

Escola Politécnica
da Universidade de São Paulo

Departamento de Engenharia Mecânica

Área de Automação e Sistemas

Área de Projeto e Fabricação

Sistema de Automação Predial

Alexandre Costa e Silva

Marcelo Proni

(~~nota~~ comparecer 3,0 22 dez 98)
OK 22 dez 98

1998

Avaliação : 6,5 (seis e cinco)
Edilson Hiroshi Tamai
21/12/1998

Alexandre Costa e Silva

Marcelo Proni

Sistema de Automação Predial

Trabalho de Formatura apresentado à

Escola Politécnica da Universidade de São Paulo

Prof. Orientador:

Prof. Dr. Edílson Hiroshi Tamai

1998

AGRADECIMENTOS

Gostaríamos de agradecer ao Prof. Dr. Edílson H. Tamai não só pela orientação e incentivo no desenvolvimento deste projeto, mas também por sua pré-disposição em nos ajudar em todos os problemas que encontramos ao longo do mesmo, sem o qual a concepção deste não seria possível.

Às nossas famílias, por todo o resto.

ÍNDICE

ÍNDICE	3
1. INTRODUÇÃO	5
2. MOTIVAÇÃO	7
3. OBJETIVOS E NECESSIDADES	8
4. DECISÃO	10
Sistema de Automação Central	14
Sistema de Automação Modular	15
5. APLICAÇÕES	17
1. Monitoração de Consumo de Energia Elétrica	17
2. Controle de Temperatura	18
3. Controle de Ar Condicionado	19
4. Detecção de Incêndio	19
5. Detecção de movimento	20
6. DELIMITAÇÃO DO PROJETO	21
7. METODOLOGIA	22
8. DECISÕES DURANTE O DESENVOLVIMENTO	24
9. CONFIGURAÇÃO DO SISTEMA	25
10. BRIEFFING DE FUNCIONAMENTO	26
11. DESCRIÇÃO DO HARDWARE	27
11.1 ESTAÇÃO CENTRAL	27
11.2 CONSTITUIÇÃO BÁSICA:	27
11.2.1 MICROCOMPUTADOR	28
A INTERFACE PARALELA 8255	30

CONVERSOR ANALÓGICO-DIGITAL (CAD 0808/0809)	32
O CONTADOR DECÁDICO 74192	33
COMPARADORES 7485	33
11.3 UNIDADES DE INTERFACE DE DADOS	34
11.3.1 MÓDULO DE MONITORAÇÃO DE CONSUMO DE ENERGIA ...	35
11.3.2 MÓDULO DE ACIONAMENTO DE CARGAS	36
11.3.3 MÓDULO DE CONTROLE DE TEMPERATURA.....	37
11.3.4 MÓDULO DE CONTROLE DA CENTRAL DE AR CONDICIONADO	37
11.3.5 MÓDULO DE DETECÇÃO DE INCÊNDIO	38
12. DESCRIÇÃO DO SOFTWARE.....	40
13. FILOSOFIA DE OPERAÇÃO.....	41
13.1 COMPOSIÇÃO DA IHM	41
13.1.1 TELA DE APRESENTAÇÃO.....	41
13.1.2 TELA ÍNDICE.....	42
MENU 1	43
MENU 2	44
TIMER	45
TEMPERATURA.....	48
DEMANDA.....	49
13.2 AQUISIÇÃO DE DADOS.....	51
14. COMENTÁRIOS.....	53
15. SIMULAÇÃO	54
15.1 PROGRAMADOR DE TIMER.....	54
15.2 CONTROLADOR DE DEMANDA.....	54
15.3 CONTROLADOR DE TEMPERATURA.....	54
15.4 CENTRAL DE AR CONDICIONADO.....	55
16. BIBLIOGRAFIA	56
ANEXO I – Diagramas Lógicos.....	58
ANEXO II - Fluxograma	63
ANEXO III - Programa	68

1. INTRODUÇÃO

Com a crescente globalização da economia mundial, tem se verificado uma intensa necessidade de implementação de processos com a finalidade de que as empresas se inserissem numa realidade mais competitiva no mercado mundial.

Para que essa realidade seja alcançada, um dos pontos mais importantes a ser adaptado é em relação aos custos. Esses, não apenas relativos à produção, como também à administração, devem ser revistos de forma a se conseguir uma redução significativa.

Com o avanço da tecnologia procedeu-se uma verdadeira revolução na indústria microeletrônica, a qual resultou no surgimento de uma nova linha de microprocessadores que associado ao aumento na capacidade e velocidade de processamento ocasionou um considerável crescimento na indústria da informática viabilizando a utilização de computadores em diversos segmentos da sociedade.

Através da utilização das diversas técnicas e processos de automatização da produção aprimoradas nos últimos anos, o segmento industrial conseguiu

evoluir suficientemente a eficiência e produtividade do processo produtivo, aliado a uma grande redução nos custos. Além disto, toda a construção está sujeita a certos tipos de acidentes, falhas, etc. que, através de um sistema baseado em observação e controle, podem ser evitados se detectados com antecedência.

A importância de se possuir sistema de controle, que monitore todos esses tipos de ocorrência, não só é importante para se evitar eventuais problemas, como também pelo fator econômico, visando a minoração dos desperdícios. No mercado estão disponíveis, atualmente, sistemas que controlam o consumo de energia elétrica, de água, entrada e saída de pessoas, entre outros.

2. MOTIVAÇÃO

Um dos grandes problemas brasileiros, principalmente no setor público, é o desperdício. Especificamente na Universidade de São Paulo, que faz parte do universo do nosso dia a dia, é verificada uma falta de controle tanto em relação aos gastos relativos à energia elétrica, como também em relação à segurança das instalações.

Com base nisso surgiu a idéia de se desenvolver um projeto visando uma maior eficiência no controle dos procedimentos empregados, não só como nos gastos com eletricidade, mas também nos controles para se evitar incêndios, acesso às instalações, etc.

Através da observação de diversos sistemas similares encontrados em diversas empresas, faculdades, escolas, condomínios, etc., os quais, muitas vezes, se preocupam apenas em alguns dos pontos citados, surgiu a idéia de se criar um modelo que pudesse integrar todas as necessidades em um controle único, ou central, baseado em módulos de forma a permitir que a sua implementação seja feita de acordo com as necessidades de cada usuário.

3. OBJETIVOS E NECESSIDADES

O sistema digital de supervisão e controle tem como objetivo principal atuar na automação de Sistemas Prediais e/ou Industriais de forma segura e concisa. Ele é constituído basicamente por uma central informatizada, unidades de coletas de dados distribuídas por pontos estratégicos da área a ser monitorada, que transmitem as informações adquiridas à estação central de forma paralela.

Este sistema deverá ser constituído, basicamente, de um Controle Central composto por um computador que recebe informações de cada módulo do sistema, processa estas informações através de um software desenvolvido especificamente para tal sistema, gerando respostas indicativas dos procedimentos a serem realizados.

O sistema deverá conter módulos que serão os responsáveis pela coleta de dados. Esses módulos serão formados por circuitos eletrônicos discretos específicos para cada variável controlada. Eles também deverão enviar esses dados para o Controle Central, onde serão analisados possibilitando uma ação rápida e imediata quanto as medidas a serem tomadas.

Além das características citadas acima, é necessário que o sistema a ser desenvolvido tenha algumas necessidades atendidas. Dentre estas, é

necessário que o sistema possua uma estrutura que seja de fácil compreensão de forma a se adaptar facilmente aos mais diversos tipos de usuários e condições de implantação. Um outro ponto importante é em relação à manutenção do sistema. Além de confiável, qualquer sistema deve ter uma manutenção rápida e fácil de modo a que os custos de manutenção e o desperdício de tempo com a manutenção sejam minimizados.

O sistema em questão, também, deve possuir uma estrutura de forma a permitir sua atualização ao longo do tempo. Tanto em relação a novos módulos, quanto em relação a novas tecnologias que, por ventura, venham a surgir.^{[8][13][14]}

4. DECISÃO

Através de um estudo na literatura levantada foi concluído que seriam dois os principais modelos de sistemas que poderiam ser implantados: o Sistema de Automação Central – projeto A – e o Sistema de Automação Modular – Projeto B.

Para se fazer uma análise eficiente, resultando em uma solução mais eficaz, foi feita uma análise quantitativa do problema a fim de que se obtivesse a melhor solução qualitativa. Para isso é necessária a atribuição de pesos a cada item que possa influenciar na escolha da “melhor solução”. Essa atribuição, item a item, é explicada a seguir:

1 – Facilidade de Implantação:

A facilidade de implantação pode ser dividida em dois itens: Primeiro, o tempo de implantação e, em segundo lugar, o tipo de máquina e conseqüente mão de obra empregada na execução. Estes dois fatores implicam diretamente no custo de implantação.

Quanto, principalmente, ao tempo de implantação, nota-se que a solução A leva uma certa vantagem, visto que o seu processo de implantação é mais simples e centralizado, embora apresente uma certa complexidade de redes e cabos.

O peso atribuído a este parâmetro será de apenas 4, devido ao fato de não ser um aspecto de grande influência na decisão da escolha do sistema.

2 – Facilidade de Operação:

A facilidade de operação também pode ser dividida em dois itens, sendo eles: facilidade de adaptação do usuário e grau de complexidade, ou seja, que o sistema apresente uma estrutura em que o usuário, no caso de sua não familiarização, compreenda rapidamente.

Ambas as soluções apresentam implementações tanto em hardware, quanto em software, bastante parecidas, visto que a diferença se comprova apenas na não necessidade de implementação conjunta de todos os módulos da solução B; porém, no que se relaciona à facilidade de adaptação, não se dimensiona diferença significativa em nenhuma solução. Nota-se ligeiramente menor o grau de complexidade relativo à solução B, uma vez que o sistema modularizado pode apresentar subsistemas divididos em blocos.

Este parâmetro, devido ao fato de estar diretamente ligado ao custo da mão de obra operacional do sistema, receberá peso 7.

3 – Manutenção:

A facilidade de manutenção é um dos fatores mais importantes ao se escolher o sistema. Isso ocorre porque a incompatibilidade entre fabricantes juntamente com a falta de uniformidade dos produtos pode relacioná-la a um custo excessivamente alto. Além disso, outro fator de relevante importância, é se pode ocorrer a necessidade de paralisação total do sistema no caso de um eventual problema em um dos controles executados, sendo que, para certas utilizações como, por exemplo, controle de temperatura para máquinas de

extrema precisão, uma paralisação do mesmo pode resultar na paralisação de diversos processos de uma fábrica podendo comprometer a entrega do produto final.

Esse item receberá peso 10.

4 – Custo:

O custo de concepção e implantação tem influência direta na escolha do sistema a ser adquirido pelo consumidor final.

A solução B tem a vantagem de, por ser um sistema subdividido em módulos, apresentar a possibilidade de escalonamento no investimento, no qual o usuário adquiriria primeiramente os módulos mais importantes e, com o passar do tempo, obteria outros adjacentes.

Devido à sua importância, atribuiremos peso 8 a esse item.

5 – Flexibilidade do Sistema:

A solução deve apresentar alternativas para a não obsolescência do sistema, visto que vivemos em um mundo que atravessa sua terceira revolução industrial com produtos cada vez mais novos e, conseqüentemente, a um preço cada vez mais acessível é oferecido no mercado.

Tendo em vista que nossos potenciais consumidores estão diretamente ligados às instalações prediais novas e que o período de obsolescência máximo permitido a essas é de cinquenta anos, a esse item será atribuído peso 9.

6 – Capacidades Empregadas:

É de vital importância a total satisfação do cliente ao adquirir o novo produto. Sendo assim, quanto maiores as possibilidades de soluções incluídas no sistema maior será o grau de satisfação alcançada.

Em vista disso, esse item receberá peso 10.

Analisando-se as duas arquiteturas, verificou-se a existência de vantagens e desvantagens de cada uma delas. Após as discussões e ponderações sobre cada um dos sistemas optou-se pelo Sistema Modular pelas razões que serão apresentadas a seguir:

	<i>Projeto A</i>	<i>Projeto B</i>	<i>Peso</i>	<i>A x Peso</i>	<i>B x Peso</i>
<i>1 – Facilidade de Implantação</i>	7	5	4	28	20
<i>2 – Facilidade de Operação</i>	7	7	7	49	49
<i>3 – Manutenção</i>	6	9	10	60	90
<i>4 – Custo</i>	8	7	8	64	56
<i>5 – Flexibilidade do Sistema</i>	5	8	9	45	72
<i>6 – Capacidades Empregadas</i>	9	9	10	90	90
<i>TOTAL</i>				336	377

Sistema de Automação Central

Neste tipo de sistema verifica-se que há a existência de uma grande multiplicidade de redes e cabos. Esta multiplicidade pode dificultar a instalação e manutenção do sistema, além de se haver a necessidade de uma paralisação total do mesmo no caso de um eventual problema com um dos controles executados. Um outro ponto importante de se salientar é que

muitas das redes não são compatíveis entre os diversos fabricantes. Tais fatores poderiam tornar o sistema mais rígido e menos propício para atualizações futuras. Além disto, há o problema de um custo de manutenção mais elevado.

Sistema de Automação Modular

Nos sistemas modulares verifica-se que, apesar de um custo de implantação superior ao sistema anteriormente citado, as suas outras características apresentam diversas vantagens que fazem que uma análise de custo-benefício o indique como o mais apropriado de acordo com os objetivos esperados.

O sistema de concepção modular permite a implantação das unidades de coleta de dados de maneira gradativa, oferecendo a possibilidade de escalonamento no investimento, de acordo com as prioridades dos usuários, gerando, assim, uma grande flexibilidade do sistema.

Todas as modificações necessárias nos sistemas já implantados serão realizadas apenas no "software", tornando desnecessário modificações no hardware, devido à composição do sistema ser baseada em circuitos eletrônicos discretos.

A interligação entre a estação central e as unidades de coletas de dados e

realizada através de "flat-cables", de modo que a partir de um cabo central podem partir futuras expansões.

Esta configuração traz como vantagem a redução de cabos, provocando uma considerável redução nos custos de implantação, já que apenas um cabo principal chega até a estação central.

As operações são realizadas automaticamente, e todas as ocorrências são apresentadas no Controle Central, através da saída do software, permitindo que o operador de plantão possua um amplo conhecimento do estado dos sistemas monitorados, e que em caso de necessidade de interação, isto ocorra de forma segura.

Qualquer tipo de ocorrência é documentada no software que pode gerar saídas de informações numa impressora através de relatórios periódicos e também armazenadas em unidades de disco.

A grande vantagem no caso de ocorrência de falhas, independentemente do módulo afetado, é que não haverá a necessidade de desativação de todo o sistema, uma vez que a manutenção pode ser realizada pela simples substituição dos módulos responsáveis pela coleta dos dados, não requerendo técnicos especializados. [8] [12] [13] [14] [16]

5. APLICAÇÕES

Definida a arquitetura do sistema, é necessário se definir quais seriam as áreas de atuação do sistema, ou seja, quais seriam os principais aspectos de uma edificação que deveriam ser monitorados pelo sistema.

As principais áreas em que a aplicação desse sistema passa a oferecer uma relação custo/benefícios vantajosa estão discriminadas a seguir:

1. *Monitoração de Consumo de Energia Elétrica*

Atualmente, a economia de energia elétrica vem sendo um fato preponderante na hora da escolha por um sistema de supervisão e controle. Baseando-se nesse fato será implementado um módulo de monitoração de consumo de energia elétrica, sempre importante em instalações industriais, devido a grande quantidade de energia requerida. Além disto, universidades, escolas, condomínios, etc. poderão diminuir seus custos relativos.

A idéia básica deste módulo origina-se no fato de que, quando um empresa "adquire" de uma concessionária uma quantia estipulada de energia elétrica a ser utilizada em um determinado período de tempo, ela esta se responsabilizando a utilizar, no máximo est. quantia adquirida. Para que não

haja um consumo acima daquele adquirido para o período em questão, a concessionária realiza, medidas em intervalos de tempo regulares a fim de fiscalizar se o patamar máximo de energia esta ou não sendo respeitado. Se, no ato da medição, este patamar tiver sido ultrapassado, é aplicada uma multa a empresa e esta tem sua cota de energia aumentada para este novo valor.

Portanto, para se evitar que tal fato ocorra este módulo do sistema é responsável pela supervisão e controle da energia elétrica sendo consumida instantaneamente pelo mesmo durante todo o processo de monitoração das demais variáveis, através da medição das variáveis determinantes:

- Corrente
- Tensão

Com a transgressão dos valores estabelecidos para estas variáveis, o Controle Central acionará uma rotina responsável pelo desligamento de cargas através de uma sequência automática previamente programada pelo operador (a qual será explicada oportunamente), cargas estas que também deverão estar sendo monitoradas pelo sistema.

2. Controle de Temperatura

Este módulo é responsável pela monitoração da temperatura vinculada a

ambientes onde esta deve ser mantida em um patamar constante, devido, por exemplo, a presença de equipamentos de precisão, onde, qualquer desvio na temperatura ambiente ao qual estão submetidos venha a alterar suas características de funcionamento.

Ele atua conjuntamente com o módulo responsável pelo controle da Central de Ar Condicionado (no caso de existir), o qual, a qualquer desvio existente na temperatura ambiente dos recintos controlados, aciona-o de forma a restabelecer seu patamar anterior.

3. Controle de Ar Condicionado

Através deste módulo pode-se monitorar o sistema central de ar condicionado, o qual deve estar vinculado ao módulo de controle de temperatura.

Estando diretamente relacionado ao controle de temperatura, a Central Ar Condicionado será acionada sempre que houver a necessidade de uma alteração na temperatura registrada em um ambiente devido a um desvio em relação a um patamar preestabelecido.

4. Detecção de Incêndio

Este módulo é responsável pela monitoração e detecção de alarme de

incêndio situados em locais estratégicos. Consiste basicamente de circuitos eletrônicos discretos com a finalidade de detectar a presença de um pequeno foco de incêndio, acionando assim um alarme geral, possibilitando que uma ação imediata e segura seja tomada a fim de solucionar o problema.

5. *Deteção de movimento*

Este módulo é responsável pela monitoração e deteção de movimento nas áreas críticas, de forma a fiscalizar a entrada de pessoas não autorizadas no setor monitorado.

[8] [12] [13] [14] [16]

6. DELIMITAÇÃO DO PROJETO

Por este se tratar de um Projeto para a conclusão do curso de Engenharia e estar limitado o seu desenvolvimento até o final deste ano, foi necessário delimitar o projeto pois da forma em que era inicialmente imaginado viria a ser inviável a sua execução num período de tempo tão limitado.

Deste forma chegou-se à conclusão que deveria constar do projeto a implantação de um hardware, que seria a interface do sistema com os atuadores. Este hardware deverá receber as informações provenientes do sistema de controle de temperatura da sala, transformar estas informações e repassá-las para o software que irá analisar tais dados e, se for necessário, de acordo com as especificações pré-determinadas, enviará um sinal de controle para a central de ar condicionado de forma a colocá-la em funcionamento ou desligá-la de acordo com a necessidade.

A outra parte a ser implantada, como já citada anteriormente, será o software que será composto de alguns módulos, entre eles, um que analisará os dados provenientes da interface e um outro responsável pelo sinal de resposta. Além disto, ele deverá estar apto a receber os outros módulos que poderão ser implantados.

7. METODOLOGIA

A metodologia a ser empregada para o desenvolvimento do projeto será a de um amplo levantamento da literatura existente referente aos sistemas de automação predial de forma a juntar subsídios para fundamentar todas as decisões necessárias.

Com o levantamento da literatura realizado é necessário que se inicie o desenvolvimento do projeto. Este desenvolvimento consiste em se analisar a literatura levantada e, com base nestas informações, definir qual será a estrutura e arquitetura, bem como a realizar a definição de como será o hardware a ser desenvolvido. Um outro ponto importante a ser definido é sobre quais serão as funções existentes no sistema projetado.

Após estes passos passa-se a parte de especificação dos componentes, ou seja, deve-se procurar no mercado quais são os componentes que mais se adaptam às características necessárias e especificar quais serão os utilizados para o desenvolvimento do protótipo.

Com os componentes definidos deve-se passar ao desenvolvimento tanto do hardware como do software. Estes desenvolvimentos devem ser simultâneos de forma a agilizar o processo e permitir testes durante o seu

desenvolvimento para sanar possíveis imperfeições.

No final do desenvolvimento do hardware e software deve-se, então, passar à construção do protótipo para a realização dos ensaios e testes.

8. DECISÕES DURANTE O DESENVOLVIMENTO

Na fase inicial do projeto foi decidido junto ao professor orientador a construção de um protótipo do módulo de controle de ar condicionado e a seleção ou construção de um dos atuadores de temperatura que seria empregado neste módulo.

Este módulo teria sua arquitetura de hardware definida segundo o padrão da placa conversora analógico – digital CAD12/36, devido à disponibilidade da mesma nos laboratórios do Departamento de Engenharia Mecânica da Escola Politécnica da USP. Em decorrência de problemas administrativos, a utilização desta placa foi substituída por uma placa por nós também desenvolvida, explicada posteriormente, mas que , devido ao custo e mão de obra de construção, nos impossibilitou o prosseguimento do protótipo, sendo ao invés disso, implementados outros circuitos inicialmente não previstos (Anexo I).

9. CONFIGURAÇÃO DO SISTEMA

O corpo principal do “Sistema de Automação Predial” é constituído basicamente de um microcomputador, um monitor de vídeo e, opcionalmente, uma impressora. Também pode estar integrado à estação central um equipamento *no-break* e um banco de baterias, a fim de garantir o funcionamento do sistema em caso de falta de energia elétrica fornecida pela concessionária.

10. BRIEFING DE FUNCIONAMENTO

As unidades de coletas de dados são responsáveis, como o próprio nome pode sugerir, pela coleta de informações dos instrumentos ou equipamentos auxiliares, componentes do sistema a ser monitorado, de modo a permitir a transmissão de dados à estação central.

A estação central tem como função básica a gestão de todas as informações provenientes das unidades de coleta de dados, fornecendo ao operador, de forma segura, as instruções necessárias à operação.

As instruções de operação, assim como as informações gerais de estado do sistema, são apresentadas ao operador através de um conjunto de telas no monitor de vídeo e de registros em impressora.

Toda a operação é automática, porém, em qualquer fase do controle supervisorio, através do teclado, é possível que o operador interfira no sistema, executando novos ajustes para as variáveis que estão sendo monitoradas, bem como a mudança na seqüência de acionamento de equipamentos controlados.

11. DESCRIÇÃO DO HARDWARE

11.1 ESTAÇÃO CENTRAL

A estação central do sistema de supervisão e controle "Sistema de Automação Predial", deverá ser instalada, de preferência, em uma sala de controle, de modo a permitir uma operação segura e adequada de todos os sistemas monitorados.

Ela é responsável pelo "Pooling" (Varredura Seqüencial Cíclica) das unidades de coletas de dados e ativação dos periféricos, a partir da ocorrência de alarmes.

11.2 CONSTITUIÇÃO BÁSICA:

Mesa de controle abrigando:

- Computador
- Monitor de vídeo
- Impressora (opcional)
- *No-break* e banco de baterias (opcionais)

11.2.1 MICROCOMPUTADOR

O microcomputador é a parte central do sistema e executa funções de gerenciamento a partir do software básico, software este que foi desenvolvido em Linguagem "C", fazendo parte integrante do arquivo AUTOEXEC.BAT a fim de que, sempre que ocorrer uma energização do sistema, ocorra também a inicialização do mesmo, não exigindo qualquer ação externa para executar operações de carga e inicialização.

CARACTERÍSTICAS:

O microcomputador utilizado como estação central é compatível com os computadores da linha "IBM-PC" com as seguintes características :

- arquitetura baseada no microprocessador Intel 8088
- Canal de comunicação para monitor de vídeo
- Interface paralela para conexão com impressora
- Interface para acionadores de disco flexível e/ou rígido
- alimentação em 110/220 Vac 60 Hz

Faz parte integrante do microcomputador, uma placa de interface por nós desenvolvida constituída principalmente por um dispositivo de interface paralela (8255), um conversor analógico-digital (CAD 0808/0809), um

contador decádico (74192) e conjuntos de endereçamento (7485), conforme visto na figura 1 abaixo e no diagrama lógico no Anexo 1. [1][2][3][6][9][10]

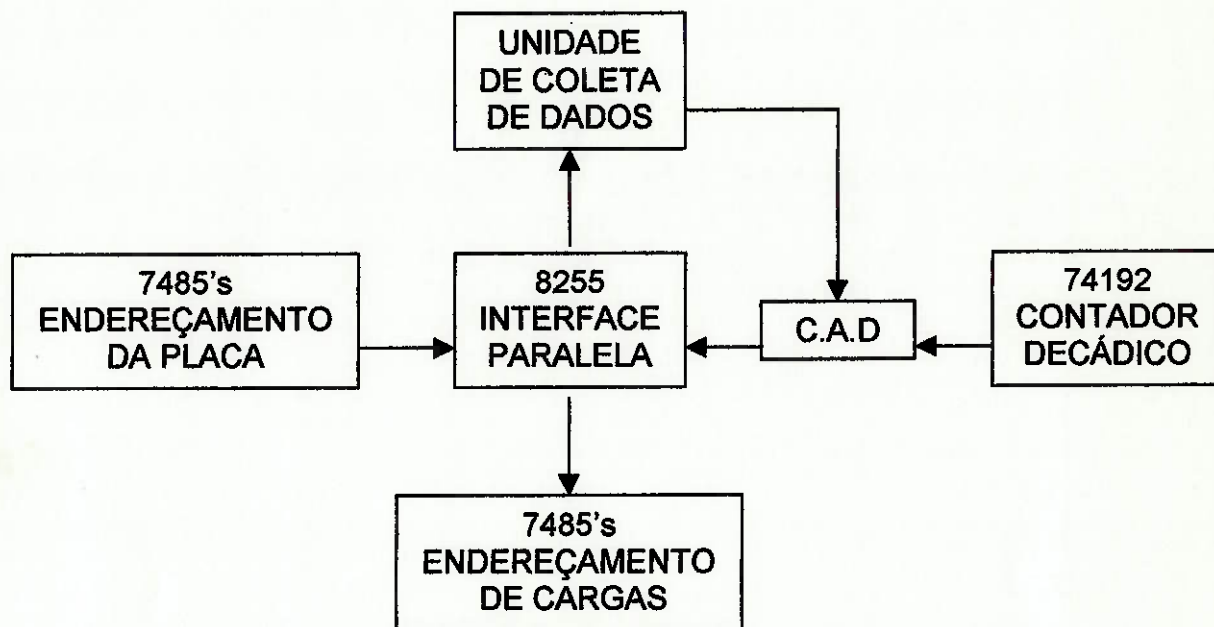


FIGURA 1

A INTERFACE PARALELA 8255

O componente 8255 é responsável pela coleta dos dados de maneira paralela, os quais serão devidamente tratados pelo software. Constitui-se basicamente por três ports de 8 bits cada que podem ser programados para atuarem distintamente como entrada/saída, de acordo com a necessidade de sua aplicação. Para a finalidade a que o projeto se propõe, sua programação se dá através da palavra de controle 83h (hexadecimal) de maneira que tenhamos:

- Port A atuando como saída
- Port B atuando como entrada
- Port C atuando da seguinte maneira:
 - entrada: 4 bits menos significativos
 - saída: 4 bits mais significativos

O endereçamento do dispositivo se faz através de comparadores que emitem um sinal, o qual é introduzido como Chip Select, habilitando-o. Comparadores estes que serão explicados posteriormente.

A correta seleção dos ports se dá através dos sinais A0 e A1 ligados, respectivamente, nos sinais A0 e A1 presentes no slot do microcomputador.

A seleção quanto à escrita ou leitura no dispositivo se dá através dos sinais

de Read e Write, ativos em nível lógico "0", respectivamente ligados aos sinais de mesmo nome do microcomputador.

O port A É responsável pelo acionamento e desligamento das cargas monitoradas pelo sistema podendo ser, no máximo, até 64 cargas, endereçadas pelos ports PA0 até PA3. O pino PA4 é responsável pelo sinal de liga/desliga das cargas em questão. É também responsável pela seleção de uma das variáveis analógicas a serem convertidas pelo conversor analógico-digital (PA7).

O port B É responsável pela coleta dos dados provenientes do conversor analógico-digital de modo que tenhamos em PB0 o bit menos significativo e em PB7, o mais significativo, resultante da conversão analógica-digital.

O port C, com os quatro bits menos significativos programados como entrada, é responsável pela coleta do sinal proveniente dos detectores de incêndio (PC0) e o sinal EOC (End of Conversion) proveniente do conversor analógico-digital (PC3), estando os demais sinais de entrada (PC1 e PC2) desconectados.

O port C, com os quatro bits mais significativos programados como saída, é responsável pela emissão do sinal de liga/desliga da central de ar condicionado (PC5), do sinal de acionamento da temperatura da central de ar condicionado (fria/quente - PC4), do sinal responsável pelo acionamento do

alarme de incêndio (PC6), e do sinal responsável pela inicialização do conversor analógico-digital (PC7). [1] [4] [6] [9] [10]

CONVERSOR ANALÓGICO-DIGITAL (CAD 0808/0809)

O conversor analógico-digital é responsável pela coleta dos dados analógicos provenientes do módulo de monitoração de consumo de energia elétrica, e do módulo de controle de temperatura. Isso é possível pois o CAD é multiplexado, possibilitando termos até oito entradas analógicas diferentes.

A seleção de qual variável analógica deva ser convertida em digital é feita através de um sinal de seleção proveniente do port A do 8255 (PA7), o qual é introduzido no pino A0 do CAD. Como temos apenas dois sinais analógicos, as demais variáveis de seleção (A1 e A2) são aterradas de forma a não interferirem na sua seleção.

O sinal responsável pelo início da conversão (START) é proveniente do port C do 8255 (PC7).

Para que haja uma conferência da total conversão do sinal analógico, o sinal EOC (End of Conversion) é introduzido no port C do 8255 (PC3) onde verifica-se se a variável foi totalmente convertida. O tempo necessário entre um pulso de START e um pulso de EOC é da ordem de 100 microsegundos.

Há também um sinal de clock responsável pela correta utilização do CAD, sinal este que deve atuar na faixa de 10 a 1280 kHz. Este sinal de clock é proveniente da saída de um contador decádico, o qual será explicado a seguir.

Vale aqui salientar, que os sinais analógicos de entrada variam em uma faixa de 0 a 5 Vcc. [1] [5] [6] [9] [10]

O CONTADOR DECÁDICO 74192

O contador decádico 74192 é responsável pela divisão por dez da frequência original do microcomputador. Sua presença se faz necessária devido ao fato de o conversor atuar na faixa de frequência de 800 kHz.

Ele tem como sinal de entrada o próprio sinal de clock do microcomputador, e como saída o sinal de "Carry-out", sinal este pulsado em nível lógico zero. Por causa disto é necessário passá-lo por uma porta inversora (7404) e a depois introduzi-lo no sinal de clock do CAD. [1] [5] [6] [9] [10]

COMPARADORES 7485

Inicialmente existem dois comparadores responsáveis pelo endereçamento da placa, endereçada como 300H (hexadecimal). A comparação realizada resulta, se os bits presentes na via de endereços coincidirem com o endereço

da placa (300H), em um sinal ativo em nível lógico "1", o qual passa por uma porta inversora (7400) para que possa, então, ser introduzido como sinal de habilitação do dispositivo 8255 (sinal de Chip Select), já que este é ativo em nível lógico "0".

Os bits utilizados como referência de comparação são os bits A9 a A3 da via de endereços do microcomputador e o sinal Address Latch Enable (ALE) do microprocessador.

Um outro conjunto de comparadores é responsável pelo correto endereçamento das cargas controladas pelo sistema. Os bits de endereçamento são provenientes do port A do 8255 (PA0 a PA3). Quando um endereço enviado coincide com o endereço especificado para uma determinada carga, o comparador libera um sinal (A=B) que será utilizado para acionar ou desligar esta carga, de acordo também, com o sinal presente no port A (PA4 - liga/desliga). ^{[1] [5] [6] [9] [10]}

11.3 UNIDADES DE INTERFACE DE DADOS

As unidades de coletas de dados tem como função principal coletar as informações provenientes do processo e enviá-las à estação central, de forma paralela e igualmente receber as informações advindas da estação central, para executar comandos interferindo no processo controlado.

Elas são constituídas basicamente dos seguintes módulos:

- módulo de monitoração de consumo de energia elétrica
- módulo de acionamento de cargas
- módulo de controle de temperatura
- módulo de controle da central de ar condicionado
- módulo de detecção de incêndio

11.3.1 MÓDULO DE MONITORAÇÃO DE CONSUMO DE ENERGIA

Este módulo constitui-se basicamente por um circuito eletrônico discreto, conforme visto no Anexo 1, responsável pela monitoração instantânea do consumo de energia elétrica a fim de evitar que este consumo ultrapasse valores pré-estipulados.

Devido ao fato de termos todas as cargas que estão sendo monitoradas dispostas de maneira paralela, podemos monitorar a corrente que está sendo consumida instantaneamente. Esta corrente passa, então, por um resistor de potência provocando assim uma queda de tensão correspondente. Esta queda de tensão é então captada por um transformador com relação de espiras de 3:1. A seguir, como a tensão originada no secundário do transformador é uma tensão AC, deve-se transformá-la em uma queda de tensão DC. Esta transformação AC/DC se dá através do circuito composto

pelo Amplificador Operacional (LM 741), originando-se assim um sinal DC, o qual será então introduzido na placa localizada no microcomputador, onde então será tratado de acordo com as especificações do software, conforme pode ser visto no Anexo I. [2] [3] [4] [15]

NOTA: O resistor de potência é dimensionado de acordo com o total de cargas a ser monitorado.

11.3.2 MÓDULO DE ACIONAMENTO DE CARGAS

Para que se possa acionar ou desligar as cargas monitoradas, foi implementado um circuito eletrônico discreto, apresentado também no Anexo I.

É composto por um flip-flop tipo D, o qual é responsável pelo "armazenamento" do sinal de liga/desliga das cargas, proveniente do Port A (PA4) e um sinal de clock responsável pelo correto endereçamento das cargas. Este sinal de clock do flip-flop é proveniente do conjunto de componentes 7485 responsáveis por este endereçamento.

Uma vez acionado o flip-flop, através do sinal de clock, sua saída polariza a base de um transistor, fazendo que, com sua condução, provoque a circulação de corrente pelo coletor acarretando no atracamento do relé ali conectado, acionando assim a carga. O mesmo ocorre quando envia-se um

sinal de desligamento da carga em questão, fazendo com que o relé, uma vez atracado, seja "desligado", desligando assim a carga. [2][3][9][10]

11.3.3 MÓDULO DE CONTROLE DE TEMPERATURA

Este módulo é responsável por efetuar o controle de temperatura em ambientes onde isto se faça necessário, devido ao aquecimento ou resfriamento do ambiente, fora dos padrões preestabelecidos.

O circuito constitui-se, basicamente, de um transistor com a característica marcante de variar sua corrente de emissor correspondente a uma variação na temperatura do mesmo. Esta variação na corrente de emissor provoca uma variação na queda de tensão em cima do resistor de emissor. Como esta variação de tensão é muito baixa, deve-se amplificá-la, através do amplificador operacional LM 741 para que seja, então, introduzida no CAD, conforme pode ser visto no Anexo I. [2][3][4][9][15]

11.3.4 MÓDULO DE CONTROLE DA CENTRAL DE AR CONDICIONADO

Este módulo é responsável pelo acionamento de uma central de ar condicionado relacionada aos ambientes onde seja feito o controle de temperatura. Ele é constituído basicamente por um circuito de lógica de seleção, o qual é responsável pelo acionamento ou não da central de ar condicionado, selecionando também se esta central deve atuar com

temperatura de resfriamento ou temperatura de aquecimento, temperaturas estas fixas, ou seja, teremos apenas um resfriamento ou aquecimento dos ambientes em questão.

O sinal responsável pelo acionamento da central de ar condicionado (liga/desliga) é proveniente do port C do 8255 (PC5) e o sinal responsável pelo acionamento do grau da temperatura (quente ou frio) é também proveniente do port C do 8255 (PC4), conforme pode ser visto no Anexo I. [1]

[2] [3] [9] [10]

11.3.5 MÓDULO DE DETECÇÃO DE INCÊNDIO

Este módulo é responsável pela detecção e alarme de incêndio e é constituído basicamente de um circuito eletrônico discreto que realiza tal função. Sua característica principal reside no fato de determinar qualquer variação brusca na temperatura de um ambiente.

Constitui-se basicamente de um par de termistores (NTC) responsáveis por esta monitoração. Um deles é hermeticamente fechado enquanto o outro está exposto à temperatura ambiente. Caso ocorra um princípio de incêndio, haverá uma variação brusca na queda de tensão sobre o termistor que está exposto à temperatura ambiente. Esta variação na queda de tensão é extremamente rápida em relação à queda de tensão no termistor que está hermeticamente fechado. Quando isto ocorre, a diferença de potencial entre

estes dois termistores é considerável, o que provoca a emissão de um sinal de nível lógico "1", o qual é então introduzido no dispositivo 8255 através de seu port C. (PC0)

Desta forma podemos ter vários dispositivos conectados em paralelo, de modo que, a qualquer disparo de um dos dispositivos, o alarme de incêndio geral será disparado, alarme este que está conectado ao sinal proveniente do port C do 8255 (PC6). ^{[1][9][11]}

Nota: Cabe aqui salientar que o desenvolvimento de um detector de incêndio específico não faz parte integrante do desenvolvimento do projeto.

12. DESCRIÇÃO DO SOFTWARE

O software foi desenvolvido completamente em Linguagem "C", ocupando uma área de memória variável conforme a complexidade do sistema.

O emprego da Linguagem "C", a otimização de algoritmos e uma adequada codificação de instruções, tornam o sistema de supervisão e controle em tempo real, um sistema de grande velocidade de respostas e alta eficiência. [7]

O software é dividido em módulos, permitindo a inclusão de futuras rotinas para expansão do sistema.

As rotinas principais são:

- Interface Homem-Máquina responsável pela comunicação entre o operador e a central, via teclado.
- Rotina de comunicação responsável pelo gerenciamento da comunicação com as unidades de coleta de dados.
- Rotinas que controlam as chamadas das telas que compõe o sistema.

13. FILOSOFIA DE OPERAÇÃO

A operação do "Sistema de Automação Predial", é realizada fundamentalmente através da interface Homem-Máquina. Esta descrição está enfocando, principalmente, o monitor de vídeo, pois é o principal componente da IHM, e é através dele que o operador manterá o diálogo com o sistema. Realizando o acompanhamento da operação dos sistemas monitorados, tomando conhecimento do estado e, se necessário, interferindo no processo.

13.1 COMPOSIÇÃO DA IHM

Todas as informações serão apresentadas ao operador através de um conjunto de telas, conforme descrição a seguir:

13.1.1 TELA DE APRESENTAÇÃO

A tela de apresentação, conforme pode ser vista a seguir, questiona primeiramente se, após a carga do sistema, o operador deseja ou não prosseguir. Em caso negativo, o sistema é automaticamente suspenso. Em caso afirmativo, a mesma tela é apresentada, só que agora requisitando que o operador entre com uma senha previamente cadastrada. As senhas válidas para o acesso ao sistema devem ser cadastradas ainda no programa fonte,

de modo que apenas o comprador do mesmo tenha acesso.

ESCOLA POLITÉCNICA DA USP	ESCOLA POLITÉCNICA DA USP
PROJETO DE FORMATURA	PROJETO DE FORMATURA
SISTEMA DE AUTOMAÇÃO PREDIAL	SISTEMA DE AUTOMAÇÃO PREDIAL
DESEJA CONTINUAR (S/N):	ENTRE COM SUA SENHA :

TELA DE APRESENTAÇÃO #1

TELA DE APRESENTAÇÃO #2

FIGURA 2

Uma vez que a senha digitada seja aceita, o sistema passa a exibir uma tela índice, caso contrário é suspenso.

13.1.2 TELA ÍNDICE

Esta tela é responsável pela apresentação ao operador, de um índice composto por dois menus de escolha através deslocamento de uma barra luminosa.

MENU 1

Consta de uma relação de todas as operações permitidas pelo sistema: ATUALIZAR O SISTEMA, INICIALIZAR O SISTEMA e SAIR, conforme vista na figura abaixo, o qual aparece no canto superior esquerdo da tela.

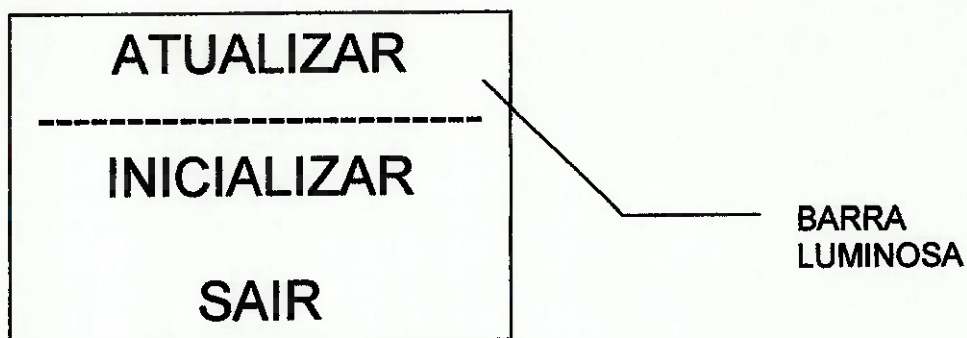


FIGURA 3

Para a realização de uma consulta ou análise, basta que o operador escolha, através da movimentação de uma barra luminosa, a função desejada. Esta movimentação se dá através das setas para cima e para baixo, e a seleção é feita através da tecla <ENTER>.

Uma vez pressionada a tecla <ENTER>, quando a barra luminosa estiver sobre a opção ATUALIZAR O SISTEMA, surgirá um novo menu (Menu 2), deslocado um pouco à direita em relação ao primeiro, com as seguintes

opções: TIMER, TEMPERATURA e DEMANDA, as quais serão explicadas oportunamente.

Uma vez pressionada a tecla <ENTER> sobre a opção INICIALIZAR O SISTEMA, o mesmo inicia todo o processo de varredura e coleta de dados, tratando-os de acordo com a necessidade instantânea dos mesmos. (O processo de varredura cíclica é explicado mais adiante).

Se o operador pressionar a tecla <ENTER> sobre a opção SAIR, o sistema indaga ao operador se realmente deseja abandonar o mesmo. Em caso afirmativo, o sistema é definitivamente suspenso, caso contrário retorna á apresentação do menu. O operador poderá, também, abandonar o sistema simplesmente pressionando a tecla <ESC> devendo proceder do mesmo modo ao anterior.

MENU 2

Este segundo menu é apresentado quando o operador opta por ATUALIZAR O SISTEMA e consta das seguintes opções: TIMER, TEMPERATURA e DEMANDA, conforme figura abaixo.

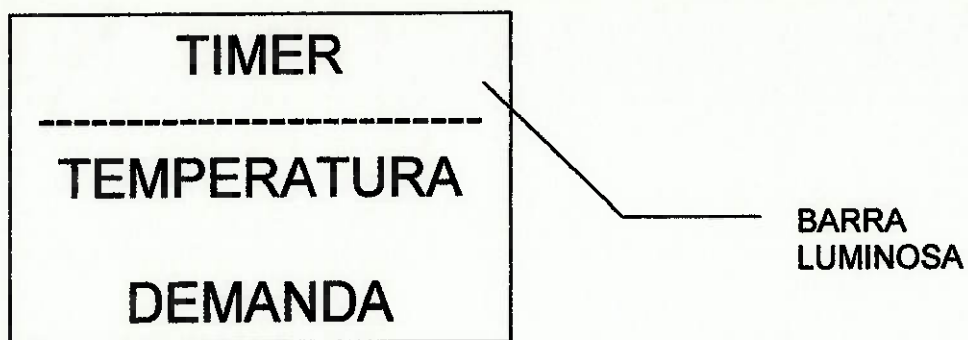


FIGURA 4

Do mesmo modo que o menu anterior, o operador poderá optar por um destes itens movendo a barra luminosa com as setas e efetuando a escolha através da tecla <ENTER>.

TIMER

Uma vez pressionada a tecla <ENTER> sobre a opção TIMER será exibido uma tela, conforme a figura abaixo, onde o operador poderá entrar com uma programação de timer para acionar ou desligar cargas em horários estabelecidos.

PROGRAMAÇÃO DE TIMER

CARGA (N)	DATA (DD/MM)	INÍCIO (HH:MM:SS)	TÉRMINO (HH:MM:SS)	PRIORIDADE (N)
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

FIGURA 5

De acordo com a disponibilidade da tabela, o operador poderá ter até oito linhas de programa, podendo-se ter múltiplas programações para uma mesma carga. Embora tenha-se apenas oito linhas nesta tabela, uma vez que ela esteja completa, o operador poderá "apagá-la" de modo a obter uma nova tabela.

Caso o operador venha a entrar com o número de uma carga que já tenha sido programada será emitido um sinal sonoro avisando-o que esta determinada carga já foi programada anteriormente, questionando-o se deseja ou não continuar com a programação para esta mesma carga.

A cada término de programação, o sistema questiona o operador se este deseja prosseguir com mais algum programa. Em caso negativo, o sistema retorna ao segundo menu, caso contrário, o operador poderá prosseguir com sua programação.

Como pôde ser visto na figura anterior, a tabela de programação é composta de diversas colunas, explicadas a seguir:

CARGA: Nesta coluna o operador deverá entrar com o número da carga a ser programada, número este que deve estar entre 0 e 63

DATA: Nesta coluna deve-se entrar com a data para a qual a carga especificada deverá ser ligada ou desligada, na forma (DD:MM), ou seja, dia e mês, ambos com dois dígitos apenas, separados por dois pontos (:).

INÍCIO: Nesta coluna deve-se entrar com o horário relativo ao qual o sistema acionará a carga correspondente, na forma (HH:MM:SS), ou seja, Hora, Minuto e Segundo, com dois dígitos cada separados por dois pontos (:).

TÉRMINO: Nesta coluna deve-se entrar com o horário relativo ao qual o sistema desligará a carga correspondente, na forma (HH:MM:SS), ou seja, Hora, Minuto e Segundo, com dois dígitos cada separados por dois pontos (:).

PRIORIDADE: Nesta coluna o operador deverá entrar com um número, de 1 a 5, que será responsável pela classificação das cargas quanto ao seu grau de prioridade para um futuro desligamento de cargas devido a um alto consumo de energia elétrica. Este desligamento é feito em ordem crescente de prioridade, caso o consumo de energia venha a ser superior ao estabelecido.

TEMPERATURA

Uma vez pressionada a tecla <ENTER> sobre a opção TEMPERATURA, será exibido uma tela conforme a figura abaixo, onde o operador poderá visualizar a atual temperatura à qual o sistema estará vinculado, podendo então atualizá-la. Deve-se aqui salientar que a temperatura a ser atualizada deve estar compreendida entre 0 e 100 graus centígrados, não sendo aceitos valores fora desta faixa de atuação. Quando isto ocorrer, será emitido um sinal sonoro e uma mensagem esclarecendo este fato.

ATUALIZAÇÃO DE TEMPERATURA



TEMPERATURA ATUAL : 20°C

NOVA TEMPERATURA :
_____ °C

FIGURA 5

DEMANDA

Se o operador pressionar a tecla <ENTER> quando a barra luminosa estiver sobre a opção DEMANDA, será exibida a tela apresentada abaixo onde o operador poderá visualizar, através de gráfico de barras e dados analíticos, a demanda acumulada do período, a demanda instantânea no ato da consulta e a tarifação acumulada no período. Aí também o operador poderá alterar o valor de patamar do consumo de energia. Este valor é o utilizado para que o sistema possa desligar cargas caso o consumo instantâneo seja superior.

ATUALIZAÇÃO DE DEMANDA

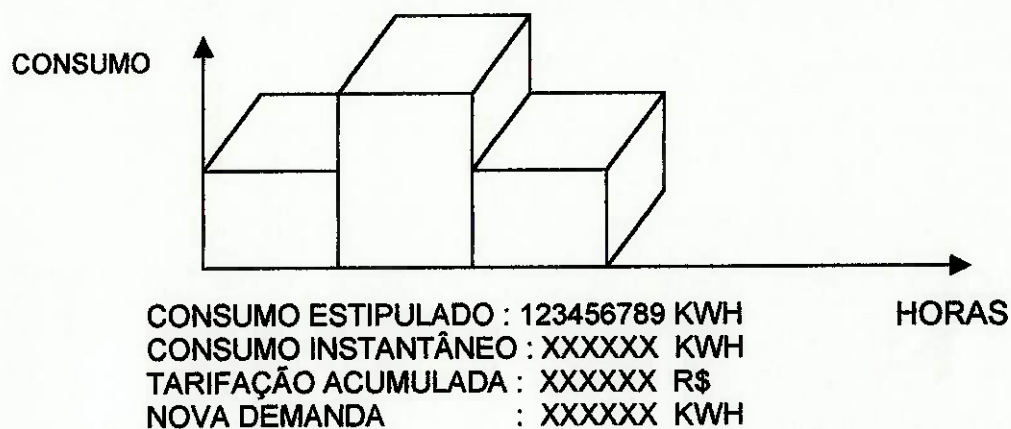


FIGURA 6

NOTA: Ao término de qualquer alteração em uma das variáveis, o sistema sempre questiona se o operador deseja continuar com mais alguma alteração ou não. Em caso negativo, o sistema retorna para o segundo menu. Ambos os menus podem ser descartados simplesmente pressionando-se a tecla <ESC>. No caso de estarmos manipulando o segundo menu, o sistema retorna ao primeiro menu. No caso de estarmos manipulando o primeiro menu, o sistema tenderá a ser cancelado, exigindo do operador que confirme tal atitude.

13.2 AQUISIÇÃO DE DADOS

A aquisição de dados é feita através do processo de *Pooling* (Varredura Seqüencial Cíclica) das unidades de coleta de dados, e o software que gerencia toda esta varredura através da placa de interface por nós desenvolvida, conforme pode ser verificado no fluxograma presente no Anexo II.

Inicialmente o software coleta os dados referentes ao consumo de energia instantâneo e o compara ao consumo estipulado para o período. Se o consumo coletado for superior ao estipulado, o sistema inicia o processo de desativação de cargas em ordem de prioridade, voltando sempre a comparar os novos valores obtidos, repetindo-se o processo até que o consumo instantâneo seja inferior ao estipulado.

Após a coleta dos dados referentes ao consumo de energia elétrica, o sistema passa a coletar os dados referentes à temperatura dos ambientes controlados, comparando-a à temperatura estipulada como padrão. Se o valor correspondente à temperatura coletada for superior à temperatura padrão, o sistema aciona a central de ar condicionado para funcionar em temperatura fria. Caso o valor coletado seja inferior ao valor estipulado, a central de ar condicionado é acionada em sua temperatura quente. Em ambos os casos a Central de Ar Condicionado fica acionada até que a temperatura ambiente atinja valores próximos ao valor padrão.

Logo após, o sistema passa à coleta dos sinais provenientes dos detectores de incêndio. Havendo a presença de sinal, o software ativa uma rotina de acionamento de alarme geral de incêndio emitindo, na tela do monitor de vídeo e, opcionalmente em unidade impressora, a ocorrência do fato a fim de que o operador possa tomar as devidas providências.

O sistema passa então a examinar se há alguma programação de timer a ser executada. Caso haja, o software coleta o horário do relógio do Sistema Operacional (DOS) e compara-o aos horários de início dos programas que eventualmente existirem. Quando estes horários coincidirem o sistema ativa a rotina responsável pelo acionamento das respectivas cargas. Somente no caso de haver um acionamento de uma ou mais cargas é que o sistema verifica o horário de desligamento das mesmas. Quando houver coincidência dos mesmos com o horário coletado do Sistema Operacional, o sistema passa a desligar as respectivas cargas.

Após o término da verificação do Timer, o sistema retorna a examinar o consumo de energia elétrica, completando-se assim o ciclo.

O processo de varredura pode ser interrompido a qualquer momento simplesmente pressionando-se uma tecla do teclado do microcomputador. Isto faz com que o sistema apresente a tela contendo o Menu 1, onde o operador poderá efetuar suas escolhas, conforme já fora explicado.

14. COMENTÁRIOS

É importante salientar que a viabilidade de um projeto como este está diretamente relacionada com o seu custo que, apesar da complexidade do sistema, está mais relacionado com o tempo disponibilizado para a sua implementação e concepção do que com conhecimentos específicos do assunto. Sistemas mais eficientes do que este estão hoje disponíveis no mercado mas, não a um custo tão acessível.

A utilização do microprocessador 8088 foi, mais do que tudo, uma decisão estratégica, uma vez que sua arquitetura é de fácil compreensão quando comparada a um microprocessador Pentium (INTEL) e já existe literatura abundante disponível no mercado. Da mesma forma foi feita a escolha da série 74 analógica de comparadores, contadores, amplificadores operacionais e portas lógicas.

É importante salientar que muito pode ainda ser desenvolvido como continuação deste projeto, como a utilização de sensores mais adequados a determinados parâmetros requeridos e a implementação de outros módulos não previstos em sua concepção básica, assim como, a sua atualização mercadológica, uma vez que será mais fácil partir de um projeto já pronto visando obter melhoras substanciais no mesmo.

15. SIMULAÇÃO

15.1 PROGRAMADOR DE TIMER

Para a simulação do programador de timer é utilizado um conjunto de três lâmpadas que atuariam como cargas quaisquer a serem ligadas e desligadas pelo microcomputador em horários pré-determinados pelo operador.

15.2 CONTROLADOR DE DEMANDA

Para a simulação do controlador de demanda é utilizado o circuito designado para esta função juntamente com o programador de timer, responsável pelo desligamento das cargas em ordem de prioridade crescente, assim que o consumo de energia ultrapassasse um valor pré-determinado para a simulação. Prioridade esta situada entre 1 e 5.

15.3 CONTROLADOR DE TEMPERATURA

Para a simulação do controlador de temperatura é utilizado o circuito apresentado no Anexo I deste manual, o qual é responsável pela variação de

tensão em sua saída correspondente à variação de temperatura em seu encapsulamento. Variação esta compreendida entre 0 e 5 volts DC.

15.4 CENTRAL DE AR CONDICIONADO

Juntamente com o circuito responsável pelo controle de temperatura, é utilizado o circuito apresentado no Anexo I responsável pela sinalização de acionamento da central de ar condicionado em sua temperatura quente ou fria, ou até mesmo desligá-la.

16. BIBLIOGRAFIA

- [1] MALVINO, Albert Paul; **Microcomputadores e microprocessadores**; São Paulo, Mc Graw Hill, 1986.
- [2] MALVINO, Albert Paul; **Eletrônica**, Volume 1; São Paulo, Mc Graw Hill, 1987.
- [3] MALVINO, Albert Paul; **Eletrônica**, Volume 2; São Paulo, Mc Graw Hill, 1987.
- [4] ANALOG DEVICES; **Amplifier Reference Manual**; Abnalog Devices; Catálogo; 1994.
- [5] LYNX; **CAD12/36 Conversor A/D e D/A para Microcomputadores**, Manual do Usuário e de Referência; LYNX Tecnologia Eletrônica Ltda; 1993.
- [6] ROSCH, Winn L.; **Desvendando o Hardware do PC**; São Paulo, Fontes, 1995.
- [7] HERBERT SCHILDT; **Turbo C ++**; São Paulo; Campus; 1996.
- [8] CONDOMINIUM CENTRO EMPRESARIAL; **Monitoração e Controle**; Apostila; 1997.
- [9] TEXAS INSTRUMENTS; **The TTL Data Book**; Manual; 1993.
- [10] INTEL; **Microsystem Components Handbook**; Manual; 1994.
- [11] SIEMENS; **Catálogo de Detectores de Incêndio**; Catálogo; 1997.
- [12] ENSEC, Divisão de Sistemas de Segurança; **Ênfase à Segurança e Integração com o Controle de Utilidades**; Revista Eletricidade Moderna, N°

220, pág. 56, Julho, 1992.

[13] SANTOS, FLÁVIO ÁLVARES DOS; **Automação Predial e Segurança**; Revista INSTEC, N° 74, pág. 20, Fevereiro; 1994.

[14] MARTE, CLÁUDIO G.; COSTA, HERBERT R. N.; FOGAGNOLI, RENATO; **A Automação e o Comportamento Energético das Edificações**, Revista Eletricidade Moderna, N° 237, pág. 58, Dezembro, 1993.

[15] UNIVERSIDADE DE SÃO PAULO; **Laboratório de Eletrônica**, Apostila; Escola Politécnica - Departamento de Engenharia Mecânica, São Paulo, 1996.

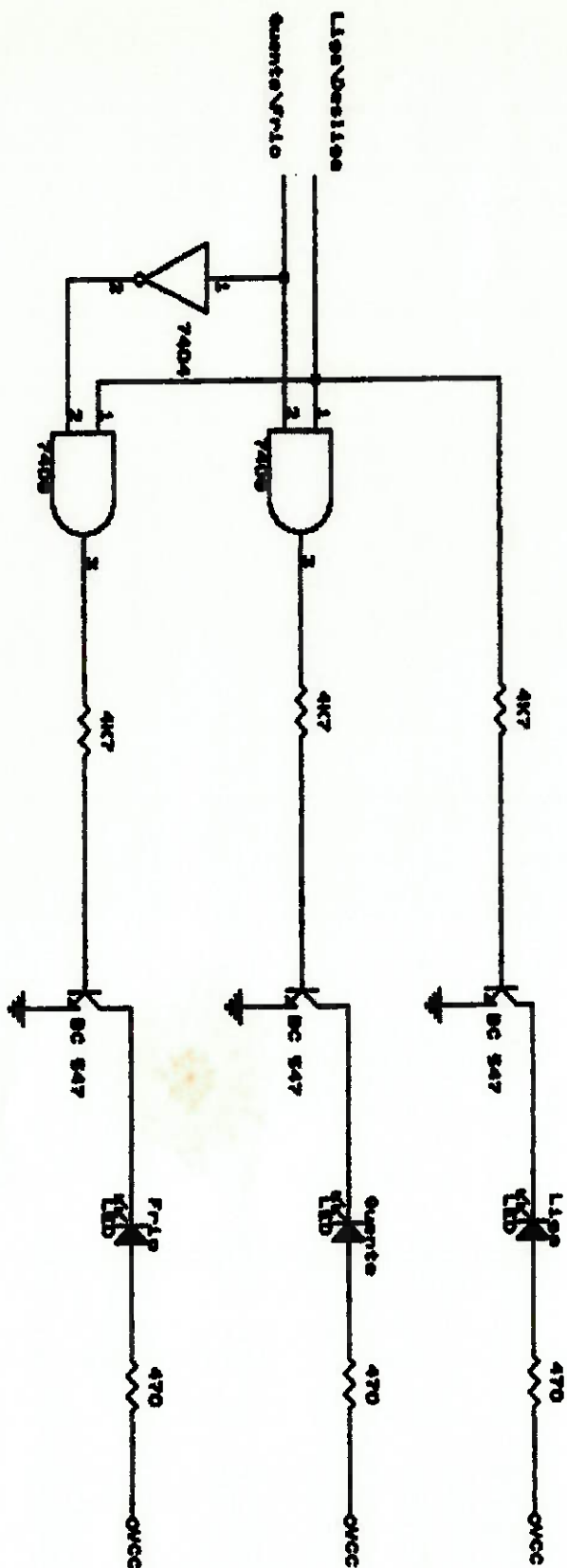
[16] JUNIOR, MOACYR MARTUCCI; **Especificação da Arquitetura de um Sistema Distribuído para Controle de Processos**, Tese de Doutorado; U.S.P. - Escola Politécnica - Departamento de Engenharia Elétrica, São Paulo, 1982.

[17] UNIVERSIDADE DE SÃO PAULO; **Diretrizes para Apresentação de Dissertações e Teses**; Escola Politécnica - Serviço de Bibliotecas, São Paulo, 1991.

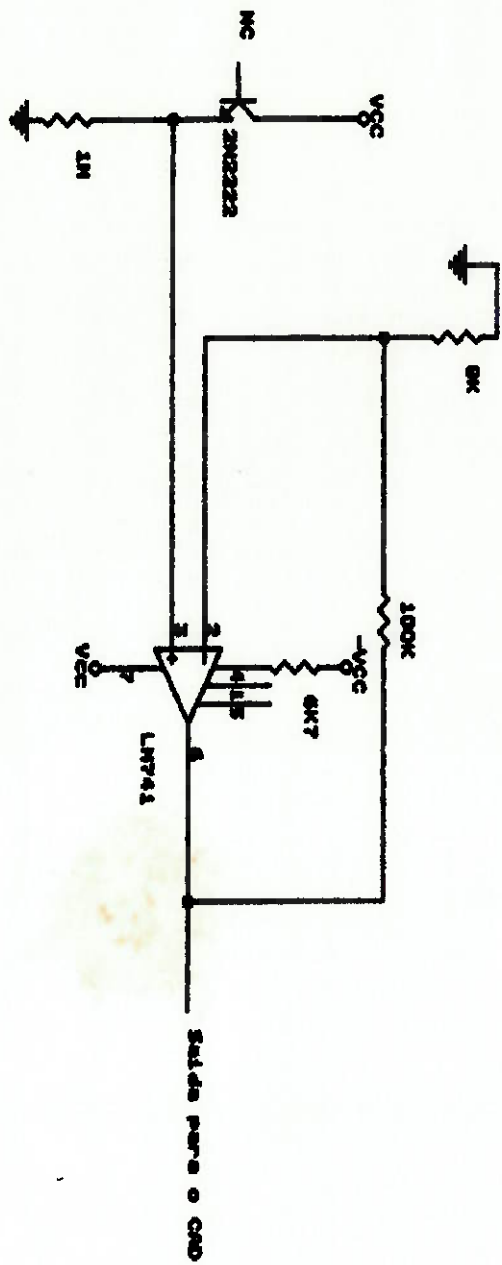
ANEXO I – Diagramas Lógicos



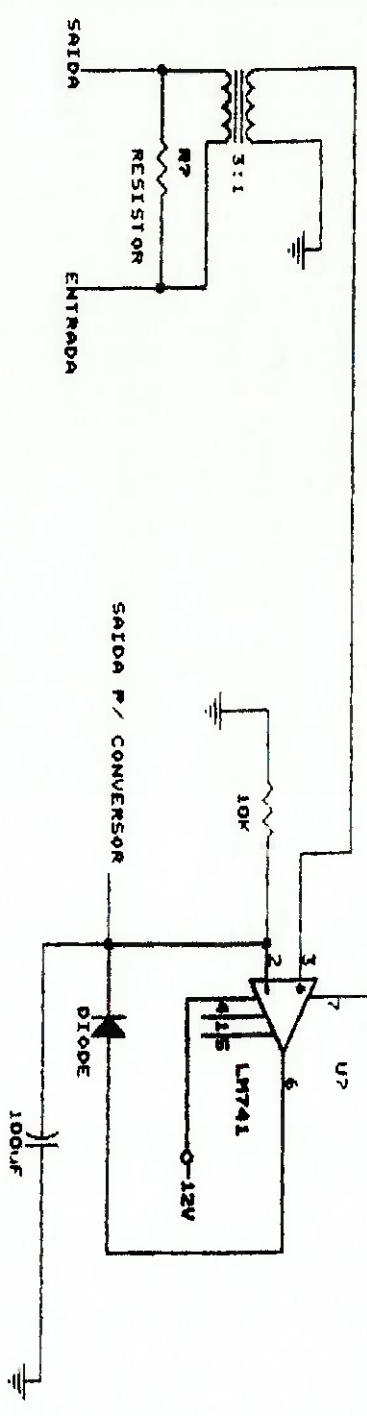
Circuito Controlador do Ar - Condicionado



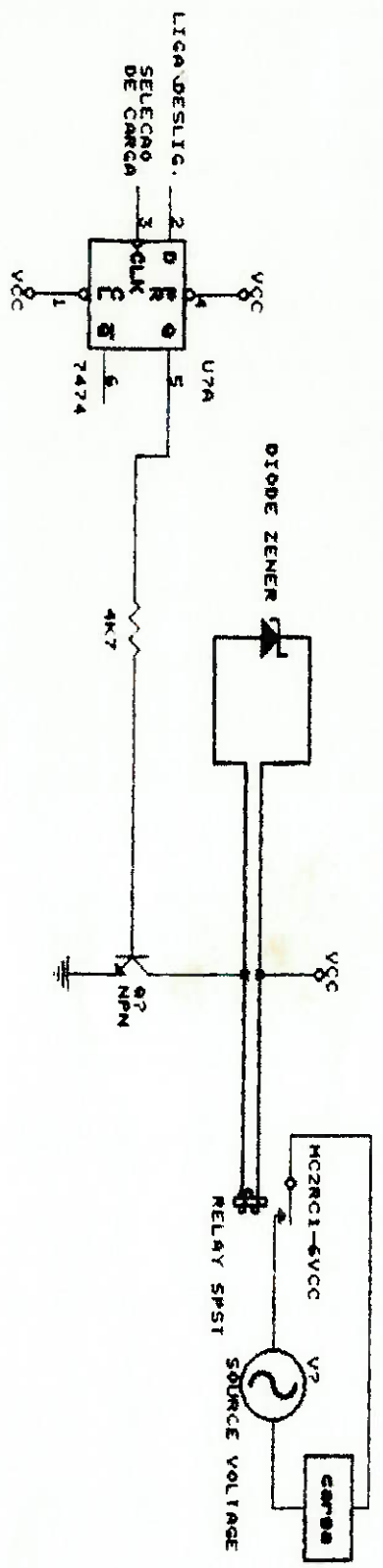
Conversor de Temperatura/Tensao



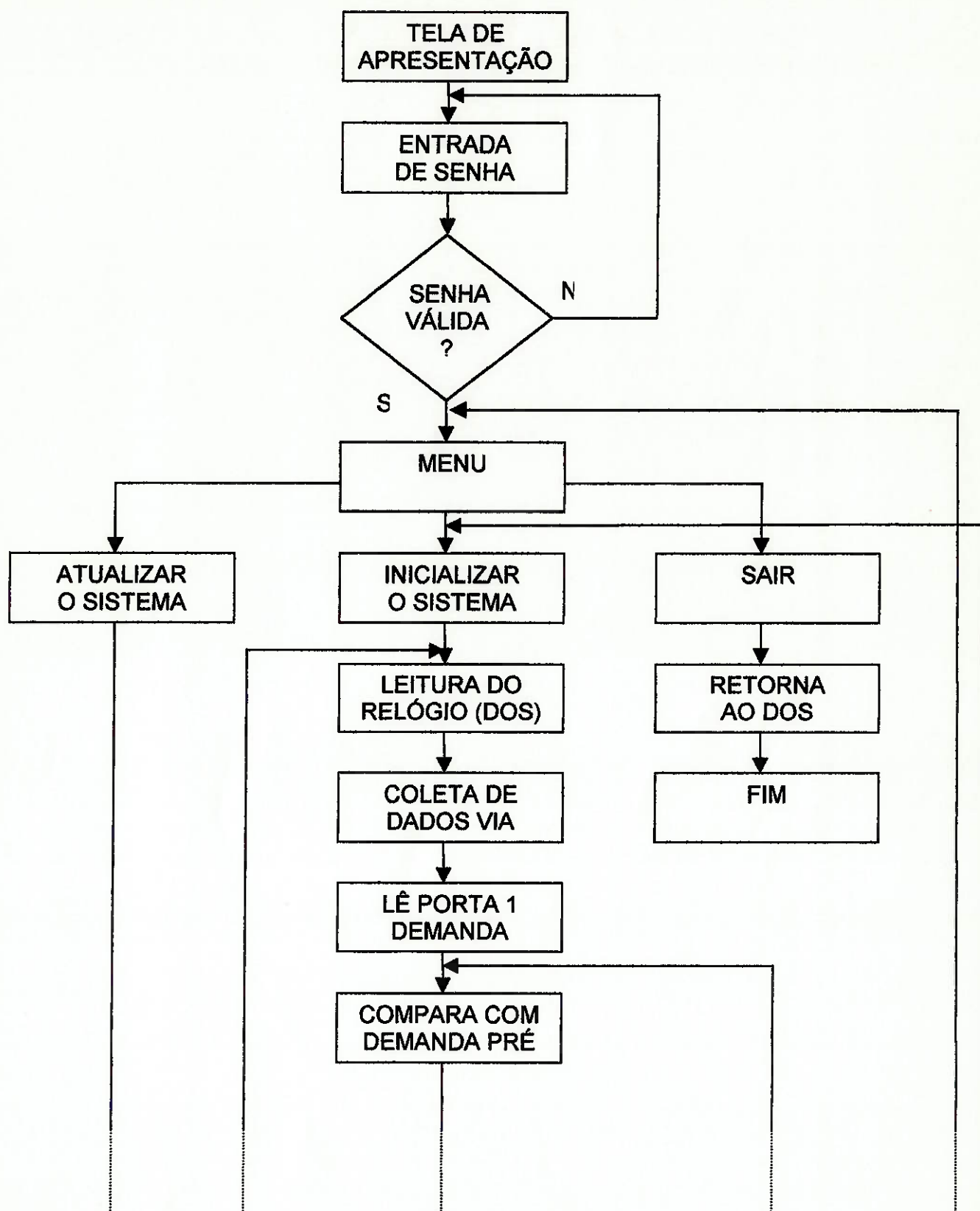
Circuito Para Controlar a Demanda

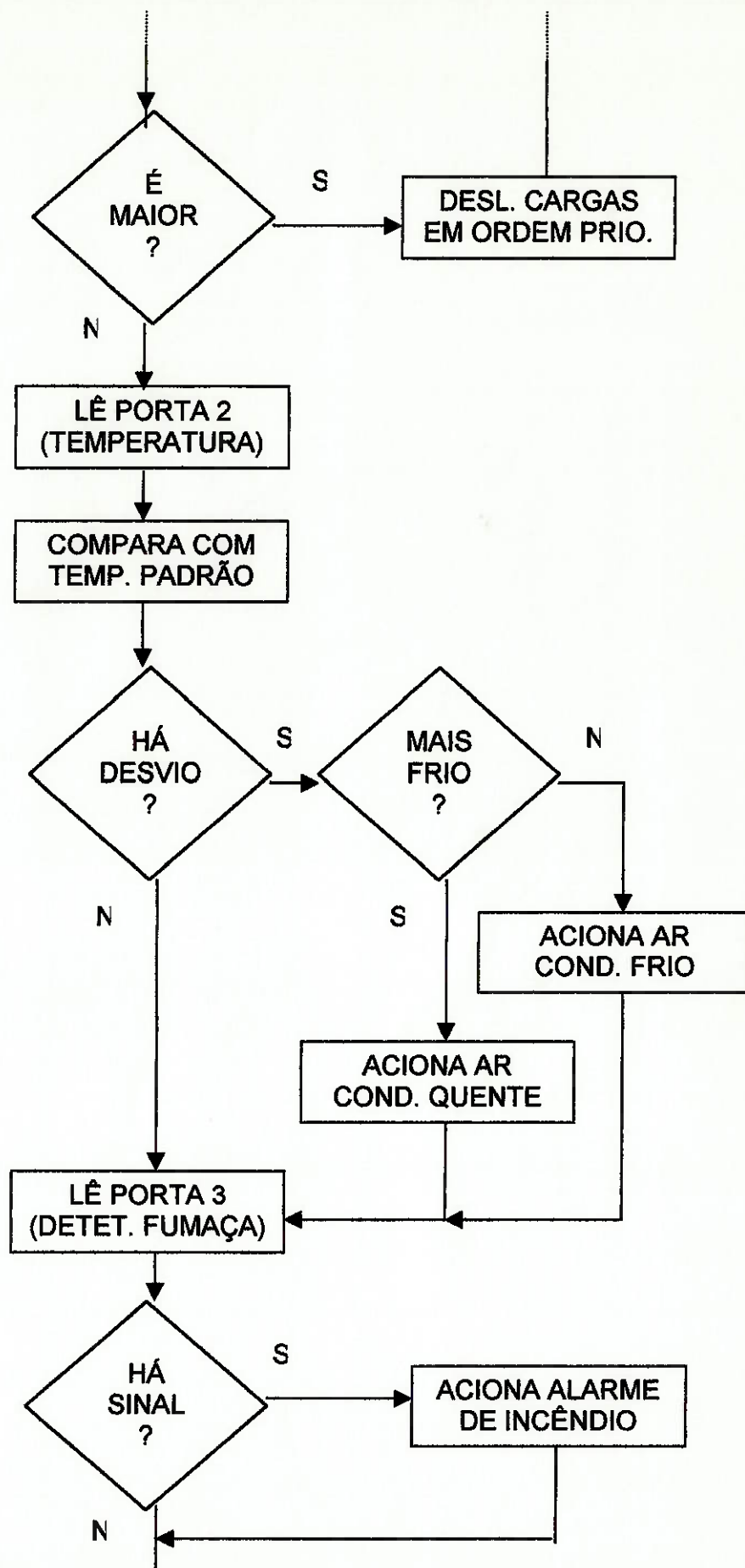


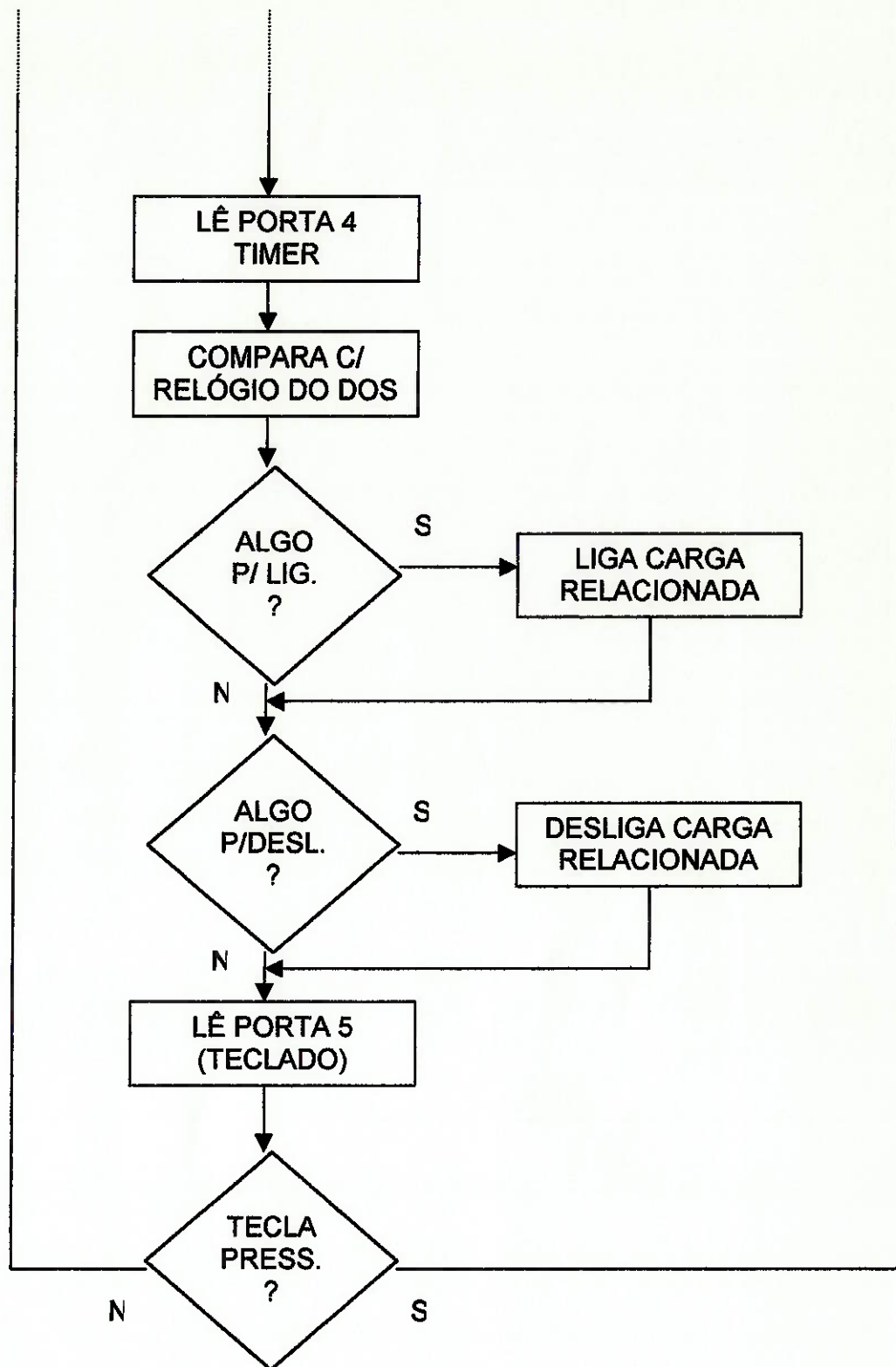
Circuito Controlador (Lig. e Desl.) de Cargas Via Timer

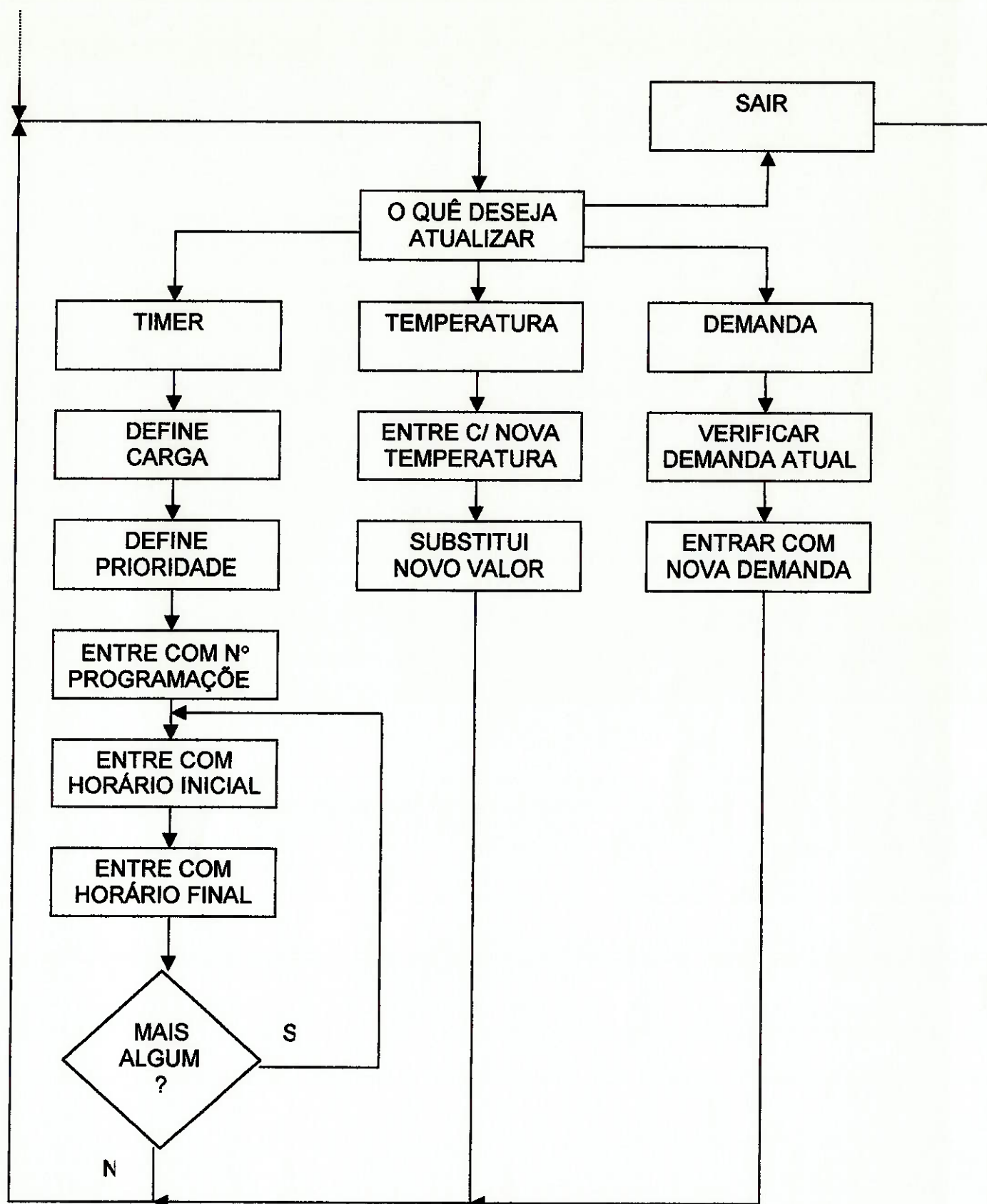


ANEXO II - Fluxograma









ANEXO III - Programa

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include <ctype.h>
#include <string.h>
#include <dos.h>
#include <alloc.h>
```

```
#define MAX 64
```

```
#define ENDER_PORT_A      0x300
#define ENDER_PORT_B      0x301
#define ENDER_PORT_C      0x302
#define ENDER_CONTROLE    0x303
#define PALAVRA_CONTROLE  0x83
```

```
void detection (void);
void start_1 (void);
void start_2 (void);
void menu (void);
```

```
void atualizar (void);
void inicializar (void);
void sair (void);
void frame ();
```

```
void timer (void);
void repeticao (void);
void moldura (void);
```

```
void temperatura (void);
void termometro ();
void demanda (void);
void grafico (void);
```

```
void consumo (void);
```

```
typedef struct {
    char *desc;
```

```

    } MenuDesc;

typedef struct {
    char *resA;
} MenuItem;

typedef struct {
    char *resB;
} MenuItemB;

typedef struct {
    char *resC;
} MenuItemC;

struct timer {
unsigned int dia;
unsigned int mes;
unsigned int h_inicio;
unsigned int m_inicio;
unsigned int s_inicio;
unsigned int h_fim;
unsigned int m_fim;
unsigned int s_fim;
unsigned int prior;
} carga[MAX][10];


int temp_atual = 20, temp_nova = 20;
unsigned int i, n, p, hi, mi, si, hf, mf;
    unsigned int sf, d, m, count(MAX), aux = 0;
    unsigned long int aux2 = 0;
    void *arrow;
    unsigned int size;
    char t;
    double demanda_estipulada = 1000;


void maim (void)
{
for (i=0 ; i<MAX ; i++)    /* inicializa dados do timer com 0 */
    {
        carga[i][0].h_inicio = 0;
    }

```



```
        count[i] = 0;
    }
```

```
detection ();
start_1();
start_2();
closegraph();
exit(0);
```

```
void detection (void)
{
    int gdriver = DETECT, gmode, errorcode;
    initgraph(&gdriver, &gmode, "");
    errorcode = graphresult();
    if (errorcode != grOk)
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit( 1);
    }
}
```

```
void start_1 (void)
{
    rectangle (0,0,getmaxx(),getmaxy());
    settxtjustify(CENTER_TEXT,CENTER_TEXT);
    settxtstyle(TRIPLEX_FONT,HORIZ_DIR,1);
    outtextxy(320,10," ESCOLA POLITECNICA DA USP");
    settxtstyle(TRIPLEX_FONT,HORIZ_DIR,1);
    outtextxy(315,40, " PROJETO DE FORMATURA ");
    settxtstyle(TRIPLEX_FONT,HORIZ_DIR,4);
    outtextxy (320,80, "SISTEMA DE AUTOMACAO");
    outtextxy(320,110, "PREDIAL");
}
```

```

void start_2 (void)
{
    char c, ptr, w;
    int maxy = getmaxy();
    char *senha1 = "27117", *senha2 = "27317", *senha3 = "25205"; char
    *password;

```

LOOP3:

```

    settextstyle(SMALL_FONT, HORIZ_DIR, 4);
    outtextxy(320, (maxy-13), "DESEJA CONTINUAR ? (S/N) :");
    gotoxy(51, 24);
    cscanf ( "%c", &c);
    c = tolower(c);
    switch(c)
    {
        case 's': cleardevice();
            start_1();
            gotoxy(30,40);
            password = getpass ( "ENTRE COM SUA SENHA :");
            ptr = strcmp (password, senha1);
            if ( !ptr )
            {
                cleardevice();
                menu();
            }
            else
            {
                ptr = strcmp (password, senha2);
                if ( !ptr )
                {
                    cleardevice();
                    menu();
                }
                else
                {
                    ptr = strcmp (password, senha3);
                    if ( !ptr )
                    {
                        cleardevice();
                        menu();
                    }
                    else
                    {
                        cleardevice();
                        gotoxy(30, 24);
                        printf (".... ATE LOGO ....");
                        getch();
                        break;
                    }
                }
            }

```

```

    }
    }
    default :    cleardeviceo();
                gotoxy(25,10);
                printf("TEM CERTEZA QUE QUER SAIR ? (S/N) :
                \a");
                w = tolower (getch());
                if (w == 'n') goto loop3;
                break;
    }

```

void menu (void)

```

{
int linha = 2,  coluna = 2 , i , length;
char tecla, tecla2, s;

MenuDesc Menu [] = {
{"ATUALIZAR O SISTEMA  "},
{"INICIALIZAR O SISTEMA  "},
{"SAIR                    "}};

closegraph();

Loop1 :
Do {
    _setcursortype ( _NOCURSOR);
    textattr (15);
    length = strlen (Menu[0].desc);
    clrscr();
    frame( length, coluna);
    textcolor (3);
    for (i=0 ; i<3 ; i++)
    {
        gotoxy(coluna, (i+2));
        cprintf ("%s", Menu[i] .desc);
    }
    gotoxy(coluna, linha);
    textattr (3+(5*16));
    cprintf( "%s" Menu[linha-2].desc);
    tecla = getch();
    if (tecla == 13)

```

```

        {
            if (linha == 2) atualizar();
            if (linha == 3) inicializar();
            if (linha == 4) sair();
        }
    if (tecla == 0)
    {
        tecla = getch();
        textattr (3);
        gotoxy (coluna, linha);
        cprintf ("%s",Menu[linha-2].desc);

        if (tecla == 80)
        {
            linha++;
            if (linha > 4) linha = 2;
        }
        if (tecla == 72)
        {
            linha--;
            if (linha == 1) linha = 4;
        }
    }
}
while (tecla != 27);
clrscr();
detection();
rectangle (0,0,getmaxx( ) ,getmaxy());
rectangle (160,70,460,80);
gotoxy(22,10);
printf ("TEM CERTEZA QUE QUER SAIR ? (S/N) : \a");
s = tolower(getch());
if (s == 'n')
{
    closegraph();
    goto loop1;
}
free (arrow);          /* apaga imagem do timer armazenada na memoria
*/
closegraph();
exit (0);
}

```

```

void atualizar (void)
{
MenuItem Menu2 [] =
    {
        {"TIMER"},
        {"TEMPERATURA"},
        {"DEMANDA"}
    };
int i, coluna = 30 , linha = 2 , length;
char tecla3;

do
    {
        length = strlen (Menu2[0].resA);
        frame( length, coluna);
        textattr (3);
        for (i=0 ; i<3 ; i++)
            {
                gotoxy(coluna, (i+2));
                cprintf ("%s", Menu2[i] .resA);
            }
        gotoxy(coluna, linha);
        textattr (3+(5*16));
        cprintf( "%s" Menu2[linha-2].resA);
        tecla3 = getch();
        if (tecla3 == 13)
            {
                if (linha == 2) timer();
                if (linha == 3) temperatura();
                if (linha == 4) demanda();
            }
        if (tecla3 == 0)
            {
                tecla3 = getch();
                textattr (3);
                gotoxy (coluna, linha);
                cprintf ("%s",Menu2[linha-2].resA);

                if (tecla3 == 80)
                    {
                        linha++;
                        if (linha > 4) linha = 2;
                    }
            }
    }

```

```

        if (tecla3 == 72)
        {
            linha--;
            if (linha == 1) linha = 4;
        }
    }

    while (tecla3 != 27);
    textattr(0);

    void inicializar (void)
    {
        struct time t;
        struct date d;

        unsigned int liga_carga , desliga_carga , maior = 1;
        unsigned int liga_ar_quente = 0x30 , valor_digital;
        unsigned int liga_ar_frio = 0x20 , desliga_ar = 0x00;
        unsigned int cad_0 = 0x00 , start = 0x80 , anterior1 = 0x00 , anterior2 =
            0x00;
        unsigned int temperatura_instant , temperatura_estipulada , incendio;
        unsigned int cad_1;
        double demanda_estipulada;
        double demanda_seguranca;
        unsigned int i, j;
        unsigned int demanda_instant;

        outportb(ENDER_CONTROLE, PALAVRA_CONTROLE);

        do {

            textattr(0);
            gotoxy(5,10);
            cprintf ("                                ");
            gotoxy(5,11);
            cprintf ("                                ");
            gotoxy(5,12);
            cprintf ("                                ");
            gotoxy(5,13);
            cprintf ("                                ");

```

```

gotoxy(5,14);
cprintf ("");
gotoxy(5,15);
cprintf ("");
gotoxy(5,16);
cprintf ("");

```

```

gettime(&t);
_setcursortype ( NOCURSOR);
gotoxy(73,1);
printf( "%2d:%02d:%02d", t.ti_hour, t.ti_min, t.ti_sec);

```

```

/*****
*****
          INICIA VARREDURA DO CONVERSOR – TEMPERATURA
*****
*****/

```

```

cad_0 = anterior1 : 0x0f;

```

```

outportb(ENDER_PORT_A, cad_0);
outportb(ENDER_PORT_C, start);
start = start & 0x7f;
outportb ( ENDER_PORT_C, start);

```

```

valor_digital = inportb(ENDER_PORT_B);

```

```

if (valor_digital < 0x68)      /* 0x68 = 2.0 volts */
{
    textattr(0);
    gotoxy(25,12);
    cprintf ("");
    textattr(BLACK + ((YELLOW*16) + 128));
    gotoxy (25,12);
    cprintf("LIGANDO AR CONDICIONADO QUENTE !!!");
    outportb(ENDER_PORT_C, liga_ar_quente);
    anterior2 = liga_ar_quente;
    start = 0x80 : anterior2;
    outportb(ENDER_PORT_C, start);
    start = start & 0x7f;
    outportb(ENDER_PORT_C, start);
}

```

```

if (valor_digital > 0x68)
{
    textattr(0);
    gotoxy(25,12);
    cprintf("                                ");
    textattr(BLACK + ((YELLOW*16) + 128));
    gotoxy (25,12);
    cprintf("LIGANDO AR CONDICIONADO FRIO !!!");
    outportb(ENDER_PORT_C, liga_ar_frio);
    anterior2 = liga_ar_frio;
    start = 0x80 : anterior2;
    outportb(ENDER_PORT_C, start);
    start = start & 0x7f;
    outportb(ENDER_PORT_C, start);
}

```

```

if (valor_digital == 0x68)
{
    textattr(0);
    gotoxy(25,12);
    cprintf("                                ");
    textattr(BLACK + ((YELLOW*16) + 128));
    gotoxy (25,12);
    cprintf("DESLIGANDO AR CONDICIONADO !!!");
    outportb(ENDER_PORT_C, desliga_ar);
    anterior2 = desliga_ar;
    start = 0x80 : anterior2;
    outportb(ENDER_PORT_C, start);
    start = start & 0x7f;
    outportb(ENDER_PORT_C, start);
}

```

/******

INICIA VARREDURA DO CONVERSOR – DEMANDA

*****/

```

cad_1 = anterior1 & 0xbf;          /* cad_1 = anterior1 : 0x40 */
outportb(ENDER_PORT_A, cad_1);
start = 0x80 : anterior2;
outportb(ENDER_PORT_C, start);

```



```

start = start & 0x7f;
outportb(ENDER_PORT_C, start);
valor_digital = inportb(ENDER_PORT_B);
if (valor_digital >= 0xb0)      /* 0xb0 = 3.5 volts */
{
    do {
        for (i = 0 ; i < 3; i++)
        {
            for (j=0; j<10; j++)
            {
                if (carga[i][count[j]].prior == maior)
                {
                    textattr(BLACK + ((YELLOW*16) + 128));
                    gotoxy(20,13);
                    cprintf("DESLIGANDO A CARGA %d ,
PRIORIDADE %d !!", i, carga[i][count[j]].prior);
                    anterior1= anterior1 & 0xef;
                    anterior1= anterior1 : 0x0f;

                    desliga_carga = anterior1 & 0xef;
                    outportb(ENDER_PORT_A, desliga_carga);
                    i = i : 0x00;
                    desliga_carga = desliga_carga & i;
                    outportb(ENDER_PORT_A, desliga_carga);
                    anterior1 = desliga_carga;
                }
            }
        }
        maior++;
        start = 0x80 : anterior2;
        outportb(ENDER_PORT_C, start);
        start = start & 0x7f;
        outportb(ENDER_PORT_C, start);

        valor_digital = inportb(ENDER_PORT_B);
    } while (( maior < 6 ) && (valor_digital >= 0xb0));
}

```

```

/*****
*****

```

INICIA VARREDURA DOS DETECTORES DE INCENDIO

```

*****
*****/

```

```

incendio = inportb(ENDER_PORT_C);
incendio = incendio & 0x01;
if (incendio == 0x01)
{
    _setcursortype(_NOCURSOR);
    textattr(BLACK + ((YELLOW*16) + 128));
    gotoxy(22,14);
    cprintf("                                ");
    gotoxy(22,15);
    cprintf("      FOI ACIONADO O ALARME DE INCENDIO !!!");
    gotoxy(22,16);
    cprintf("                                ");
    sound(300);
    delay(300);
    sound (500);
    delay(300);
    sound (700);
    delay (300);
    nosound ();
    textattr(LIGHTGRAY);
}

```

```

/*****
*****
                INICIA VARREDURA DO TIMER
*****
*****/

```

```

getdate(&d);
gettime(&t);
for (i = 0 ; i < MAX; i++)
{
    for (j=0; j<10; j++)
    {
        if (carga[i][count[j]].dia == d.da_day &&
            carga[i][count[j]].mes == d.da_mon )
        {
            if (carga[i][count[j]].h_inicio == t.ti_hour &&

```

```

        carga[i][count[j]].m_inicio == t.ti_min    )
    {
        _setcursortype(_NOCURSOR);
        textattr(BLACK + ((YELLOW*16) + 128));
        gotoxy(26,10);
        cprintf("LIGANDO A CARGA %d ", i );
        liga_carga = 0x1f : anterior1;
        outportb(ENDER_PORT_A, liga_carga);
        i =i : 0x10;
        liga_carga = liga_carga & i ;
        outportb(ENDER_PORT_A, liga_carga);
        anterior1 = liga_carga;
    }
}
}

for (i = 0 ; i < MAX; i++)
{
    for (j=0; j<MAX; j++)
    {
        if (carga[i][count[j]].dia == d.da_day &&
            carga[i][count[j]].mes == d.da_mon    )
        {
            if (carga[i][count[j]].h_fim == t.ti_hour &&
                carga[i][count[j]].m_fim == t.ti_min    )
            {
                _setcursortype(_NOCURSOR);
                textattr(BLACK + ((YELLOW*16) + 128));
                gotoxy(26,11);
                cprintf("DESLIGANDO A CARGA %d ", i );
                desliga_carga = 0x0f : anterior1;
                outportb(ENDER_PORT_A, desliga_carga);
                i =i : 0x00;
                desliga_carga = desliga_carga & i ;
                outportb(ENDER_PORT_A, desliga_carga);
                anterior1 = desliga_carga;
            }
        }
    }
}

} while (!kbhit());

```

```
return;  
}
```

```
void sair (void)
```

```
{  
    char s;  
    clrscr();  
    detection();  
    rectangle (0,0,getmaxx( ) ,getmaxy());  
    rectangle (160,70,460,80);  
    gotoxy(22,10);  
    printf ("TEM CERTEZA QUE QUER SAIR ? (S/N) : \a");  
    s = tolower(getch());  
    if (s == 'n')  
    {  
        closegraph();  
        menu();  
    }  
    free (arrow);          /* apaga imagem do timer armazenada na memoria */  
    closegraph();  
    exit (0);  
}
```

```
void frame (int length , int coluna)
```

```
{  
    int i;  
    gotoxy((coluna-1),1);  
    printf (" ");  
    gotoxy((coluna-1),2);  
    for (i=0 ; i < 3 ; i++)  
    {  
        printf ("||");  
        gotoxy( (coluna-1), (i + 3));  
    }  
    printf ( " ");  
    for (i=0 ; i< length ; i++) printf("=");  
    printf (" ");  
    for (i=1 ; i<5 ; i++)  
    {
```

```

    gotoxy ( (length + coluna), i );
    printf (" ");
}
gotoxy(coluna, 1);
for (i=0 ; i< length ; i++) printf("=");
printf (" ");

textattr(15);
gotoxy (30, 20);
cprintf ("ESCOLHA SUA OPCAO :");
gotoxy (10, 25);
cprintf ("SETAS : move");
gotoxy (31, 25);
cprintf ("ENTER : seleciona");
gotoxy (56, 25);
cprintf ("ESC : abandona");
}

void timer (void)
{
char x, y, z ;
char i;
detection();
loop5: cleardevice();

aux2++;
if (aux2 >= 2)
{
cleardevice();
putimage (0, 0, arrow, COPY_PUT);
}

moldura();

loop4 :
size = imagesize (0, 0, getmaxx(), getmaxy());
arrow = malloc(size);
getimage (0, 0, getmaxx(), getmaxy(), arrow);

do {

```

```

gotoxy(20, 24);
printf ("SE DESEJAR CONTINUAR, PRESSIONE < ENTER >");
gotoxy (20, 25);
printf ("SE DESEJAR SAIR, PRESSIONE < ESC >");

x= getch();
if (x == 13)
{
    if (aux == 16)
    {
        putimage (0, 0, arrow, COPY_PUT);
        gotoxy (20, 24);
        printf("NAO HA MAIS ESPACO PARA
PROGRAMACAO\");
        gotoxy (20, 25);
        printf ( "PRESSIONE QUALQUER TECLA");
        getch();
        cleardevice();
        putimage (0, 0, arrow, COPY_PUT);
        gotoxy (20, 24);
        printf("DESEJA APAGAR OS DADOS DA TABELA ?
(S/N) :");

        z = tolower(getch());
        if (z == 's')
        {
            cleardevice();
            moldura();
            for (i=0 ; i<MAX ; i++) /* inicializa dados do
timer com 0 */

            {
                carga[i][0].h_inicio = 0;
                count[i] = 0;
            }
            aux = 0;
            aux2 = 0;
            goto loop5;
        }
        closegraph();
        return;
    }
    aux+=2;
    cleardevice();
    Putimage(0, 0, arrow,COPY_PUT);

```

```

gotoxy(10, (6+aux));
scanf ("%d", &n);
if (carga[n][count[n]].h_inicio != 0)
{
    while (carga[n][count[n]].h_inicio != 0)
    {
        repeticao (count);
        goto loop4;
    }
}
gotoxy(24, (6+aux) );
scanf ("%d/%d", &d, &m);
carga[n][count[n]].dia = d;
carga[n][count[n]].mes = m;
gotoxy (37, (6+aux));
scanf ("%d: %d: %d", &hi, &mi, &si);
carga[n][count[n]].h_inicio = hi;
carga[n][count[n]].m_inicio = mi;
carga[n][count[n]].s_inicio = si;
gotoxy (53, (6+aux));
scanf ("%d: %d: %d", &hf, &mf, &sf);
carga[n][count[n]].h_fim = hf;
carga[n][count[n]].m_fim = mf;
carga[n][count[n]].s_fim = sf;
gotoxy (70, (6+aux));
scanf ("%d", &p);
carga[n][count[n]].prior = p;
size = imagesize(0, 0, getmaxx(), getmaxy());
arrow = malloc(size);
getimage (0, 0, getmaxx(), getmaxy(), arrow);
gotoxy(25, 24);
printf ("DESEJA CONTINUAR ? (S/N) : ");
y = getch();
}
if (x == 27)
{
    closegraph();
    return;
} while (y == 's');
closegraph();
return;
}

```


void moldura (void)

```
{
rectangle (0, 0, getmaxx(), getmaxy());
settextjustify (CENTER_TEXT, CENTER_TEXT);
settextstyle (TRIPLEX_FONT, HORIZ_DIR, 1);
outtextxy((getmaxx() / 2), 10, "PROGRAMACAO DE TIMER");
line(190, 20, 440, 20);
rectangle (20, 30, (getmaxx() - 20), (getmaxy() - 20));
line (20, 50, (getmaxx () - 20), 50);
line (140, 30, 140, (getmaxy () - 20);
line (260, 30, 260, (getmaxy () - 20);
line (380, 30, 380, (getmaxy () - 20);
line (500, 30, 500, (getmaxy () - 20);
line (20, 52, (getmaxx () - 20), 52);
line (20, 68, (getmaxx () - 20), 68);
line (20, 84, (getmaxx () - 20), 84);
line (20, 100, (getmaxx () - 20), 100);
line (20, 116, (getmaxx () - 20), 116);
line (20, 132, (getmaxx () - 20), 132);
line (20, 148, (getmaxx () - 20), 148);
line (20, 164, (getmaxx () - 20), 164);
gotoxy(9, 5);
printf ("CARGA");
gotoxy(9, 6);
printf ("NUM.");
gotoxy(23, 5);
printf ("DATA");
gotoxy(22, 6);
printf ("(DD/MM)");
gotoxy(37, 5);
printf ("INICIO");
gotoxy(35, 6);
printf("(HH:MM:SS)");
gotoxy(52, 5);
printf ("TERMINO");
gotoxy (51, 56);
printf("(HH:MM:SS)");
gotoxy(66, 5);
printf("PRIORIDADE");
gotoxy(69, 6);
printf ("NUM.");
```



```
}
```

```
void repeticao (int count[])
```

```
{
char k;
putimage(0, 0, arrow, COPY_PUT);
count[n]++;
gotoxy(20, 24);
printf ("VOCE JA PROGRAMOU ESTA CARGA %d VEZ(es) \a",
count[n]);
gotoxy(15, 25);
printf ("DESEJA CONTINUAR PROGRAMANDO A MESMA CARGA ?
(S/N) : ");
k = getch();
switch (k)
{
case 's' :
        putimage(0, 0, arrow, COPY_PUT);
        gotoxy(10, (6+aux));
        printf ("%d", n);
        gotoxy(24, (6+aux) );
        scanf ("%d/%d", &d, &m);
        carga[n][count[n]].dia = d;
        carga[n][count[n]].mes = m;
        gotoxy (37, (6+aux));
        scanf ("%d: %d: %d", &hi, &mi, &si);
        carga[n][count[n]].h_inicio = hi;
        carga[n][count[n]].m_inicio = mi;
        carga[n][count[n]].s_inicio = si;
        gotoxy (53, (6+aux));
        scanf ("%d: %d: %d", &hf, &mf, &sf);
        carga[n][count[n]].h_fim = hf;
        carga[n][count[n]].m_fim = mf;
        carga[n][count[n]].s_fim = sf;
        gotoxy (70, (6+aux));
        scanf ("%d", &p);
        carga[n][count[n]].prior = p;
        size = imagesize(0, 0, getmaxx(), getmaxy());
        arrow = malloc(size);
        getimage (0, 0, getmaxx(), getmaxy(), arrow);
}
```

```
break;
```

```
case "n" :
```

```
    putimage (0, 0, arrow, COPY_PUT);  
    count[n]--;  
    aux- = 2;  
    return;
```

```
}
```

```
cleardevice();
```

```
putimage (0, 0, arrow, COPY_PUT);
```

```
return;
```

```
}
```

```
void temperatura (void)
```

```
{
```

```
detection();
```

```
loop2 :
```

```
do {
```

```
    cleardevice();
```

```
    rectangle (0, 0, getmaxx() ,getmaxy());
```

```
    settextjustify(CENTER_TEXT, CENTER_TEXT);
```

```
    settextstyle(TRIPLEX_FONT, HORIZ_DIR, 1);
```

```
    outtextxy(( getmaxx() / 2), 10, "ATUALIZACAO DE  
TEMPERATURA");
```

```
    termometro(temp_atual);
```

```
    settextstyle (DEFAULT_FONT, HORIZ_DIR, 1);
```

```
    outtextxy(350, 68, "TEMPERATURA ATUAL : ");
```

```
    gotoxy(55, 9);
```

```
    printf ("%d °C", temp_atual);
```

```
    outtextxy (370, 100, "ENTRE COM A NOVA TEMPERATURA :
```

```
");
```

```
    line (350, 120, 370, 120);
```

```
    gotoxy(48, 15);
```

```
    printf (" °C ");
```

```
    gotoxy(45, 15);
```

```
    setcursortype( _NOCURSOR);
```

```
    scanf("%d", &temp_nova);
```

```

if (temp_nova<0 :: temp_nova>100)
{
gotoxy(35, 20);
printf ("TEMPERATURA DEVE ESTAR ENTRE");
gotoxy(42, 21);
printf ("0 °C E 100 °C \a");
getch();
goto loop2;
}
temp_atual = temp_nova;
termometro(temp_atual);
gotoxy(35, 20);
printf("DESEJA CONTINUAR ? (S/N) :");
t = tolower(getch());
} while (t == 's');

```

```

_setcursortype( _NORMALCURSOR);
closegraph();
return;
}

```

```

void termometro (int temp_atual)

```

```

{
setcolor (getmaxcolor());
line (160, 20, 470, 20);
line (120,50,120,152);
arc (125,155,110,60,10);
line (130, 50 ,130 ,152);
arc (125, 50, 0, 180, 5);
line (120, (150 – temp_atual), 130, (150 – temp_atual));
setfillstyle (SOLID_FILL, getmaxcolor());
floodfill (125, 155, getmaxcolor());
}

```

```

void demanda (void)

```

```

{
detection();

```

```

grafico();
closegraph();
return;
}

```

void grafico (void)

```

{
struct time t;
cleardevice();
rectangle (0, 0, getmaxx() ,getmaxy());
settextjustify(CENTER_TEXT, CENTER_TEXT);
settextstyle(TRIPLEX_FONT, HORIZ_DIR, 1);
outtextxy(( getmaxx() / 2), 10, "ATUALIZACAO DE DEMANDA");
line (180, 20, 450, 20);
line (170, 30, 170, 132);
line ( 168, 130, 490, 130);
line ( 270, 129, 270, 132);
line ( 370, 129, 370, 132);
line ( 470, 129, 470, 132);
setcolor(getmaxcolor());
bar3d (170, 70, 269, 130, 10, 1);
bar3d (271, 40, 369, 130, 10, 1);
bar3d (371, 70, 469, 130, 10, 1);
settextstyle(SMALL_FONT, HORIZ_DIR, 4);
outtextxy(170, 134, "0");
outtextxy(270, 134, "8");
outtextxy(370, 134, "16");
outtextxy(470, 134, "24");
outtextxy(510, 130, "HORAS");
outtextxy(145, 30, "CONSUMO");
gotoxy(20, 20);
gettime(&t);
gotoxy (6, 10);
printf ("%2d:%02d:%02d", t.ti_hour, t.ti_min, t.ti_sec);
consumo();
}

```

void consumo (void)

```

{
double nova_demanda;
char r, x;
do {
    gotoxy(20, 20);
    printf("CONSUMO ESTIPULADO PARA O PERIODO : %lf
KWH", demanda_estipulada);
    gotoxy(20, 21);
    printf("CONSUMO INSTANTANEO : ????? KWH");
    gotoxy(20, 22);
    printf("TARIFACAO ACUMULADA : R$ ?????");
    gotoxy(20, 24);
    printf("SE DESEJAR CONTINUAR, PRESSIONE < ENTER >");
    gotoxy(20, 25);
    printf("SE DESEJAR SAIR, PRESSIONE < ESC >");
    x = getch();
    if (x == 13)
    {
        gotoxy(20, 24);
        printf(" ");
        gotoxy(20, 25);
        printf(" ");
        line(0, getmaxy(), getmaxx(), getmaxy());
        gotoxy(20, 23);
        printf("ENTRE COM NOVO VALOR : ");
        line(430, 185, 510, 185);
        gotoxy(65, 23);
        printf("KWH");
        gotoxy(56, 23);
        scanf("%lf", &nova_demanda);
        demanda_estipulada = nova_demanda;
        gotoxy(29, 24);
        printf("DESEJA CONTINUAR ? (S/N) :");
        r = tolower(getch());
    }
    if (x == 27)
    {
        closegraph();
        return;
    }
} while (r == 's');
return;

```