: SCR 1, D11. Controlada pela placa de CPU, fica isolada desta pelo acoplador óptico MOC3041.

5.4. Descrição da placa de sensores de temperatura.

O circuito de conformação do sinal de temperatura, visto no apêndice B-5, funciona ajustando a reta de transferencia temperatura versus tensão. Os dois graus de liberdade necessários para este ajuste foram obtidos por meio de "trimpots" batizados de "zero" e "ganho".

O "trimpot" de "ZERO" está inserido em um dos braços da ponte.

O "trimpot" de "GANHO" está inserido no braço de realimentação do segundo amplificador.

5.5. Descrição da placa de balança e sua calibração.

Conforme podemos observar no esquema do apêndice B-4, a calibração da balança se faz por meio da tensão que alimenta a célula de carga.

Cada "bit" do conversor A/D corresponde a 2 gramas. Desta forma a capacidade máxima da balança (tara + peso útil) corresponde a : $2^{12} \times 2$ gramas = 8192 gramas.

Assim sendo a sensibilidade do conjunto célula mais amplificador tem que ser: 5 volts / 4096 / 2 gramas, ou seja, 0,61 milivolts/grama.

A calibração na prática é muito simples: após certificar-se que a bandeja está bem posicionada (apoiada apenas pelos quatro pinos), liga-se a incubadora, seleciona-se 28°C e aguarda-se terminar a fase de pré aquecimento. Após esta fase, a balança é automaticamente zerada por "software", o qual considera tudo que se encontrava sobre a célula de carga como TARA. Deve ser colocado, então, sobre a bandeja um peso padrão (precisão de gramas). Ajusta-se o "trimpot" de GANHO do amplificador da balança, até obter-se no "display", o valor real do peso padrão.

Para verificar se a calibração está correta e se existe repetibilidade na medida, retira-se o peso padrão, colocando outro qualquer em seu lugar. Através da tecla "C" no painel da balança zera-se novamente a mesma. Coloca-se novamente o peso padrão. O valor indicado tem que ser o mesmo.

Deve também ser verificado se o valor indicado é o mesmo para diferentes pontos de apoio do peso sobre a bandeja.

O sistema ainda memoriza, a cada minuto, medidas de pesos em um "buffer" de 22 horas. Isto permite à enfermeira verificar a evolução do peso devido a alimentação ou excreção

É importante observar que a soma dos pesos sobre a célula de carga não pode superar 8 kg.

5.6. Descrição da placa de acionamento dos motores.

Um sistema eletromecânico movimenta a bandeja do colchão para as posições de "Trendelemburg", "Fowler", horizontal alta e horizontal baixa.

Pressionada a tecla, o lado correspondente da bandeja é movimentado até um limite pré-fixado que, uma vez atingido, desliga automaticamente o sistema elétrico acionado.

Neste sistema, as funções são acionadas uma de cada vez, pelo que, enquanto estiver uma tecla pressionada, nenhuma das de menor prioridade funcionará.

Esta hierarquia de acionamentos é uma característica desejável pois obriga o operador estar atento a apenas um movimento de cada vez.

Por outro lado, não é esta a principal razão para esta característica funcional. Suponhamos que os três motores pudessem ser acionados ao mesmo tempo. Neste caso o circuito de potência teria obrigatoriamente que ser dimensionado levando-se em conta a possibilidade dos três motores serem acionados simultaneamente.

Com o circuito implementado, conseguimos reduzir consideravelmente a dimensão, o peso e o custo do transformador e do retificador.

6. CONCLUSÃO

6.1. Resultados obtidos

De acordo com a norma AAMI no seu capítulo 4 (tests), cujos tópicos estão traduzidos entre aspas, os seguintes resultados foram obtidos:

Fig. 5 - Resultados de acordo com a norma ANSI/AAMI II36-1991

a) **TEMPO DE SUBIDA DA TEMPERATURA** "Com a incubadora em uma temperatura ambiente de $20 \pm 1,0^{\circ}\text{C}$, selecione a temperatura de controle em seu valor máximo. Meça o tempo requerido para a TEMPERATURA DA INCUBADORA aumentar 10°C " (de $20,0^{\circ}\text{C}$ a $30,0^{\circ}\text{C}$) = **18 minutos.**

b) **OVERSHOOT** "Com a incubadora em uma temperatura ambiente de $25 \pm 1,0^{\circ}\text{C}$, deixe ela atingir O EQUILÍBRIO DE TEMPERATURA com a temperatura de controle no modo ATC, selecionada em 36°C . Reselecione para 32°C e deixe ela atingir O EQUILÍBRIO DE TEMPERATURA. Meça com precisão de $0,1^{\circ}\text{C}$ a TEMPERATURA DA INCUBADORA, calculando a diferença entre a média e a máxima" ($T_{\text{ambiente}} = 25^{\circ}\text{C}$; de $32,0^{\circ}\text{C}$ a $36,0^{\circ}\text{C}$) = **+1,5°C.**

UNDERSHOOT "Com a incubadora em uma temperatura ambiente de $25 \pm 1,0^{\circ}\text{C}$, deixe ela atingir o EQUILÍBRIO DE TEMPERATURA com a temperatura de controle no modo ATC, selecionada em 36°C . Reselecione para 32°C e deixe ela atingir o EQUILÍBRIO DE TEMPERATURA. Meça com precisão de $0,1^{\circ}\text{C}$ a TEMPERATURA DA INCUBADORA, calculando a diferença entre a média e a mínima" ($T_{\text{ambiente}} = 25^{\circ}\text{C}$; de $36,0^{\circ}\text{C}$ a $32,0^{\circ}\text{C}$) = $0,9^{\circ}\text{C}$.

c) Tempo requerido para alcançar a TEMPERATURA DE EQUILÍBRIO:

"Tempo requerido para a temperatura atingir o equilíbrio, em modo ATC, seguindo o teste descrito nos item **b)** acima, medido com precisão de minuto."

($36,0^{\circ}\text{C}$ até $32,0^{\circ}\text{C}$) = **45 min.**

($32,0^{\circ}\text{C}$ até $36,0^{\circ}\text{C}$) = **72 min.**

VARIABILIDADE DA TEMPERATURA: "As diferenças entre as temperaturas máxima e mínima da incubadora durante o EQUILÍBRIO em 32°C e em 36°C , medidas com precisão de $0,1^{\circ}\text{C}$."

para 32°C , = **$0,1^{\circ}\text{C}$** .

para 36°C , = **$0,1^{\circ}\text{C}$** .

d) TEMPO DE RECUPERAÇÃO DE TEMPERATURA EM INTERVENÇÕES:

"Partindo da TEMPERATURA DE EQUILÍBRIO em 36°C , abrir as portinholas para ter acesso ao paciente com as duas mãos. Deixe as portinholas abertas por 15 minutos, fechando-as em seguida. Repita o procedimento abrindo totalmente a cúpula.

Medir: A TEMPERATURA DA INCUBADORA mínima, medida com precisão de $0,1^{\circ}\text{C}$; tempo de recuperação, medido com precisão de minuto, a partir do instante de fechamento do acesso."

TEMPERATURA DA INCUBADORA mínima = $35,6^{\circ}\text{C}$.

tempo de recuperação (portinholas)= 4 minutos.

tempo de recuperação (cúpula)= 6 minutos.

e) FAIXA DE CONTROLE TEMPERATURA:

modo ATC = (de $28,0^{\circ}\text{C}$ até $39,0^{\circ}\text{C}$)

modo ITC = (de $35,0^{\circ}\text{C}$ até $37,0^{\circ}\text{C}$)

f) UNIFORMIDADE DE TEMPERATURAS: "Cinco sensores de temperatura calibrados, devem ser dispostos em um plano paralelo e 10 cm acima do colchão, sendo 4 nos centros de cada uma das quatro áreas definidas por linhas que dividem ao meio o colchão nos sentidos da largura e do comprimento, e um último acima da intercessão destas linhas."

Durante o equilíbrio de temperatura, tanto em 32 como em 36°C, medir com precisão de 0,1°C a temperatura de cada sensor com o colchão na horizontal e nas demais posições limites recomendadas pelo fabricante.

- g) Tempo para alarmar após ocorrência de falha na circulação de ar: 2 seg.
- h) Nível máximo de ruído sonoro no compartimento:
- i) operação normal 56 dB;
- j) alarmando 58 dB.
- k) Nível máximo do alarme: 68 dB.

Após os testes de laboratório duas incubadoras foram enviadas para unidades de cuidado intensivo para um longo período de testes "in vivo", no Hospital das Clínicas de Ribeirão Preto. A segurança e operacionalidade dos equipamentos foram comprovadas em uso diário de rotina.

16 recém-nascidos pré-termo pesando entre 1000 g e 1760 g com idade gestacional entre 27 e 32 semanas, foram manejados cuidadosamente durante períodos contínuos de pelo menos 12 horas. Todo paciente tratado no modelo SCL, era pesado em uma balança com precisão de uma grama. Em nenhum caso houve discrepância maior que uma grama entre o valor fornecido por esta balança e o da incubadora. Toda a equipe de enfermagem, incluindo estagiárias, tomaram contato com a incubadora e preencheram um formulário dando origem à um relatório de avaliação. Tal relatório,

encaminhado para a indústria, indicou que o equipamento foi aprovado para início de produção.

O objetivo do trabalho foi plenamente atingido tendo em vista que o projeto construção e teste, deu origem à uma família totalmente inovadora de incubadoras para recém nascidos de risco.

6.2. *Propostas para trabalhos futuros*

Como sugestões para melhoria da incubadora podemos citar os seguintes trabalhos:

- Substituição da CPU 8088 e memórias por um microcontrolador de 16 bits. O investimento na substituição seria pago com a redução no custo de produção.
- Inclusão de um mecanismo que permitisse o posicionamento manual do colchão do paciente em caso de falta de energia
- Como sofisticação já é possível pensar em uma interface mais amigável com os operadores da incubadora, tendo em vista a redução do custo dos "displays" gráficos.

Separar a resistência de aquecimento em 2(dois) braços, de tal forma que em 110V estes fiquem em paralelo e em 220V em série. Com isto aumentamos a área para troca de calor entre a resistência e o fluxo de ar, diminuindo a impedância térmica de condução.

7. REFERÊNCIAS BIBLIOGRÁFICAS

- 1 Brezina Z., Gawlas W., Gebauerova D., Fricova V., Hloskova V., Mackowska E., Torsova V., Varekova E., Zarska Z. "Which disinfectant should be used in incubators?", *Cesk Pediatr.* 1990 Dec. 45(12). P 745.
- 2 Kjelsberg AB., Krogsaeter R. "Cleaning and disinfection of incubators. New guidelines from the National Institute for Public Health", *Sykepleien.* 1988 Aug 11. 76(15). P 28-30.
- 3 Daschner F., Helwig H. "Disinfection of incubators: Draeger-Aseptor or scrub and wipe disinfection?", *Monatsschr Kinderheilkd.* 1985 Jan. 133(1)P 56.
- 4 Heuschkel HJ. Felscher D. "Iatrogenic inhalation of disinfectants [formalin (formaldehyde, methanol)], phenol during incubator care and CPAP respiratory assistance of a newborn infant with respiratory distress syndrome", *Z Arztl Fortbild (Jena).* 1983. 77(2). P 88-91.
- 5 Hashimoto T. "A comparison of the amount of dust produced by disposable and cloth diapers in incubators.", *Pediatrician.* 1987. 14 Suppl 1P 44-7.
- 6 Ziegler P. Zwacka G. Holzapfel G. Senf L. Schimmel G. Jung B. Gabler L. "Effectively decreasing residues following use of formaldehyde disinfection of incubators", *Kinderarztl Prax.* 1986 Jul. 54(7). P 381-6.
- 7 Ziegler P., Frenzel J., Schoel U. "Relevance of hospital hygiene for incubators", *Kinderarztl Prax.* 1986 Jan. 54(1). P 21-4.
- 8 Konrad Diem "Documnta Geigy Tablas Científicas.", Editora Geigy, Buenos Aires, Sexta edición ,1965.
- 9 Topper WH., Stewart TP. "Thermal support for the very-low-birth-weight infant: role of supplemental conductive heat", *J Pediatr.* 1984 Nov. 105(5). P 810-4.
- 10 Thompson MH., Stothers JK., McLellan NJ. "Weight and water loss in the neonate in natural and forced convection", *Arch Dis Child.* 1984 Oct. 59(10). P 951-6.
- 11 Harpin VA., Rutter N. "Humidification of incubators", *Arch Dis Child.* 1985 Mar. 60(3). P 219-24.
- 12 Fitch CW. Korones SB. "Heat shield reduces water loss", *Arch Dis Child.* 1984 Sep. 59(9). P 886-8.
- 13 Hardy JD, Soderstorm GF. "Heat loss from the nude body and peripheral blood flow at temperatures of 22 to 35 °C", *J Nutr* 16:493-510, 1938.
- 14 Wesley E.Snyder "Industrial Robots", McGraw-Hill, New York

7. REFERÊNCIAS BIBLIOGRÁFICAS

- 1 Brezina Z., Gawlas W., Gebauerova D., Fricova V., Hloskova V., Mackowska E., Torsova V., Varekova E., Zarska Z. "Which disinfectant should be used in incubators?", *Cesk Pediatr.* 1990 Dec. 45(12). P 745.
- 2 Kjelsberg AB., Krogsaeter R. "Cleaning and disinfection of incubators. New guidelines from the National Institute for Public Health", *Sykepleien.* 1988 Aug 11. 76(15). P 28-30.
- 3 Daschner F., Helwig H. "Disinfection of incubators: Draeger-Aseptor or scrub and wipe disinfection?", *Monatsschr Kinderheilkd.* 1985 Jan. 133(1)P 56.
- 4 Heuschkel HJ. Felscher D. "Iatrogenic inhalation of disinfectants [formalin (formaldehyde, methanol)], phenol during incubator care and CPAP respiratory assistance of a newborn infant with respiratory distress syndrome", *Z Arztl Fortbild (Jena).* 1983. 77(2). P 88-91.
- 5 Hashimoto T. "A comparison of the amount of dust produced by disposable and cloth diapers in incubators.", *Pediatrician.* 1987. 14 Suppl 1P 44-7.
- 6 Ziegler P. Zwacka G. Holzapfel G. Senf L. Schimmel G. Jung B. Gabler L. "Effectively decreasing residues following use of formaldehyde disinfection of incubators", *Kinderarztl Prax.* 1986 Jul. 54(7). P 381-6.
- 7 Ziegler P., Frenzel J., Schoel U. "Relevance of hospital hygiene for incubators", *Kinderarztl Prax.* 1986 Jan. 54(1). P 21-4.
- 8 Konrad Diem "Documnta Geigy Tablas Científicas.", Editora Geigy, Buenos Aires, Sexta edición ,1965.
- 9 Topper WH., Stewart TP. "Thermal support for the very-low-birth-weight infant: role of supplemental conductive heat", *J Pediatr.* 1984 Nov. 105(5). P 810-4.
- 10 Thompson MH., Stothers JK., McLellan NJ. "Weight and water loss in the neonate in natural and forced convection", *Arch Dis Child.* 1984 Oct. 59(10). P 951-6.
- 11 Harpin VA., Rutter N. "Humidification of incubators", *Arch Dis Child.* 1985 Mar. 60(3). P 219-24.
- 12 Fitch CW. Korones SB. "Heat shield reduces water loss", *Arch Dis Child.* 1984 Sep. 59(9). P 886-8.
- 13 Hardy JD, Soderstorm GF. "Heat loss from the nude body and peripheral blood flow at temperatures of 22 to 35 °C", *J Nutr* 16:493-510, 1938.

1985.

- 15 AAMI Associatio for the Advancement of Medical Instrumentation. "American National Standard Infant incubators" II36-1991
- 16 International Electrotechnical Commision. "Particular requirements for safety of baby incubators". 601-2-19, 1990.
- 17 International Electrotechnical Commision. "General requirements for safety of Electromedical Devices". 601-01-19, 1990.

**APÊNDICE A - TRABALHO APRESENTADO NO
" WORLD CONGRESS ON MEDICAL PHYSICS AND
BIOMEDICAL ENGINEERING AGO 94 "**

Giuseppe M. STEPHAN¹, C. A. PELA², Salim MOYSES J.³, Arthur L. GONÇALVES³
 1) OLIDEF Av. Patriarca, 2223 CEP. 14031-580 RIB. PRETO-SP, Brazil, Fax+55166371741
 2) CIDRA-FFCL USP RP 3) Hospital das Clinicas Faculdade de Medicina USP RP.

[INTRODUCTION] A new concept of incubator, aiming to make easier and safer the care for risk newborns was developed, constructed and tested.

[MATERIALS AND METHODS] The incubator's hardware is an interchangeable module consisting of a 8088 CPU, 32K EPROM, 8K RAM and 8 chanel 12 bit analog to digital converter. The innovatory properties of the device are: a continuos air renovation; an aperture system that gives access to both sides and head of the baby; a quick and easy positioning X-ray chassis; a continuous infant wheight monitoring system; an electromechanical dispositive to change infant position; besides of a baby compartment without rugosities, reducing contamination risk.

[RESULTS] The performance evaluation was made based on the ANSI/AAMI I136-1991 and IEC 601-2-19 STANDARDS. 16 pre-term infants (gestatinal age 27 to 32 weeks and birth-weight 1000 g to 1760 g) were managed carefully and thoroughly during continuous periods of at least 12 hours in special care baby units (enviroment temperature 20 to 26 C). The nurses aproved performance and reability and gave some ergonomic sugestions. Results bellow:

Fig 1 - Results according to ANSI/AAMI I136-1991

Fig 2 - Mattress positions.

[CONCLUSION] Sugestions to make use easier are being implemented for new tests.

APÊNDICE B - DIAGRAMAS DO HARDWARE

- B1- DIAGRAMA EM BLOCOS
- B2 - PLACA DE CPU
- B3- FONTE DE ALIMENTAÇÃO
- B4 - AMPLIFICADOR DA BALANÇA
- B5- SENSOR DE TEMPERATURA
- B6 - ACIONAMENTO DOS MOTORES

**APÊNDICE A - TRABALHO APRESENTADO NO
" WORLD CONGRESS ON MEDICAL PHYSICS AND
BIOMEDICAL ENGINEERING AGO 94 "**

1 2 3
Giuseppe M. STEPHAN¹, C. A. PELA², Salim MOYSES J.³, Arthur L. GONÇALVES³
1) OLIDEF Av. Patriarca, 2223 CEP. 14031-580 RIB. PRETO-SP, Brazil, Fax+55166371741
2) CIDRA-FFCL USP RP 3) Hospital das Clinicas Faculdade de Medicina USP RP.

[INTRODUCTION] A new concept of incubator, aiming to make easier and safer the care for risk newborns was developed, constructed and tested.

[MATERIALS AND METHODS] The incubator's hardware is an interchangeable module consisting of a 8088 CPU, 32K EPROM, 8K RAM and 8 channel 12 bit analog to digital converter. The innovatory properties of the device are: a continuous air renovation; an aperture system that gives access to both sides and head of the baby; a quick and easy positioning X-ray chassis; a continuous infant weight monitoring system; an electromechanical dispositive to change infant position; besides of a baby compartment without rugosities, reducing contamination risk.

[RESULTS] The performance evaluation was made based on the ANSI/AAMI I136-1991 and IEC 601-2-19 STANDARDS. 16 pre-term infants (gestational age 27 to 32 weeks and birth-weight 1000 g to 1760 g) were managed carefully and thoroughly during continuous periods of at least 12 hours in special care baby units (environment temperature 20 to 26 C). The nurses approved performance and reliability and gave some ergonomic suggestions. Results below:

Fig 1 - Results according to ANSI/AAMI I136-1991

Fig 2 - Mattress positions.

[CONCLUSION] Suggestions to make use easier are being implemented for new tests.

APÊNDICE B - DIAGRAMAS DO HARDWARE

- B1- DIAGRAMA EM BLOCOS
- B2 - PLACA DE CPU
- B3- FONTE DE ALIMENTAÇÃO
- B4 - AMPLIFICADOR DA BALANÇA
- B5- SENSOR DE TEMPERATURA
- B6 - ACIONAMENTO DOS MOTORES

APÊNDICE C - LISTAGEM DO SOFTWARE

IBM PC-DOS PL/M-86 V2.0 COMPILATION OF MODULE INCUBA
OBJECT MODULE PLACED IN INCUBA.OBJ
COMPILER INVOKED BY: PLM86 INCUBA.DOC

```
/* **** */
/* data: 16.JUN.1993 tarde */
/* NOME DO ARQUIVO: INCUBA.PLM */
/* FUNCAO: MONITORAR TECLADO, FUNCIONAMENTO DA INCUBADORA */
/* CONTROLAR TEMPERATURA. */
/* **** */
```

```
$OVERFLOW
$CODE
$NOINTVECTOR
$LARGE
$DEBUG
$OPTIMIZE(2)
```

```
/* tipo de controle empregado: */
```

```
$ set(ligdes=0h)
$ set(prop=0h)
$ set(int=0h)
$ set(amb=0h)
$ set(difere=0ffh)
```

```
$ set(amos=0ffh)
```

```
$ set(ekg=0h)
```

```
$ set(bal=0ffh)
```

```
$ set(ser=0h)
```

```
$ set(bl=0FFh)
```

```
$ set(so2=0h)
```

```
$ set(s=0h)
```

```
$ set(cir=0FFh)
```

```
$ set(teste=0h)
```

```
$ set(loop=0h)
```


PL/M-86 COMPILER INCUBA
10/11/95 07:38:23 PAGE 2

```

1      INCUBA:
      DO;

      /******
      /*          VARIÁVEIS GLOBAIS          */
      /******

2 1    DECLARE
      A      BYTE,

      /******
      /*          IMAGENS DE SAIDAS          */
      /******
      IMSAIDASRW      BYTE,          /* LEDS DO PAINEL , CONTROLE DA RESISTENCIA */
                                          /* DE AQUECIMENTO, ENDEREÇOS DO MUX A/D e */
                                          /* BUZZER. */

      /******
      /*          ENDEREÇOS DE VARIÁVEIS      */
      /******

      $ if bal = Offh
      $ if ser = Offh
          nada      byte,
          point     BYTE,
          SERIAL    BYTE,
      $ endif

          POINTIN   INTEGER,
          tara      integer,
          media     INTEGER,
          atual     INTEGER,
          PESO(1380) INTEGER, /* 23 horas x 60 minutos = 1380 */

          ciclo     BYTE,
          seg       INTEGER,
          min       INTEGER,
          hora      INTEGER,

          cursor    integer,

      $ endif

      $ if ekq = Offh
          ecg        INTEGER,

      $ endif

          TEMPSARS1(7) integer,
          TEMPSARS2(7) integer,
          TEMPSARSREAL integer,
          TEMPSARS1SFILT integer,
          TEMPSARS2SFILT integer,

          temp$amb   integer,

```

PL/M-86 COMPILER INCUBA
10/11/95 07:38:23 PAGE 3

```

/* TEMP$RNS1(7)    integer,    */
/* TEMP$RNS2(7)    integer,    */
/* TEMP$RNS$REAL    integer,    */
/* TEMP$RNS1$FILT   integer,    */
/* TEMP$RNS2$FILT   integer,    */

AR$DESEJ           integer,
RN$DESEJ           integer,

SET$POINT          integer,
temp$real          integer,
faixar            integer,
ERRO              integer,
dif              integer,
TEMPO1            integer,      /* tempo do alarme silenciado */
TEMPO2            integer,      /* bouce da tecla modo */
TEMPO3            integer,      /* bouce da tecla adianta */
TEMPO4            integer,      /* bouce da tecla atrasa */
TEMPO5            integer,      /* bouce da tecla aumenta */
TEMPO6            integer,      /* bouce da tecla diminui */
TEMPO7            byte,         /* espera S&H estabilizar */
TEMPO8            byte,         /* bouce da silencia alarme */

MODO      BYTE,
senrn     byte,

$ if so2 = Offh

    esto2  BYTE,    /* 0 = com fluxo, 1 = sem fluxo */

$ endif

I      integer,
i1     byte,
i2     byte,
i3     byte,
i4     byte,
i5     byte,
i6     byte,
iv     byte,
ca     byte,

/* ----- */
/* ca |          CAUSA DO ALARME */
/* ----- */
/* 01 | SENSOR DO AR 1 COM DEFEITO */
/* ----- */
/* 02 | SENSOR DO AR 2 COM DEFEITO */
/* ----- */
/* 03 | AR 1.C +FRIO modo ar */
/* ----- */
/* 04 | AR 1.C +QUENTE modo ar */
/* ----- */
/* 05 | RN HIPOTERMICO (< setpoint -.5.C) */
/* ----- */
/* 06 | RN HIPERTERMICO (> setpoint +.5.C) */
/* ----- */
/* 08 | OXIGENIO */
/* ----- */
/* 09 | AR NAO CIRCULA */
/* ----- */
/* 13 | AR 4.C +FRIO modo RN */
/* ----- */
/* 14 | AR 4.C +QUENTE modo RN */
/* ----- */
/* 38 | TEMPERATURA DESEJADA DO AR > 38.C */
/* ----- */
/* FE | FALHA NA ENERGIA */
/* ----- */
/* FF | BACK LIGHT */
/* ----- */

NL      BYTE;    /* Numero de ciclos Ligado ou desligado */

```

PL/M-86 COMPILER INCUBA
10/11/95 07:38:23 PAGE 4

```

/*****
/*          CONSTANTES          */
*****/

3 1 declare
    fora    LITERALLY  '000H',
    ativo   LITERALLY  'OFFH',

    AR      LITERALLY  '000H',
    RN      LITERALLY  'OFFH';

/*****
/* TABELA DE LINEARIZACAO NTC 60k */
*****/

4 1 DECLARE
    linear05C LITERALLY  '409',
    linear10C LITERALLY  '819',
    linear15C LITERALLY  '1228',
    linear20C LITERALLY  '1638',
    linear25C LITERALLY  '2047',
    linear30C LITERALLY  '2457',
    linear35C LITERALLY  '2866',
    linear40C LITERALLY  '3276',
    linear45C LITERALLY  '3685',
    linear50C LITERALLY  '4095';

/*****
/*          MENSAGENS          */
*****/

5 1 DECLARE
    CALIBRA1(*)  BYTE  DATA ('retira sens.RN e'),
    CALIBRA2(*)  BYTE  DATA ('p',5,'e CALIBR. '),
    $ if s=Offh
        CALIBRA3(*)  BYTE  DATA ('mod. S 16.JUN.93'),
    $ endif
    $ if s=0h
        CALIBRA3(*)  BYTE  DATA ('mod.SC 16.JUN.93'),
    $ endif
    CALIBRA4(*)  BYTE  DATA ('*** DESLIGUE ***'),
    HIPO(*)      BYTE  DATA ('HIPOTERM. ',6),
    HIPER(*)     BYTE  DATA ('HIPERTERM.',6),
    CONHECE1(*)  BYTE  DATA ('Ja usou a matrix'),
    CONHECE2(*)  BYTE  DATA ('? sim ^ nao v '),
    ESCOLHE1(*)  BYTE  DATA ('QUAL TEMPERATURA'),
    ESCOLHE2(*)  BYTE  DATA ('DESEJADA ? '),
    AGUARDE(*)   BYTE  DATA ('c',3,4,'ula com '),
    ARCUPULA(*)  BYTE  DATA ('ESPERE AT',7,' '),
    PRONTO1(*)   BYTE  DATA (' APERTE A TECLA '),
    PRONTO2(*)   BYTE  DATA (' SILENCIA ALARME'),
    $ if s = Offh
        TEMPERATURAS(*) BYTE  DATA ('c',3,4,'ula com '),
    $ endif
    $ if s = 0h
        TEMPERATURAS(*) BYTE  DATA ('RN __,_,0,' ar'),
    $ endif
    DESEJ$RN(*)  BYTE  DATA ('Deseja RN ',2),
    DESEJ$AR(*)  BYTE  DATA ('Deseja ar ',2),
    CULA(*)      BYTE  DATA (' ar n',1,'o CIRCULA '),
    ARSENSOR1(*) BYTE  DATA ('sensor AR 1 fora'),
    ARSENSOR2(*) BYTE  DATA ('sensor AR 2 fora'),
    RSENSOR1(*)  BYTE  DATA ('sensor RN 1 fora'),
    RSENSOR2(*)  BYTE  DATA ('sensor RN 2 fora'),
    FRIO(*)      BYTE  DATA ('AR FRIO ',6),
    QUENTE(*)    BYTE  DATA ('AR QUENTE ',6),
    CALCULO(*)   BYTE  DATA ('CALCULOS ERRADOS'),
    ENERGIA(*)   BYTE  DATA ('FALHA DE ENERGIA'),
    ENERGIA2(*)  BYTE  DATA ('DESLIGUE e LIGUE'),
    ASCII(*)     BYTE  DATA ('0123456789-+'),
    $ if bal = Offh
        DEFEITO1(*)  BYTE  DATA ('DEFEITO NA PORTA'),
        DEFEITO2(*)  BYTE  DATA ('SERIAL DO PESO'),
        BALANCA(*)   BYTE  DATA ('000:00:00 0000 g'),
        ZERA1(*)     BYTE  DATA ('antes de colocar'),
        ZERA2(*)     BYTE  DATA ('o RN, tecle "C" '),
        ANTES(*)     BYTE  DATA ('00h antes ---- g'),
    $ endif
    $ if so2 = Offh
        AVISO02(*)   BYTE  DATA ('VEJA O OXIGENIO '),
    $ endif

```


PL/M-86 COMPILER INCUBA
10/11/95 07:38:23 PAGE 5

CGRAMRW(*)	BYTE	DATA
		(10000b,
		00110b,
		01001b,
		01000b,
		01000b,
		01001b,
		00110b,
		00000b,
		01101b,
		10010b,
		01110b,
		00001b,
		01111b,
		10001b,
		01111b,
		00000b,
		10000b,
		00000b,
		01110b,
		00001b,
		01111b,
		10001b,
		01111b,
		00000b,
		00110b,
		00100b,
		10001b,
		10001b,
		10001b,
		10011b,
		01101b,
		00000b,
		00000b,
		00000b,
		11110b,
		10001b,
		11110b,
		10000b,
		10000b,
		10000b,
		01101b,
		10010b,
		01110b,
		10001b,
		10001b,
		10001b,
		01110b,
		00000b,
		00100b,
		01110b,
		11111b,
		00000b,
		11111b,
		01110b,
		00100b,
		00000b,
		00011b,
		00010b,
		11111b,
		10000b,
		11111b,
		10000b,
		11111b,
		00000b),

```
$ if bal = 0ffh
```

[illegible]

```

/*****
/* ENDERECOS DE ENTRADA E SAIDA */
*****/

```

```

          CRISTALRW      LITERALLY  '00006H',      /* '08000H',  */
          CRISTALBAL     LITERALLY  '08000H',      /* '00006H',  */

          SERIALSDADO    LITERALLY  '08002H',
          SERIALSCOMANDO LITERALLY  '08003H',
          SERIALSSTAT    LITERALLY  '08003H',

          ENTRADAS       LITERALLY  '00004H',

TADIANTA */      Tmodo      LITERALLY  '00000001b', /* ( ICTI = RN ou ATCI = AR )   era
era TATRASA */    Tsilencia LITERALLY  '00000010b', /* silencia ALARME POR 30 SEG

          FALTA          LITERALLY  '00000100b', /* FALTA ENERGIA */
          TADIANTA       LITERALLY  '00001000b',
          O2             LITERALLY  '00010000b', /* SENSOR DE O2 */
          TATRASA        LITERALLY  '00100000b', /*

era modo */      Taumenta   LITERALLY  '01000000b',
          Tdiminui      LITERALLY  '10000000b',

          SAIDASRW       LITERALLY  '00002H',

          OOBUZ          LITERALLY  '10000000b', /* BUZZER - ALARME */
          OORESAQ        LITERALLY  '00001000b', /* RESISTENCIA DE AQUECIMENTO */
          OOMODRN        LITERALLY  '00000100b', /* LED vd. MODO RN */
          OOMODAR        LITERALLY  '00000010b', /* LED vd. MODO AR */
          OOSIL          LITERALLY  '00000001b', /* LED vm. SILENCIA ALARME */

          QABUZ          LITERALLY  '01111111b', /* BUZZAR - ALARME */
          QARESQA        LITERALLY  '11110111b', /* RESISTENCIA DE AQUECIMENTO */
          QAMODRN        LITERALLY  '11111011b', /* LED vd. MODO RN */
          QAMODAR        LITERALLY  '11111101b', /* LED vd. MODO AR */
          QASIL          LITERALLY  '11111110b', /* LED vm. SILENCIA ALARME */

/*
          D4 = CONTROLE DO MUX A
          D5 = CONTROLE DO MUX B
          D6 = CONTROLE DO MUX C
*/

          ADLSB          LITERALLY  '00000H',
          ADMSB          LITERALLY  '00001H';

```

SEJECT


```

/*****
/*      PROCEDIMENTOS COMUNS      */
*****/

$ if ser = Offh

LESER:  PROCEDURE integer REENTRANT;

        DO WHILE ((INPUT(SERIAL$STAT) and 02H)=0); END;

        RETURN (DOUBLE(INPUT(SERIAL$DADO))-48);

END LESER;

/*****
/*      DEFEITO:  PROCEDURE REENTRANT ;
*****/

        DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
        OUTPUT(CRISTALBAL) = 080H; /*ENDERECO*/
        DO I = 0 TO 15;
            DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
            OUTPUT(CRISTALBAL+1) = DEFEITO1(I);
        END;
        DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
        OUTPUT(CRISTALBAL) = 0COH; /*ENDERECO*/
        DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
        DO I = 0 TO 15;
            DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
            OUTPUT(CRISTALBAL+1) = DEFEITO2(I);
        END;

END DEFEITO;

/*****
$ endif
*****/

6 1  CONVERTE:  PROCEDURE (k)integer REENTRANT;
7 2  DECLARE k  BYTE,
      lsb byte,
      msb byte;
8 2      k = k AND 07H;
9 2      IMSAIDASRW = (IMSAIDASRW and 10001111b) or (k*10000b);
10 2     OUTPUT(SAIDASRW) = IMSAIDASRW; /*escolhe canal*/
11 2     if k = 0 then k = 20; else k = 10; /* da mais tempo `a balanca */
14 3     DO WHILE TEMPO7 < k; TEMPO7 = TEMPO7 + 1; END; /* 10 */
17 2     TEMPO7 = 0;
18 2     lsb = input(adlsb);
/*     ESPERA fim de conversao */
19 2     msb = input(admsb);
20 2     if msb > 7 then return ((double(lsb)+double(msb)*256)-2048);
22 2     else RETURN ((DOUBLE(LSB)+DOUBLE(MSB)*256)+2048); /*conversao */
23 2     END CONVERTE;
/*****

```

PL/M-86 COMPILER INCUBA
10/11/95 07:38:23 PAGE 9

```

/*****/
24 1  LINEARIZA1: PROCEDURE (V)integer REENTRANT;
25 2  DECLARE V integer;

26 2  return v/4+270*4;      /* 27,0 .C a 39,7 .C => /8 x 4 */
27 2  END LINEARIZA1;

/*****/

28 1  LINEARIZAb: PROCEDURE (V)integer REENTRANT;
29 2  DECLARE V integer;

30 2  return v/64+280;      /* 28,0 .C a 34,3 .C */
31 2  END LINEARIZAb;

/*****/

32 1  LINEARIZAa: PROCEDURE (V)integer REENTRANT;
33 2  DECLARE V integer;

34 2  return v/64+334;      /* 33,4 .C a 39,7 .C */
35 2  END LINEARIZAa;

/*****/

36 1  LINEARIZA2: PROCEDURE (V)integer REENTRANT;
37 2  DECLARE V integer;

38 2  return v/32+190;      /* 19,0 .C a 31,8 .C */

$ if 1=2
    if V < linear05C then return (150*V/linear05C)/3;
    else
    if V < linear10C then RETURN ((V-linear05C)*50)/(linear10C-linear05C)+50;
    else
    if V < linear15C then RETURN ((V-linear10C)*50)/(linear15C-linear10C)+100;
    else
    if V < linear20C then RETURN ((V-linear15C)*50)/(linear20C-linear15C)+150;
    else
    if V < linear25C then RETURN ((V-linear20C)*50)/(linear25C-linear20C)+200;
    else
    if V < linear30C then RETURN ((V-linear25C)*50)/(linear30C-linear25C)+250;
    else
    if V < linear35C then RETURN ((V-linear30C)*50)/(linear35C-linear30C)+300;
    else
    if V < linear40C then RETURN ((V-linear35C)*50)/(linear40C-linear35C)+350;
    else
    if V < linear45C then RETURN ((V-linear40C)*50)/(linear45C-linear40C)+400;
    else
    if V < (linear50C+1) then RETURN ((V-linear45C)*150)/(linear50C linear45C)+450*3)/3;
    else CALL ESTOURO;

$ endif
39 2  END LINEARIZA2;

```

```

/*****/
40 1  ALARME : PROCEDURE REENTRANT ;
41 2      IF (TEMPO1<>0) THEN RETURN; ELSE;
44 2      IMSAIDASRW = IMSAIDASRW and OABUZ ; /* and OASIL */;
45 2      OUTPUT(SAIDASRW) = IMSAIDASRW; /* LIGA BUZINA e LED */
46 2      END ALARME;
/*****/
47 1  DESALARMA: PROCEDURE REENTRANT ;
48 2      IMSAIDASRW = IMSAIDASRW OR OOBUZ;
49 2      IF CA <> 0ffh then IMSAIDASRW = IMSAIDASRW OR OOSIL;
51 2      OUTPUT(SAIDASRW) = IMSAIDASRW; /* DESLIGA BUZINA e APAGA LED */
52 2      END DESALARMA;
/*****/

53 1  LIGA: PROCEDURE REENTRANT ;
54 2      IMSAIDASRW = IMSAIDASRW AND OARESAQ;
55 2      OUTPUT(SAIDASRW) = IMSAIDASRW; /* LIGA RESITENCIA*/
56 2      END LIGA;
/*****/

57 1  DESLIGA:PROCEDURE REENTRANT ;
58 2      IMSAIDASRW = IMSAIDASRW OR OORESQA;
59 2      OUTPUT(SAIDASRW) = IMSAIDASRW; /* DESLIGA RESITENCIA*/
60 2      END DESLIGA;
/*****/

61 1  AUMENTA: PROCEDURE REENTRANT ;
62 2      TEMPO5 = TEMPO5 + 10;
63 2      if set$point>(380*4) then
64 2          do;
65 3          tempo5=tempo5-7; /* -9 e' muito lento */
66 3          ca = 38;
67 3          IMSAIDASRW = IMSAIDASRW and OASIL;
68 3          OUTPUT(SAIDASRW) = IMSAIDASRW;
69 3          end;
70 2          else if ca = 38 then ca=0;
72 2      IF TEMPO5 > 500 THEN
73 2          DO;
74 3          IMSAIDASRW = IMSAIDASRW and OABUZ;
75 3          OUTPUT(SAIDASRW) = IMSAIDASRW; /* LIGA BUZZER */
76 3          TEMPO5 = 0;
77 3          SET$POINT = SET$POINT + 4;
78 3          if modo = ar then do;
79 4              if set$point>(390*4) then
80 5                  do;
81 5                      IMSAIDASRW = IMSAIDASRW OR OOBUZ;
82 5                      OUTPUT(SAIDASRW) = IMSAIDASRW; /* DESLIGA BUZZER */
83 5                      set$point=(390*4);
84 5                      end;
85 5                  end;
86 4              end;
87 3          else do;
88 4              if set$point > (370*4) then
89 5                  do;
90 5                      IMSAIDASRW = IMSAIDASRW OR OOBUZ;
91 5                      OUTPUT(SAIDASRW) = IMSAIDASRW; /* DESLIGA BUZZER */
92 5                      set$point =(370*4);
93 5                      end;
94 4                  end;
95 3          call escltempdes(SET$POINT);
96 3          IMSAIDASRW = IMSAIDASRW OR OOBUZ;
97 3          OUTPUT(SAIDASRW) = IMSAIDASRW; /* DESLIGA BUZZER */
98 3          END;
99 2      END AUMENTA;
/*****/

```



```

100 1      DIMINUI: PROCEDURE REENTRANT;
101 2          if ((set$point<380*4)and(ca = 38)) then ca=0;
103 2          TEMPO6 = TEMPO6 + 10;
104 2      IF TEMPO6 > 500 THEN
105 2          DO;
106 3              IMSAIDASRW = IMSAIDASRW and OABUZ;
107 3              OUTPUT(SAIDASRW) = IMSAIDASRW; /* LIGA BUZZER */
108 3              TEMPO6 = 0;
109 3              SET$POINT = SET$POINT - 4;          /* 310 */
110 3              if modo = ar then do;
112 4                  if set$point < (280*4) then
113 4                      do;
114 5                          IMSAIDASRW = IMSAIDASRW OR OOBUZ;
115 5                          OUTPUT(SAIDASRW) = IMSAIDASRW; /* DESLIGA BUZZER */
116 5                          set$point=(280*4);
117 5                      end;
118 4                  end;
119 3              else do;
120 4                  if set$point < (350*4) then
121 4                      do;
122 5                          IMSAIDASRW = IMSAIDASRW OR OOBUZ;
123 5                          OUTPUT(SAIDASRW) = IMSAIDASRW; /* DESLIGA BUZZER */
124 5                          set$point =(350*4);
125 5                      end;
126 4                  end;
127 3              call esc$empdes(SET$POINT);
128 3              IMSAIDASRW = IMSAIDASRW OR OOBUZ;
129 3              OUTPUT(SAIDASRW) = IMSAIDASRW; /* DESLIGA BUZZER */
130 3          END;
131 2      END DIMINUI;

/*****/

$ if bal = Offh

132 1      ADIANTA: PROCEDURE REENTRANT;
133 2          TEMPO3 = TEMPO3 + 1;
134 2      IF TEMPO3 > 40 THEN
135 2          DO;
136 3              IMSAIDASRW = IMSAIDASRW and OABUZ;
137 3              OUTPUT(SAIDASRW) = IMSAIDASRW; /* LIGA BUZZER*/
138 3              TEMPO3 = 0;
139 3              CURSOR = CURSOR + 60;
140 3              if cursor > 22*60 then cursor = 22*60;
142 3              CALL ESC$tant(cursor/60); /* (23*60-cursor) */
143 3              IF ( POINTIN - CURSOR ) < 0 THEN
144 3                  call esc$pesant(peso(23*60 + POINTIN - CURSOR ));
145 3                  ELSE
146 3                      call esc$pesant(peso( POINTIN - CURSOR ));
147 3              IMSAIDASRW = IMSAIDASRW OR OOBUZ;
148 3              OUTPUT(SAIDASRW) = IMSAIDASRW; /* DESLIGA BUZZER */
149 2          END;
      END ADIANTA;

/*****/

```

INCUBADORA MICROPROCESSADA COM CONTROLE TÉRMICO ADAPTATIVO

PL/M-86 COMPILER INCUBA
10/11/95 07:38:23 PAGE 12

```

150 1  ATRASA: PROCEDURE REENTRANT;
151 2      TEMPO4 = TEMPO4 + 1;
152 2      IF TEMPO4 > 40 THEN
153 3          DO;
154 3              IMSAIDASRW = IMSAIDASRW and OABUZ;
155 3              OUTPUT(SAIDASRW) = IMSAIDASRW; /* LIGA BUZZER*/
156 3              TEMPO4 = 0;
157 3              CURSOR = CURSOR - 60;
158 3              if cursor < 0 then cursor = 0; /* (23*60-cursor) */
159 3              CALL ESCtant(cursor/60);
160 3              IF ( POINTIN - CURSOR ) < 0 THEN
161 3                  call escpesant(peso( 23*60 + POINTIN - CURSOR ));
162 3                  ELSE
163 3                      call escpesant(peso( POINTIN - CURSOR ));
164 3              IMSAIDASRW = IMSAIDASRW OR OOBUZ;
165 3              OUTPUT(SAIDASRW) = IMSAIDASRW; /* DESLIGA BUZZER */
166 3          END;
167 2  END ATRASA;
/*****
168 1  ESCtant: PROCEDURE (T) REENTRANT;
169 2      DECLARE D  INTEGER,
              U  INTEGER,
              t  INTEGER;

170 2          d = t/10;
171 2          u = t-d*10;
172 3          DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
173 3          OUTPUT(CRISTALBAL) = 0COH; /*ENDERECO*/
174 2          DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
175 3          DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END; /*ESCREVE DEZENA*/
176 2          OUTPUT(CRISTALBAL+1) = ASCII(D);
177 3          DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
178 2          OUTPUT(CRISTALBAL+1) = ASCII(U); /*ESCREVE UNIDADE*/
179 3          DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
180 2          OUTPUT(CRISTALBAL+1) = ASCII(U); /*ESCREVE UNIDADE*/
181 2  END ESCtant;
/*****
182 1  ESCSEG: PROCEDURE (T) REENTRANT;
183 2      DECLARE D  INTEGER,
              U  INTEGER,
              t  INTEGER;

184 2          d = t/10;
185 2          u = t-d*10;
186 3          DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
187 3          OUTPUT(CRISTALBAL) = 87H; /*ENDERECO*/
188 2          DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
189 3          DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END; /*ESCREVE DEZENA*/
190 2          OUTPUT(CRISTALBAL+1) = ASCII(D);
191 3          DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
192 2          OUTPUT(CRISTALBAL+1) = ASCII(U); /*ESCREVE UNIDADE*/
193 3          DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
194 2          OUTPUT(CRISTALBAL+1) = ASCII(U); /*ESCREVE UNIDADE*/
195 2  END ESCSEG;
/*****
196 1  ESCMIN: PROCEDURE (T) REENTRANT;
197 2      DECLARE D  INTEGER,
              U  INTEGER,
              t  INTEGER;

198 2          d = t/10;
199 2          u = t-d*10;
200 3          DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
201 3          OUTPUT(CRISTALBAL) = 84H; /*ENDERECO*/
202 2          DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
203 3          DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END; /*ESCREVE DEZENA*/
204 2          OUTPUT(CRISTALBAL+1) = ASCII(D);
205 3          DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
206 2          OUTPUT(CRISTALBAL+1) = ASCII(U); /*ESCREVE UNIDADE*/
207 3          DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
208 2          OUTPUT(CRISTALBAL+1) = ASCII(U); /*ESCREVE UNIDADE*/
209 2  END ESCMIN;
/*****
210 1  ESCHORA: PROCEDURE (T) REENTRANT;
211 2      DECLARE C integer,
              D integer,
              U integer,
              t integer;

/* disable; */
212 2          C = t/100;
213 2          D = (t-C*100)/10;
214 2          U = t-(C*100+D*10);
215 3          DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
216 3          OUTPUT(CRISTALBAL) = 80H; /*ENDERECO*/
217 2          DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
218 3          DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END; /*ESCREVE CENTENA*/
219 2          OUTPUT(CRISTALBAL+1) = ASCII(C);
220 3          DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
221 2          DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END; /*ESCREVE DEZENA*/
222 3          OUTPUT(CRISTALBAL+1) = ASCII(D);
223 2          DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
224 3          DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END; /*ESCREVE UNIDADE*/
225 2          OUTPUT(CRISTALBAL+1) = ASCII(U);
226 2          /* enable; */
227 2  END ESCHORA;

```

PL/M-86 COMPILER INCUBA
10/11/95 07:38:23 PAGE 13

```

228 1      zerado:      PROCEDURE REENTRANT;
229 2                  CALL ESCHORA(hora);
230 3                  DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
232 2                  OUTPUT(CRISTALBAL+1) = ':';
233 2                  CALL ESCMIN(min);
234 3                  DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
236 2                  OUTPUT(CRISTALBAL+1) = ':';
237 2                  CALL ESCSEG(seg);
238 3                  DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
240 2                  OUTPUT(CRISTALBAL+1) = ' ';
241 3                  DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
243 2                  OUTPUT(CRISTALBAL) = 08EH; /*ENDERECO*/
244 3                  DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
246 2                  DO I = 14 TO 15;
247 4                  DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
249 3                  OUTPUT(CRISTALBAL+1) = BALANCA(I);
250 3                  END;
251 2                  CALL ESCtant(cursor/60); /* (1380-cursor) */
252 3                  DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
254 2                  DO I = 2 TO 9;
255 4                  DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
257 3                  OUTPUT(CRISTALBAL+1) = ANTES(I);
258 3                  END;
259 2                  IF ( POINTIN - CURSOR ) < 0 THEN
260 2                      call escpesant(peso(23*60 + POINTIN - CURSOR ));
261 2                      ELSE
262 2                          call escpesant(peso( POINTIN - CURSOR ));
263 4                  DO I = 14 TO 15;
265 3                  DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
266 3                  OUTPUT(CRISTALBAL+1) = ANTES(I);
267 2                  END;
end zerado;

```

BIBLIOTECA
FACULDADE DE FILOSOFIA
CIÊNCIAS E LETRAS DE
RIBEIRÃO PRETO DA U.S.P.

PL/M-86 COMPILER INCUBA
10/11/95 07:38:23 PAGE 14

```

/*****
268 1  zere:      PROCEDURE REENTRANT;
269 2          IMSAIDASRW = IMSAIDASRW AND OABUZ;
270 2          OUTPUT(SAIDASRW) = IMSAIDASRW; /* LIGA BUZINA */

271 3          DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
273 2          OUTPUT(CRISTALBAL) = 080H; /*ENDERECO*/
274 3          DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
276 2          DO I = 0 TO 15;
277 4          DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
279 3          OUTPUT(CRISTALBAL+1) = zera1(I);
280 3          END;

281 3          DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
283 2          OUTPUT(CRISTALBAL) = 0C0H; /*ENDERECO*/
284 3          DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
286 2          DO I = 0 TO 15;
287 4          DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
289 3          OUTPUT(CRISTALBAL+1) = zera2(I);
290 3          END;
291 2          IMSAIDASRW = IMSAIDASRW OR OOBUZ;
292 2          OUTPUT(SAIDASRW) = IMSAIDASRW; /* DESLIGA BUZINA */

293 2  end zere;

/*****
294 1  ESCPESO:      PROCEDURE (T) REENTRANT;
295 2      DECLARE C  integer,
                D  integer,
                U  integer,
                F  integer,
                t  integer;

/* disable; */
296 2      IF t<0 then call zere;
298 2      ELSE DO;
299 3          if ( t > (4050 - tara) ) then
300 3              do;
301 4                  C = 11;
302 4                  D = 11;
303 4                  U = 11;
304 4                  F = 11;
305 4                  end;

306 3              else do;
307 4                  C = t/1000;
308 4                  D = (t MOD 1000)/100;
309 4                  U = (t-(C*1000+D*100))/10;
310 4                  F = t-(C*1000+D*100+U*10);
311 4                  end;

312 3          CALL ZERADO;

313 4          DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
315 3          OUTPUT(CRISTALBAL) = 8AH; /*ENDERECO*/
316 4          DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
318 3          OUTPUT(CRISTALBAL+1) = ASCII(C); /*ESCREVE CENTENA*/
319 4          DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
321 3          OUTPUT(CRISTALBAL+1) = ASCII(D); /*ESCREVE DEZENA*/
322 4          DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
324 3          OUTPUT(CRISTALBAL+1) = ASCII(U); /*ESCREVE UNIDADE*/
325 4          DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
327 3          OUTPUT(CRISTALBAL+1) = ASCII(F); /*ESCREVE DECIMOS*/

328 3          END;
329 2  enable;

330 2  END ESCPESO;

```

PL/M-86 COMPILER INCUBA
10/11/95 07:38:23 PAGE 15

```

/*****
331 1  ESCPESant:  PROCEDURE (T) REENTRANT;
332 2      DECLARE M      integer,
              C      integer,
              D      integer,
              U      integer,
              t      integer;

              /* disable; */
333 2      if((M < 0) or (M > 4095)) then M = 0;
335 2          M = t/1000;
336 2          C = (t MOD 1000)/100;
337 2          D = (t-(M*1000+C*100))/10;
338 2          U = t-(M*1000+C*100+D*10);
339 3      DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
341 2      OUTPUT(CRISTALBAL) = 0CAH; /*ENDERECO*/
342 3      DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
344 2      OUTPUT(CRISTALBAL+1) = ASCII(M); /*ESCREVE MILHAR*/
345 3      DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
347 2      OUTPUT(CRISTALBAL+1) = ASCII(C); /*ESCREVE CENTENA*/

348 3      DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
350 2      OUTPUT(CRISTALBAL+1) = ASCII(D); /*ESCREVE DEZENA*/
351 3      DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
353 2      OUTPUT(CRISTALBAL+1) = ASCII(U); /*ESCREVE UNIDADE*/
354 2  /* enable; */
      END ESCPESant;

/*****

$ endif

/*****

355 1  ESCTEMPRN:  PROCEDURE (T) REENTRANT;
356 2      DECLARE D integer,
              U integer,
              C integer,
              t integer;

357 2      if (t < (300*4)) then
358 2          do;
359 3              d = 10;
360 3              u = 10;
361 3              c = 10;
362 3              end;
363 2      else
364 3          do;
365 3              if ( t > (526*4-2)) then
366 4                  do;
367 4                      d = 11;
368 4                      u = 11;
369 4                      c = 11;
370 4                      end;
371 4                      else do;
372 4                          t = t/4;
373 4                          d = t/100;
374 4                          u = (t-d*100)/10;
375 4                          c = t-(d*100+u*10);
376 3                      end;

377 3      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
379 2      OUTPUT(CRISTALRW) = 83H; /*ENDERECO*/
380 3      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
382 2      OUTPUT(CRISTALRW+1) = ASCII(D); /*ESCREVE DEZENA*/
383 3      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
385 2      OUTPUT(CRISTALRW+1) = ASCII(U); /*ESCREVE UNIDADE*/
386 3      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
388 2      OUTPUT(CRISTALRW+1) = ','; /*ESCREVE VIRGULA*/
389 3      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
391 2      OUTPUT(CRISTALRW+1) = ASCII(C); /*ESCREVE DECIMOS*/

392 2  END ESCTEMPRN;

/*****

```

PL/M-86 COMPILER INCUBA
10/11/95 07:38:23 PAGE 16

```

393 1  ESCTEMPDES: PROCEDURE (T) REENTRANT;
394 2      DECLARE D integer,
          U integer,
          C integer,
          t integer;

395 2      if (t < (270*4+2)) then
396 2          do;
397 3              d = 10;
398 3              u = 10;
399 3              c = 10;
400 3              end;
401 2      else
          do;
402 3          if ( t > (398*4-2)) then
403 3              do;
404 4                  d = 11;
405 4                  u = 11;
406 4                  c = 11;
407 4                  end;
408 3              else do;
409 4                  t = t/4;
410 4                  d = t/100;
411 4                  u = (t-d*100)/10;
412 4                  c = t-(d*100+u*10);
413 4                  end;
414 3              end;

415 3      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
417 2      OUTPUT(CRISTALRW) = OCBH; /*ENDereco*/
418 3      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
420 2      OUTPUT(CRISTALRW+1) = ASCII(D); /*ESCREVE DEZENA*/
421 3      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
423 2      OUTPUT(CRISTALRW+1) = ASCII(U); /*ESCREVE UNIDADE*/
424 3      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
426 2      OUTPUT(CRISTALRW+1) = ','; /*ESCREVE VIRGULA*/
427 3      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
429 2      OUTPUT(CRISTALRW+1) = ASCII(C); /*ESCREVE DECIMOS*/

430 2  END ESCTEMPDES;

/*****/

```

PL/M-86 COMPILER INCUBA
10/11/95 07:38:23 PAGE 17

```

431 1      ESCTEMPAR: PROCEDURE (T) REENTRANT;
432 2          DECLARE D integer,
                U integer,
                C integer,
                t integer;
433 2          if (t < (270*4+2)) then
434 2              do;
435 3                  d = 10;
436 3                  u = 10;
437 3                  c = 10;
438 3                  end;
439 2          else
440 3              do;
441 3                  if ( t > (526*4-2)) then
442 4                      do;
443 4                          d = 11;
444 4                          u = 11;
445 4                          c = 11;
446 3                      end;
447 4                      else do;
448 4                          t = t/4;
449 4                          d = t/100;
450 4                          u = (t-d*100)/10;
451 4                          c = t-(d*100+u*10);
452 3                      end;
453 3              end;
454 3          DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
455 2          OUTPUT(CRISTALRW) = 08BH; /*ENDereco*/
456 3          DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
457 2          OUTPUT(CRISTALRW+1) = ASCII(D); /*ESCREVE DEZENA*/
458 3          DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
459 2          OUTPUT(CRISTALRW+1) = ASCII(U); /*ESCREVE UNIDADE*/
460 3          DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
461 2          OUTPUT(CRISTALRW+1) = ','; /*ESCREVE VIRGULA*/
462 3          DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
463 2          OUTPUT(CRISTALRW+1) = ASCII(C); /*ESCREVE DECIMOS*/
464 3          DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
465 2          OUTPUT(CRISTALRW+1) = ASCII(C); /*ESCREVE DECIMOS*/
466 3          DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
467 2          OUTPUT(CRISTALRW+1) = ASCII(C); /*ESCREVE DECIMOS*/
468 2      END ESCTEMPAR;
      /*****/

```


PL/M-86 COMPILER INCUBA
10/11/95 07:38:23 PAGE 18

```

469 1 CALIBRA: PROCEDURE REENTRANT;
470 2 DECLARE A BYTE,
          C integer,
          D integer,
          U integer,
          F integer,
          t integer;

471 2 disable;
          /* LIGA TODOS OS LEDS */
472 2 IMSAIDASRW = OARESAQ and OAMODRN and OAMODAR and OASIL;
473 2 OUTPUT(SAIDASRW) = IMSAIDASRW;

474 3 DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
476 2 OUTPUT(CRISTALRW) = 080H; /*ENDERECO*/
477 3 DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
479 2 DO I = 0 TO 15;
480 4 DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
482 3 OUTPUT(CRISTALRW+1) = CALIBRA1(I);
483 3 END;
484 3 DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
486 2 OUTPUT(CRISTALRW) = 0C0H; /*ENDERECO*/
487 3 DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
489 2 DO I = 0 TO 11;
490 4 DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
492 3 OUTPUT(CRISTALRW+1) = CALIBRA2(I);
493 3 END;

494 2 DO WHILE ((INPUT(ENTRADAS) and (Taumenta or Tdiminui)) <> 0);
495 3 t = CONVERTE(3);
496 3 do i = 0 to 1000;
497 4 t = (((CONVERTE(3)+t)/2+t)/2+t)/2+t)/2;
498 4 end;
499 3 C = t/1000;
500 3 D = (t MOD 1000)/100;
501 3 U = (t-(C*1000+D*100))/10;
502 3 F = t-(C*1000+D*100+U*10);

503 4 DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
505 3 OUTPUT(CRISTALRW) = 0CCH; /*ENDERECO*/
506 4 DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
508 3 OUTPUT(CRISTALRW+1) = ASCII(C); /*ESCREVE CENTENA*/
509 4 DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
511 3 OUTPUT(CRISTALRW+1) = ASCII(D); /*ESCREVE DEZENA*/
512 4 DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
514 3 OUTPUT(CRISTALRW+1) = ASCII(U); /*ESCREVE UNIDADE*/
515 4 DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
517 3 OUTPUT(CRISTALRW+1) = ASCII(F); /*ESCREVE DECIMOS*/
518 3 END;
519 2 disable;
520 2 do while 2 > 1;
521 3 IMSAIDASRW = IMSAIDASRW and ( OABUZ and OASIL );
522 3 OUTPUT(SAIDASRW) = IMSAIDASRW; /* LIGA BUZINA e LED */
523 4 DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
525 3 OUTPUT(CRISTALRW) = 080H; /*ENDERECO*/
526 4 DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
528 3 DO I = 0 TO 15;
529 5 DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
531 4 OUTPUT(CRISTALRW+1) = CALIBRA3(I);
532 4 END;
533 4 DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
535 3 OUTPUT(CRISTALRW) = 0C0H; /*ENDERECO*/
536 4 DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
538 3 DO I = 0 TO 15;
539 5 DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
541 4 OUTPUT(CRISTALRW+1) = CALIBRA4(I);
542 4 END;
543 3 end;
544 2 HALT;
545 2 END CALIBRA;

```

PL /M-86 COMPILER INCUBA
10/11/95 07:38:23 PAGE 19

\$ if so2 = Offh

```

/*****/
SENSO2:  PROCEDURE REENTRANT;
DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
OUTPUT(CRISTALRW) = 0COH; /*ENDERECO*/
DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
DO I = 0 TO 15;
DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
OUTPUT(CRISTALRW+1) = AVISO02(I);
END;
END SENSO2;

```

```

/*****/
$ endif

```

```

546 1  /*****/
547 3  CIRCULA:  PROCEDURE REENTRANT;
549 2  DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
550 3  OUTPUT(CRISTALRW) = 0COH; /*ENDERECO*/
552 2  DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
553 4  DO I = 0 TO 15;
555 3  DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
556 3  OUTPUT(CRISTALRW+1) = CULA(I);
557 2  END;
END CIRCULA;

```

```

/*****/
/*      ROTINAS DE TRATAMENTO DE INTERRUPCAO      */
/*****/

```

\$ if ser = Offh

NMI: PROCEDURE INTERRUPT 2 REENTRANT;

```

IF (INPUT(SERIAL$STAT) and 0011010b) <> 02H THEN /* EXISTE ERRO ? */
DO;
call estouro;
OUTPUT (SERIAL$COMANDO) = 00010110b;
RETURN;
END;

```

```

SERIAL = INPUT(SERIAL$DADO);
IF SERIAL = '.' THEN DO; POINT = 10; RETURN; END;

```

IF POINT < 10 THEN RETURN;

```

DO WHILE (INPUT(CRISTALBAL) OR 7FH)=OFFH;END;
OUTPUT(CRISTALBAL) = 080H + POINT;
DO WHILE (INPUT(CRISTALBAL) OR 7FH)=OFFH;END;
OUTPUT(CRISTALbal+1) = SERIAL;

```

POINT = POINT +1;

IF POINT < 14 THEN RETURN;

```

POINT = 0;
RETURN;

```

END NMI;

PL/M-86 COMPILER INCUBA
10/11/95 07:38:23 PAGE 20

```

$ endif

/*****
558 1 NMI: PROCEDURE INTERRUPT 2 REENTRANT;
/*****
/*FUNCAO: ALARMAR EM CASO DE QUEDA DA TENSÃO DA REDE */
/*****

559 2 enable;
560 2 ca = 0FEh;
561 2 IMSAIDASRW = IMSAIDASRW AND OABUZ;
562 2 OUTPUT(SAIDASRW) = OABUZ; /* LIGA BUZINA */

563 3 DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
565 2 OUTPUT(CRISTALRW) = 080H; /*ENDERECO*/

566 2 DO I1 = 0 TO 15;
567 4 DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
569 3 OUTPUT(CRISTALRW+1) = ENERGIA(I1);
570 3 END;

571 3 do while ((INPUT(ENTRADAS) and Tsilencia) <> 0);end;
573 2 OUTPUT(SAIDASRW) = OFFH; /* DESLIGA BUZINA */

/* silencia alarme quando tecla de silencia for apertada */

574 2 enable;
575 2 HALT;
576 2 CALL DESALARMA;
577 2 enable;
578 2 RETURN;

579 2 END NMI;
/*****
580 1 ESTOURO: PROCEDURE INTERRUPT 4 REENTRANT;
/*****
/*FUNCAO: ALARMAR EM CASO DE DIVISAO POR ZERO OU ESTOURO*/
/*****

581 2 IMSAIDASRW = IMSAIDASRW AND OABUZ;
582 2 OUTPUT(SAIDASRW) = IMSAIDASRW; /* LIGA BUZINA */
583 3 DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
585 2 OUTPUT(CRISTALRW) = 0COH; /*ENDERECO*/
586 3 DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
588 2 DO I2 = 0 TO 15;
589 4 DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
591 3 OUTPUT(CRISTALRW+1) = CALCULO(I2);
592 3 END;
593 2 CALL DESLIGA;
/* HALT; */
594 2 END ESTOURO;
/*****

595 1 INTERRUPT: PROCEDURE INTERRUPT 80 reentrant;

```

PL/M-86 COMPILER INCUBA
10/11/95 07:38:23 PAGE 21

```

/*****
/*  FUNCAO:  CONTROLAR TEMPERATURA E MONITORAR          */
/*  FUNCIONAMENTO DA INCUBADORA                        */
*****/

/*****/

596  2  disable;
597  3  if ca = 0FEh then do; ca = 0; enable; RETURN;end;

$ if bal = Offh

603  2  IMSAIDASRW = (MSAIDASRW and 10001111b);
604  2  OUTPUT(SAIDASRW) = IMSAIDASRW; /*escolhe o canal 0 */

                                           /* variacao */
                                           /* 16 4 64 */
605  2  RELOGIO:                                           /*128 32 0 */
DO;

$ if ser =0

/* media = 2*media; */
/* media=(((((CONVERTE(0)*2-tara)+media)/4+media)/4)+media)/32)*8; */
/* media=(((((CONVERTE(0)-tara)+media)/4+media)/4)+media)/4; */
/* media=(((((CONVERTE(0)-tara)+media)/4+media)/4)+media)/8)*2; */
/* media=(((((CONVERTE(0)-tara)+media)/3+media)/3); */
/* media=((CONVERTE(0)-tara)+media)/2; 4g */
606  3  atual = converte(0) - tara;
607  3  IF (MEDIA-100<atual) and (atual<MEDIA+100) THEN media=(atual+7*media)/8;
                                           /*VAR 2g*/

/* media = CONVERTE(0); VAR 10g */

$ endif

$ if ser = 0fh

media = 0;
MARCA2:

IF ((INPUT(SERIAL$STAT) and 02H)=0) THEN GOTO AINDANAO;
IF (INPUT(SERIAL$DADO) <> 02H) THEN GOTO AINDANAO;

DO WHILE ((INPUT(SERIAL$STAT) and 02H)=0); END;
/* IF (INPUT(SERIAL$DADO) <> 36H) THEN GOTO AINDANAO; */
nada = INPUT(SERIAL$DADO);

DO WHILE ((INPUT(SERIAL$STAT) and 02H)=0); END;
/* IF (INPUT(SERIAL$DADO) <> 2EH) THEN GOTO AINDANAO; */
nada = INPUT(SERIAL$DADO);

$ if 100 = 200

media = media + leser * 1000;
media = media + leser * 100;
media = media + leser * 10;
media = media + leser;

```


PL/M-86 COMPILER INCUBA
10/11/95 07:38:23 PAGE 22

```

DO WHILE ((INPUT(SERIAL$STAT) and 02H)=0); END;
nada = INPUT(SERIAL$DADO);

DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
OUTPUT(CRISTALBAL) = 89H; /*ENDERECO*/
DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
OUTPUT(CRISTALBAL+1) = NADA; /*ESCREVE*/

$ endif

DO WHILE ((INPUT(SERIAL$STAT) and 02H)=0); END;
nada = INPUT(SERIAL$DADO);

DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
OUTPUT(CRISTALBAL) = 8AH; /*ENDERECO*/
DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
OUTPUT(CRISTALBAL+1) = NADA; /*ESCREVE*/

DO WHILE ((INPUT(SERIAL$STAT) and 02H)=0); END;
nada = INPUT(SERIAL$DADO);

DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
OUTPUT(CRISTALBAL) = 8BH; /*ENDERECO*/
DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
OUTPUT(CRISTALBAL+1) = NADA; /*ESCREVE*/

DO WHILE ((INPUT(SERIAL$STAT) and 02H)=0); END;
nada = INPUT(SERIAL$DADO);

DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
OUTPUT(CRISTALBAL) = 8CH; /*ENDERECO*/
DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
OUTPUT(CRISTALBAL+1) = NADA; /*ESCREVE*/

DO WHILE ((INPUT(SERIAL$STAT) and 02H)=0); END;
nada = INPUT(SERIAL$DADO);

DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
OUTPUT(CRISTALBAL) = 8DH; /*ENDERECO*/
DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
OUTPUT(CRISTALBAL+1) = NADA; /*ESCREVE*/

$ if 100 = 200

DO WHILE ((INPUT(SERIAL$STAT) and 02H)=0); END;
nada = INPUT(SERIAL$DADO);

DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
OUTPUT(CRISTALBAL) = 8EH; /*ENDERECO*/
DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
OUTPUT(CRISTALBAL+1) = NADA; /*ESCREVE*/

DO WHILE ((INPUT(SERIAL$STAT) and 02H)=0); END;
nada = INPUT(SERIAL$DADO);

DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
OUTPUT(CRISTALBAL) = 8FH; /*ENDERECO*/
DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
OUTPUT(CRISTALBAL+1) = NADA; /*ESCREVE*/

$ endif

AINDANAO:

$ endif

```

PL/M-86 COMPILER INCUBA
10/11/95 07:38:23 PAGE 23

```

609 3      ciclo = ciclo+1;
610 3      if ciclo > 119 then
611 3          do;
612 4          ciclo = 0;
        /* CALL ESCSEG(seg); */
613 4          seg = seg+1;

        $ if ser = 0
614 4          IF (atual<MEDIA-100) or (atual>MEDIA+100) THEN media = atual;
        $ endif
616 4          CALL ESCPESO((media/2)*2);

617 4      end;
618 3          if seg > 58 then
619 3              do;
620 4              seg = -1;
        /*          CALL ESCMIN(min); */
621 4              min = min + 1;
                /* ATUALIZA BUFFER DE PESO DO RN */
622 4              POINTIN = POINTIN + 1;
623 4              IF POINTIN > 1380 THEN POINTIN = 0;
625 4              peso(POINTIN) = media;
626 4              IF ( POINTIN - CURSOR ) < 0 THEN
627 4                  call escpesant(peso( 23*60 + POINTIN - CURSOR ));
628 4                  ELSE
                        call escpesant(peso(POINTIN-CURSOR));

629 4          if min > 59 then
630 4              do;
631 5              min = 0;
        /*          CALL ESCHORA(hora); */
632 5              hora = hora + 1;

633 5          if hora > 999 then hora = 0;
635 5          end;
636 4      end;
637 3      END RELOGIO;

        $ endif

        $ if cir = Offh

        /* ( Verifica se ar circula ) */

638 2          IF CONVERTE(7) < 800 THEN          /* 800 */
639 2              do;
640 3              ca = 9;
641 3              CALL CIRCULA;
642 3              end;
643 2          else if ((ca = 9) and ( CONVERTE(7) > 400)) then ca = 0;

        $ endif
        /* cir */

        $ if teste = 0h

```

```

/* FILTROS: */
/* DO; */
/* temp$ar$1$filt = temp$ar$1(7); */
/*
((TEMP$AR$1(0)+TEMP$AR$1(2))/2+(TEMP$AR$1(1)+TEMP$AR$1(3))/2+(TEMP$AR$1(4)+TEMP$AR$1(6))/2+(TEMP
$AR$1(5
-)+TEMP$AR$1(7))/2)/4; */
/* temp$ar$2$filt = temp$ar$2(7); */
/*
((TEMP$AR$2(0)+TEMP$AR$2(2))/2+(TEMP$AR$2(1)+TEMP$AR$2(3))/2+(TEMP$AR$2(4)+TEMP$AR$2(6))/2+(TEMP
$AR$2(5
-)+TEMP$AR$2(7))/2)/4; */

/* temp$rn$1$filt = temp$rn$1(7); */
/*
((TEMP$RN$1(0)+TEMP$RN$1(2))/2+(TEMP$RN$1(1)+TEMP$RN$1(3))/2+(TEMP$RN$1(4)+TEMP$RN$1(6))/2+(TEMP
$RN$1(5
-)+TEMP$RN$1(7))/2)/4; */
/* temp$rn$2$filt = temp$rn$2(7); */
/*
((TEMP$RN$2(0)+TEMP$RN$2(2))/2+(TEMP$RN$2(1)+TEMP$RN$2(3))/2+(TEMP$RN$2(4)+TEMP$RN$2(6))/2+(TEMP
$RN$2(5
-)+TEMP$RN$2(7))/2)/4; */
/* END; */

$ endif
/* endif amostra*/

$ if ekg = Offh
/* ATUALIZA BUFFER DO ECG. */
IF (converte(1)-ecg > 30) then
do;
/* IMSAIDASRW = IMSAIDASRW AND 7EH; */
/* OUTPUT(SAIDASRW) = IMSAIDASRW; LIGA LED e BUZINA */

/* do Iv = 0 to 100; end; */
/* call desalarma; */
end;
ecg = converte(1);

$ endif

```

\$ if 1=2

```
MEDIARN:
/*****
/* CALCULA E INDICA A MEDIA DAS ULTIMAS AMOSTRAS DE */
/* TEMPERATURA DO RN, ALARMANDO NO CASO DE */
/* DISCREPANCIA MAIOR DO QUE 1.C ENTRE OS SENSORES */
*****/
DO;
IF IABS(TEMP$RN$1$FILT - TEMP$RN$2$FILT) > 200 THEN
DO;
CA = 1;
IF IABS(TEMP$RN$1(7) - TEMP$RN$1(5)) > 200 THEN
DO;
/* ALARMA E ELIMINA SENSOR 1 */
DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
OUTPUT(CRISTALRW) = 0COH; /*ENDereco*/
DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
DO I4 = 0 TO 15;
DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
OUTPUT(CRISTALRW+1) = RSENSOR1(I4);
END;
TEMP$RN$REAL = TEMP$RN$2$FILT;
END;

else IF IABS(TEMP$RN$2(7) - TEMP$RN$2(5)) > 200 THEN
DO;
/* ALARMA E ELIMINA SENSOR 2 */
DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
OUTPUT(CRISTALRW) = 0COH; /*ENDereco*/
DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
DO I4 = 0 TO 15;
DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
OUTPUT(CRISTALRW+1) = RSENSOR2(I4);
END;
TEMP$RN$REAL = TEMP$RN$1$FILT;
END;
END;
ELSE do;
if ca = 1 then CA = 0;
TEMP$RN$REAL = (TEMP$RN$1$FILT+TEMP$RN$2$FILT)/2;
end;
END MEDIARN;

$ endif
```


PL/M-86 COMPILER INCUBA
10/11/95 07:38:23 PAGE 26

```

647 2    TEMP$RNSREAL = lineariza1(CONVERTE(3));

$ if 1=2

MEDIAAR:
/*****
/* CALCULA E INDICA A MEDIA DAS ULTIMAS AMOSTRAS DE */
/* TEMPERATURA DO AR,          ALARMANDO NO CASO DE */
/* DISCREPANCIA MAIOR DO QUE 10.C ENTRE OS SENSORES */
*****/

DO;
IF IABS(TEMP$AR$1$FILT - TEMP$AR$2$FILT) > 100*4 THEN
DO;
  CA = 2;
  IF IABS(TEMP$AR$1(7) - TEMP$AR$1(5)) > 100*4 THEN
  DO;
    /* ALARMA E ELIMINA SENSOR 1 */
    DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
    OUTPUT(CRISTALRW) = 0COH; /*ENDERECO*/
    DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
    DO I4 = 0 TO 15;
      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
      OUTPUT(CRISTALRW+1) = ARSENSOR1(I4);
    END;
  /* TEMP$AR$REAL = TEMP$AR$2$FILT; */
  END;

  else IF IABS(TEMP$AR$2(7) - TEMP$AR$2(5)) > 100*4 THEN
  DO;
    /* ALARMA E ELIMINA SENSOR 2 */
    DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
    OUTPUT(CRISTALRW) = 0COH; /*ENDERECO*/
    DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
    DO I4 = 0 TO 15;
      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
      OUTPUT(CRISTALRW+1) = ARSENSOR2(I4);
    END;
  /* TEMP$AR$REAL = TEMP$AR$1$FILT; */
  END;
END;
ELSE do;
  if ca = 2 then CA = 0;
  TEMP$AR$REAL = (TEMP$AR$1$FILT+TEMP$AR$2$FILT)/2;
end;

```

PL/M-86 COMPILER INCUBA
10/11/95 07:38:23 PAGE 27

```

END MEDIAAR;

$ endif

648 2    temp$ar$real = (linearizal(CONVERTE(2))+linearizal(CONVERTE(1)))/2;
649 2    LEESTADORN:      /* MONITORA ESTADO DO Recem Nascido */
        DO;
        /* verifica se o sensor do rn esta ativo */
650 3    if senrn = ativo then
651 3    do; /* desvio se sensor do rn fora */
652 4        faixar = 25*4; /* Faixa de desvio para alarmar ar quente ou frio = 2,5.C. */
        /* verifica se temperatura do rn 0,5 .C menor que o set point */
653 4        IF (TEMP$RN$REAL < (rn$desej - 5*4))
            THEN
654 4            DO;
655 5                CALL LIGA;
656 5                CA = 5;
657 6                DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
659 5                OUTPUT(CRISTALRW) = 0COH; /*ENDERECO*/
660 6                DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
662 5                DO I5 = 0 TO 10;
663 7                    DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
665 6                    OUTPUT(CRISTALRW+1) = HIPO(I5);
666 6                    END;
667 5                CALL esctempdes(rn$desej);
668 6                DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
670 5                OUTPUT(CRISTALRW+1) = 0;
671 5                RETURN;
672 5            END;
673 4            ELSE if ((CA = 5) and (TEMP$RN$REAL > (rn$desej - 4*4))) then ca = 0;

        /* verifica se temperatura do rn 0,5 .C maior que o set point */
675 4        IF (TEMP$RN$REAL > (rn$desej + 5*4)) THEN
676 4            DO;
677 5                CALL DESLIGA;
678 5                CA = 6;
679 6                DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
681 5                OUTPUT(CRISTALRW) = 0COH; /*ENDERECO*/
682 6                DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
684 5                DO I5 = 0 TO 10;
685 7                    DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
687 6                    OUTPUT(CRISTALRW+1) = HIPER(I5);
688 6                    END;
689 5                CALL esctempdes(rn$desej);
690 6                DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
692 5                OUTPUT(CRISTALRW+1) = 0;
693 5                RETURN;
694 5            END;
695 4            ELSE if ((ca = 6) and (TEMP$RN$REAL < (rn$desej + 4*4))) then ca = 0;

697 4        end;

        /* desvio se sensor do rn fora */
698 3    else faixar = 10*4; /*Faixa de desvio para alarmar ar quente ou frio = 1,0C.*/

699 3    END LEESTADORN;

```

PL/M-86 COMPILER INCUBA
10/11/95 07:38:23 PAGE 28

```

700 2  LEESTADOAR:          /* MONITORA ESTADO DA INCUBADORA */
      DO;
      /* verifica se temperatura do ar menor que o set point - faixa */
701 3  IF (TEMP$AR$REAL < (ar$desej - faixar))
      THEN
702 3      DO;
703 4          CALL LIGA;
704 4          CA = 3;
705 5          DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
707 4          OUTPUT(CRISTALRW) = 0COH; /*ENDereco*/
708 5          DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
710 4              DO I5 = 0 TO 10;
711 6                  DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
713 5                  OUTPUT(CRISTALRW+1) = FRIO(I5);
714 5              END;
715 4          CALL esctempdes(ar$desej);
716 5          DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
718 4          OUTPUT(CRISTALRW+1) = 0;
      /* if ((modo = rn) and (tempol=0))then ca = 13; */
      /* else if ca=13 then ca = 0; */
719 4          RETURN;
720 4      END;
721 3      ELSE if ((CA = 3)and(TEMP$AR$REAL >(ar$desej - faixar+4))) then ca=0;

/* verifica se temperatura do ar maior que o set point +faixa */
723 3  IF (TEMP$AR$REAL > (ar$desej + faixar))
      THEN
724 3      DO;
725 4          CALL DESLIGA;
726 4          CA = 4;
727 5          DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
729 4          OUTPUT(CRISTALRW) = 0COH; /*ENDereco*/
730 5          DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
732 4              DO I5 = 0 TO 10;
733 6                  DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
735 5                  OUTPUT(CRISTALRW+1) = QUENTE(I5);
736 5              END;
737 4          CALL esctempdes(ar$desej);
738 5          DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
740 4          OUTPUT(CRISTALRW+1) = 0;
      /* if ((modo = rn) and (tempol=0))then ca = 14; */
      /* else if ca=14 then ca = 0; */
741 4          RETURN;
742 4      END;
743 3      ELSE if ((CA = 4)and(TEMP$AR$REAL <(ar$desej + faixar-4))) then ca=0;

745 3  END LEESTADOAR;

```

PL/M-86 COMPILER INCUBA
10/11/95 07:38:23 PAGE 29

```

/******
/*      CALCULA O ESTADO DA RESISTENCIA
/******
746  2  CONTROLA:

do;

$ if ligdes = Offh

if (temp$real > (set$point-2)) then call desliga;
else call liga;

$ endif

747  4      if temp$real < (270*4+1) then do; call liga; return; end;

$ if prop = Offh
  IF NL = 0 THEN erro = (SET$POINT - temp$real+12);
$ endif

$ if amb = Offh
  IF NL = 0 THEN erro = (SET$POINT-temp$real)*(SET$POINT-temp$amb)/2;
$ endif

$ if int = Offh
  IF NL = 0 THEN erro = (SET$POINT-temp$real+erro/2);
$ endif

/* 110 v */

/******
/* controle | Temp$des | Temp$amb | Temp$ar$real | Temp pes | Temp inc.merc*/
/* lig.desl. | 32,0 .C | 24,0 .C | 32,05/,15 .C | .C | .C */
/* prop+12 | 31,0 .C | 25,2+-0,3 | 31,0/,2 .C | .C | .C */
/* int-dif/32 | 32,0 .C | 26,5+-0,5 | 31,9/,2 .C | 31,4 .C | .C */
/* int-dif/32 | 35,0 .C | 26,5+-0,5 | 34,8/,1 .C | 34,1 .C | .C */
/* int-dif/32 | 38,0 .C | 26,5+-0,5 | 37,8/,1 .C | 36,6 .C | .C */
/* int-dif/32 | 38,0 .C | 31,5+-0,5 | 37,9/,1 .C | 36,7 .C | .C */
/* int-dif/32 | 38,0 .C | 38,5+-0,5 | 38,0/,1 .C | , .C | .C */
/* int-dif/16 | 31,0 .C | 22,5+ 0,3 | 30,8/,9 .C | 29,6/7.C | 30,1/1 .C */
/* int-dif/16 | 38,0 .C | 23,2/,7 | 37,8/,9 .C | 35,9/36,0 | 37,2/2 .C */
/* int-dif/4 | 31,0 .C | 24,0- 0,3 | 30,8/31,0 .C | 30,1/2.C | 30,3/1 .C */
/* prop+12 | , .C | , .C | , .C | , .C | , .C */
/* prop+12 | 38,0 .C | 24,9 .C | 37,85.C | 37,9 .C | 37,8 .C */
/******
/* controle | Temp$des | Temp$amb | Temp$ar$real | FILRO AR | Temp inc.merc*/
/* i/16-dif/16 | 36,0 .C | 24, | 35,4/,5 .C | S | 34,4/,6.C */
/* i/16-dif/16 | 37,0 .C | 23,0 | 36,3 .C | S | 34,6 .C */
/* i/16-dif/16 | 37,0 .C | 23,0 | 35,1 .C | N | 34,3 .C */
/* ERR/2-dif/4 | 36,0 .C | 24, | 35,2/,3 .C | N | 35,4/,6.C */
/******

```


PL/M-86 COMPILER INCUBA
10/11/95 07:38:23 PAGE 30

```

$ if difere = Offh

752 3      IF NL = 0 THEN erro = SET$POINT-temp$real+erro; /*2,4,8,16 */

754 3      dif = (linearizal(CONVERTE(2))-linearizal(CONVERTE(1)))/2; /*1,2,16;*/
/* obs: dif e' negativo quando ambiente + frio: sensor 2 e' mais externo */
755 3      if dif > 0 then dif = 0;

757 4      IF ERRO > ( 80 + dif ) THEN do; erro=80+dif; call liga; RETURN;end;
763 4      IF ERRO < dif THEN do;erro=dif;call desliga; RETURN;end;
769 3      NL = NL + 1;
770 3      IF NL = 80 THEN NL = 0;

772 3      DO CASE (ERRO-dif); /* temp real = */
773 4      CALL DESLIGA; /* SETPOINT + 10 */
774 4      IF NL < 2 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT + 9 */
777 4      IF NL < 3 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT + 8 */
780 4      IF NL < 4 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT + 7 */
783 4      IF NL < 5 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT + 6 */
786 4      IF NL < 6 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT + 5 */
789 4      IF NL < 7 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT + 4 */
792 4      IF NL < 8 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT + 3 */
795 4      IF NL < 9 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT + 2 */
798 4      IF NL <10 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT + 1 */
801 4      IF NL <11 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT */
804 4      IF NL <12 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 1 */
807 4      IF NL <13 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 2 */
810 4      IF NL <14 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 3 */
813 4      IF NL <15 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 4 */
816 4      IF NL <16 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 5 */
819 4      IF NL <17 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 6 */
822 4      IF NL <18 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 7 */
825 4      IF NL <19 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 8 */
828 4      IF NL <20 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 9 */
831 4      IF NL <21 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT */
834 4      IF NL <22 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 1 */
837 4      IF NL <23 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 2 */
840 4      IF NL <24 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 3 */
843 4      IF NL <25 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 4 */
846 4      IF NL <26 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 5 */
849 4      IF NL <27 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 6 */
852 4      IF NL <28 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 7 */
855 4      IF NL <29 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 8 */
858 4      IF NL <30 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 9 */
861 4      IF NL <31 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT */
864 4      IF NL <32 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 1 */

```

PL/M-86 COMPILER INCUBA
10/11/95 07:38:23 PAGE 31

```

867 4      IF NL <33 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 2 */
870 4      IF NL <34 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 3 */
873 4      IF NL <35 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 4 */
876 4      IF NL <36 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 5 */
879 4      IF NL <37 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 6 */
882 4      IF NL <38 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 7 */
885 4      IF NL <39 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 8 */
888 4      IF NL <40 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 9 */
891 4      IF NL <41 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 10 */
894 4      IF NL <42 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 11 */
897 4      IF NL <43 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 12 */
900 4      IF NL <44 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 13 */
903 4      IF NL <45 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 14 */
906 4      IF NL <46 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 15 */
909 4      IF NL <47 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 16 */
912 4      IF NL <48 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 17 */
915 4      IF NL <49 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 18 */
918 4      IF NL <50 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 19 */
921 4      IF NL <51 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 20 */
924 4      IF NL <52 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 21 */
927 4      IF NL <53 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 22 */
930 4      IF NL <54 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 23 */
933 4      IF NL <55 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 24 */
936 4      IF NL <56 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 25 */
939 4      IF NL <57 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 26 */
942 4      IF NL <58 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 27 */
945 4      IF NL <59 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 28 */
948 4      IF NL <60 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 29 */
951 4      IF NL <61 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 30 */
954 4      IF NL <62 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 31 */
957 4      IF NL <63 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 32 */
960 4      IF NL <64 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 33 */
963 4      IF NL <65 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 34 */
966 4      IF NL <66 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 35 */
969 4      IF NL <67 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 36 */
972 4      IF NL <68 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 37 */
975 4      IF NL <69 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 38 */
978 4      IF NL <70 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 39 */
981 4      IF NL <71 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 40 */
984 4      IF NL <72 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 41 */
987 4      IF NL <73 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 42 */
990 4      IF NL <74 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 43 */
993 4      IF NL <75 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 44 */
996 4      IF NL <76 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 45 */
999 4      IF NL <77 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 46 */
1002 4     IF NL <78 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 47 */
1005 4     IF NL <79 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 48 */
1008 4     IF NL <80 THEN CALL LIGA; ELSE CALL DESLIGA; /* SETPOINT - 49 */
1011 4     CALL LIGA;

1012 4     end;

$ endif
/* lig.desl. */

1013 3     RETURN;

1014 3     END CONTROLA;
1015 2     enable;
/* HALT; */

$ endif
/* teste */

1016 2     END INTERRUPTAO;

```

PL/M-86 COMPILER INCUBA
10/11/95 07:38:23 PAGE 32

```

/*****
/*      INICIALIZACOES      ( ENDEREÇO DE JUMP DO RESET )      */
*****/

1017 1  INICIO: PROCEDURE REENTRANT PUBLIC;
1018 2  DISABLE;
/*****
/* INICIALIZA ENDEREÇO DE TRATAMENTO DE INTERRUPCAO */
*****/

1019 2  CALL SET$INTERRUPT( 0h,ESTOURO);
1020 2  CALL SET$INTERRUPT( 2h,NMI);
1021 2  CALL SET$INTERRUPT( 4h,ESTOURO);
1022 2  CALL SET$INTERRUPT( 16,ESTOURO);
1023 2  CALL SET$INTERRUPT(80h,INTERRUPCAO);

/* if ca <> 0FEh then */

/*****
/* INICIO FRIO: NAO POR CAUSA DE FALHA NA ENERGIA */
*****/

/* do;*/
1024 2  senrn = ativo;
1025 2  faixar = 10*4; /* Faixa de desvio para alarmar ar quente ou frio = 1,0C. */
1026 2  MODO = AR;
1027 2  RN$DESEJ = 365*4;
1028 2  AR$DESEJ = 290*4;
/* TEMP$ARSREAL = 300*4; */
1029 2  temp$rn$real = 365*4;
1030 2  SET$POINT = 300*4;

$ if so2 = Offh
    esto2 = 02;
$ endif

/* end;*/

/*****
/* INICIO QUENTE: POR CAUSA DE FALHA NA ENERGIA */
*****/

1031 2  erro = 0;
1032 2  ca = 0;
1033 2  TEMPO1 = 0;
1034 2  TEMPO2 = 0;
1035 2  TEMPO3 = 0;
1036 2  TEMPO4 = 0;
1037 2  TEMPO5 = 0;
1038 2  TEMPO6 = 0;
1039 2  TEMPO7 = 0;
1040 2  TEMPO8 = 0;
1041 2  NL = 0;

```


PL/M-86 COMPILER INCUBA
10/11/95 07:38:23 PAGE 33

```

/* INICIALIZA BUFFERS DE TEMPERATURA DO AR */
1042 2      DO I = 0 TO 7;
1043 3      TEMPSAR$1(I) = 300*4;
1044 3      END;
1045 2      DO I = 0 TO 7;
1046 3      TEMPSARS$2(I) = 300*4;
1047 3      END;

/* INICIALIZA BUFFERS DE TEMPERATURA DO RN */
/*      DO I = 0 TO 7;          */
/*      TEMPSRNS$1(I) = 355*4;  */
/*      END;                    */

/*****/

1048 2      IMSAIDASRW = OABUZ and OAMODRN and OAMODAR and OASIL;
1049 2      OUTPUT(SAIDASRW) = IMSAIDASRW; /*LIGA BUZINA E ACENDE LEDS*/
1050 3      do while tempol<32000; tempol = tempol+1; end;
1053 2      tempol=0;
1054 2      IMSAIDASRW = OOBUZ or OOMODRN or OOMODAR or OORESAQ;
1055 2      OUTPUT(SAIDASRW) = IMSAIDASRW; /* PARA BUZINA E APAGA LEDS */
/*                                /* EXCETO O LED VERMELHO */
/*****/

/*****/
/* INICIALIZA DISPLAYs */
/*****/

/*****/
/* INICIALIZA CG RAM */
/*****/

1056 3      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1058 2      OUTPUT(CRISTALRW) = 40H;
1059 2      DO I = 0 TO 63;
1060 4      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1062 3      OUTPUT(CRISTALRW+1) = CGRAMRW(I);
1063 3      END;

$ if bal = Offh
1064 2      POINTIN = 0;
1065 2      cursor = 0;

1066 3      DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
1068 2      OUTPUT(CRISTALBAL) = 40H;

1069 2      DO I = 0 TO 63;
1070 4      DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
1072 3      OUTPUT(CRISTALBAL+1) = CGRAMBAL(I);
1073 3      END;

1074 2      OUTPUT(CRISTALBAL) = 38H;
1075 3      DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
1077 2      OUTPUT(CRISTALBAL) = 38H; /* 8 bits*/
1078 3      DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
1080 2      OUTPUT(CRISTALBAL) = 06H; /*MODO*/
1081 3      DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
1083 2      OUTPUT(CRISTALBAL) = 0CH; /*desloca p/direita*/
1084 3      DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
1086 2      OUTPUT(CRISTALBAL) = 01H; /*CLEAR*/
1087 3      DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
1089 2      OUTPUT(CRISTALBAL) = 03H; /*HOME*/
1090 3      DO WHILE (INPUT (CRISTALBAL) OR 7FH)=OFFH;END;
1092 2      OUTPUT(CRISTALBAL) = 80H; /*HOME*/

1093 2      call zere;

```


PL/M-86 COMPILER INCUBA
10/11/95 07:38:23 PAGE 34

\$ if ser = 0

1094 2 tara = converte(0);

\$ endif

\$ if ser = 0ffh

/******
/* INICIALIZA 8251 */
/******

OUTPUT (SERIAL\$COMANDO) = 0;
OUTPUT (SERIAL\$COMANDO) = 0;
OUTPUT (SERIAL\$COMANDO) = 0;
OUTPUT (SERIAL\$COMANDO) = 40H; /* RESET */
OUTPUT (SERIAL\$COMANDO) = 11001110b; /* 2 SB,s/parid.,8bits,x16 */
OUTPUT (SERIAL\$COMANDO) = 00111110b; /* zera erro, recebendo */

\$ if 1 = 2

MARCA1:

media = 0;
DO WHILE ((INPUT(SERIAL\$STAT) and 02H)=0); END;
IF (INPUT(SERIAL\$DADO) <> 02H) THEN GOTO MARCA1;

DO WHILE ((INPUT(SERIAL\$STAT) and 02H)=0); END;
IF (INPUT(SERIAL\$DADO) <> 36H) THEN GOTO MARCA1;

DO WHILE ((INPUT(SERIAL\$STAT) and 02H)=0); END;
IF (INPUT(SERIAL\$DADO) <> 2EH) THEN GOTO MARCA1;

media = media + leser * 1000;
media = media + leser * 100;
media = media + leser * 10;
media = media + leser;
DO WHILE ((INPUT(SERIAL\$STAT) and 02H)=0); END;
IF (INPUT(SERIAL\$DADO) <> 0DH) THEN CALL DEFEITO;

\$ endif

\$ endif

\$ endif

PL/M-86 COMPILER INCUBA
10/11/95 07:38:23 PAGE 35

```

1095 3      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1097 2      OUTPUT(CRISTALRW) = 38H;
1098 3      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1100 2      OUTPUT(CRISTALRW) = 38H; /* 8 bits*/
1101 3      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1103 2      OUTPUT(CRISTALRW) = 06H; /*MODO*/
1104 3      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1106 2      OUTPUT(CRISTALRW) = 0cH; /*desloca p/direita*/
1107 3      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1109 2      OUTPUT(CRISTALRW) = 01H; /*CLEAR*/
1110 3      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1112 2      OUTPUT(CRISTALRW) = 03H; /*HOME*/
1113 3      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1115 2      OUTPUT(CRISTALRW) = 80H; /*HOME*/

$ if teste = 0

/******
/* LE VALOR DESEJADO PARA TEMPERATURA DO AR NA CUPULA /\ \ /
/******

1116 3      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1118 2      OUTPUT(CRISTALRW) = 080H; /*ENDERECO*/
1119 3      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1121 2      DO I = 0 TO 15;
1122 4          DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1124 3          OUTPUT(CRISTALRW+1) = ESCOLHE1(I);
1125 3      END;

/* verifica se deve diminuir ou aumentar O SET$POINT */
/* (tempo2 < 32000) AND */
1126 2      DO WHILE ((tempo2 < 10000) AND ((INPUT(ENTRADAS) and Tsilencia) <> 0));
1127 4          IF((INPUT(ENTRADAS) and Tdiminui)=0) THEN do;CALL DIMINUI;tempo6=tempo6-
5;TEMPO2=0;end;
1133 4          ELSE IF((INPUT(ENTRADAS) and Taumenta)=0) THEN do;CALL
AUMENTA;tempo5=tempo5-5;TEMPO2=0;end;
1139 3          tempo2 = tempo2+1; /* 19 */

1140 4      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1142 3      OUTPUT(CRISTALRW) = 0COH; /*ENDERECO*/
1143 4      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1145 3      DO I = 0 TO 0AH;
1146 5          DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1148 4          OUTPUT(CRISTALRW+1) = ESCOLHE2(I);
1149 4          END;
1150 3          CALL ESCTEMPDES (SET$POINT);
1151 4          DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1153 3          OUTPUT(CRISTALRW+1) = 0;

```

PL/M-86 COMPILER INCUBA
10/11/95 07:38:23 PAGE 36

```

/******
/* MONITORA SE QUER CALIBRAR */
/******

1154 3      IF((INPUT(ENTRADAS) and (Taugmenta or Tsilencia))=0)
1155 4      THEN DO; CALL CALIBRA; HALT; END;

$ if loop = Offh
/******
/* MONITORA FALHA NA ENERGIA */
/******

IF((INPUT(ENTRADAS) and FALTA) = FALTA) THEN
DO;
  disable;
  ca = OFEh;
  IMSAIDASRW = IMSAIDASRW AND OABUZ;
  OUTPUT(SAIDASRW) = OABUZ; /* LIGA BUZINA */
  DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
  OUTPUT(CRISTALRW) = 080H; /*ENDERECO*/

  DO I1 = 0 TO 15;
  DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
  OUTPUT(CRISTALRW+1) = ENERGIA(I1);
  END;

  DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
  OUTPUT(CRISTALRW) = 0C0H; /*ENDERECO*/

  DO I1 = 0 TO 15;
  DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
  OUTPUT(CRISTALRW+1) = ENERGIA2(I1);
  END;

  OUTPUT(SAIDASRW) = OFFh; /* DESLIGA BUZINA */

  HALT;
  DISABLE;
LOOP1:
  GO TO LOOP1;

  END;
$ endif
/* loop */

```

PL/M-86 COMPILER INCUBA
10/11/95 07:38:23 PAGE 37

```
$ if cir = Offh

    /* MONITORA CIRCULACAO DO AR NA CUPULA */
    /* ( Verifica se ar circula ) */
1159 3      DO WHILE CONVERTE(7) < 80; /* 800;*/
1160 4      IMSAIDASRW = IMSAIDASRW and ( OABUZ and OASIL );
1161 4      OUTPUT(SAIDASRW) = IMSAIDASRW; /* LIGA BUZINA e LED */
1162 4      CALL CIRCULA;
1163 4      END;
1164 3      IMSAIDASRW = IMSAIDASRW OR OOBUZ;
1165 3      OUTPUT(SAIDASRW) = IMSAIDASRW; /* DESLIGA BUZINA */

$ endif
/* cir */

1166 3      END;
1167 2      tempo2 = 0;

$ IF 10=20

    DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
    OUTPUT(CRISTALBAL) = 0FH; /*desloca p/direita*/

$ ENDIF
/*******/

1168 2      IMSAIDASRW = OABUZ and OAMODRN and OAMODAR and OASIL;
1169 2      OUTPUT(SAIDASRW) = IMSAIDASRW; /*LIGA BUZINA E ACENDE LEDS*/
1170 3      do while tempol<32000; tempol = tempol+1; end;
1173 2      tempol=0;
1174 2      IMSAIDASRW = OOBUZ or OOMODRN or OOMODAR or OORESAQ or OOSIL;
1175 2      OUTPUT(SAIDASRW) = IMSAIDASRW; /*PARA BUZINA E APAGA LEDS*/

    /*******/
    /* ESPERA TEMPERATURA DO AR NA CUPULA ATINGIR VALOR DESEJADO */
    /*******/

1176 2      ar$desej = set$point;
1177 2      temp$ar$real = (linearizal(CONVERTE(2))+linearizal(CONVERTE(1)))/2;
1178 2      DO WHILE (temp$ar$real < SET$POINT);

        /*******/
        /* MONITORA SE QUER CALIBRAR */
        /*******/

1179 3      IF((INPUT(ENTRADAS) and (Taumenta or Tsilencia))=0)
1180 4      THEN DO; CALL CALIBRA; HALT; END;
```


PL/M-86 COMPILER INCUBA
10/11/95 07:38:23 PAGE 38

\$ if loop = Offh

```

/*****
/* MONITORA FALHA NA ENERGIA */
*****/
IF((INPUT(ENTRADAS) and FALTA)= FALTA) THEN
DO;
  disable;
  ca = 0FEh;
  IMSAIDASRW = IMSAIDASRW AND OABUZ;
  OUTPUT(SAIDASRW) = OABUZ; /* LIGA BUZINA */
  DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
  OUTPUT(CRISTALRW) = 080H; /*ENDERECO*/

  DO I1 = 0 TO 15;
  DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
  OUTPUT(CRISTALRW+1) = ENERGIA(I1);
  END;

  DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
  OUTPUT(CRISTALRW) = 0COH; /*ENDERECO*/

  DO I1 = 0 TO 15;
  DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
  OUTPUT(CRISTALRW+1) = ENERGIA2(I1);
  END;

  OUTPUT(SAIDASRW) = OFFh; /* DESLIGA BUZINA */

HALT;
DISABLE;
LOOP2:
GO TO LOOP2;

END;

$ endif

/* loop */

/*****
/* MONITORA CIRCULACAO DO AR NA CUPULA */
*****/

/* ( Verifica se ar circula ) */

1184 3          DO WHILE CONVERTE(7) < 80; /* 800; */
1185 4          IMSAIDASRW = IMSAIDASRW and ( OABUZ and OASIL );
1186 4          OUTPUT(SAIDASRW) = IMSAIDASRW; /* LIGA BUZINA e LED */
1187 4          CALL CIRCULA;
1188 4          END;
1189 3          IMSAIDASRW = IMSAIDASRW OR OOBUZ;
1190 3          OUTPUT(SAIDASRW) = IMSAIDASRW; /* DESLIGA BUZINA */

1191 3          CALL LIGA;

1192 4          DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1194 3          OUTPUT(CRISTALRW) = 080H; /*ENDERECO*/
1195 4          DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1197 3          DO I = 0 TO 10;
1198 5          DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;

```

PL/M-86 COMPILER INCUBA
10/11/95 07:38:23 PAGE 39

```

1200 4      OUTPUT(CRISTALRW+1) = AGUARDE(I);
1201 4      END;
1202 3      CALL ESCTEMPPar (temp$ar$real);
1203 4      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1205 3      OUTPUT(CRISTALRW+1) = 0;

1206 4      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1208 3      OUTPUT(CRISTALRW) = 0COH; /*ENDEREÇO*/
1209 4      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1211 3      DO I = 0 TO 0AH;
1212 5      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1214 4      OUTPUT(CRISTALRW+1) = ARCUPULA(I);
1215 4      END;
1216 3      temp$ar$real = (linearizal(CONVERTE(2))+linearizal(CONVERTE(1)))/2;
1217 3      CALL ESCTEMPDES (set$point);
1218 4      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1220 3      OUTPUT(CRISTALRW+1) = 0;

      $ if bl = Offh

/* verifica se deve ligar/desligar back light */
1221 3      IF (((INPUT(ENTRADAS) and Tsilencia)=0) AND ((IMSAIDASRW or OABUZ) = Offh))
      then
1222 3          do;
1223 4              tempo8 = tempo8 + 1;
1224 4              if tempo8 > 100 then
1225 4                  do;
1226 5                      IMSAIDASRW = IMSAIDASRW xor OOSIL;
1227 5                      OUTPUT(SAIDASRW) = IMSAIDASRW;
1228 5                      tempo8 = 0;
1229 5                      end;
1230 4              end;
      $ endif

1231 3      END;

/******
/* QUANDO TEMPERATURA DO AR NA CUPULA ATINGIU O VALOR DESEJADO */
/******

1232 3      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1234 2      OUTPUT(CRISTALRW) = 080H; /*ENDEREÇO*/
1235 3      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1237 2      DO I = 0 TO 15;
1238 4      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1240 3      OUTPUT(CRISTALRW+1) = PRONTO1(I);
1241 3      END;

1242 3      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1244 2      OUTPUT(CRISTALRW) = 0COH; /*ENDEREÇO*/
1245 3      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1247 2      DO I = 0 TO 15;
1248 4      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1250 3      OUTPUT(CRISTALRW+1) = PRONTO2(I);
1251 3      END;

```

PL/M-86 COMPILER INCUBA
10/11/95 07:38:23 PAGE 40

```

1252 2      IMSAIDASRW = 087H; /* desliga buzina, liga resistencia */
          /******
          /* ENQUANTO A TECLA silencia alarme NAO E PRESSIONADA , */
          /* MANTEM A TEMPERATURA DO AR NA CUPULA NO VALOR DESEJADO */
          /******
1253 2      do while ((Tsilencia and input(entradas)) <> 0);
1254 3          IMSAIDASRW = IMSAIDASRW xor ( OOBUZ or OOSIL );
1255 3          OUTPUT(SAIDASRW) = IMSAIDASRW; /* PISCA BUZINA E LED */
1256 3          do while (tempo2<32000) AND ((Tsilencia and input(entradas)) <> 0);
1257 4              tempo2 = tempo2+1;
1258 4          end;
1259 3          tempo2=0;
1260 3          temp$ar$real = (linearizal(CONVERTE(2))+linearizal(CONVERTE(1)))/2;
1261 3          if ( temp$ar$real > (set$point-2)) then call desliga;
1263 3          else call liga;
1264 3          end;

1265 2      IMSAIDASRW = ( OOBUZ or OOMODRN or OOSIL or OORESQA );
          /* LED DO MODO AR LIGADO, RESISTENCIA */
1266 2      OUTPUT(SAIDASRW) = IMSAIDASRW; /* e BUZZER DESLIGADOS, ENDERECOS DO */
          /* MUX em ar1 */

$ endif

/* if teste */

$ if bal = Offh

1267 2      ciclo = 0;
1268 2      seg = 0;
1269 2      min = 0;
1270 2      hora = 0;
          /* INICIALIZA BUFFER DE PESO DO RN */
1271 2      DO I = 0 TO 1380;
1272 3          PESO(I) = 000;
1273 3          END;
$ endif

/*end;*/

```


PL/M-86 COMPILER INCUBA
10/11/95 07:38:23 PAGE 41

```

/*****
/* LE O PAINEL e RETORNA MODO DE FUNCIONAMENTO */
/* ALARMANDO ENTRADA OU INTERRUPCAO NA CIRCULACAO */
/* DE AR ou ALTERACAO NO FLUXO DE O2. */
*****/

1274 2    do while 2>1;

        /* IF((INPUT(ENTRADAS) and (Taumenta or Tsilencia))=0) */
        /* THEN DO; CALL CALIBRA; HALT; END; */

        $ if bal = Offh

1275 3            IF((INPUT(ENTRADAS) and (Tadiana or Tatrassa))=0)
1276 3            THEN do;
1277 4                call zerado;

        $ if ser = 0
1278 4                media = 0;
1279 4                tara = converte(0);

        $ endif

1280 4                end;
1281 3            ELSE DO;
1282 4                IF((INPUT(ENTRADAS) and Tatrassa)=0) THEN call atrassa;
1283 4                IF((INPUT(ENTRADAS) and Tadiana)=0) THEN call adianta;
1284 4                END;
1285 4            $ endif

1287 3    do;

        $ if so2 = Offh

        /* ( Verifica se entra ou sai O2 em quaisquer modos) */

        IF (INPUT(ENTRADAS) and O2 <> esto2) THEN

                do;
                ca = 8;
                CALL SENSO2;
                esto2 = INPUT(ENTRADAS) and O2;
                end;

        else if ((ca = 8) and ( tempol <> 0)) then ca = 0;

        $ endif

        /* silencia alarme se tecla apertada */
        /*(tecla apertada em quaisquer modos AND alarme ativado ) OU (tempol<>0)*/
1288 4    IF (((INPUT(ENTRADAS) and Tsilencia)=0) AND((IMSAIDASRW AND 80H) = 0)) OR
        (TEMPOL<>0)) then
1289 4        do;
1290 5            IMSAIDASRW = (IMSAIDASRW AND OASIL) OR OOBUZ;
1291 5            OUTPUT(SAIDASRW) = IMSAIDASRW; /*PARA BUZINA E ACENDE LED vermelho */
1292 5            if ((ca = 5) or (ca = 6)) then
1293 5                do;
1294 6                    senrn = fora;
1295 6                    ca = 0;

```


PL/M-86 COMPILER INCUBA
10/11/95 07:38:23 PAGE 42

```

1296 6      modo = ar;
1297 6      IMSAIDASRW = IMSAIDASRW or OOMODRN;
1298 6      IMSAIDASRW = IMSAIDASRW and (OABUZ and OAMODAR);
1299 6      end;
1300 5      else do;
1301 6          TEMPO1 = TEMPO1 + 1;
1302 6          IF TEMPO1 > 32760 THEN
1303 6              DO;
1304 7                  TEMPO1 = 0;
1305 7                  IMSAIDASRW = (IMSAIDASRW AND OABUZ);
1306 7          IF CA <> 0ffh then IMSAIDASRW = IMSAIDASRW OR OOSIL;
1307 7          OUTPUT(SAIDASRW) = IMSAIDASRW; /* LIGA BUZINA E APAGA LED */
1308 7          end;
1309 7      end;
1310 6      end;
1311 5      end;

/******
/* Verifica se existe alguma causa de alarme em quaisquer modos */
/******

1312 4      IF ((ca <> 0) and (ca <> 0ffh)) THEN call alarme;

/******
/* INICIO DO caso nao exista causa de alarmes */
/******

1314 4      ELSE
1315 4      DO;

$ if bl = 0ffh

/* verifica se deve ligar back light */
1315 5      IF (INPUT(ENTRADAS) and Tsilencia)=0 /* AND((IMSAIDASRW or OABUZ) = 0ffh) */
1316 5      then
1317 5          do;
1318 6              tempo8 = tempo8 + 1;
1319 6              if tempo8 > 50 then
1320 6                  do;
1321 7                      ca = not (ca);
1322 7                      IMSAIDASRW = IMSAIDASRW xor OOSIL;
1323 7                      IMSAIDASRW = IMSAIDASRW and OASIL; /*
1324 7                      OUTPUT(SAIDASRW) = IMSAIDASRW;
1325 6                      tempo8 = 0;
1326 5                      end;
1327 6                      end;
1328 5                      else
1329 5                      $ endif
1330 5                      do; call desalarma; end;
1331 5                      tempo1 = 0;

/******
/* escreve temperatura DESEJADA */
/******

```

PL/M-86 COMPILER INCUBA
10/11/95 07:38:23 PAGE 43

```

1330 6      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1332 5      OUTPUT(CRISTALRW) = 0COH; /*ENDERECO*/

1333 5      IF MODO = AR THEN
1334 5      DO;
1335 7          DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1337 6              DO I = 0 TO 10;
1338 8                  DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1340 7                      OUTPUT(CRISTALRW+1) = desej$ar(I);
1341 7                          END;
1342 6      END;

1343 5      else if MODO = RN then
1344 5      do;
1345 7          DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1347 6              DO I = 0 TO 10;
1348 8                  DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1350 7                      OUTPUT(CRISTALRW+1) = desej$rn(I);
1351 7                          END;
1352 6      end;

/******
/*      escreve temperaturas REAIS      */
/******

1353 6      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1355 5      OUTPUT(CRISTALRW) = 080H; /*ENDERECO*/
1356 6      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1358 5          DO I = 0 TO 2;
1359 7              DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1361 6                  OUTPUT(CRISTALRW+1) = temperaturas(I);
1362 6                  END;

$ if s=Offh
DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
OUTPUT(CRISTALRW) = 083H; /*ENDERECO*/
DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
DO I = 3 TO 6;
DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
OUTPUT(CRISTALRW+1) = temperaturas(I);
END;
$ endif

1363 6      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1365 5      OUTPUT(CRISTALRW) = 087H; /*ENDERECO*/
1366 5      DO I = 7 TO 10;
1367 7          DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1369 6              OUTPUT(CRISTALRW+1) = TEMPERATURAS(I);
1370 6              END;
1371 6      do while (input (cristalrw) or 7fh)=Offh;end;
1373 5      output(cristalrw) = 8fh;
1374 6      DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1376 5      OUTPUT(CRISTALRW+1) = 0;

1377 5      END;

```

BIBLIOTECA
FACULDADE DE FILOSOFIA
CIÊNCIAS E LETRAS DE
RIBEIRÃO PRETO DA U.S.P.

PL/M-86 COMPILER INCUBA
10/11/95 07:38:23 PAGE 44

```

/*****
/*      FIM DO caso nao exista causa de alarmes      */
*****/

1378  4      call esctempar(temp$ar$real);
1379  4      $ if s=0h      call esctempnrn(TEMP$rn$real);
      $ endif

1380  4      /* verifica se a tecla de modo esta pressionada */
1381  4      IF((INPUT(ENTRADAS) and Tmodo)=0) THEN
1382  5      DO;
1383  5          IF TEMPO2 > 50 THEN /* 20 */
1384  6              DO;
1385  6                  MODO = NOT (MODO);
1386  6                  if modo=AR then
1387  7                      do;
1388  7                          IMSAIDASRW = IMSAIDASRW or OOMODRN;
1389  7                          IMSAIDASRW = IMSAIDASRW and (OABUZ and OAMODAR);
1390  7                          senrn = ativo;
1391  6                      end;
1392  6                  else
1393  7                      do;
1394  7                          IMSAIDASRW = IMSAIDASRW or OOMODAR;
1395  7                          IMSAIDASRW = IMSAIDASRW and (OABUZ and OAMODRN);
1396  6                      end;
1397  6                      OUTPUT(SAIDASRW) = IMSAIDASRW; /* LIGA BUZINA e LED MODO */
1398  6                      TEMPO2 = 0;
1399  6                      IMSAIDASRW = IMSAIDASRW OR OOBUZ;
1400  6                      OUTPUT(SAIDASRW) = IMSAIDASRW; /* DESLIGA BUZINA */
1401  5                      END;
1402  5                  ELSE TEMPO2 = TEMPO2 + 1;
1403  5                  END;

/*****
1402  4      IF MODO = AR THEN
1403  4      DO;
1404  5          temp$real = temp$ar$real;
1405  5          SET$POINT = AR$DESEJ;
1406  5      END;

1407  4      else if MODO = RN then
1408  4      do;
1409  5          senrn = ativo;
1410  5          temp$real = temp$rn$real;
1411  5          SET$POINT = RN$DESEJ;
1412  5      end;

/* verifica se deve diminuir ou aumentar O SET$POINT */
1413  5      IF((INPUT(ENTRADAS) and Tdiminui)=0) THEN do; CALL DIMINUI;end;
1417  5      ELSE IF((INPUT(ENTRADAS) and Taumenta)=0) THEN do; CALL AUMENTA;end;

```

PL/M-86 COMPILER INCUBA
10/11/95 07:38:23 PAGE 45

```

/*****/

/* ATUALIZA VALOR DESEJADO */
1421 4      IF MODO = AR THEN AR$DESEJ = SET$POINT;
1423 4      else if MODO = RN then
1424 4      RN$DESEJ = SET$POINT;

1425 4      CALL esctempdes(SET$POINT);
1426 5          DO WHILE (INPUT (CRISTALRW) OR 7FH)=OFFH;END;
1428 4          OUTPUT(CRISTALRW+1) = 0;

/*****/

/* IF((INPUT(ENTRADAS) and FALTA)= FALTA) THEN CALL NMI; */
1429 4      END; /* CALIBRA */
1430 3      enable;
1431 3      end;

1432 2      END INICIO;
1433 1      halt;
1434 1      END INCUBA;

```