

TRABALHO DE GRADUAÇÃO

SISTEMA SUPERVISÓRO EM ACTIONVIEW PARA IMPLEMENTAÇÃO DE CONFORTO TÉRMICO EM REDES WIRELESS

Marco Willian Amaral Cardoso

Brasília, Março de 2010



**ENGENHARIA
MECATRÔNICA**
UNIVERSIDADE DE BRASÍLIA

TRABALHO DE GRADUAÇÃO

**SISTEMA SUPERVISÓRIO EM ACTIONVIEW
PARA IMPLEMENTAÇÃO DE CONFORTO
TÉRMICO EM REDES WIRELESS**

POR,

Marco Willian Amaral Cardoso

Relatório submetido como requisito parcial para obtenção do grau de Engenheiro de Controle e Automação.

Banca Examinadora

Prof. Adolfo Bauchspiess, UnB/ ENE (Orientador)

Prof. João Yoshiyuki Ishihara , UnB/ ENE

Prof. Lélío Ribeiro Soares Júnior, UnB/ ENE

Brasília, Março de 2010

FICHA CATALOGRÁFICA

CARDOSO, MARCO WILLIAN AMARAL	
Sistema supervisorio em ActionView para implementação de conforto térmico em redes wireless, [Distrito Federal] 2010.	
xi, 80p., 297 mm (FT/UnB, Engenheiro, Controle e Automação, 2009). Trabalho de Graduação – Universidade de Brasília. Faculdade de Tecnologia.	
1.Sistema supervisorio	2.Conforto térmico
3.Wireless	4.Índice PMV
I. Mecatrônica/FT/UnB	II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

CARDOSO, M. W. A., (2009). Sistema supervisorio em ActionView para implementação de conforto térmico em redes wireless. Trabalho de Graduação em Engenharia de Controle e Automação, Publicação FT.TG-nº 028 , Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 80p.

CESSÃO DE DIREITOS

AUTOR: Marco Willian Amaral Cardoso.

TÍTULO DO TRABALHO DE GRADUAÇÃO: Sistema supervisorio em ActionView para implementação de conforto térmico em redes wireless.

GRAU: Engenheiro

ANO: 2009

É concedida à Universidade de Brasília permissão para reproduzir cópias deste Trabalho de Graduação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desse Trabalho de Graduação pode ser reproduzida sem autorização por escrito do autor.

Marco Willian Amaral Cardoso
QMSW 02 conjunto C lote 27 – Sudoeste.
70680-200 Brasília – DF – Brasil.

AGRADECIMENTOS

A minha mãe, pela dedicação, renúncia e força para proporcionar os meus estudos.

Ao professor Adolfo, pela orientação, confiança, reconhecimento e compreensão das dificuldades enfrentadas ao longo da realização desse trabalho.

Aos grandes amigos do curso, da empresa júnior Mecajun e do LAVSI, em especial ao Paulo, Helger, Luis Felipe, Rodrigo, Pedro e Raphael pelo grande auxílio e acompanhamento durante este projeto e na graduação.

Marco Willian Amaral Cardoso.

RESUMO

Dentro do conceito de *Ambient Intelligence*, que tem como objetivo estabelecer o conforto aos usuários destes ambientes através da interação dos diversos sistemas que o compõem, este trabalho apresenta os parâmetros para determinar o conforto térmico por meio do índice PMV (*Predicted Mean Vote*). Nestes ambientes são necessários sistemas para gerenciar as informações, controlar os equipamentos e apresentar de forma simplificada os dados, obtidos de sensores e atuadores com comunicação *wireless*, logo é considerado a utilização de um software supervisor.

Palavras Chave: sistema supervisor, conforto térmico, *wireless*, índice *PMV*.

ABSTRACT

Considering "Ambient Intelligence", which have as one of its purpose to provide comfort for the users of these environments through interaction of the various systems that compose it, this work presents the parameters determining the thermal comfort using the *Predicted Mean Vote* (PMV) index. In these environments require systems to manage information, control equipment and provide a simplified data obtained from sensors and actuators with wireless communication, so it is considered the use of a supervisory software.

Keywords: supervisory system, thermal comfort, wireless, PMV

SUMÁRIO

1. INTRODUÇÃO	1
1.1. OBJETIVO	1
1.2. ESTRUTURA DO TEXTO	2
2. CONFORTO TÉRMICO	3
2.1. VISÃO GERAL	3
2.2. NORMAS TÉCNICAS	4
2.2.1. ASHRAE 55	4
2.2.2. ASHRAE 62	4
2.2.3. ISO 7726	4
2.2.4. ISO 7730	4
2.2.5. ISO 7993	5
2.2.6. ISO 8996	5
2.2.7. ISO 9920	5
2.2.8. Outras normas	6
2.3. ÍNDICE PMV	6
3. REDES DE SENSORES SEM FIO	10
3.1. VISÃO GERAL	10
3.2. ZIGBEE	10
3.2.1. Topologias de redes ZigBee	11
3.3. XBEE	12
3.3.1. Modos de operação	13
3.3.2. X-CTU	14
3.3.3. CON-USBBEE	15
4. SISTEMAS SUPERVISÓRIOS	16
4.1. VISÃO GERAL	16
4.2. PLANEJAMENTO DO SUPERVISÓRIO	17
4.2.1. Entendimento do processo a ser automatizado	17
4.2.2. Planejamento da tomada de dados	17
4.2.3. Planejamento do banco de dados	17
4.2.4. Planejamento de alarmes	18
4.2.5. Planejamento a hierarquia de navegação entre telas	18
4.2.6. Desenho de telas	18
4.2.7. Gráfico de tendências	18
4.2.8. Segurança	19
4.2.9. Padrão industrial de desenvolvimento	19
4.3. ACTIONVIEW	19
4.3.1. Visão Geral	19
4.3.2. Módulo de configuração - AVStudio	20
4.3.3. Módulo <i>run-time</i>	20
4.3.4. Módulos de comunicação	23
4.3.5. Módulo de desenvolvimento	23
5. DESENVOLVIMENTO	25
5.1. CONFIGURAÇÃO DO SUPERVISÓRIO	25
5.1.1. Entendimento do processo	25
5.1.2. Planejamento da tomada de dados	26
5.1.3. Planejamento do banco de dados	27

5.1.4.	Planejamento dos alarmes.....	27
5.1.5.	Planejamento de hierarquia de navegação entre telas	27
5.1.6.	Desenho da telas.....	27
5.1.7.	Gráficos de tendências.....	30
5.1.8.	Segurança.....	31
5.1.9.	Padrão industrial de desenvolvimento.....	31
5.2.	COMUNICAÇÃO DO SUPERVISÓRIO.....	31
5.2.1.	Elaboração de protocolo simplificado	31
5.2.2.	Utilização de protocolo BACnet criado no LAVSI	32
5.2.3.	Processamento paralelo de programas em tempo real.....	32
6.	CONCLUSÃO	35
	REFERÊNCIAS BIBLIOGRÁFICAS	36
	ANEXOS.....	39
	ANEXO I	40
	ANEXO II	43
	ANEXO III	53
	ANEXO IV	66
	ANEXO V	69

LISTA DE FIGURAS

Figura 2.1 - Escala de percepções térmicas da ISO 10551/95.	8
Figura 2.2 – Relação entre os índices PMV e PPD.	8
Figura 3.1 – Aplicações do padrão Zigbee.	11
Figura 3.2 – Topologias de redes ZigBee.	12
Figura 3.3 – Módulo Xbee.	12
Figura 3.4 – Estrutura de comando enviado	14
Figura 3.5 – Tela do programa X-CTU da Digi®.	15
Figura 3.6 – Placa CON-USBEE.	15
Figura 4.1 – Módulo configurador – AVStudio.	20
Figura 4.2 – Tela sumário de eventos e alarmes correntes.	21
Figura 4.3 – Tela de tendência histórica de variáveis.	22
Figura 4.4 – Tela de tendência em tempo real.	22
Figura 4.5 - Tela de árvore do sistema.	23
Figura 5.1 – Planta baixa do laboratório LAVSI.	25
Figura 5.2 – Disposição dos equipamentos na planta piloto.	26
Figura 5.3 – Planejamento da disposição dos sensores e atuadores.	26
Figura 5.4 – Tela do índice PMV	28
Figura 5.5 – Tela de temperatura	28
Figura 5.6 – Tela de umidade relativa do ar.	29
Figura 5.7– Tela de velocidade do vento.	29
Figura 5.8 – Tela de parâmetros do controlador.	30
Figura 5.9 – Tela de configuração de parâmetros.	30
Figura 5.10 – Tela do supervísório elaborado em [20].	33
Figura AIII.1 – Fluxograma de funcionamento do software supervísório.	54

LISTA DE TABELAS

Tabela 3.1 – Descrição dos pinos do XBee.	13
Tabela AI.1 – Taxas metabólicas segundo a norma ISO 7730.	40
Tabela AI.2 – Resistência térmica da vestimenta	42
Tabela AII.1 – Variáveis utilizadas.....	43
Tabela AIII.1 – Configuração dos módulos da rede	53

LISTA DE SÍMBOLOS

Símbolos Latinos

C	Calor perdido por convecção	$[W/m^2]$
E_{dif}	Calor perdido por difusão de vapor de água através da pele	$[W/m^2]$
E_{rl}	Calor perdido por respiração latente	$[W/m^2]$
E_{rs}	Calor perdido por respiração sensível	$[W/m^2]$
E_{trans}	Calor perdido por transpiração	$[W/m^2]$
f_{vest}	Fator de vestuário	$[m^2K/W]$
I_{vest}	Resistência térmica da vestimenta	$[m^2K/W]$
L	Carga térmica	$[W/m^2]$
p_a	Pressão de vapor	$[kPa]$
p_{sat}	Pressão de saturação	$[kPa]$
R	Calor perdido por radiação	$[W/m^2]$
T	Período de amostragem	$[s]$
T_{ar}	Temperatura do ar	$[°C]$
T_{ar1}	Temperatura do ar do ambiente 1	$[°C]$
T_{ar2}	Temperatura do ar do ambiente 2	$[°C]$
T_{ar3}	Temperatura do ar da sala de reunião	$[°C]$
T_{met}	Taxa metabólica	$[W/m^2]$
T_{rad}	Temperatura radiante média	$[°C]$
T_{rad1}	Temperatura média radiante do ambiente 1	$[°C]$
T_{rad2}	Temperatura média radiante do ambiente 2	$[°C]$
T_{vest}	Temperatura da vestimenta	$[°C]$
V	Velocidade do vento	$[m/s]$

Grupos Adimensionais

PMV	<i>Predicted Mean Vote</i>
PPD	<i>Predicted Percentage of Dissatisfied</i>
UR	Umidade relativa do ar

Siglas

ABNT	Associação Brasileira de Normas Técnicas
ADC	Analog to Digital Converter
API	Application Programming Interface
ASHRAE	American Society of Heating, Refrigerating and Air-Conditioning Engineers

bps	bits por segundo
BTU	<i>British Thermal Unit</i>
°C	Graus Celsius
DLL	<i>Dynamic Library Link</i>
FFD	<i>Full Function Device</i>
Hz	Hertz
HVAC	<i>Heating Ventilating and Air Conditioning</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IHM	Interface Homem-Máquina
ISM	<i>Industrial, Scientific and Medical</i>
ISO	<i>International Organization for Standardization</i>
LARA	Laboratório de Robótica e Automação
LAVSI	Laboratório de Visão e Sistemas Inteligentes
MS-DOS	<i>Microsoft Disk Operation System</i>
RF	Radio Freqüência
RFD	<i>Reduced Function Device</i>
SCADA	<i>Supervisory Control and Data Acquisition</i>
USB	<i>Universal Serial Bus</i>
UnB	Universidade de Brasília
UTR	Unidade Terminal Remota
<i>Zigbee</i>	Padrão IEEE 802.15.4 de comunicação sem fio

1. INTRODUÇÃO

Os ambientes inteligentes objetivam proporcionar serviços para o conforto dos ocupantes através de diversos sistemas que interagem entre si [5]. Estes ambientes diferem dos ambientes convencionais no modo de gerenciamento e utilização dos diversos sistemas relacionados ao conforto.

Desta forma, os ambientes convencionais operam seus diversos sistemas independentemente, isto é, um sistema de ar condicionado não tem interação com o controle de acesso, logo não permite variar a potência do aparelho de ar condicionado quando se tem maior ou menor fluxo de pessoas, assim identificando como parâmetro principal a temperatura e não otimizando o sistema para considerar todos os fatores relacionados ao conforto térmico.

O estudo de conforto térmico é motivado por 3 fatores: proporcionar satisfação ao homem quanto a sensação térmica, melhorar a performance humana em atividades intelectuais, manuais e perceptivas, pois alguns estudos mostram maior rendimento destas atividades em ambientes com conforto térmico, e para otimização da eficiência energética, principalmente devido ao alto gasto de energia dos aparelhos de ar condicionado [1].

Para implementar ambientes confortáveis termicamente, é necessário a utilização de sistemas de supervisão, os quais são voltados a processar as informações do sistema e apresentá-las de forma simplificada aos operadores deste *software*, estabelecer o controle dos equipamentos e dos modos de atuação no processo. Considerando ainda estes ambientes, justifica-se a utilização de redes sem fio, tanto para novos projetos quanto para projetos antigos que normalmente necessitam de retrofitting para continuar atendendo especificações de conforto térmico.

1.1. OBJETIVO

O objetivo deste trabalho é apresentar as variáveis do processo relacionadas a conforto térmico do LAVSI através do sistema supervisor ActionView que foi desenvolvido pela empresa SPIN Engenharia de Automação Ltda. Para obtenção dos parâmetros de conforto, deve se utilizar os sensores e atuadores já disponíveis no laboratório, que foram elaborados em trabalhos anteriores da área de automação predial sem fio. Desta forma, o projeto visa demonstrar o funcionamento desse hardware e registrar em banco de dados a evolução temporal das variáveis medidas.

1.2. ESTRUTURA DO TEXTO

Este trabalho é composto por seis capítulos. O primeiro apresenta o contexto do projeto, descrevendo a motivação do estudo e os objetivos propostos para a realização do mesmo.

O segundo capítulo aborda alguns conceitos de conforto térmico, resume as principais normas utilizadas e traz um detalhamento do índice PMV e da ISO 7730.

O terceiro capítulo resume a teoria de sistemas supervisórios, apresentam uma metodologia de planejamento eficaz e o software supervisório ActionView.

No capítulo quatro é apresentado uma visão geral de redes sem fio, abrangendo o protocolo ZigBee e os módulos de comunicação XBee.

O quinto capítulo descreve o trabalho desenvolvido e no capítulo seis, são apresentadas as conclusões e sugestões para trabalhos futuros.

2. CONFORTO TÉRMICO

Este capítulo apresenta uma introdução aos conceitos de conforto térmico, resumo das principais normas e detalhamento do índice PMV e da ISO 7730.

2.1. VISÃO GERAL

Segundo a ISO 7730, “conforto térmico é o estado de alma que expressa satisfação com o ambiente térmico” [4].

O desempenho durante qualquer atividade física ou mental pode ser otimizado, desde que o ambiente propicie condições de conforto. O ambiente térmico é constituído principalmente pelos seguintes parâmetros [10]:

- Temperatura do ar, que influencia a troca de calor por convecção;
- Temperatura média radiante, que influencia a troca de calor por radiação;
- Velocidade relativa do ar, que influencia a troca de calor na convecção forçada;
- Umidade, que influencia a troca de calor por respiração e por evaporação de água ou suor.

Aliado a estes fatores ambientais, a troca de calor do homem com o ambiente também é influenciado por outros fatores:

- Metabolismo, com a produção interna de calor do corpo;
- Resistência térmica das roupas, a qual gera uma resistência na troca de calor por convecção.

Outros diversos fatores influenciam no conforto térmico, como a geometria do ambiente afinal o ar condicionado produz correntes de ar e através das janelas pode-se ter grande incidência solar e assim provocando desconforto.

É importante determinar quais as combinações das variáveis acima que levam o conforto térmico do homem. Dois métodos são bastante utilizados para ambientes térmicos moderados, o primeiro é sugerido pela ASHRAE 55 que é baseado em experiências onde pessoas são submetidas a diversas condições votando em relação a sua sensação térmica. O segundo método é baseado na ISO 7730 o qual consiste em equacionar a troca de calor do homem com o ambiente, introduzindo condições de conforto fisiológicas da temperatura da pele e da quantidade de suor e transpiração [10].

A seguir, é apresentada uma breve síntese das principais normas utilizadas para se estabelecer um conforto térmico em ambientes fechados.

2.2. NORMAS TÉCNICAS

2.2.1. ASHRAE 55

A norma ASHRAE 55 especifica as características de ambientes internos que, associadas a fatores pessoais, geram condições térmicas aceitáveis para no mínimo 80% dos ocupantes do local.

Esta norma utiliza como padrões físicos do ambiente: temperatura, umidade do ar, radiação térmica e velocidade do ar, e para os parâmetros pessoais, o tipo de vestimenta utilizada e o nível de atividade exercida no ambiente. É indispensável à medição de todos os critérios em conjunto, afinal para se estabelecer o conforto leva-se em consideração a interação de todos os fatores [8].

2.2.2. ASHRAE 62

A norma ASHRAE 62 especifica a taxa mínima de ventilação e a qualidade do ar que sejam aceitáveis para evitar efeitos adversos à saúde humana. Esta norma é aplicada a todos os espaços fechados onde não necessite uma norma específica em relação à ventilação devido a requisitos maiores ou quando estiver em conflito com a norma ASHRAE 55. Outra regulamentação da ASHRAE 62 é relacionada ao controle da quantidade de partículas dispersas no ar, determinando taxas de ventilação aceitáveis ou a instalação de filtros de ar e coletores de poeira no sistema de ar condicionado [8].

2.2.3. ISO 7726

A ISO 7726 especifica os requisitos mínimos dos instrumentos de medição de grandezas físicas, assim como apresenta métodos de medição destes parâmetros a fim de caracterizar um ambiente. Ela não objetiva definir um índice global de conforto térmico ou stress, mas simplesmente tem a finalidade de padronizar o processo de obtenção de informações que a leve a determinação destes índices, sendo citada em diversas normas.

Esta norma especifica duas classes de equipamentos, os da classe C para métodos e medições em ambientes moderados que se aproximam do conforto térmico e a classe S para métodos e medições em ambientes sujeitos à grande stress térmico [4].

2.2.4. ISO 7730

A norma ISO 7730, baseada nos estudos de Fanger (1970), especifica que a porcentagem de pessoas insatisfeitas seja inferior a 10% para este ambiente proporcionar condições térmicas aceitáveis de conforto térmico. Esta norma apresenta um método para previsão da sensação térmica e do grau de desconforto de pessoas expostas a ambientes de temperatura moderada. Além disso, especifica as condições térmicas aceitáveis para o conforto [8].

A sensação térmica humana está relacionada ao equilíbrio térmico corporal. Este equilíbrio é influenciado por vários parâmetros, como o nível de atividade física, tipo de vestimenta,

temperatura do ar, temperatura média radiante, velocidade do vento e umidade do ar. Com a medição ou estimativa destes fatores, pode-se prever a sensação térmica corporal calculando o índice PMV (*Predicted Mean Vote*), o qual representa a porcentagem esperada de pessoas satisfeitas para diferentes condições térmicas do ambiente. Através de uma correlação com o PMV, pode-se estimar o índice PPD (*Predicted Percentage of Dissatisfied*) [8].

De acordo com o manual proposto em [22], elaborado pelo Ministério do Trabalho para especificar condições de conforto térmico em ambientes de trabalho, pode-se verificar a necessidade de aplicação no Brasil e em [13] confirma-se que a ISO 7730 é uma das normas mais aceitas em meios acadêmicos relacionado a conforto térmico.

Na seção 2.3 deste trabalho será novamente abordado esta norma e o índice PMV para um melhor detalhamento.

2.2.5. ISO 7993

Esta norma determina um método analítico de avaliação e interpretação do stress térmico em um ambiente quente. Ela descreve um método de cálculo do balanço de calor, bem como a taxa de suor que o corpo humano deveria produzir para manter este equilíbrio estabelecendo uma taxa de suor desejada. Este padrão faz com que seja possível determinar qual parâmetro ou grupo de parâmetros devem ser modificados, e em que medida, a fim de reduzir o risco à saúde [8].

2.2.6. ISO 8996

A taxa metabólica é a conversão de energia química em mecânica e térmica, dimensionando o custo energético da carga muscular e informa um índice numérico de atividade. O conhecimento da taxa metabólica é necessário para medir a produção de calor metabólico para a avaliação da regulação do calor humano.

A norma apresenta três abordagens de diferentes métodos para determinar a taxa metabólica, especificando a sua exatidão e o tipo de estudo exigido. A norma prevê tabelas indicando a taxa metabólica para atividades típicas de uma pessoa normal com base em medições. Ela fornece também uma descrição detalhada dos métodos para a determinação direta (por medição) da taxa metabólica, medindo o consumo de oxigênio ou através da frequência cardíaca [8].

2.2.7. ISO 9920

A norma ISO 9920 especifica a estimativa da resistência evaporativa e do isolamento térmico de diferentes tipos de roupas. Desta forma, estabelece uma relação da vestimenta básica com a sensação térmica percebida pelas pessoas, entretanto desconsidera os desgastes dos tecidos ao longo do tempo, a absorção de suor pelo tecido, os vestuários

especiais (como *dry-fit*, impermeabilizados, entre outros) e distribuições não uniformes de vestimentas no corpo [8].

2.2.8. Outras normas

Existe ainda uma vasta quantidade de normas para regulamentar as características de ambientes e algumas têm relacionamento indireto com as que foram apresentadas acima como a ISO 7243 que é complementar a ISO 7730, a qual somente pode ser utilizada em ambientes moderados enquanto a outra é utilizada para ambientes com stress térmico[18]. Logo verifica-se o relacionamento entre outras normas como a ISO 7933, que especifica métodos de avaliação e interpretação analítica do stress térmico e utiliza alguns requisitos da ISO 7243 [1].

Além da ASHRAE 62, que especifica a qualidade necessária de ventilação em um ambiente com conforto térmico, tem-se a norma ASHRAE 113 a qual define métodos para testes de ensaios para o desempenho da difusão do ar em ambientes fechados. E assim auxilia no desenvolvimento de um layout para os sistemas HVAC (*Heating Ventilating and Air Conditioning*), proporcionando um ambiente agradável termicamente com base no deslocamento de ar, na velocidade do vento e na temperatura do ar relacionados às zonas de aquecimento e resfriamento, entretanto não podendo ser aplicada em ambientes com apenas uma saída para o fluxo do ar [8].

Outra norma bastante utilizada é a NR-15 a qual estabelece limites a exposição de calor em ambientes de trabalho de stress térmico para trabalhadores no desempenho de suas atividades, examinando o tempo de exposição a estes locais. Considera-se o ciclo de trabalho em relação ao tempo de descanso, prevendo repousos, em alguns casos, em local diferente do que ambiente de trabalho, a fim de não comprometer a saúde do trabalhador [1].

2.3. ÍNDICE PMV

A ISO 7730 normatiza o modelo de predição do conforto térmico, bem como a avaliação da aceitabilidade térmica de ambientes internos [21]. Esta norma objetiva criar um ambiente em que todos estejam em conforto térmico, porém existem alguns parâmetros subjetivos que impossibilitam a completa satisfação de todos os usuários. Neste caso, busca-se uma porcentagem máxima de pessoas satisfeitas que não necessitem nem de estar mais frio nem mais quente no ambiente local, e assim estabelecendo uma neutralidade térmica [9].

De acordo com [1], o corpo dos seres humanos é homeotérmico, afinal, mesmo com alterações no meio ambiente, consegue manter a temperatura corporal constante devido a mecanismos que promovem trocas de calor com o ambiente, dissipando a energia gerada através das atividades físicas que exercemos. Neste contexto, verifica-se que um organismo

exposto por longo tempo a um ambiente térmico constante e moderado tenderá a um equilíbrio térmico com esse ambiente, isto é, a produção de calor pelo organismo, através de seu metabolismo, será igual à perda de calor do mesmo para o ambiente, através das diversas formas de transferência de calor.

O índice PMV é obtido por meio do balanço térmico entre o homem e o ambiente, onde a geração de calor pelo organismo devido à execução de uma atividade deve ser dissipada em igual proporção ao ambiente, através de dois meios de trocas térmicas: pele e respiração. A dissipação de calor através da pele pode ser dividida em perda de calor por convecção (C), radiação (R), difusão de vapor de água (E_{dif}) e transpiração (E_{trans}). A dissipação de calor através da respiração divide-se em perda de calor por respiração sensível (E_{rs}) e por respiração latente (E_{rl}) [21].

Considerando esta necessidade do corpo humano em obter um equilíbrio ou neutralidade térmica, existe a produção de calor pelo corpo, dada pela taxa metabólica (M), em reação às condições climáticas e a carga térmica (L) a qual é referente à realização de algum trabalho pelo corpo ou ao nível de atividade, isto é, quantidade extra de calor produzido ou armazenado na troca de calor com o ambiente em regime estacionário do balanço térmico. A partir destas condições, o equilíbrio ou balanço térmico corporal é dado por [4]:

$$M - L = E_{dif} + E_{trans} + E_{rl} + E_{rs} + R + C \quad (1)$$

Considera-se o trabalho desenvolvido como nulo para a maioria das atividades humanas que não tenha algum grande esforço do corpo, logo pode-se definir que há um equilíbrio do calor produzido e o dissipado, sendo nestes casos, definido o balanço térmico corporal por:

$$M = E_{dif} + E_{trans} + E_{rl} + E_{rs} + R + C \quad (2)$$

Dependendo da carga térmica à qual um corpo é submetido, este irá alterar sua temperatura média da pele e a taxa de suor para manter sua temperatura interna, o que altera a sensação térmica. Sendo assim, é possível inferir que existe uma relação entre a sensação térmica demonstrada por uma pessoa e a carga térmica à qual está submetida. A carga térmica, por sua vez, é influenciada pela produção de calor corporal interna. Logo, podemos dizer que a sensação térmica tem relação estreita com a produção de calor corporal interna [4].

Para o cálculo do índice PMV, utilizam-se quatro variáveis físicas (temperatura do ar, velocidade do ar, temperatura média radiante e umidade relativa) e duas variáveis pessoais (isolamento da vestimenta e nível de atividade) para prever a sensação térmica média de um conjunto de pessoas [19]. Portanto, a Fig. 2.1 apresenta a escala de sensação térmica, utilizada nos estudos de Fanger e incorporada à ISO 10551/95, para estabelecer um método de estimar qual a sensação térmica que a pessoa está sentindo no momento dos testes.

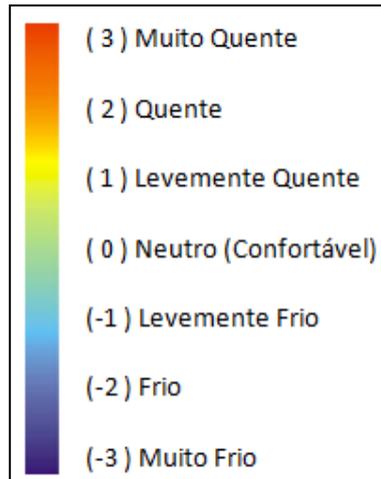


Figura 2.1 - Escala de percepções térmicas da ISO 10551/95 [20].

Através da expressão de equilíbrio térmico proposta na Eq. 1, a qual leva em consideração a relação do acúmulo de energia do corpo com o metabolismo e também considerando a análise estatística dos resultados experimentais de Fanger, em que utilizava-se o voto baseado na escala de sensação, propõe-se a equação do índice PMV descrita abaixo [21], [18]:

$$PMV = (0,303e^{-0,036 * M} + 0,028) * L \quad (3)$$

Para compreensão dos fatores que alteram o cálculo do voto médio estimado (PMV), substitui na Eq. 3 o valor da carga térmica (L) descrito na Eq. 1, tem se:

$$PMV = (0,303e^{-0,036 * M} + 0,028) * (M - E_{dif} - E_{trans} - E_{rl} - E_{rs} - R - C) \quad (4)$$

O cálculo de cada termo da Eq. 4 será descrito no Anexo 1, elaborado por [4].

As análises efetuadas com relação ao índice PPD foram feitas tomando-se por base os votos fornecidos pelas pessoas na escala de sensação térmica, tendo Fanger feito a seguinte consideração: somente se encontravam insatisfeitas as pessoas que votaram +3, +2, -3 e -2 na escala de sensações. Os votos +1 e -1 não caracterizava pessoas insatisfeitas com o ambiente [21]. A relação do índice PMV com a porcentagem de pessoas não satisfeitas (PPD) é apresentada pelo gráfico da Fig. 2.2

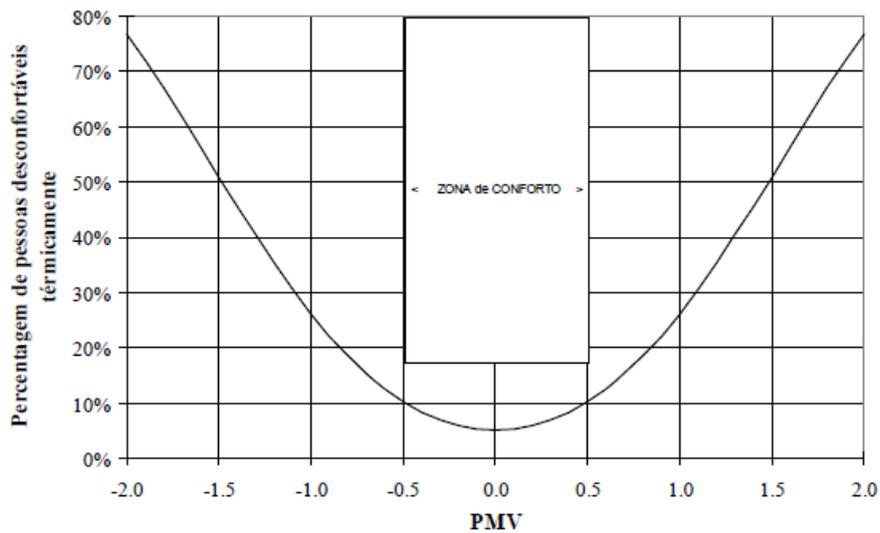


Figura 2.2 – Relação entre os índices PMV e PPD [18].

Analisando a Fig. 2.2, pode se constatar que mesmo no momento de conforto térmico, ou seja, quando o PMV é igual a zero, o índice PPD apresenta uma quantidade de pessoas insatisfeitas. Isto se deve a influência de fatores não mensuráveis ou pessoais que a norma está sujeita como a preferência de algumas pessoas por ambientes mais quentes e outras por mais frios. Portanto, pode ser estabelecida uma zona de conforto térmico, ao invés de ponto de conforto térmico, na qual o valor para o PMV pode estar entre -0,5 e 0,5 e ainda se satisfazer o percentual de pessoas insatisfeitas menor que 10%.

No Anexo IV, obtido através de [1], é proposto um roteiro com algumas exigências mínimas para padronização e validação de trabalhos e testes relacionados a conforto térmico.

3. REDES DE SENSORES SEM FIO

Este capítulo aborda alguns conceitos de redes sem fio, do protocolo ZigBee e dos módulos XBee.

3.1. VISÃO GERAL

Segundo [13], ambientes inteligentes objetivam melhorar a qualidade de vida do dia a dia, sendo que essa melhoria se dá através de criação de redes, módulos sem fio e unidades de processamento, contudo monitoram os parâmetros necessários para o ambiente se adaptar as necessidades de seus ocupantes.

Nesse contexto de automação predial, vem crescendo bastante a utilização de redes sem fio devido a diversos fatores, como o incentivo por parte das empresas no desenvolvimento de protocolos e equipamentos, por exemplo, estimou-se que o custo em 2005 de cada nó ZigBee foi de US\$ 1,10 para os grandes fabricantes [13]. Além disso, em ambientes desconfortáveis para se habitar ou trabalhar é necessário realizar o *retrofitting*, em que se propõem melhorias nessas infra-estruturas, como substituição de equipamentos (sensores, atuadores, entre outros) com baixa eficiência energética por outros sem fio, que não comportam novos cabeamentos e o alto custo para se desenvolver a construção de novas instalações se torna inviável.

Mesmo com o avanço na utilização das redes *wireless*, é importante alertar que ainda existem algumas lacunas no conhecimento e desenvolvimento das mesmas, em diversos aspectos como a confiabilidade e eficiência destas redes no âmbito da segurança de transmissão da informação. Dessa forma, o padrão ZigBee, que ainda está em desenvolvimento, surgiu como uma tecnologia relativamente simples, utilizando um protocolo de pacotes de dados com características específicas e projetado para oferecer flexibilidade quanto aos tipos de dispositivos que pode controlar, buscando soluções de baixo custo, baixo consumo de energia e que estabeleça confiabilidade de segurança [19].

3.2. ZIGBEE

O ZigBee é um padrão de comunicação em desenvolvimento para rede de sensores de curto alcance para aplicações com baixo consumo de energia pelos módulos e de baixo custo. Este protocolo apresenta baixa velocidade de transmissão em relação a outros protocolos, todavia, é suficiente para aplicações em projetos de automação predial como segurança, controle de acesso, iluminação, sensores e atuadores que são demonstrados na Fig. 3.1 [24].

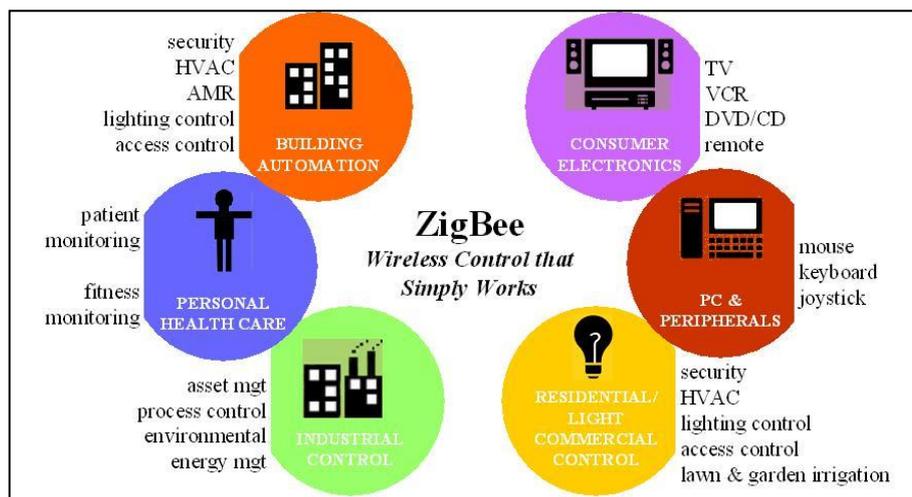


Figura 3.1 – Aplicações do padrão Zigbee [24].

Este padrão garante comunicações robustas e opera na frequência ISM (*Industrial, Scientific and Medical*), sendo utilizada a frequência de 868 MHz com um canal na Europa, 915 MHz com dez canais nos Estados Unidos e 2,4 GHz com dezesseis canais em outras partes do mundo, e não requerem licença para funcionamento. As Redes ZigBee oferecem uma excelente imunidade contra interferências, e a capacidade de hospedar milhares de dispositivos numa rede (mais que 65.000), com taxas de transferências de dados variando entre 20Kbps a 250Kbps [22].

3.2.1. Topologias de redes ZigBee

Neste protocolo existem três classes de dispositivos:

- Coordenador: dispositivo do tipo FFD (*Full Function Device*) responsável pela inicialização, distribuição manutenção da rede, reconhecimento de todos os nós e outras funções.
- Roteador: dispositivo FFD (*Full Function Device*) com a função de estabelecer a comunicação com outras redes ou até mesmo amplificar o sinal dentro da mesma rede sem precisar do coordenador.
- *End device*: dispositivo RFD (*Reduced Function Device*) ou FFD que apresenta limitada funcionalidade a um baixo custo e complexidade. Em alguns textos esse dispositivo também é chamado de *end point*.

Conforme mencionado, existem dois tipos de dispositivos para esse padrão, definidos como:

- *Full Function Device* (FFD) – dispositivos de construção mais complexa, devido à necessidade de desenvolvimento de um hardware acoplado para a implantação da pilha de protocolos, e desempenham a função de coordenador, roteador ou end device. O FFD consome mais energia devido ao hardware.
- *Reduced Function Device* (RFD) – dispositivo limitado pela simplicidade de sua uma configuração, atuando somente como *end device* e podendo comunicar com roteadores ou coordenadores de uma rede [22].

A Figura 3.2 mostra três diferentes topologias para redes ZigBee que são descritas como [19].

- Topologia estrela: topologia mais simples onde se tem um coordenador e os *end device*, indicada para ambientes que ofereçam poucos obstáculos para a transmissão dos sinais. A principal vantagem é a facilidade de implementação e coordenação, já como desvantagem tem a presença de um único coordenador, limitando o alcance do sinal e gerando dependência de todos os *end point*.
- Topologia árvore: topologia é formada por redes paralelas comunicando entre si através de roteadores e também é possível haver mais de um coordenador dentro da mesma topologia, contanto que cada um gerencie uma rede diferente. A aplicação principal está na comunicação de dispositivos situados em andares diferentes ou entre salas distantes entre si, separadas por paredes. O alcance do sinal de RF deste tipo de rede é maior, tendo em vista que um coordenador pode controlar um cliente ligado em outra rede através de uma solicitação ao outro coordenador.
- Topologia malha: essa topologia permite que, com a entrada de novos dispositivos, a rede se ajuste automaticamente durante sua inicialização, otimizando o tráfego de dados.

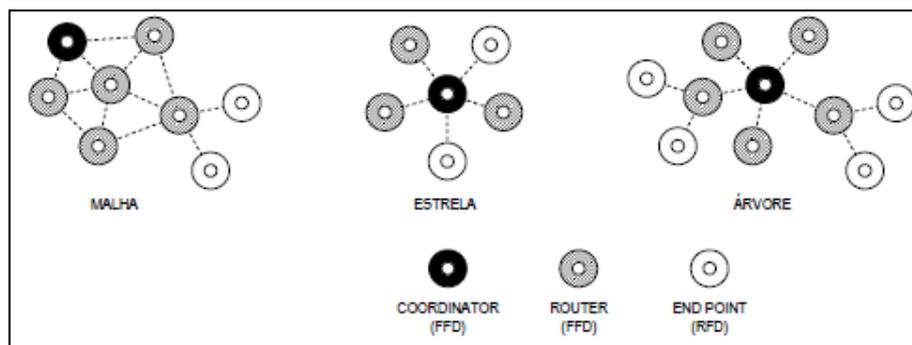


Figura 3.2 – Topologias de redes ZigBee [25].

3.3. XBEE

O módulo XBee, fabricado pela Digi (antiga MaxStream), é um transmissor configurado para atender o padrão IEEE 802.15.4, o qual corresponde ao ZigBee.

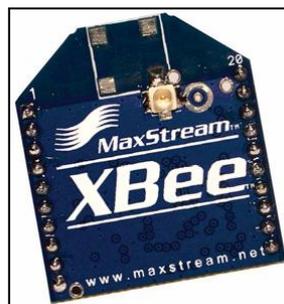


Figura 3.3 – Módulo Xbee [23].

O XBee, Figura 3.3, possui as seguintes características [23]:

- Faixa de transmissão em área interna: 30 metros;
- Faixa de transmissão em área externa: 90 metros;
- Taxa de transmissão: 250 Kbps;
- Potência de transmissão: 1 mW;
- Sensibilidade do receptor: - 92dBm;
- Tensão de alimentação: 3,3V
- Frequência de operação: 2,4 GHz.

A Tabela 3.1 mostra as especificações de cada um dos pinos do XBee, em que se pode verificar que os pinos 2 (DOUT) e 3 (DIN) são os responsáveis pela comunicação serial com outros dispositivos. Além disso, pode se destacar que o módulo contém entradas de conversão analógica, canais PWM e controle de fluxo.

Tabela 3.1 – Descrição dos pinos do XBee [23].

Pin #	Name	Direction	Description
1	VCC	-	Power supply
2	DOUT	Output	UART Data Out
3	DIN / CONFIG	Input	UART Data In
4	DO8*	Output	Digital Output 8
5	RESET	Input	Module Reset (reset pulse must be at least 200 ns)
6	PWM0 / RSSI	Output	PWM Output 0 / RX Signal Strength Indicator
7	PWM1	Output	PWM Output 1
8	[reserved]	-	Do not connect
9	DTR / SLEEP_RQ / DI8	Input	Pin Sleep Control Line or Digital Input 8
10	GND	-	Ground
11	AD4 / DIO4	Either	Analog Input 4 or Digital I/O 4
12	CTS / DIO7	Either	Clear-to-Send Flow Control or Digital I/O 7
13	ON / SLEEP	Output	Module Status Indicator
14	VREF	Input	Voltage Reference for A/D Inputs
15	Associate / AD5 / DIO5	Either	Associated Indicator, Analog Input 5 or Digital I/O 5
16	RTS / AD6 / DIO6	Either	Request-to-Send Flow Control, Analog Input 6 or Digital I/O 6
17	AD3 / DIO3	Either	Analog Input 3 or Digital I/O 3
18	AD2 / DIO2	Either	Analog Input 2 or Digital I/O 2
19	AD1 / DIO1	Either	Analog Input 1 or Digital I/O 1
20	AD0 / DIO0	Either	Analog Input 0 or Digital I/O 0

3.3.1. Modos de operação

O XBee possui quatro modos de operação [2], [11], [22] :

- Modo transparente: este é o modo padrão de operação do XBee, onde os dados são transmitidos e recebidos da mesma forma que a comunicação serial RS-232 padrão. Sendo que todos os dados que são inseridos pelo pino de entrada (DIN) da comunicação serial é transmitido pela antena, já todos os dados recebido pela antena é enviado pelo pino de saída (DOUT) da comunicação serial.
- Modo de comando: este modo permite acessar várias configurações do módulo. Por meio de uma seqüência específica de caracteres (“+++”), o XBee entra em um estado no qual parâmetros podem ser configurados e monitorados por meio de seqüências de caracteres, enviadas e recebidas através da comunicação serial do

módulo. Entre estes parâmetros, pode-se alterar, por exemplo, o canal de transmissão, o endereço do módulo na rede e o endereço de destino. A Figura 3.4 apresenta a estrutura dos comandos a serem enviados ao XBee neste modo de operação e o exemplo mostra a alteração do endereço de destino para o valor 1F:

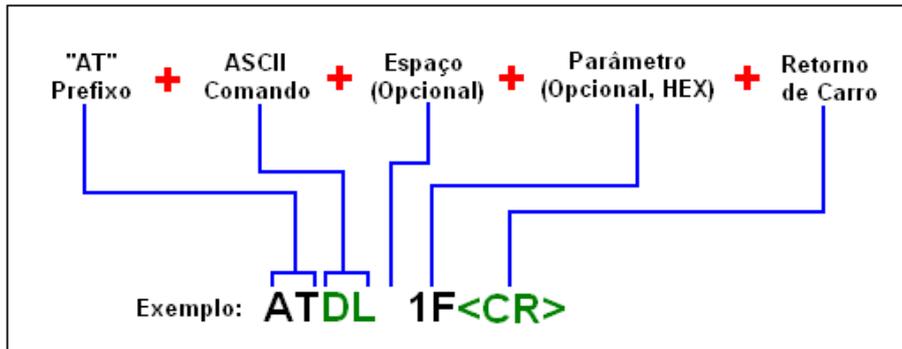


Figura 3.4 – Estrutura de comando enviado [22].

- Modo Sleep: este é o modo de maior economia de energia e pode entrar ou sair deste estado, por hardware com a utilização do pino 9 descrito na Tabela 3.1 ou por software – através do modo de comando - especificando o intervalo de tempo que o módulo ficará nesse estado. Também, é possível utilizar esses dois métodos em conjunto.
- Modo API: este modo é uma alternativa para o modo transparente. Neste modo, os dados transmitidos e recebidos estão contidos em frames, que definem operações ou eventos dentro do módulo. Desta forma, é possível um determinado módulo enviar endereço fonte, endereço destino, nome de um determinado nó, receber mensagens de sucesso ou falha após cada envio de dados e identificar o endereço do módulo que enviou a mensagem recebida.

3.3.2. X-CTU

O programa X-CTU (Figura 3.5) é disponibilizado gratuitamente pela Digi, por meio de sua página na internet (www.digi.com), e apresenta uma interface simples para configurar os parâmetros do módulo, atualizar o firmware e outras funções. O programa disponibiliza um terminal que possibilita a leitura de dados e registradores do módulo pela comunicação serial, a execução de comandos e o envio de dados a outros módulos [2].

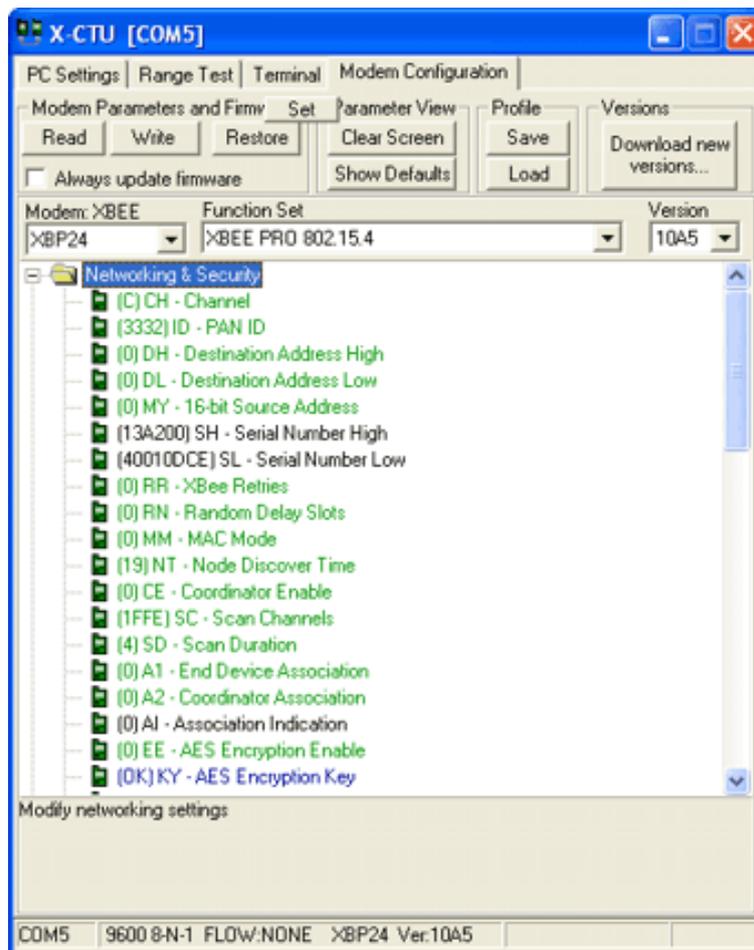


Figura 3.5 – Tela do programa X-CTU da Digi® [22].

3.3.3. CON-USBEE

A empresa Rogercom desenvolveu a placa CON-USBEE, visualizada na Fig. 3.6, para facilitar a conexão do módulo XBee ao computador pela porta USB a qual é muito mais freqüente do que a porta serial, tanto em computadores pessoais quanto notebooks [22].

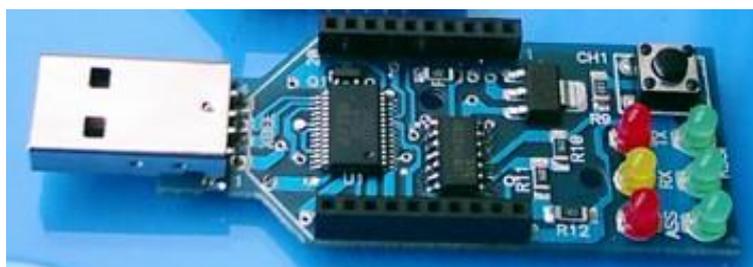


Figura 3.6 – Placa CON-USBEE [22].

Para a utilização da CON-USBEE é necessário a instalação do driver deste equipamento, o qual é disponibilizado pelo site do fabricante (www.rogercom.com) e é compatível com os seguintes sistemas operacionais: Windows 98, ME, 2000, XP, Vista, Linux e Mac. Conectando este dispositivo ao computador é criada uma porta de comunicação virtual, COMx (na maioria das vezes sendo COM4 ou COM5), similar a uma porta serial padrão RS232 [11].

4. SISTEMAS SUPERVISÓRIOS

Este capítulo introduz conceitos de sistemas supervisórios, apresenta um modo de planejamento eficaz e o software supervisório ActionView.

4.1. VISÃO GERAL

Os sistemas supervisórios permitem que sejam monitoradas e rastreadas informações de um processo. Utilizando equipamentos de aquisição de dados, se coletam informações as quais são manipuladas, analisadas, armazenadas e finalmente apresentadas ao usuário. Outra importante característica destes sistemas é a prevenção de falhas, através do reconhecimento das mesmas antes que ocorram efetivamente e a verificação das condições de alarmes, identificando quando o valor de alguma variável ultrapassa a sua faixa pré-estabelecida, sendo possível programar a gravação de registros em bancos de dados, a ênfase através da ativação de som, mensagem, mudança de cores, envio de mensagens por e-mail, celular, ou outros meios [6].

Estes sistemas também são chamados de SCADA (*Supervisory Control and Data Acquisition*). Os primeiros sistemas de supervisão eram basicamente telemétricos e permitiam informar periodicamente o estado corrente do processo industrial monitorando apenas sinais de medidas e estados de dispositivos através de um painel de lâmpadas e indicadores, sem que houvesse qualquer interface de aplicação com o operador. Com a evolução da tecnologia, os computadores pessoais (*desktop*) se tornaram mais acessíveis e passaram a ter um papel importante na supervisão dos sistemas, coletando e tornando disponíveis os dados do processo. Com o acesso remoto aos dados, facilita-se tanto o monitoramento quanto o controle do processo, fornecendo, em tempo útil, o estado atual do sistema através de gráficos, previsões ou relatórios, desta forma viabilizando a tomadas de decisões, seja automaticamente ou por iniciativa do operador [5].

Os supervisórios identificam as *tags*, que são todas as variáveis numéricas ou alfanuméricas envolvidas na aplicação, podendo executar funções computacionais (operações matemáticas, lógicas, com vetores ou strings) ou representar pontos de entrada/saída de dados do processo que está sendo controlado. Neste caso, correspondem às variáveis do processo real (ex: temperatura, umidade, velocidade do vento), se comportando como a ligação entre o controlador e o sistema. É com base nos valores das *tags* que os dados coletados são apresentados ao usuário [12].

4.2. PLANEJAMENTO DO SUPERVISÓRIO

Devido à complexidade de alguns sistemas de automação, se faz necessário a criação de um software para facilitar o trabalho do operador. Este software denominado de Supervisório ou Interface Homem-Máquina (IHM) deve ter uma interface bastante amigável, ou seja, tem de ter uma visualização tão eficiente que seja possível monitorar todos os dados relevantes da planta ou sistema de trabalho quanto deve propiciar condições para manipular os dados com simplicidade. A necessidade de simplicidade em trabalhar com esse tipo de software é de extrema importância afinal será implantado em algum ambiente de trabalho e os operadores têm de acostumar com essa nova ferramenta.

Segundo [3], nove etapas são recomendadas para o planejamento de um sistema supervisório para uma determinada planta:

4.2.1. Entendimento do processo a ser automatizado

Esta etapa constitui em um levantamento completo e detalhado do sistema a ser automatizado, sendo necessário conversar com operadores ou com especialistas do processo e também com o corpo administrativo para tomar conhecimento das operações da planta e anotar todas as informações relevantes as quais devem estar disponíveis para cada tipo de operador [3].

O processo deve ser quebrado em etapas, as quais são nomeadas de forma padronizada, levando em consideração os significados físicos, em seguida, deve-se determinar as variáveis de cada etapa, nomeando as conforme o padrão estabelecido.

4.2.2. Planejamento da tomada de dados

Consiste em escolher o protocolo de comunicação e considerar apenas os dados essenciais para que o sistema se torne conciso. É necessário estabelecer um limite da quantidade de dados, assim evitando o excesso de tráfego na rede [3].

4.2.3. Planejamento do banco de dados

O banco de dados deve conter informações sobre o fluxo do processo, lista de endereços dos dados e lista de alarmes, podendo ser elaborado em programas para aplicações mais simples em Microsoft® Access e para aplicações mais complexas com MySQL, Oracle, dentre outros.

As variáveis devem manter o padrão lógico estabelecido na etapa inicial onde se nomeia algumas variáveis, escolher a classe de varredura (*scan*), a qual é a velocidade de leitura das variáveis e utilizar pastas de arquivos que organizam as variáveis facilitando no manuseio e agrupando de maneira mais significativa, como exemplo aparelhos similares ou dentro da mesma planta [3].

4.2.4. Planejamento de alarmes

Visando garantir maior segurança no processo, um sistema de supervisão é capaz de gerar alarmes a partir da ocorrência de algum evento específico. Os alarmes são classificados por níveis de prioridade em função da sua gravidade, sendo reservada a maior prioridade para os alarmes relacionados com questões de segurança. Os sistemas SCADA armazenam em arquivos as informações relativas a todos os alarmes gerados, de modo a permitir que posteriormente, se necessário, uma análise mais detalhada das circunstâncias que os originou, prevenindo falhas no sistema [5].

Para o planejamento dos alarmes, é imprescindível verificar as condições de acionamento, a forma de notificação dos operadores, as providências a serem tomadas e o envio de mensagens. A análise e filtragem de eventos devem levar em consideração o nível de prioridade e o número de ocorrências do sinal, evitando o aparecimento de um número elevado de alertas simultaneamente, assim distribuindo os alertas e otimizando a eficácia do supervisor. Esta filtragem pode conter alguns scripts que tratam e automatizam a decisão de alguns alertas [3].

4.2.5. Planejamento a hierarquia de navegação entre telas

A hierarquia de navegação consiste em uma série de telas que fornecem progressivamente detalhes das plantas e seus constituintes à medida que se navega através do aplicativo. A boa organização da navegação torna o sistema claro e condizente com a realidade, guiando o serviço dos operadores. Geralmente, nas telas são projetadas barras de navegação com botões em que estão com nomes ou símbolos que identificam o conteúdo da tela a ser chamada [4].

4.2.6. Desenho de telas

Nesta etapa, tem de começar colocando os títulos, nomes chaves para as *tags* e botões de navegação das telas, sendo essencial organizar com cautela e de forma clara as partes constituintes das telas para auxiliar os operadores e aumentar a eficiência do sistema supervisor. Para tornar a interpretação e visualização do processo clara, deve se evitar o uso de abreviações difíceis de compreender e também o excesso de informações, dispondo somente o estritamente necessário.

Dessa forma, deve-se utilizar com consistência e padronização dos símbolos, cores e os nomes dos botões e tomando o cuidado para colocar os botões sempre nas mesmas posições em todas as telas, a fim de não comprometer o chaveamento entre telas [3].

4.2.7. Gráfico de tendências

Os gráficos de tendências mostram como determinadas variáveis mudam no decorrer do tempo, sendo possível apresentar os dados em tempo real durante a navegação do

processo. Se estes gráficos apresentarem mais de uma variável, devem ser plotados com várias cores para facilitar a visualização e interpretação. Deste modo, pode se analisar tendências do processo, monitorar a eficiência do sistema e arquivar variáveis de processo para garantir conformidade com algumas regulamentações [3].

4.2.8. Segurança

No planejamento da segurança deve-se listar todos os tipos de operadores que utilizarão o software, estabelecendo uma política de acesso, a qual restringirá as funcionalidades necessárias tanto no supervisório quanto no sistema operacional. Além de estabelecer uma senha para o acesso de cada operador, para garantir a confidencialidade, integridade e disponibilidade de acesso aos operadores devidamente autorizados, os procedimentos de segurança devem garantir que uma falha de segurança não resulte em conseqüências desastrosas ou catastróficas. Deste modo o sistema tem de ser confiável e controlável [7].

4.2.9. Padrão industrial de desenvolvimento

Recomenda-se desenvolver o sistema supervisório baseado em um padrão de IHM similar ao do Windows da Microsoft® devido a sua larga utilização por todos os tipos de usuários e assim reduzindo o tempo de aprendizagem do operador e também a integração com outros softwares do sistema operacional [3].

4.3. ACTIONVIEW

O ActionView é um software do tipo SCADA para a aplicação em sistemas de supervisão e controle de processos desenvolvido pela SPIN Engenharia de Automação Ltda. e executado em ambiente Windows.

4.3.1. Visão Geral

O ActionView é disponibilizado com os seguintes módulos principais:

- Módulo configurador - AVStudio: este módulo é utilizado para a geração e manutenção da base de dados de parâmetros do software SCADA, contendo a definição dos pontos supervisionados, das telas para visualização em tempo real, bem como as demais características da planta ou do processo controlado.[16].
- Módulo *run-time*: este módulo é utilizado para a implementação da interface gráfica de apresentação de telas, alarmes e estados atualizados de pontos supervisionados pelo sistema.
- Módulos de Comunicação: compatíveis com os protocolos dos principais fabricantes de UCL's (Unidades de Controle Local), UTR's (Unidades Terminais Remotas), CLP's (Controladores Lógicos Programáveis), relés e outros DEI's (Dispositivos Eletrônicos Inteligentes) [15].

- Kit de desenvolvimento - AVDK: este módulo apresenta os aspectos principais da estrutura interna ActionView, o relacionamento entre seus módulos, e, principalmente, a interface pela qual outros programas aplicativos podem se relacionar em tempo real com os dados e módulos do sistema.

4.3.2. Módulo de configuração - AVStudio

Este módulo corresponde a um ambiente de desenvolvimento que pode ser utilizado *off-line*, durante a fase de projeto, ou *on-line*, durante a execução da supervisão e controle em tempo real. Ele permite que o usuário faça a geração e manutenção do banco de dados de parâmetros do *software* SCADA, bem como a configuração das telas de processos. [16]

A Figura 4.1 mostra uma tela exemplo do AVStudio onde, no topo existem menus com todas as funções disponibilizadas e barras de ferramenta com atalhos para as principais funções. À esquerda, é apresentada a árvore do projeto em que o usuário pode definir estações de trabalho, canais de comunicação, usuários, perfis de acesso, base de dados de tempo real, telas de processo, telas de medidas e bibliotecas de objetos de visualização [14].

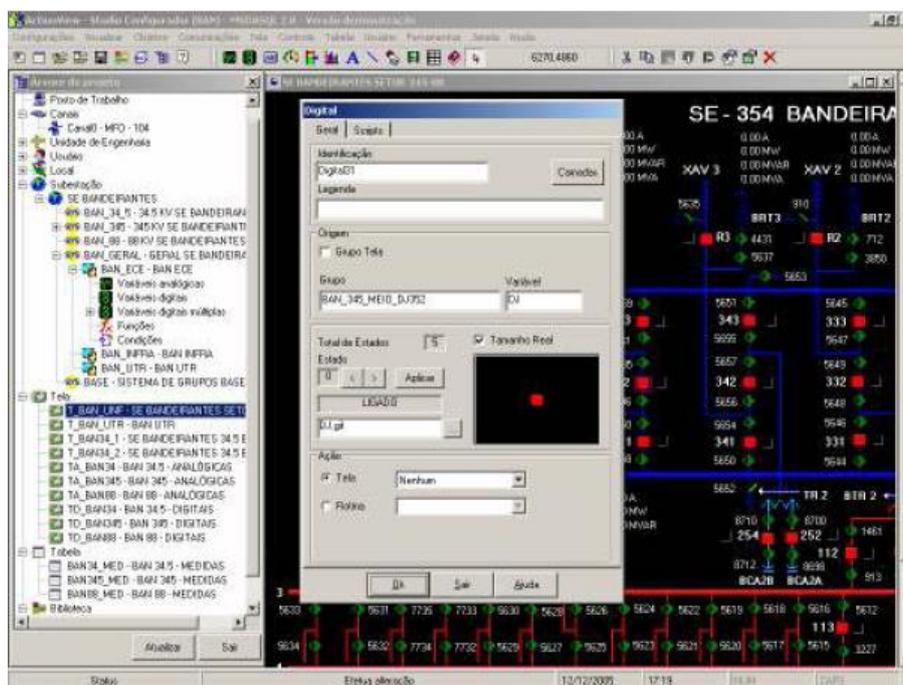


Figura 4.1 – Módulo configurador - AVStudio [14].

Outra importante funcionalidade do ActionView é a possibilidade de programação interna dos objetos através de scripts. Com esta programação é possível alterar-se o comportamento normal da visualização e monitoração dos pontos, através da escrita de procedimentos que serão executados na ocorrência de determinados eventos [14].

4.3.3. Módulo *run-time*

O módulo *run-time* é executado durante o monitoramento da planta, sendo responsável pela coleta (*scan*), tratamento, armazenamento e apresentação dos dados para o operador.

Portanto, este módulo é composto pelos seguintes processos: *AVServer* (servidor de comunicação e BDTR), *AVHistory* (servidor de dados históricos), *AVviewer* (servidor de IHM) e *AVWeb* (servidor de web) [14].

As principais telas disponibilizadas pelo *AVviewer* são descritas a seguir:

- Telas de processo: estas telas que contém diagramas gerais do sistema, animados através dos objetos de visualização que serão modificados em tempo real [15].
- Telas de medidas: estas telas são constituídas por tabelas cujas células podem ser textos constantes, cabeçalhos, valores atuais de variáveis analógicas, valores máximos ou mínimos diários, atualizados em tempo real [16].
- Telas sumário de eventos: estas telas apresentam a lista dos últimos eventos com o tamanho configurado pelo usuário. Dessa forma, pode se observar que um evento é qualquer mudança de estado de um equipamento, diferentemente de um alarme, que é uma mudança indesejável.

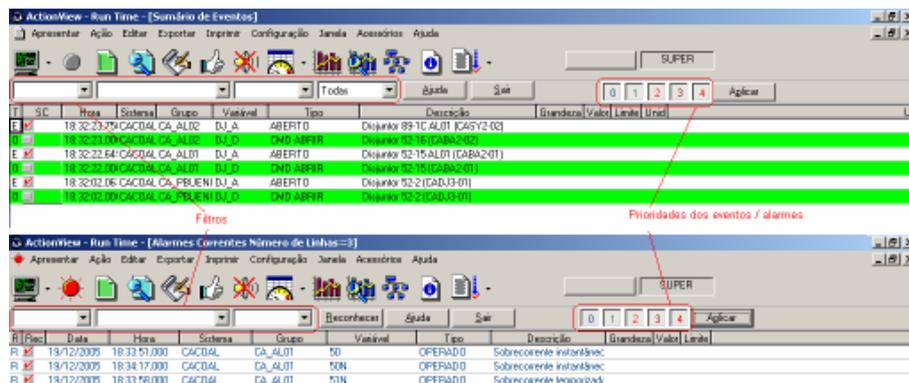


Figura 4.2 – Tela sumário de eventos e alarmes correntes [14].

- Telas de alarmes correntes: estas telas apresentam as mensagens referentes a alarmes correntes (não normalizados), reconhecidos ou não, conforme apresentado na Fig. 4.2 [14].
- Telas de tendência histórica de variáveis: estas telas que mostram gráficos ou tabelas com o comportamento de variáveis analógicas, a partir de arquivos históricos, em que as informações são apresentadas em gráficos e tabelas [14].

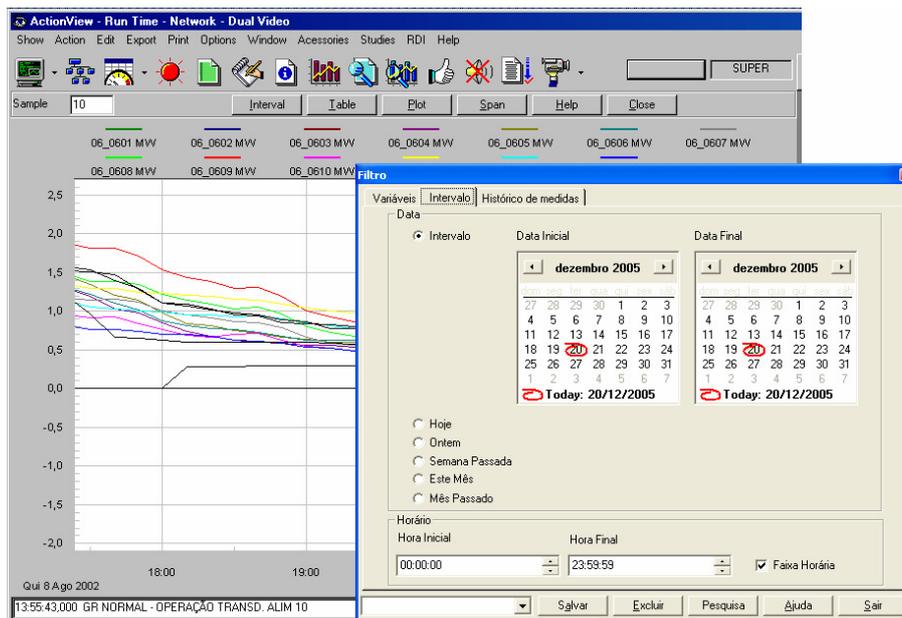


Figura 4.3 – Tela de tendência histórica de variáveis [14].

- Telas de tendência em tempo real: estas telas, Fig. 4.4, mostram gráficos de tendência de variáveis analógicas selecionadas pelo usuário, em tempo real. Estes gráficos possuem as mesmas características dos gráficos de tendência histórica.

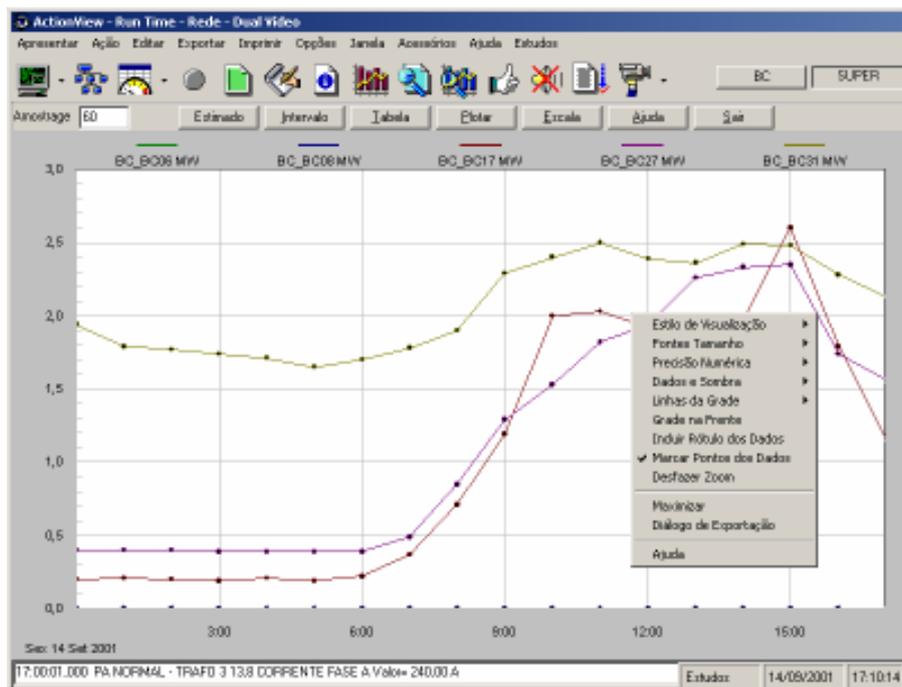


Figura 4.4 – Tela de tendência em tempo real [14].

- Telas de árvore do sistema: estas telas, Fig. 4.5, mostram o sistema em forma de árvore e, ao atingir-se o último nível, um duplo clique apresenta a janela de parâmetros da variável selecionada [14].

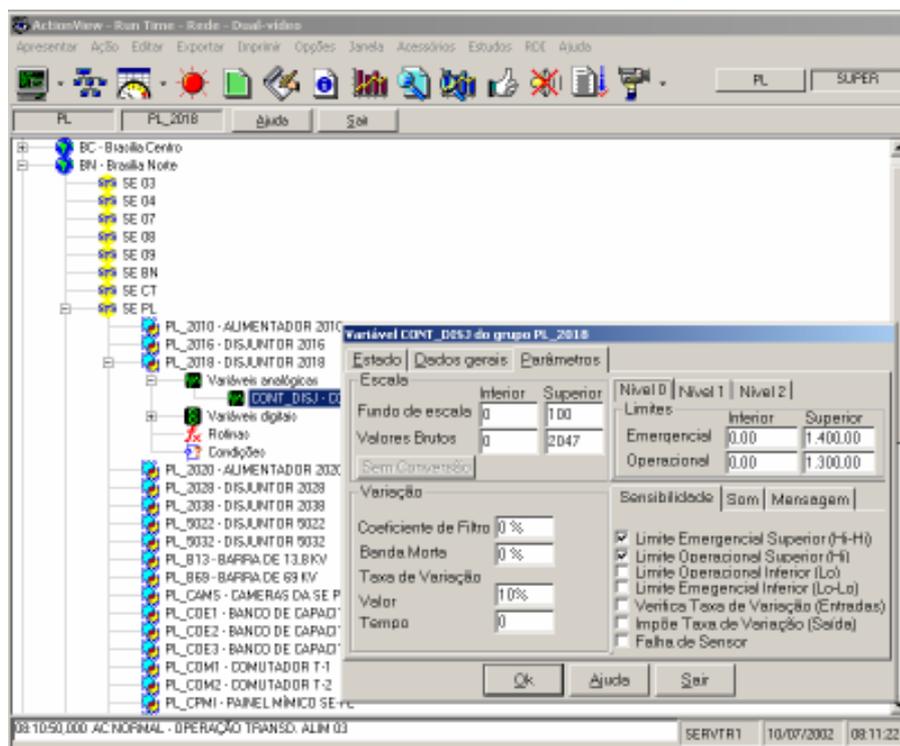


Figura 4.5 - Tela de árvore do sistema [14].

4.3.4. Módulos de comunicação

O AVServer pode se comunicar com diferentes equipamentos, em protocolo ponto-a-ponto ou multiponto. A comunicação é implementada através de uma biblioteca de sub-rotinas residente em memória (DLL), existindo uma para cada protocolo implementado. Este servidor de comunicação é multiprotocolo, o que permitindo ao usuário associar diferentes protocolos para diferentes pontos ou mesmo diferentes protocolos para um mesmo ponto [14].

A SPIN disponibiliza, por meio de seu site (www.spinengenharia.com.br), um manual focando todos os protocolos de comunicação que pode ser utilizado no software ActionView.

4.3.5. Módulo de desenvolvimento

O ActionView, além dos “scripts” permite o desenvolvimento de novas funcionalidades, ligadas em tempo real, através de bibliotecas do tipo DLL ou programas executáveis (*.EXE). Esta facilidade é utilizada SPIN, para desenvolver novas funções solicitadas pelos usuários, com características globais. Assim, métodos associados a objetos de visualização e funcionalidades do tipo: manobras, anotações, gráfico de tendência em tempo real de objetos de visualização analógicos, eventos de um objeto de visualização, etc. podem ser implementados [14].

Desta forma, o sistema em tempo real aproveita a arquitetura de DLLs do MS-Windows para a implementação de uma base de dados em memória e rotinas para a manipulação destes dados. Neste sentido, há dois tipos de DLLs: o primeiro com métodos e dados de tempo

real, e o segundo que estabelece o protocolo de comunicação entre o ActionView e o hardware de aquisição de dados. A seguir segue a descrição da biblioteca utilizada neste trabalho [17].

- AVRUNT50.DLL - Biblioteca principal que cria a base de dados em tempo real a partir da leitura da base de dados de parâmetros em MS-Access. Mantém os estados e valores atuais de pontos analógicos e digitais, incluindo todo seu status com respeito a alarmes, últimas atualizações, parâmetros para alarmes, etc. Recebe chamadas dos módulos de comunicação, tempo real e de aplicativos dos usuários. Assim como recebe chamadas das outras DLLs, que passam os dados recém lidos de UTRs para serem atualizados dentro da base de dados em tempo real.

A partir de [17], pode se estudar as diversas funções e selecionar duas funções para implementar a comunicação no projeto.

- GetVarRef: desenvolve um apontador para o objeto que define um ponto monitorado/comandado dentro da base de dados em memória, na AVRUNT50, a partir de sua identificação de grupo e identificação de variável por strings de caracteres.
- PutDataValue: altera o valor atual de uma variável em tempo real, não fazendo considerações quanto ao modo ou estado atual ou intertravamentos.

5. DESENVOLVIMENTO

O desenvolvimento deste projeto foi dividido em duas etapas, uma para configuração o software supervisor e outra para estabelecer a comunicação com os sensores e atuadores.

5.1. CONFIGURAÇÃO DO SUPERVISÓRIO

Para a configuração do ActionView, levou em consideração as etapas de planejamento descrito na seção 4.2 e também o padrão estabelecido em [4].

5.1.1. Entendimento do processo

O software supervisor será utilizado nas instalações do LAVSI, que possui área de 66 m², apresentada na Fig. 5.1. O laboratório dispõe de dois equipamentos de ar condicionado com a mesma configuração: tipo *split*, da marca Springer, modelo MAXIFLEX, com capacidade de 22000 BTUs/h. Estes equipamentos somente possibilitam ligar ou desligar o compressor ao longo do tempo de operação. Deste modo, não permitindo selecionar um nível de operação para manter o valor desejado da temperatura do ambiente [20].

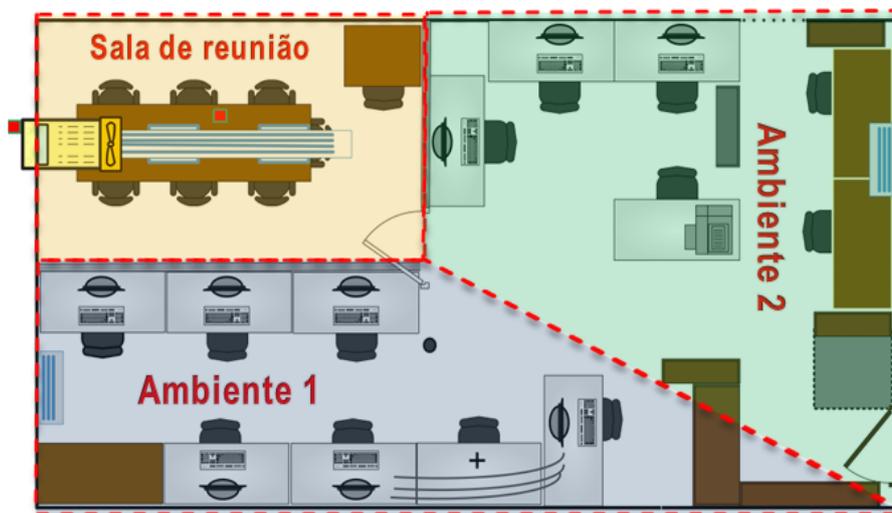


Figura 5.1 – Planta baixa do laboratório LAVSI.

É possível verificar que o LAVSI apresenta uma passagem para o laboratório LARA, que deve ser considerada afinal interfere bastante no sistema por favorecer a troca de calor entre os ambientes.

Dividiu-se o laboratório em três ambientes, sendo que entre os ambiente 1 e 2 não há nenhuma divisão física, sendo considerado somente para facilitar a metodologia do projeto. A sala de reunião possui outro sistema de ar condicionado e não será abordado neste trabalho.



Figura 5.2 – Disposição dos equipamentos na planta piloto [4].

A Figura 5.2, mostra os equipamentos, sensores e atuadores planejados para as instalações do laboratório no projeto PROMOVE, logo adaptando para o contexto do projeto, sugeriu a configuração da Fig. 5.3.

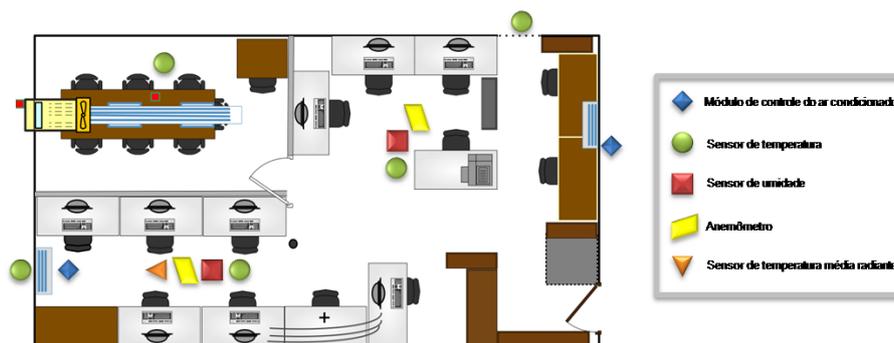


Figura 5.3 – Planejamento da disposição dos sensores e atuadores.

Neste trabalho, sugere-se o monitoramento das temperaturas internas e externas, a umidade relativa do ar, as velocidades do vento e a temperatura média radiante em cada ambiente, exceto na sala de reunião. Estas variáveis do processo serão medidas através de sensores e atuadores elaborados em projetos anteriores, a partir de todos os dados necessários, propõe o cálculo do índice de conforto térmico PMV em cada ambiente, implementando diferentes tipos de controladores.

No Anexo I, obtido através de [4], são mostradas as equações necessárias para o cálculo do PMV.

5.1.2. Planejamento da tomada de dados

Os módulos sensores e de atuação, já elaborados no LAVSI, utilizam o XBee para estabelecer a comunicação *wireless*. Considerou-se a implementação de uma rede sem fio

do tipo estrela, para simplificar esta tomada de dados, em que o nó coordenador ficaria no computador sendo utilizado pelo software supervisor. Com os XBee operando em modo transparente e utilizando a CON-USBBEE para emular uma porta COM virtual no computador e assim comunicando como o padrão RS-232. Após a configuração de todo o software, pode se comprovar que mesmo utilizando os módulos XBee em modo transparente e com a placa CON-USBBEE, é necessário um protocolo de comunicação compatível para estabelecer a comunicação com o *software* ActionView.

Portanto, para contornar este entrave na comunicação, foi proposto utilizar soluções alternativas que serão descritas na seção 5.2.

5.1.3. Planejamento do banco de dados

O ActionView utiliza por default, em aplicações mais simples, o Microsoft® Access para criar e acessar seu banco de dados. Este banco de dados conterà todas as informações pertinentes que foram configuradas no supervisor como listas de endereços das variáveis, tipos de variáveis, os protocolos disponíveis para comunicação, etc. Conterà as informações necessárias para que a aplicação seja executada.

Seguindo um padrão do software, a criação das variáveis torna a automática organização em grupos, correspondendo à finalidade de sua criação, e assim facilitando a compreensão da aplicação. A lista com todas as encontra se de forma detalhada no Anexo II deste trabalho.

5.1.4. Planejamento dos alarmes

Por se tratar de uma aplicação simples de monitoramento de algumas condições de conforto térmico, validação da comunicação dos módulos sem fio com o software e elaboração de estratégias de controle tomando como resposta o índice PMV, nenhum alarme será utilizado. Ao interagir o sistema de conforto térmico com outros sistemas, como o sistema de controle de acesso, será necessário a incorporação destes alarmes.

5.1.5. Planejamento de hierarquia de navegação entre telas

Seguindo o mesmo padrão estabelecido pelo trabalho [4], foram criadas as seguintes telas: configuração de parâmetros, parâmetros do controlador, índice PMV, temperatura ambiente das salas, umidade relativa do ambiente, velocidade do vento e posição dos sensores. A tela inicial do módulo *Run-time* será a tela do índice PMV, e de qualquer tela poderemos acessar às outras ou voltar para a tela inicial.

5.1.6. Desenho da telas

As telas foram criadas considerando a necessidade tanto de padronizar quanto possibilitar uma identidade visual a todos os sistemas de supervisão configurados por meio do laboratório, portanto este padrão estabelecido nas telas, através de alguns conceitos

importantes como as cores da logomarca da UnB, identifica os sistemas configurados pelo LAVSI. A tela mostrando a localização de todos os sensores é a mesma da Figura 5.3 e é visualizada provisoriamente ao selecionar a logo do LAVSI.

A tela inicial identificando o valor do índice PMV calculado em cada ambiente é mostrada na Fig. 5.4.

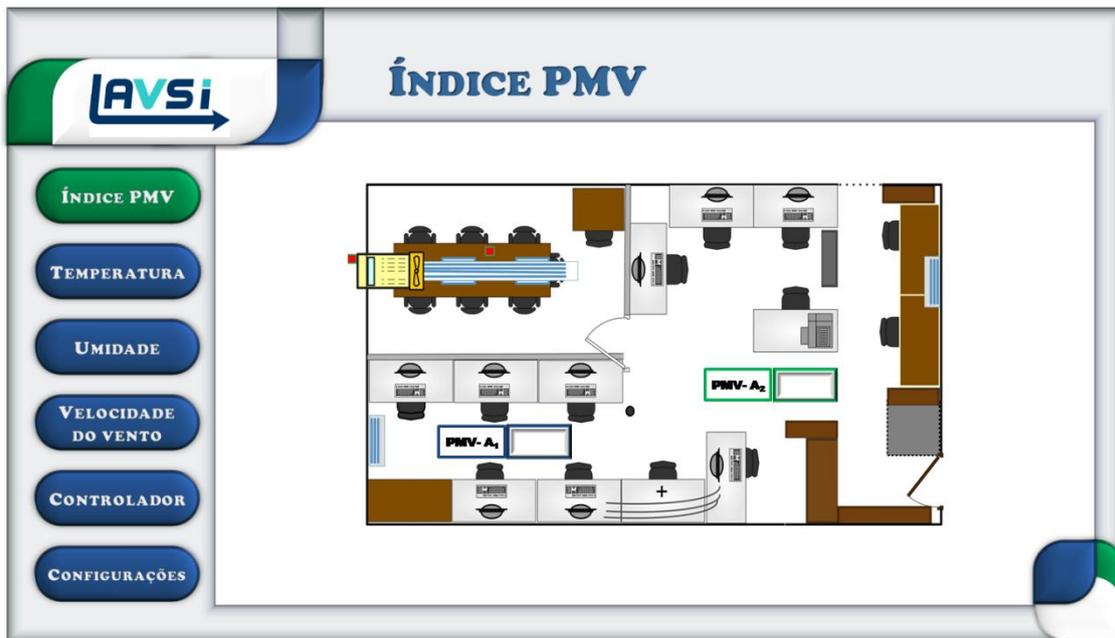


Figura 5.4 – Tela do índice PMV.

A tela de temperatura é mostrada na Figura. 5.5, e exibe os valores das temperaturas internas e externas dos ambientes dados em °C, incluindo também o da sala de reuniões que foi considerado como ambiente 3.

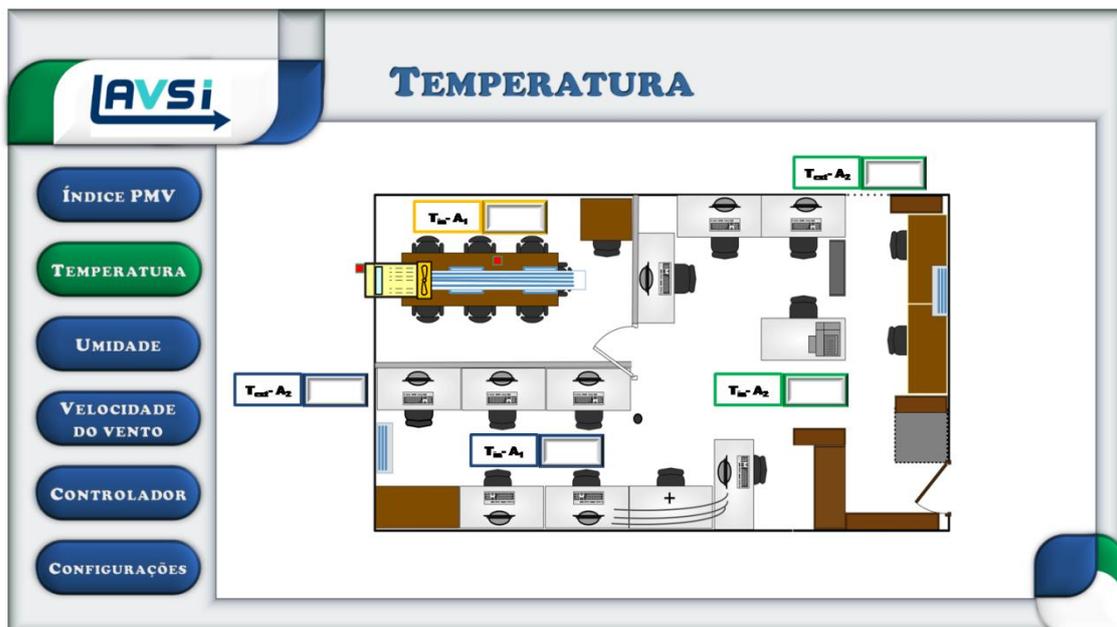


Figura 5.5 – Tela de temperatura.

A tela de umidade relativa, Figura 5.6, mostra os valores da umidade relativa dados em porcentagem.

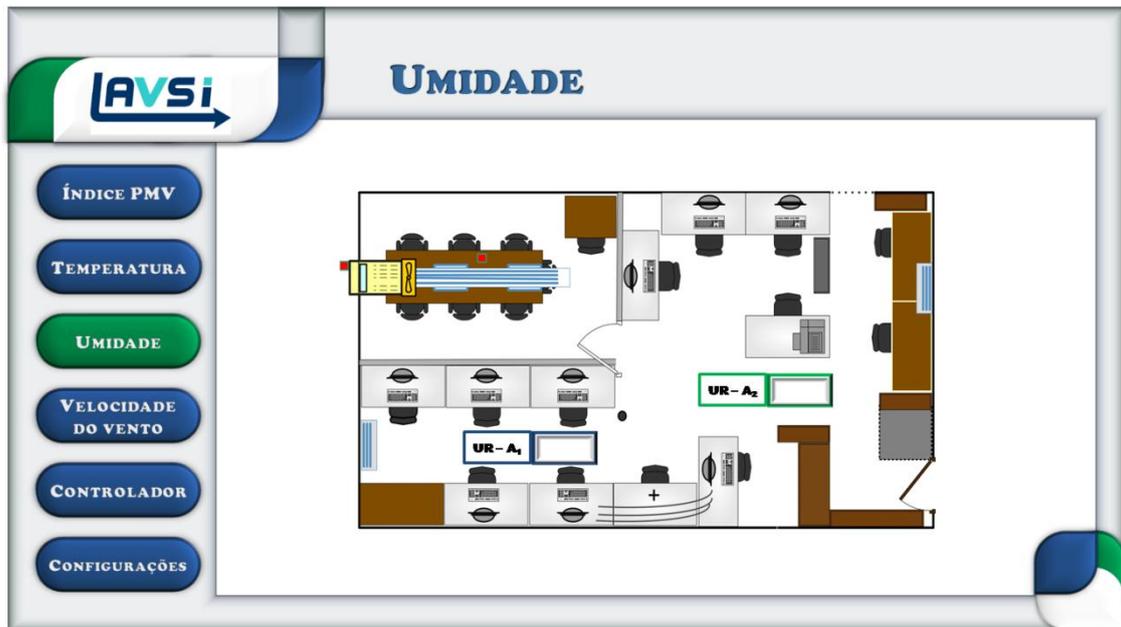


Figura 5.6 – Tela de umidade relativa do ar.

A Figura 5.7 mostra a tela de velocidade do vento, que apresenta os valores medidos por anemômetros e dados em m/s.

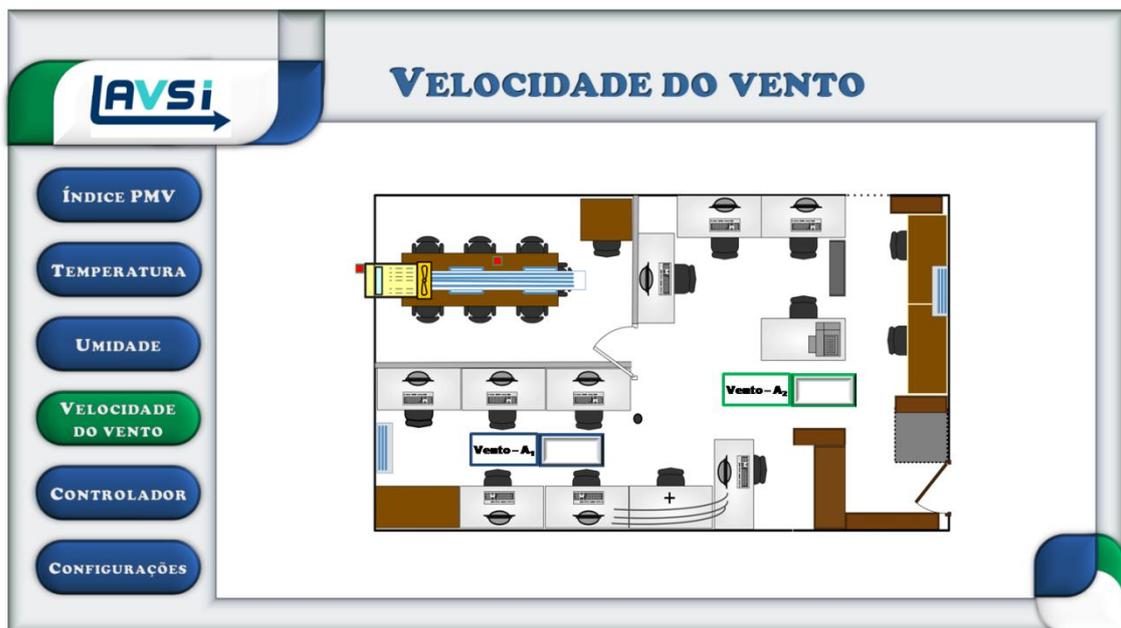


Figura 5.7– Tela de velocidade do vento.

A tela sobre o tipo de controle, Figura 5.8, mostra a seleção do tipo de controlador e também o status do controlador, se está ligado ou desligado.

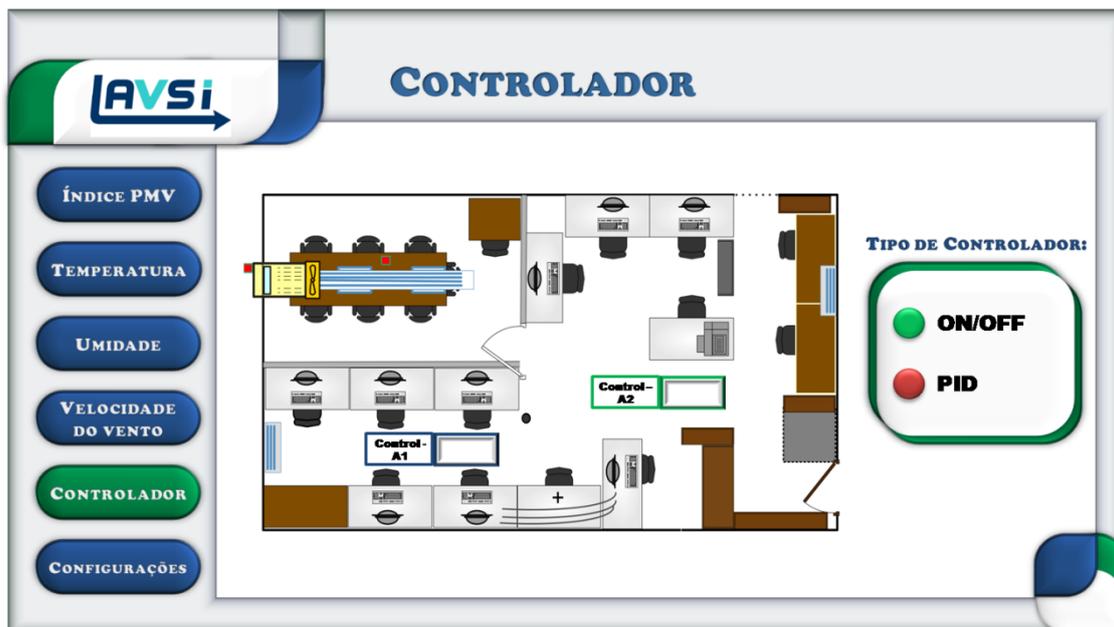


Figura 5.8 – Tela de parâmetros do controlador.

A tela de configuração de parâmetros é mostrada na Figura 5.9 e apresenta os campos para entrada dos dados sobre o tipo de vestimenta utilizado, o nível de atividade realizada, quantidade de pessoas e a temperatura média radiante (dada em °C) de cada ambiente.

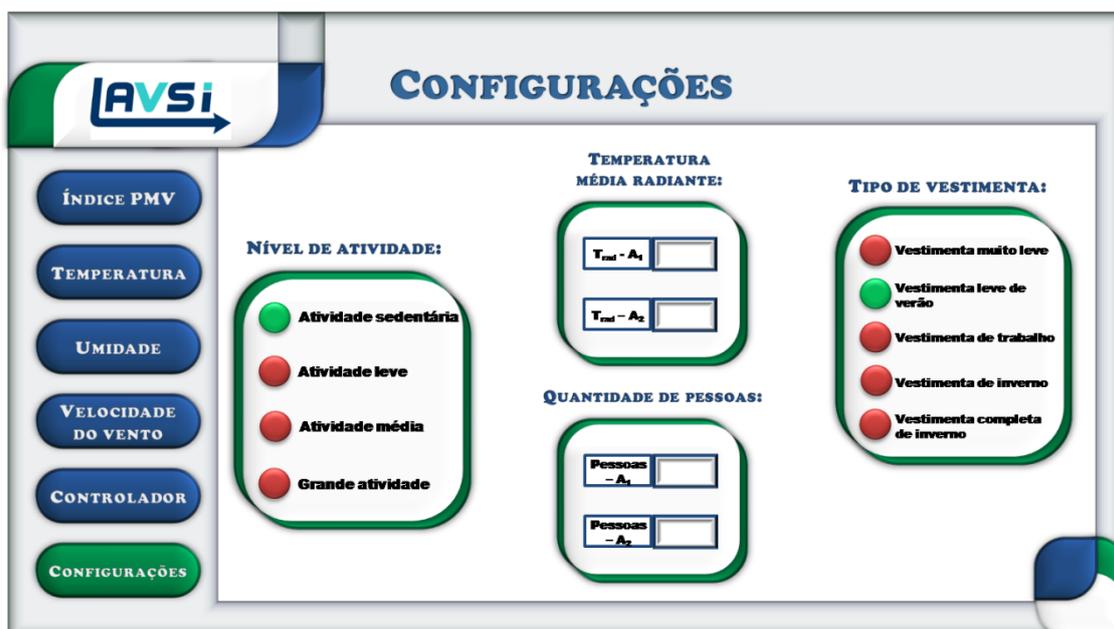


Figura 5.9 – Tela de configuração de parâmetros.

5.1.7. Gráficos de tendências

Os gráficos de tendências, que mostram como determinadas variáveis do processo mudam com o decorrer do tempo, poderão ser exibidos durante a navegação pelo processo em tempo real [4]. Neste projeto, não foi criada nenhuma tela ou atalho para visualizar estes gráficos afinal no modo *run-time*, é possível selecionar o intervalo de tempo que se deseja plotar os valores de cada variável medida.

5.1.8. Segurança

A configuração do ActionView para o LAVSI, neste projeto, é voltada a visualizar e supervisionar os dados dos parâmetros disponíveis para proporcionar o conforto térmico e assim não é necessário nem cadastrar vários operadores nem planejar restrições para utilizar o supervisor.

5.1.9. Padrão industrial de desenvolvimento

O ActionView é um supervisor que atende esta especificação por ser comercialmente utilizado a vários anos. Para também atender essa especificação, configurou-se o software em suas telas para ser o mais simples e próximo a um ambiente como o do Windows.

5.2. COMUNICAÇÃO DO SUPERVISÓRIO

O projeto da configuração do supervisor é bastante acessível após o estudo dos manuais [14],[15] e [16] disponíveis pela empresa SPIN (www.spinengenharia.com.br), e também é facilitado ao seguir os nove passos de planejamento descrito na seção 4.2 deste trabalho. Entretanto, houve um erro conceitual em não se diferenciar o padrão de comunicação RS-232 de um protocolo de comunicação para redes sem fio.

Todo supervisor necessita de um protocolo para gerenciar as informações recebidas pelos módulos sensores e atuadores, logo verificou-se na seção 5.1.2, após toda a configuração do *software* supervisor, que a comunicação com este tipo de *software* necessita de um protocolo, e em se tratando de comunicação *wireless* essencial e ainda em desenvolvimento para diversos protocolos.

Para solucionar esse entrave na comunicação, juntamente com a empresa SPIN, foi proposto três alternativas para a comunicação.

5.2.1. Elaboração de protocolo simplificado

Esta primeira solução consistiu em selecionar um protocolo padrão do ActionView que fosse bastante consistente, para ser simplificado e assim selecionou-se o MODBUS. Dessa forma, teria de entender e simplificar este protocolo, programar algumas funções específicas para atender a comunicação com módulos XBee e realizar uma série de testes de confiabilidade, e fim de garantir a da informação lida pois se não houvesse um tratamento devido da informação lida o ActionView poderia travar e parar o gerenciamento do processo. Esta solução foi descartada para este projeto por necessitar de bastante tempo para a comprovação da eficácia deste protocolo simplificado.

5.2.2. Utilização de protocolo BACnet criado no LAVSI

Esta solução permeia a utilização dos módulos de sensores *wireless* elaborados durante um projeto de graduação no LAVSI, o qual atende os requisitos do protocolo BACnet. De acordo com a SPIN, este protocolo está em fase de desenvolvimento no ActionView e ainda não está bem consolidado. Logo, não é possível a utilização devido a possibilidade de acarretar erros de comunicação e assim gerar falhas no supervisório.

5.2.3. Processamento paralelo de programas em tempo real

Esta proposta iniciou com a possibilidade de executar paralelamente programas elaborados através de outros projetos do LAVSI com o ActionView. Estes programas paralelos teriam a função de tratamento, gerenciamento e envio das informações lidas dos módulos atuadores e sensores ao ActionView e assim proporcionando a comunicação.

Segundo os manuais do supervisório, esta seria uma aplicação mais complexa devido colocar em funcionamento o supervisório com programas paralelos em tempo real, acarretando em restrições a utilização do banco de dados padrão. Para possibilitar solução deste entrave, verificou-se as etapas de migração do Access para o SQL, contudo a SPIN aconselhou continuar utilizando o banco de dados padrão do ActionView devido a sua simplicidade na manipulação dos dados e também da pequena quantidade de variáveis medidas.

O primeiro programa selecionado foi o elaborado no projeto descrito em [20] que propõe um supervisório para redes utilizando módulos MeshBean, a escolha desse projeto é acarretada ao fato de disponibilizar uma grande quantidade de tipos de sensores em perfeito funcionamento para o conforto térmico. O software supervisório deste trabalho foi elaborado em linguagem Visual Basic e a sua IHM é apresentada na Fig. 5.10. Durante as alterações deste programa para enviar o valor lido do sensor a cada variável, considerando que este programa gerenciaria a escrita no banco de dados do ActionView, houveram alguns erros ao acessar o banco de dados afinal alguns campos de determinadas tabelas tinham restrições e faziam um relacionamento entre outras tabelas do banco de dados, impossibilitando de serem acessados ou ocasionando erros. Devido a esta dificuldade em acessar diretamente o banco de dados por meio do programa auxiliar, a empresa SPIN sugeriu uma solução que já é utilizada pela empresa e é mais robusta.

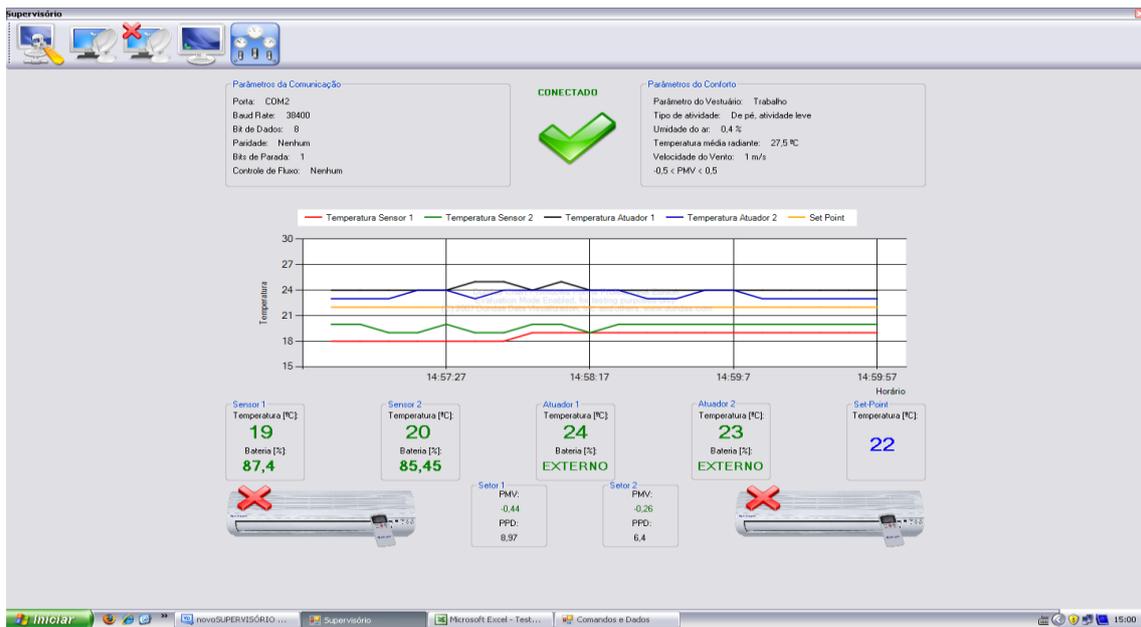


Figura 5.10 – Tela do supervisório elaborado em [20].

Esta solução é voltada a incorporação de bibliotecas, ligadores e funções que proporcionem o programa externo utilizar a DLL AVRUNT50, a qual é a principal ferramenta para criar a base de dados em tempo real a partir da leitura da base de dados de parâmetros em MS-Access. No manual apresentado em [17], várias funções são descritas para estas soluções alternativas, as quais tem de elaborar um programa auxiliar para propiciar o funcionamento ou comunicação do software ActionView com o sistema

Considerando esta solução alternativa, selecionou duas funções para coletar e enviar os dados ao ActionView. Ao aplicar as alterações, diagnosticou uma incompatibilidade na linguagem dos programas, afinal o software realizando no LAVSI utiliza VB dotnet que é a última versão da linguagem Visual Basic e a biblioteca AvrunAPI.bas é escrita em VB6. Logo, verificou-se, para a utilização deste programa supervisório, a necessidade de atualizar o código da biblioteca do ActionView e através desta atualização poderia haver alguns erros devido a interligação desta biblioteca com outras do programa.

Deste modo, foi selecionado outro supervisório elaborado também no LAVSI e disponibilizado em [11], que apresenta o código em linguagem de programação C e é bem mais simples de modificações. Este programa somente gerencia dois módulos sensores e dois atuadores, com a possibilidade de expansão para outros módulos de sensores. A lógica e o código deste programa é apresentado no Anexo III.

Para o funcionamento da biblioteca, que permite a comunicação do programa com a DLL, foi necessário migrar o código do programa utilizado de plataforma, afinal o código do supervisório proposto em [11] utiliza o DevC++, que tem como compilador o GCC, e na SPIN é utilizado *Microsoft Visual Studio*.

Após todas as alterações necessárias e os vários testes para a comprovação e consolidação da comunicação entre esse programa, o programa sempre apresentou erros na compilação . Os erros de compilação devem se ao fato da migração do programa para o

Visual Studio e modificando o programa para ser utilizado no DevC++, não foi possível estabelecer o link afinal a biblioteca não está em uma extensão apropriada para o GCC.

Mesmo após diversas tentativas, não se obteve sucesso na comunicação do projeto com os módulos, e pode se constatar que o projeto conclui sem apresentar o valor dos parâmetros de conforto térmico no ActionView.

Portanto, este projeto se tornou importante em proporcionar tanto o conhecimento da teoria dos tópicos referentes a ambientes inteligentes, conforto térmico, rede de sensores, sistemas supervisórios necessários para ser elaborado, quanto o alcance a temas de diversas áreas relacionadas, como as vários modos de se resolver alguns obstáculos na comunicação de supervisórios com módulos de aquisição de dados, incompatibilidade de diferentes versões da mesma linguagem, limitações dos protocolos a módulos wireless, dentre outros. Constata se que o projeto proporcionou com as diversas dificuldades, um ganho de conhecimento e experiência na solução dos problemas apresentados.

6. CONCLUSÃO

A solução proposta pelo projeto de interagir dois programas, ActionView e o programa criado em [11] com as alterações mencionadas no capítulo anterior, funcionando paralelamente em tempo real, não obteve sucesso devido erros na execução de ambos programas acarretando falhas de comunicação entre o programa adaptado e a DLL do programa ActionView.

Ainda se tivesse apresentado sucesso nesta interação, e completando parcialmente o objetivo do projeto ao mostrar o valor dos sensores e atuadores através do software supervisorio, limitaria a utilização do programa criado pela SPIN. Esta limitação deve se a fato de que modo run-time do supervisorio, exerceria somente a função de visualização ou supervisão das variáveis medidas, ou seja, o supervisorio atuaria somente como uma IHM mais amigável ao operador. Assim, não seria possível nem atuar, nem controlar o processo, e considerando pela definição de sistemas supervisorios que estas duas características são necessárias para que seja essencial a utilização de um supervisorio, afinal o custo de implantação destes sistemas é bastante elevado e tem de ter algum retorno do ponto de vista.

Portanto, este trabalho possibilitou além do aprendizado sobre os conceitos e metodologias referentes a sistemas supervisorios, redes de sensores sem fio, conforto térmico e, conseqüentemente, ambientes inteligentes, verificar a limitação na utilização de equipamentos wireless, os quais necessitam de protocolos completamente desenvolvidos, em sistemas de automação predial como o *software* supervisorio.

Dessa forma, propõe se que seja desenvolvido (ou aprimorado) e consolidado em parceria com a SPIN um protocolo para o ActionView, por exemplo o BACnet que proporciona a interligação dos diversos sensores e atuadores, afinal é específico para aplicações em automação e controle predial, ou até mesmo o ZigBee o qual permitiria a utilização de módulos como o XBee em modo transparente.

Visando a padronização dos projetos realizados no GRAV relacionados a automação predial sugere se, conforme descrito no projeto apresentado em [13], onde se define um sistema de automação predial como um sistema híbrido, por ter uma combinação de componentes discretos orientados a eventos e processos térmicos distribuídos contínuos no tempo, um modelamento dos eventos que compõem os sistemas do *Ambient Intelligence* proposto, através de ferramentas como redes de Petri. Através da simulação deste modelamento, tornaria mais simples o desenvolvimento e interação de todos os sistemas que proporcionam conforto neste ambientes, prevenindo falhas devido a implementações empíricas. A evidência da importância deste modelamento pode ser vista em [10] e [26].

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] LAMBERTS, R, XAVIER, A. A. P. **Conforto Térmico e Stress Térmico. Laboratório de Eficiência Energética em Edificações**, Universidade Federal de Santa Catarina, Florianópolis/SC, 2002. 111pp.
- [2] COUTO, F. L., FIGUEREDO, L. F. C., (2008). **Medição Móvel de Conforto Térmico para Rede de Automação Predial Wireless**. Trabalho de Graduação em Engenharia de Controle e Automação, Publicação FT. TG-nº 011/2008, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 85pp.
- [3] MORAES, C. C. de., CASTRUCCI, P. de L. **Engenharia de Automação Industrial**. Rio de Janeiro/RJ: Editora LTC, 2ª Ed. 2001. 295 pp.
- [4] GALLO, E. A., RIBEIRO, F. N., (2007). **Índice de conforto térmico ISO 7730 em Automação Predial**. Trabalho de Graduação em Engenharia de Controle e Automação, Publicação FT.TG-nº, Faculdade de Tecnologia, Universidade de Brasília, Brasília/DF, 98pp.
- [5] SOUZA, R. B. de., (2005) **Uma Arquitetura para Sistemas Supervisórios Industriais e sua Aplicação em Processos de Elevação Artificial de Petróleo**. Dissertação (Mestrado em Engenharia Elétrica). Universidade Federal do Rio Grande do Norte. Natal/RN, 53pp.
- [6] SILVA, A. P. G. da., SALVADOR, M. **O que são sistemas supervisórios?** 2005. Disponível em <<http://www.centralmat.com.br/Artigos/Mais/oQueSaoSistemasSupervisorios.pdf>> Acesso em: 11 de janeiro de 2010.
- [7] PIRES, P. S. M., OLIVEIRA, L. A. H. G. de., BARROS, D. N.(2004) **Aspectos de segurança em sistemas SCADA – Uma visão geral**. Publicado na revista Controle & Instrumentação – Edição nº 104 – Maio de 2005. Disponível em <ftp://users.dca.ufrn.br/artigos/2004/isa_scada_2004.pdf>. Acesso em: 11 de janeiro de 2010
- [8] MARKOV, D. **Standards in thermal comfort**. In: **ANNUAL INTERNATIONAL COURSE: VENTILATION AND INDOOR CLIMATE**, Technical University of Sofia. Ed. P. Stankov . 2002. pp. 147 – 157. ISBN 954-9782-27-1. Disponível em <<http://www.cfdc.tu-sofia.bg/publications/lec18Markov.pdf>> Acesso em: 03 de agosto de 2008.
- [9] MARKOV, D., **Practical evaluation of the thermal comfort parameters, Annual International Course: Ventilation and Indoor climate**, Technical University of Sofia. Ed. P.

Stankov. 2002. pp. 158 – 170. ISBN 954-9782-27-1. Disponível em <<http://www.cfdc.tu-sofia.bg/publications/lec19Markov.pdf>>. Acesso em: 03 de agosto de 2008.

[10] VILLANI, E.(2000). **Abordagem Híbrida para Modelagem de Sistemas de Ar Condicionado em Edifícios Inteligentes..** Dissertação (Mestrado em Engenharia Mecânica). Escola Politécnica da Universidade de São Paulo. São Paulo/SP. 168pp.

[11] FILHO, P. R. M., DIAS, Y. F. G., (2008). **Acionamento de potência para rede de automação wireless.** Trabalho de graduação em Engenharia de Controle e Automação, Publicação FT.TG-nº 012, Faculdade de Tecnologia, Universidade de Brasília, Brasília/DF,67pp.

[12] BATISTA, R. S.,(2007) **Sistema de Monitoramento do Bombeamento de Líquidos controlados entre bacias..** Trabalho de graduação em Engenharia de Controle e Automação. Universidade Federal de Ouro Preto. Ouro Preto/MG.74pp.

[13] BAUCHSPIESS, A. **Rede de automação predial distribuída para racionalização de energia segundo o paradigma *Ambient Intelligence*.** IN: Edital CT-Energ/MCT/CNPq nº 28/2006. Brasília, 2006.

[14] SPIN Engenharia de Automação. **ActionView – Especificação Funcional.** Brasília/DF, março de 2008. Disponível em <<http://www.spinengenharia.com.br/downloads.aspx>> Acesso em 10 de janeiro de 2010.

[15] SPIN Engenharia de Automação. **ActionView – Introdução.** Brasília/DF, março de 2008. Disponível em <<http://www.spinengenharia.com.br/downloads.aspx>> Acesso em 10 de janeiro de 2010.

[16] SPIN Engenharia de Automação. **ActionView – Módulo configurador.** Brasília/DF, março de 2008. Disponível em <<http://www.spinengenharia.com.br/downloads.aspx>> Acesso em 10 de janeiro de 2010.

[17] SPIN Engenharia de Automação. **ActionView – Desenvolvimento de Aplicativos.** Brasília/DF, outubro de 2003.

[18] ÁGUAS, M. P. N. **Conforto térmico.** IN: **Módulo da Disciplina de Mestrado Métodos Instrumentais em Energia e Ambiente.** Instituto Superior Técnico. Lisboa, 2000/2001. 25 pp.

[19] URZÊDA, C. C. de.,(2006) **Software SCADA como plataforma para racionalização inteligente de energia elétrica em automação predial.** Tese de mestrado em Engenharia Elétrica, Publicação, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF. 91 pp.

[20] QUEIROZ, R. B. de., AZEVEDO, R. C. de A., (2009). **Rede de sensores sem fio para automação predial com módulos Meshbean .** Trabalho de Graduação em Engenharia de Elétrica, Faculdade de Tecnologia, Universidade de Brasília, Brasília/DF, 81pp.

[21] XAVIER, A. A. P., LAMBERTS, R., VOLPATO, C.G.,(2000) **Relacionamento entre sensações térmicas e percentagem de pessoas insatisfeitas com o ambiente em**

estudos de campo, ENTAC 2000 – VIII Encontro Nacional de Tecnologia do Ambiente Construído – Modernidade e Sustentabilidade, Salvador/BA. Laboratório de eficiência energética em edificações. Universidade Federal de Santa Catarina, Florianópolis/SC, 8pp.

[22] MESSIAS, A. R. **Controle remoto e aquisição de dados via XBee/ZigBee (IEEE 802.15.4)**. Disponível em <<http://www.rogercom.com/ZigBee/ZigBee.htm>> Acesso em: 29 de outubro de 2009.

[23] DIGI International. **XBee®/XBee-PRO® DigiMesh™ 2.4 OEM RF Modules**. Disponível em <http://ftp1.digi.com/support/documentation/90000991_a.pdf> Acesso em: 29 de outubro de 2009.

[24] CUNHA, P. S., CARVALHO, L. O. M. de, **Seminário de redes de computadores II – 802.15.4 e ZigBee**. Disponível em <<http://www.gta.ufrj.br/~rezende/cursos/eel879/trabalhos/zigbee/index.html>> Acesso em : 29 de outubro de 2009.

[25] MALAFAYA, H.; T., L., SOUSA, J. P.(2006). **Sensorização sem fios sobre ZigBee e IEEE 802.15.4**. IN: **Terceiras Jornadas de Engenharia de Electrónica e Telecomunicações e de Computadores**. Departamento de Engenharia Electrotécnica e de Computadores. Faculdade de Engenharia da Universidade do Porto. Lisboa/Portugal. 5pp

[26] CURZEL, J. L., (2008) **Síntese e implementação de controle supervisorio em uma célula flexível de manufatura didática**. Tese de Mestrado em Engenharia Elétrica. Departamento de Engenharia Elétrica, Universidade do Estado de Santa Catarina, Joinville/SC. 106pp.

ANEXOS

Anexo I – Cálculo dos termos da equação do PMV [4].

Anexo II – Lista de variáveis e scripts utilizados no ActionView

Anexo III – Software supervisorio [11].

Anexo IV – Roteiro para o trabalho de conforto térmico [1].

Anexo V – Descrição do conteúdo do cd

ANEXO I

Este anexo apresenta as fórmulas para o cálculo dos termos individuais da equação do índice de conforto térmico PMV, elaborado por [4] através de [1] e [18]

AI.1 CÁLCULO DA TAXA METABÓLICA (T_{met})

A taxa metabólica M é um fator que varia de acordo com o nível de atividade física que um indivíduo realiza. Quanto maior a atividade física, maior será a taxa metabólica. A Tabela AI.1 apresenta alguns valores para a taxa metabólica em função do nível de atividade física realizada:

Tabela AI.1 – Taxas metabólicas segundo a norma ISO 7730 [18].

ATIVIDADES	TAXAS METABÓLICAS	
	[W/m ²]	[met]
Atividade sedentária (escola, residência, escritório, laboratório)	70	1,2
Atividade leve em pé (compras, laboratório, indústria leve)	93	1,6
Atividade média em pé (balconista, trabalho doméstico, trabalho em máquinas)	117	2,0
Atividade intensa	175	3,0

Observando a Tabela AI.1, podemos perceber que a taxa metabólica também pode ser expressa em termos da unidade [met]. Além disso, fazendo a proporção, concluímos que 1 met é igual a 58 W/m².

AI.2 CÁLCULO DO CALOR PERDIDO POR DIFUSÃO (E_{dif})

O calor perdido por difusão de vapor de água através da pele E_{dif} é calculado através da Eq. AI.1 dada a seguir:

$$E_{dif} = 3,05 * (5,73 - 0,007 * T_{met} - p_a) \quad (AI.1)$$

Na Equação AI.1, temos a presença da pressão de vapor p_a , cujo cálculo pode ser encontrado a seguir.

AI.2.1 Cálculo da pressão de vapor (p_a)

A pressão de vapor p_a pode ser calculada através da Eq. AI.2 que se encontra a seguir:

$$p_a = UR * p_{sat} \quad (AI.2)$$

Na Equação AI.2, UR é a umidade relativa do ar, que pode ser medida através de sensores de umidade conhecidos como higrômetros. O termo p_{sat} pode ser calculado através da fórmula encontrada na seqüência.

AI.2.2 Cálculo da pressão de saturação (p_{sat})

A pressão de saturação p_{sat} deve ser calculada da seguinte maneira:

$$p_{sat} = e^{\frac{17,269 * T_{ar}}{237,3 + T_{ar}}} \quad (AI.3)$$

Na Equação AI.3, T_{ar} é a temperatura do ar dada em [°C], a qual pode ser medida através de diversos tipos de sensores.

AI.3 CÁLCULO DO CALOR PERDIDO POR TRANSPIRAÇÃO (E_{trans})

O calor perdido por transpiração E_{trans} é calculado da seguinte maneira:

$$E_{trans} = 0,42 * (T_{met} - 58,15) \quad (AI.4)$$

AI.4 CÁLCULO DO CALOR PERDIDO POR RESPIRAÇÃO LATENTE (E_{rl})

O calor perdido por respiração latente E_{rl} é dado pela Eq. AI.5 a seguir:

$$E_{rl} = 0,0173 * T_{met} * (5,87 - p_a) \quad (AI.5)$$

AI.5 CÁLCULO DO CALOR PERDIDO POR RESPIRAÇÃO SENSÍVEL (E_{rs})

O calor perdido por respiração sensível E_{rs} pode ser obtido através da seguinte equação:

$$E_{rl} = 0,0014 * T_{met} * (34 - T_{ar}) \quad (AI.6)$$

AI.6 CÁLCULO DO CALOR PERDIDO POR RADIAÇÃO (R)

O calor perdido por radiação R é calculado por meio da Eq. AI.6 dada a seguir:

$$R = 3,96 * 10^{-8} * f_{vest} [(T_{vest} + 273)^4 - (T_{rad} + 273)^4] \quad (AI.7)$$

Na Equação AI.7, f_{vest} é o fator de vestuário dado em [m^2K/W], T_{vest} é a temperatura da vestimenta dada em [°C] e T_{rad} é a temperatura radiante média dada em [°C]. Os cálculos destes termos individualmente podem ser encontrados na seqüência.

AI.6.1 Cálculo do fator de vestuário (f_{vest})

O fator de vestuário f_{vest} é uma função com definições diferentes para dois intervalos distintos de sua variável independente. É calculado como se segue:

$$\begin{cases} f_{vest} = 1,00 + 1,29 * I_{vest} , para I_{vest} < 0,078 m^2K/W \\ f_{vest} = 1,05 + 0,645 * I_{vest} , para I_{vest} \geq 0,078 m^2K/W \end{cases} \quad (AI.8)$$

Na Equação AI.8, I_{vest} é a resistência térmica da vestimenta dada em [m^2K/W]. O valor de I_{vest} pode ser encontrado em tabelas como a que se encontra a seguir.

AI.6.2 Cálculo da resistência térmica da vestimenta (I_{vest})

A resistência térmica da vestimenta I_{vest} varia de acordo com o tipo de roupa que uma pessoa está usando. Quanto mais peças de roupa, maior será a resistência térmica da vestimenta. A Tabela AI.2 apresenta alguns valores de I_{vest} em função do tipo de vestimenta utilizado.

Tabela Al.2 – Resistência térmica da vestimenta [18].

VESTIMENTA	RESISTÊNCIA TÉRMICA	
	[m ² K/W]	[clo]
Vestuário tropical	0,047	0,3
Vestuário leve de verão	0,078	0,5
Vestuário de trabalho	0,124	0,7
Vestuário de inverno para ambientes internos	0,155	1,0
Vestuário completo	0,233	1,5

Observando a Tabela Al.2, podemos perceber que a resistência térmica da vestimenta também pode ser expressa em termos da unidade [clo]. Além disso, vemos que 1 clo é igual a 0,155 m²K/W.

Al.6.3 Cálculo da temperatura da vestimenta (T_{vest})

A temperatura da vestimenta T_{vest} é calculada através da Eq. Al.9 dada a seguir:

$$T_{vest} = 35,7 - 0,0275 * T_{met} - 0,155 * I_{vest} * (R + C) \quad (Al.9)$$

Al.7 CÁLCULO DO CALOR PERDIDO POR CONVECÇÃO (C)

O calor perdido por convecção C deve ser calculado da seguinte maneira:

$$C = 12,1 * f_{vest} * (T_{vest} - T_{ar}) * \sqrt{v} \quad (Al.10)$$

Na Equação Al.10, v é a velocidade do vento dada em [m/s]

Al.8 ANÁLISE DE RECURSIVIDADE

Observando as Eqs Al.7, Al.9 e Al.10, podemos notar que elas possuem recursividade entre si, em termos das variáveis T_{vest} , R e C. Uma maneira de se contornar este problema (maneira esta que foi usada neste projeto) é definir um valor inicial para uma das variáveis (no nosso caso, T_{vest}) e fazer iterações de cálculo das variáveis, sempre atualizando os valores das variáveis ao realizar novas iterações.

ANEXO II

Este anexo apresenta a lista de todas as variáveis criadas no ActionView e os scripts utilizados.

AIII.1 VARIÁVEIS UTILIZADAS

A Tabela All.1 mostra as variáveis utilizadas neste projeto, sendo que o campo, “Unidade” apresenta a unidade de medida da variável, o campo “Tipo” informa se a variável é de entrada (I), saída (O) ou interna (U), e o campo “A/D/M” informa se a variável é analógica (A), digital (D) ou digital múltipla(M).

Tabela All.1 – Variáveis utilizadas

Grupo	Variável	Descrição	Unidade	A/D/M	Tipo
ANEMOMETROS	ANEM_1	Anemômetro - Ambiente 1	m/s	A	I
ANEMOMETROS	ANEM_2	Anemômetro - Ambiente 2	m/s	A	I
APOIO	C_1	Convecção - Ambiente 1	W/m ²	A	I
APOIO	C_2	Convecção - Ambiente 2	W/m ²	A	I
APOIO	D_VAP_1	Difusão de vapor - Ambiente 1	W/m ²	A	I
APOIO	D_VAP_2	Difusão de vapor - Ambiente 2	W/m ²	A	I
APOIO	F_VEST	Fator de vestuário	m ² K/W	A	I
APOIO	L_1	Acumulação de calor - Ambiente 1	W/m ²	A	I
APOIO	L_2	Acumulação de calor - Ambiente 2	W/m ²	A	I
APOIO	PA_1	Pressão de vapor - Ambiente 1	kPa	A	I
APOIO	PA_2	Pressão de vapor - Ambiente 2	kPa	A	I
APOIO	PMV_1	Índice PMV - Ambiente 1	-	A	I
APOIO	PMV_1S	Índice PMV - Ambiente 1 - OUT	-	A	O
APOIO	PMV_2	Índice PMV - Ambiente 2	-	A	I
APOIO	PMV_2S	Índice PMV - Ambiente 2 - OUT	-	A	O
APOIO	PSAT_1	Pressão de saturação - Ambiente 1	kPa	A	I

APOIO	PSAT_2	Pressão de saturação - Ambiente 2	kPa	A	I
APOIO	RAD_1	Radiação - Ambiente 1	W/m ²	A	I
APOIO	RAD_2	Radiação - Ambiente 2	W/m ²	A	I
APOIO	RL_1	Respiração latente - Ambiente 1	W/m ²	A	I
APOIO	RL_2	Respiração latente - Ambiente 2	W/m ²	A	I
APOIO	RS_1	Respiração sensível - Ambiente 1	W/m ²	A	I
APOIO	RS_2	Respiração sensível - Ambiente 2	W/m ²	A	I
APOIO	TRANSP	Transpiração	W/m ²	A	I
APOIO	TVEST_1	Temperatura da vestimenta - Ambiente 1	°C	A	I
APOIO	TVEST_2	Temperatura da vestimenta - Ambiente 2	°C	A	I
APOIO	X	Variável de apoio - contador	-	A	U
PARAMETROS	ACT_1	Atividade sedentária	W/m ²	M	U
PARAMETROS	ACT_2	Atividade leve	W/m ²	M	U
PARAMETROS	ACT_3	Atividade média	W/m ²	M	U
PARAMETROS	ACT_4	Grande atividade	W/m ²	M	U
PARAMETROS	I_VEST	Resistência térmica da vestimenta	m ² K/W	A	U
PARAMETROS	M	Metabolismo	W/m ²	A	U
PARAMETROS	T_RAD_1	Temperatura média radiante - Ambiente 1	°C	A	I
PARAMETROS	T_RAD_2	Temperatura média radiante - Ambiente 2	°C	A	U
PARAMETROS	TR_A1_P	Número de pessoas - Ambiente 1	-	A	U
PARAMETROS	TR_A2_P	Número de pessoas - Ambiente 2	-	A	U
PARAMETROS	VEST_1	Vestimenta muito leve	m ² K/W	M	U
PARAMETROS	VEST_2	Vestimenta leve de verão	m ² K/W	M	U
PARAMETROS	VEST_3	Vestimenta de trabalho	m ² K/W	M	U
PARAMETROS	VEST_4	Vestimenta de inverno	m ² K/W	M	U

PARAMETROS	VEST_5	Vestimenta completa de inverno	m ² K/W	M	U
PID	AC_1	Estado do ar condicionado - Ambiente 1	-	A	U
PID	AC_2	Estado do ar condicionado - Ambiente 2	-	A	U
PID	AC1	Estado do ar condicionado - Ambiente 1	-	D	I
PID	AC2	Estado do ar condicionado - Ambiente 2	-	D	I
TEMPERATURA	T_EXT_1	Temperatura externa - Ambiente 1	°C	A	I
TEMPERATURA	T_EXT_2	Temperatura externa - Ambiente 2	°C	A	I
TEMPERATURA	T_INT_1	Temperatura interna - Ambiente 1	°C	A	I
TEMPERATURA	T_INT_2	Temperatura interna - Ambiente 2	°C	A	I
TEMPERATURA	T_INT_3	Temperatura interna - Ambiente 3	°C	A	I
UMIDADE	UR_1	Umidade relativa - Ambiente 1	%	A	I
UMIDADE	UR_2	Umidade relativa - Ambiente 2	%	A	I

AI.2 SCRIPTS UTILIZADOS

Baseado no projeto desenvolvido em [4], utilizou-se para o cálculo e a execução do ActionView dez scripts, sendo um do tipo OnTimer, executado periodicamente e os outros nove do tipo OnMouseClicked, associados a botões, que são executados sob um clique do mouse. A programação destes scripts é realizada na linguagem VBscript, porém o ActionView não interpreta funções mais avançadas, sendo necessário elaborar uma programação bem simplificada.

O *script* APP_OnTimer possui o código a seguir:

```
'
```

```
Sub OnTimer()
```

```
'Utilização de variável analógica para gravação do comportamento histórico da variável digital:
```

```
Var.Value("PID","AC_1") = Var.Value("PID","AC1")
```

```
Var.Value("PID","AC_2") = Var.Value("PID","AC2")
```

```
'Atualizando os valores de I_VEST e M:
```

If Var.Value("PARAMETROS","VEST_1")=1 Then
 Var.Value("PARAMETROS","I_VEST")=0.047
 If Var.Value("PARAMETROS","VEST_2")=1 Then
 Var.Value("PARAMETROS","I_VEST")=0.078
 If Var.Value("PARAMETROS","VEST_3")=1 Then
 Var.Value("PARAMETROS","I_VEST")=0.124
 If Var.Value("PARAMETROS","VEST_4")=1 Then
 Var.Value("PARAMETROS","I_VEST")=0.155
 If Var.Value("PARAMETROS","VEST_5")=1 Then
 Var.Value("PARAMETROS","I_VEST")=0.233

If Var.Value("PARAMETROS","ACT_1")=1 Then Var.Value("PARAMETROS","M")=70
 If Var.Value("PARAMETROS","ACT_2")=1 Then Var.Value("PARAMETROS","M")=93
 If Var.Value("PARAMETROS","ACT_3")=1 Then Var.Value("PARAMETROS","M")=117
 If Var.Value("PARAMETROS","ACT_4")=1 Then Var.Value("PARAMETROS","M")=175

'Calculando o valor do índice PMV do ambiente 1:

'Pressão de Saturação:

Var.Value("APOIO","PSAT_1")=0.61078*(2.71828^((17.269*Var.Value("TEMPERATURA","T_INT_1"))/(237.3+Var.Value("TEMPERATURA","T_INT_1"))))

'Pressão de Vapor:

Var.Value("APOIO","PA_1")=Var.Value("UMIDADE","UR_1")*Var.Value("APOIO","PSAT_1")

'Difusão de Vapor:

Var.Value("APOIO","D_VAP_1")=3.05*(5.73-0.007*Var.Value("PARAMETROS","M")-
 Var.Value("APOIO","PA_1"))

'Transpiração:

Var.Value("APOIO","TRANSP")=0.42*(Var.Value("PARAMETROS","M")-58.15)

'Respiração Latente:

Var.Value("APOIO","RL_1")=0.0173*Var.Value("PARAMETROS","M")*(5.87-
 Var.Value("APOIO","PA_1"))

'Respiração Sensível:

Var.Value("APOIO","RS_1")=0.0014*Var.Value("PARAMETROS","M")*(34-
Var.Value("TEMPERATURA","T_INT_1"))

'Fator de Vestuário:

If Var.Value("PARAMETROS","I_VEST") > 0.078 Then

Var.Value("APOIO","F_VEST")=1.05+0.645*Var.Value("PARAMETROS","I_VEST")

If Var.Value("PARAMETROS","I_VEST") < 0.078 Then

Var.Value("APOIO","F_VEST")=1.00+1.290*Var.Value("PARAMETROS","I_VEST")

If Var.Value("PARAMETROS","I_VEST") = 0.078 Then

Var.Value("APOIO","F_VEST")=1.05+0.645*Var.Value("PARAMETROS","I_VEST")

'Radiação:

Var.Value("APOIO","RAD_1")=(3.96/10^(8))*Var.Value("APOIO","F_VEST")*(((Var.Value("A
POIO","TVEST_1")+273)^4)-(Var.Value("PARAMETROS","T_RAD_1")+273)^4)

'Convecção:

Var.Value("APOIO","C_1")=Var.Value("APOIO","F_VEST")*12.1*SQR(Var.Value("ANEMOM
ETROS","ANEM_1"))*(Var.Value("APOIO","TVEST_1")-

Var.Value("TEMPERATURA","T_INT_1"))

'Temperatura de Vestimenta:

Var.Value("APOIO","TVEST_1")=35.7-0.0275*Var.Value("PARAMETROS","M")-

0.155*Var.Value("PARAMETROS","I_VEST")*(Var.Value("APOIO","RAD_1")+Var.Value("AP
OIO","C_1"))

'Acumulação de Calor:

Var.Value("APOIO","L_1")=Var.Value("PARAMETROS","M")-

Var.Value("APOIO","D_VAP_1")-Var.Value("APOIO","TRANSP")-

Var.Value("APOIO","RL_1")-Var.Value("APOIO","RS_1")-Var.Value("APOIO","RAD_1")-

Var.Value("APOIO","C_1")

'Índice PMV_1:

Var.Value("APOIO","PMV_1")=(0.303/(2.71828^(0.036*Var.Value("PARAMETROS","M")))+0.
028)*Var.Value("APOIO","L_1")

'

'Calculando o valor do índice PMV do ambiente 2:

'Pressão de Saturação:

Var.Value("APOIO","PSAT_2")=0.61078*(2.71828^((17.269*Var.Value("TEMPERATURA","T_INT_2")))/(237.3+Var.Value("TEMPERATURA","T_INT_2"))))

'Pressão de Vapor:

Var.Value("APOIO","PA_2")=Var.Value("UMIDADE","UR_2")*Var.Value("APOIO","PSAT_2")

'Difusão de Vapor:

Var.Value("APOIO","D_VAP_2")=3.05*(5.73-0.007*Var.Value("PARAMETROS","M")-
Var.Value("APOIO","PA_2"))

'Transpiração:

Var.Value("APOIO","TRANSP")=0.42*(Var.Value("PARAMETROS","M")-58.15)

'Respiração Latente:

Var.Value("APOIO","RL_2")=0.0173*Var.Value("PARAMETROS","M")*(5.87-
Var.Value("APOIO","PA_2"))

'Respiração Sensível:

Var.Value("APOIO","RS_2")=0.0014*Var.Value("PARAMETROS","M")*(34-
Var.Value("TEMPERATURA","T_INT_2"))

'Fator de Vestuário:

If Var.Value("PARAMETROS","I_VEST") > 0.078 Then

Var.Value("APOIO","F_VEST")=1.05+0.645*Var.Value("PARAMETROS","I_VEST")

If Var.Value("PARAMETROS","I_VEST") < 0.078 Then

Var.Value("APOIO","F_VEST")=1.00+1.290*Var.Value("PARAMETROS","I_VEST")

If Var.Value("PARAMETROS","I_VEST") = 0.078 Then

Var.Value("APOIO","F_VEST")=1.05+0.645*Var.Value("PARAMETROS","I_VEST")

'Radiação:

Var.Value("APOIO","RAD_2")=(3.96/10^(8))*Var.Value("APOIO","F_VEST")*(((Var.Value("APOIO","TVEST_2")+273)^4)-(Var.Value("PARAMETROS","T_RAD_2")+273)^4)

'Convecção:

Var.Value("APOIO","C_2")=Var.Value("APOIO","F_VEST")*12.1*SQR(Var.Value("ANEMOMETROS","ANEM_2"))*(Var.Value("APOIO","TVEST_2")-
Var.Value("TEMPERATURA","T_INT_2"))

'Temperatura de Vestimenta:

```
Var.Value("APOIO","TVEST_2")=35.7-0.0275*Var.Value("PARAMETROS","M")-  
0.155*Var.Value("PARAMETROS","I_VEST")*(Var.Value("APOIO","RAD_2")+Var.Value("APOIO","C_2"))
```

'Acumulação de Calor:

```
Var.Value("APOIO","L_2")=Var.Value("PARAMETROS","M")-  
Var.Value("APOIO","D_VAP_2")-Var.Value("APOIO","TRANSP")-  
Var.Value("APOIO","RL_2")-Var.Value("APOIO","RS_2")-Var.Value("APOIO","RAD_2")-  
Var.Value("APOIO","C_2")
```

'Índice PMV_2:

```
Var.Value("APOIO","PMV_2")=(0.303/(2.71828^(0.036*Var.Value("PARAMETROS","M")))+0.028)*Var.Value("APOIO","L_2")
```

End Sub

'

O *script*, APP_OnTimer, inicia atualizando o valor das variáveis analógicas AC_1 e AC_2 com os valores das variáveis digitais AC1 e AC2, respectivamente, que correspondem ao estado do ar condicionado dos ambientes 1 e 2. Este procedimento é necessário para poder acessar a tendência histórica do estado dos ar condicionados, afinal existe a limitação do ActionView em somente registrar corretamente, no histórico do programa, as variáveis analógicas.

Em seguida este *script* atualiza os valores da resistência térmica da vestimenta, I_VEST, e da taxa de metabolismo, M. Isto é feito verificando qual das variáveis VEST_X e ACT_X estão selecionadas pelo operador, sendo que os valores para cada uma dessas variáveis foi descrito no Anexo I deste trabalho.

Os valores de ACT_X e VEST_X são atualizados através de *scripts OnMouseClicked*, associados a botões, que representam as escolhas do nível de atividade e do tipo de vestimenta, feita pelo operador na tela de configurações. Os códigos destes *scripts* são bastante semelhantes e apresentam a seguinte lógica: um clique em um dos botões faz com que o valor associado à variável seja alterado e assim considera-se que somente uma variável de cada tipo está selecionada. Os textos desses *scripts* encontram-se a seguir:

'

CONFIG_Digital0_OnMouseClicked:

```
Sub OnMouseClicked()
```

```
If Var.Value("PARAMETROS","ACT_1")=0 Then Var.Value("PARAMETROS","ACT_1")=1
If Var.Value("PARAMETROS","ACT_1")=1 Then Var.Value("PARAMETROS","ACT_1")=0
If Var.Value("PARAMETROS","ACT_1")=1 Then Var.Value("PARAMETROS","ACT_2")=0
If Var.Value("PARAMETROS","ACT_1")=1 Then Var.Value("PARAMETROS","ACT_3")=0
If Var.Value("PARAMETROS","ACT_1")=1 Then Var.Value("PARAMETROS","ACT_4")=0
End Sub
```

CONFIG_Digital1_OnMouseClicked:

```
Sub OnMouseClicked()
If Var.Value("PARAMETROS","ACT_2")=0 Then Var.Value("PARAMETROS","ACT_2")=1
If Var.Value("PARAMETROS","ACT_2")=1 Then Var.Value("PARAMETROS","ACT_2")=0
If Var.Value("PARAMETROS","ACT_2")=1 Then Var.Value("PARAMETROS","ACT_1")=0
If Var.Value("PARAMETROS","ACT_2")=1 Then Var.Value("PARAMETROS","ACT_3")=0
If Var.Value("PARAMETROS","ACT_2")=1 Then Var.Value("PARAMETROS","ACT_4")=0
End Sub
```

CONFIG_Digital2_OnMouseClicked:

```
Sub OnMouseClicked()
If Var.Value("PARAMETROS","ACT_3")=0 Then Var.Value("PARAMETROS","ACT_3")=1
If Var.Value("PARAMETROS","ACT_3")=1 Then Var.Value("PARAMETROS","ACT_3")=0
If Var.Value("PARAMETROS","ACT_3")=1 Then Var.Value("PARAMETROS","ACT_1")=0
If Var.Value("PARAMETROS","ACT_3")=1 Then Var.Value("PARAMETROS","ACT_2")=0
If Var.Value("PARAMETROS","ACT_3")=1 Then Var.Value("PARAMETROS","ACT_4")=0
End Sub
```

CONFIG_Digital3_OnMouseClicked:

```
Sub OnMouseClicked()
If Var.Value("PARAMETROS","ACT_4")=0 Then Var.Value("PARAMETROS","ACT_4")=1
If Var.Value("PARAMETROS","ACT_4")=1 Then Var.Value("PARAMETROS","ACT_4")=0
If Var.Value("PARAMETROS","ACT_4")=1 Then Var.Value("PARAMETROS","ACT_1")=0
If Var.Value("PARAMETROS","ACT_4")=1 Then Var.Value("PARAMETROS","ACT_2")=0
If Var.Value("PARAMETROS","ACT_4")=1 Then Var.Value("PARAMETROS","ACT_3")=0
End Sub
```

CONFIG_Digital4_OnMouseClicked:

Sub OnMouseClicked()

If Var.Value("PARAMETROS","VEST_1")=0 Then Var.Value("PARAMETROS","VEST_1")=1

If Var.Value("PARAMETROS","VEST_1")=1 Then Var.Value("PARAMETROS","VEST_1")=0

If Var.Value("PARAMETROS","VEST_1")=1 Then Var.Value("PARAMETROS","VEST_2")=0

If Var.Value("PARAMETROS","VEST_1")=1 Then Var.Value("PARAMETROS","VEST_3")=0

If Var.Value("PARAMETROS","VEST_1")=1 Then Var.Value("PARAMETROS","VEST_4")=0

If Var.Value("PARAMETROS","VEST_1")=1 Then Var.Value("PARAMETROS","VEST_5")=0

End Sub

'

CONFIG_Digital5_OnMouseClicked:

Sub OnMouseClicked()

If Var.Value("PARAMETROS","VEST_2")=0 Then Var.Value("PARAMETROS","VEST_2")=1

If Var.Value("PARAMETROS","VEST_2")=1 Then Var.Value("PARAMETROS","VEST_2")=0

If Var.Value("PARAMETROS","VEST_2")=1 Then Var.Value("PARAMETROS","VEST_1")=0

If Var.Value("PARAMETROS","VEST_2")=1 Then Var.Value("PARAMETROS","VEST_3")=0

If Var.Value("PARAMETROS","VEST_2")=1 Then Var.Value("PARAMETROS","VEST_4")=0

If Var.Value("PARAMETROS","VEST_2")=1 Then Var.Value("PARAMETROS","VEST_5")=0

End Sub

'

CONFIG_Digital6_OnMouseClicked:

Sub OnMouseClicked()

If Var.Value("PARAMETROS","VEST_3")=0 Then Var.Value("PARAMETROS","VEST_3")=1

If Var.Value("PARAMETROS","VEST_3")=1 Then Var.Value("PARAMETROS","VEST_3")=0

If Var.Value("PARAMETROS","VEST_3")=1 Then Var.Value("PARAMETROS","VEST_1")=0

If Var.Value("PARAMETROS","VEST_3")=1 Then Var.Value("PARAMETROS","VEST_2")=0

If Var.Value("PARAMETROS","VEST_3")=1 Then Var.Value("PARAMETROS","VEST_4")=0

If Var.Value("PARAMETROS","VEST_3")=1 Then Var.Value("PARAMETROS","VEST_5")=0

End Sub

'

CONFIG_Digital7_OnMouseClicked:

Sub OnMouseClicked()

If Var.Value("PARAMETROS","VEST_4")=0 Then Var.Value("PARAMETROS","VEST_4")=1

If Var.Value("PARAMETROS","VEST_4")=1 Then Var.Value("PARAMETROS","VEST_4")=0

If Var.Value("PARAMETROS","VEST_4")=1 Then Var.Value("PARAMETROS","VEST_1")=0

```
If Var.Value("PARAMETROS","VEST_4")=1 Then Var.Value("PARAMETROS","VEST_2")=0
If Var.Value("PARAMETROS","VEST_4")=1 Then Var.Value("PARAMETROS","VEST_3")=0
If Var.Value("PARAMETROS","VEST_4")=1 Then Var.Value("PARAMETROS","VEST_5")=0
End Sub
```

CONFIG_Digital8_OnMouseClicked:

```
Sub OnMouseClicked()
```

```
If Var.Value("PARAMETROS","VEST_5")=0 Then Var.Value("PARAMETROS","VEST_5")=1
If Var.Value("PARAMETROS","VEST_5")=1 Then Var.Value("PARAMETROS","VEST_5")=0
If Var.Value("PARAMETROS","VEST_5")=1 Then Var.Value("PARAMETROS","VEST_1")=0
If Var.Value("PARAMETROS","VEST_5")=1 Then Var.Value("PARAMETROS","VEST_2")=0
If Var.Value("PARAMETROS","VEST_5")=1 Then Var.Value("PARAMETROS","VEST_3")=0
If Var.Value("PARAMETROS","VEST_5")=1 Then Var.Value("PARAMETROS","VEST_4")=0
End Sub
```

ANEXO III

Este anexo apresenta o software supervisor elaborado em [11] com as adaptações para comunicar com o ActionView.

AIII.1 – Programa supervisor

O supervisor que foi desenvolvido em linguagem C, utilizando o compilador GCC, e coordena o fluxo de dados da rede por meio do nó coordenador conectado ao computador. A rede é do tipo estrela e constituída por um nó coordenador, dois nós sensores e dois nós atuadores.

A Tabela AIII.1 apresenta a configuração dos sensores e atuadores que possuem endereçamento fixo, e os módulos sensores somente enviam dados enquanto os módulos enviam e recebem dados para o coordenador. A comunicação é feita em modo *unicast*, e assim o programa altera endereço pessoal e destino do coordenador para que o mesmo envie e receba dados de cada nó da rede.

Tabela AIII.1 – Configuração dos módulos da rede [11].

	Coordenador	SENSOR 1	SENSOR 2	ATUADOR 1	ATUADOR 2
ATMY	VARIÁVEL	201	205	199	204
ATDL	VARIÁVEL	200	203	202	198
ATIA	FFFF	FFFF	FFFF	FFFF	FFFF
ATIU	1	1	1	1	1
ATIR		9C4	9C4		
ATIT		3	3		
ATDO		2	2		

Este programa tem somente uma temperatura de referência para o cálculo do controle ON/OFF, entretanto este cálculo é feito separadamente para cada ambiente do laboratório através da temperatura de cada sensor e assim é enviado o *status* do atuador para o nó determinado. A Figura AIII.1 apresenta o fluxograma do funcionamento deste programa.

Para minimizar os erros de leitura da temperatura, o software faz um tratamento três amostras de temperaturas obtidas dos sensores e assim calculando a média destes valores. Se a diferença par a par absoluta entre os três valores lidos for menor ou igual a 1 ou todas seja maiores que 1, a temperatura considerada será a média dos valores. Senão, a temperatura considerada será a média entre as duas amostras em que a diferença seja menor que 1 [11].

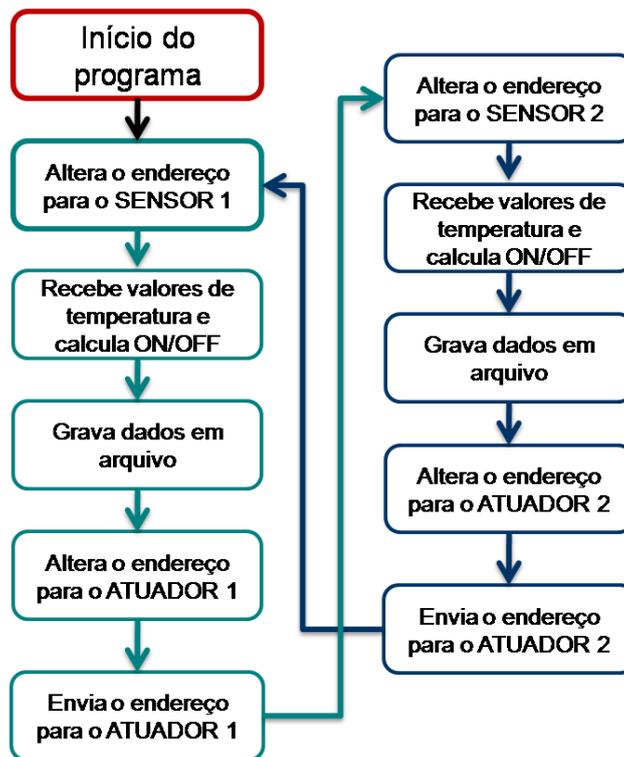


Figura AIII.1 – Fluxograma de funcionamento do software supervisor [11].

O código foi modificado de acordo com a sintaxe do Visual Studio, inserindo as funções GetVarRef e PutDataValue para enviar os valores de temperatura do programa ao ActionView. Se mostrou necessário a utilização do arquivo AVRUN50.lib como *link* e também adicionar a biblioteca AVRUNTAPI.h, ambos disponibilizados pela SPIN.

AIII.2 –Código adaptado para o Visual Studio 2010

```

/*****
Nome do arquivo: supervisorio_v4_API.c
Autores: Pedro Ramos Mateus Filho
        Yuri Ferreira Gomes Dias
Data: 21/05/2008
Descrição: Software supervisorio de controle de temperatura em rede de automa-
cao wireless. Ele coordena dois nós sensores e dois nós atuadores, processando
valores de temperatura e realizando os dados do controlador liga-desliga. Este
software foi utilizado no Trabalho de Graduação dos autores.”
Última modificação: 17/03/2010 por Marco Willian Amaral Cardoso.
*****/

```

```

/*Bibliotecas incluídas no programa*/

```

```

#include <stdafx.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>
#include <windows.h>
#include "AVRUNTAPI.h"

```

```

/*Funcoes criadas*/

/*Converte hexa para inteiro*/
int ConvHEXDEC(char *Hexa);

/*Altera enderecos para comunicacao com o Atuador1*/
void altera_atmy_atuador1(FILE *Serial);

/*Altera enderecos para comunicacao com o Atuador2*/
void altera_atmy_atuador2(FILE *Serial);

/*Altera enderecos para comunicacao com o Sensor1*/
void altera_atmy_remote1(FILE *Serial);

/*Altera enderecos para comunicacao com o Sensor2*/
void altera_atmy_remote2(FILE *Serial);

/*Envia para cada atuador o seu status, ON-OFF*/
void envia_temperatura(char status_atuador, FILE *Serial);

/*Recebe e processa os dados de temperatura recebidos*/
float recebe_temperatura (FILE *Serial,unsigned char c, float temp1,float temp2,float
temp3,float t_ref,char status_atuador);

/*Limpa a serial e entrada padrao de dados*/
void limpa_Serial(FILE *Serial);

/*Imprime cabecalho*/
void cabecalho();

/*Função para comunicação com ActionView*/
void SendPoint( void*      pVD,
               float      floValor,
               unsigned long unslTime,
               short       shoMilisecs,
               short       shoError // intError
               );

/*Calculos do controlador liga-desliga*/
char calculo_ON_OFF (float t_ref, float t_media, char status_atuador);

/*Volta ao modo API 1*/
void modo_API(FILE *Serial);

int main(){

/*Variáveis utilizadas*/
FILE *Serial;
FILE *Dados;
unsigned char c;
unsigned char *string =(unsigned char *)malloc(sizeof(char)*300);
float temp1=0,temp2=0,temp3=0,t_ref=0,t_media=0;
int canal = 0,duracao = 0,i = 0,tamanho=0;
long int hora = 0;
char status_atuador1 = 'L',status_atuador2 = 'L';

```

```

printf ("*****ACIONAMENTO DE POTENCIA EM REDE DE AUTOMACAO
WIRELESS*****");
printf("\n\nAbrindo porta serial...");
printf("\n\nAbrindo porta serial...");

/*Abre porta serial*/
Serial=fopen("COM4", "r+");
if (Serial==NULL)
{
printf(" Falhou!");
}
else
{
printf(" Concluido!!\n\n");
}

printf("Abrindo arquivo de gravacao de dados...");

/*Abre arquivo onde sao armazenados os dados de controle*/
Dados=fopen("Dados.txt", "w");
if (Dados==NULL)
{
printf(" Falhou!");
}
else
{
printf(" Concluido!!\n\n");
}

/*Estabelece o valor da referencia em 22 graus Celsius */
//printf("DIGITE O VALOR DA REFERENCIA DESEJADA:");
//scanf("%f",&t_ref);
system("cls");
t_ref =22;
cabecalho();
limpa_Serial(Serial);
modo_API(Serial);

/*LOOP que coordena a chamada das funcoes de comunicacao com os modulos e
grava os dados de controle em arquivo*/
while (duracao < 60000){
printf("\n\nREFERENCIA = %0.2f",t_ref);
printf("\nDURACAO = %d",duracao);

/*Chama funcao para comunicacao com o sensor1*/
altera_atmy_remote1(Serial);

/*Chama funcao que recebe e processa os dados de temperatura*/
t_media = recebe_temperatura(Serial,c,temp1,temp2,temp3,t_ref,status_atuador1);

/*Grava no arquivo de dados*/
fprintf(Dados, "%f",t_ref);
fprintf(Dados, "\t");
fprintf(Dados, "%f",t_media);
fprintf(Dados, "\t");
status_atuador1 = calculo_ON_OFF(t_ref,t_media,status_atuador1);
}

```

```

fprintf(Dados,"%c",status_atuador1);
fprintf(Dados,"\t");

/*Chama funcao para comunicacao com o atuador1*/
altera_atmy_atuador1(Serial);

/*Chama funcao enviar status ao atuador1*/
envia_temperatura(status_atuador1,Serial);

altera_atmy_remote2(Serial);

/*Chama funcao que recebe e processa os dados de temperatura*/
t_media = recebe_temperatura(Serial,c,temp1,temp2,temp3,t_ref,status_atuador2);

/*Pega hora*/
hora = clock();

/*Grava no arquivo de dados*/
fprintf(Dados,"%f",t_media);
fprintf(Dados,"\t");
status_atuador2 = calculo_ON_OFF(t_ref,t_media,status_atuador2);
fprintf(Dados,"%c",status_atuador2);
fprintf(Dados,"\t");
fprintf(Dados,"%d",hora);
fprintf(Dados,"\n");

/*Chama funcao para comunicacao com o atuador2*/
altera_atmy_atuador2(Serial);

/*Chama funcao enviar status ao atuador2*/
envia_temperatura(status_atuador2,Serial);

/*Incrementa variavel que define a duracao do experimento*/
duracao++;

/*Dependendo do valor selecionado, consulta sobre a alteracao de
referencia*/
if (duracao == 60000){
    printf("\n\nDESEJA ALTERAR A REFERENCIA? s/n: ");
    c = getchar();
    if (c == 's'){
        printf("\nDIGITE O VALOR DA REFERENCIA DESEJADA:");
        scanf("%f",&t_ref);
    }
}
}

/*Fecha arquivos e encerra programa*/
fclose(Dados);
fclose(Serial);
printf("\n\n");
system("PAUSE");
return 0;
}

/*Funcao que converte hexadecimal para inteiro: por Fellipe Lopes Couto*/

```

```

int ConvHEXDEC(char *Hexa)
{
    int k;
    k=strlen(Hexa);
    return (k);
}

```

/*Funcao que altera endereco para comunicacao com atuador1. Ela entra no modo de comando do Xbee corrdenador e seta ATMY = 202 e ATDL = 199 e apos encerra modo de comando. O retorno lido do Xbee é recebido e pode ser usado para debugar o programa.*/

```

void altera_atmy_atuador1(FILE *Serial){

```

```

    putc(0x7E , Serial);
    putc(0x00 , Serial);
    putc(0x08 , Serial);
    putc(0x08 , Serial);
    putc(0x00 , Serial);
    putc(0x4D , Serial);
    putc(0x59 , Serial);
    putc(0x00 , Serial);
    putc(0x00 , Serial);
    putc(0x02 , Serial);
    putc(0x02 , Serial);
    putc(0x4D , Serial);

```

```

    putc(0x7E , Serial);
    putc(0x00 , Serial);
    putc(0x08 , Serial);
    putc(0x08 , Serial);
    putc(0x00 , Serial);
    putc(0x44 , Serial);
    putc(0x4C , Serial);
    putc(0x00 , Serial);
    putc(0x00 , Serial);
    putc(0x01 , Serial);
    putc(0x99 , Serial);
    putc(0xCD , Serial);

```

```

}

```

/*Funcao que altera endereco para comunicacao com atuador2. Ela entra no modo de comando do Xbee corrdenador e seta ATMY = 198 e ATDL = 204 e apos encerra modo de comando. O retorno lido do Xbee é recebido e pode ser usado para debugar o programa.*/

```

void altera_atmy_atuador2(FILE *Serial){

```

```

    putc(0x7E , Serial);
    putc(0x00 , Serial);
    putc(0x08 , Serial);
    putc(0x08 , Serial);
    putc(0x00 , Serial);
    putc(0x4D , Serial);
    putc(0x59 , Serial);
    putc(0x00 , Serial);
    putc(0x00 , Serial);
    putc(0x01 , Serial);

```

```
putc(0x98 , Serial);
putc(0xB8 , Serial);
```

```
putc(0x7E , Serial);
putc(0x00 , Serial);
putc(0x08 , Serial);
putc(0x08 , Serial);
putc(0x00 , Serial);
putc(0x44 , Serial);
putc(0x4C , Serial);
putc(0x00 , Serial);
putc(0x00 , Serial);
putc(0x02 , Serial);
putc(0x04 , Serial);
putc(0x61 , Serial);
```

```
}
```

/*Funcao que altera endereco para comunicacao com sensor1. Ela entra no modo de comando do Xbee corrdenador e seta ATMY = 200 e apos encerra modo de comando. O retorno lido do Xbee é recebido e pode ser usado para debugar o programa.*/

```
void altera_atmy_remote1(FILE *Serial){
```

```
putc(0x7E , Serial);
putc(0x00 , Serial);
putc(0x08 , Serial);
putc(0x08 , Serial);
putc(0x00 , Serial);
putc(0x4D , Serial);
putc(0x59 , Serial);
putc(0x00 , Serial);
putc(0x00 , Serial);
putc(0x02 , Serial);
putc(0x00 , Serial);
putc(0x4F , Serial);
```

```
}
```

/*Funcao que altera endereco para comunicacao com sensor2. Ela entra no modo de comando do Xbee corrdenador e seta ATMY = 203 e apos encerra modo de comando. O retorno lido do Xbee é recebido e pode ser usado para debugar o programa.*/

```
void altera_atmy_remote2(FILE *Serial){
```

```
putc(0x7E , Serial);
putc(0x00 , Serial);
putc(0x08 , Serial);
putc(0x08 , Serial);
putc(0x00 , Serial);
putc(0x4D , Serial);
putc(0x59 , Serial);
putc(0x00 , Serial);
putc(0x00 , Serial);
putc(0x02 , Serial);
putc(0x03 , Serial);
```

```

    putc(0x4C , Serial);
}

/*Funcao que envia ao atuador o seu status ligado - H ou desligado - L
Apos o envio ela altera o endereco para o sensor2*/
void envia_temperatura(char status_atuador, FILE *Serial){

    putc(0x7E , Serial);
    putc(0x00 , Serial);
    putc(0x08 , Serial);
    putc(0x08 , Serial);
    putc(0x00 , Serial);
    putc(0x41 , Serial);
    putc(0x50 , Serial);
    putc(0x00 , Serial);
    putc(0x66 , Serial);

    char c;
    limpa_Serial(Serial);
    putc(status_atuador,Serial);
    /*fputc(13,Serial);*/

    Sleep(50);
    limpa_Serial(Serial);

    c = fgetc(Serial);
    /*printf("\n\nFFFFFFFFFFFFFF%c",c);*/

    limpa_Serial(Serial);

    modo_API(Serial);
}

/*Funcao que recebe todo o FRAME API enviado pelos sensores e o processa
identificando os valores de temperatura. Estes valores sao verificados para
eliminacao de erros de medicao.*/
float recebe_temperatura (FILE *Serial, unsigned char c, float temp1,float temp2,float
temp3,float t_ref,char status_atuador){

    int endereco;
    float temp_media = 0;
    char atuador;
    int j=0,t=0;
    temp1=0,temp2=0,temp3=0;
    char estado_atuador;

    limpa_Serial(Serial);
    c = fgetc(Serial);

    /*Aguarda até a chegada do FRAME API, que identificado pelo HEXA 7E*/

```

```

while (t==0){
while (c != 0x7E){
    limpa_Serial(Serial);
    c = fgetc(Serial);
    }

//printf("\n\nvalor: %c",c);

/*As linhas abaixo retira do frame apenas os dados que tem interesse,
endereço e temperatura*/
Sleep(10);
c = fgetc(Serial);

if (c == 0){
    t=0;
    }

for(j = 1; c != 255 ; c=fgetc(Serial),j++){

if (j==4){
    ConvHEXDEC((char*)&c);
    endereco = c*100;
    }
if (j==5){
    ConvHEXDEC(&c);
    endereco += c;
    //printf("\n\nEndereço do sensor: %d",endereco);
    }

if (j==12){ //12
    ConvHEXDEC(&c);
    temp1 = c + temp1;
    temp1 = temp1/3.28;
    /*Imprime amostra1*/
    //printf(" T1 = %0.2f",temp1);
    }

if (j==14){ //14
    ConvHEXDEC(&c);
    temp2 = c + temp2;
    temp2 = temp2/3.28;
    /*Imprime amostra2*/
    //printf(" T2 = %0.2f",temp2);
    }

if (j==16){ //16
    ConvHEXDEC(&c);
    temp3 = c + temp3;
    temp3 = temp3/3.28;
    /*Imprime amostra2*/
    //printf(" T3 = %0.2f",temp3);
    }
}
Sleep(10);

```

```

}

if (fabs(temp1-temp2)>1 && fabs(temp3-temp2)>1 && fabs(temp1-temp3)>1) {
    t = 0;
} else {
    if (temp1==temp2 && temp2==temp3 && temp1 == temp3 && temp1 == 0) {
        t = 0;
    } else {
        t = 1;
    }
}

printf("\n\nEndereco do sensor: %d",endereco);
printf(" T1 = %0.2f",temp1);
printf(" T2 = %0.2f",temp2);
printf(" T3 = %0.2f",temp3);

/*Funcao que elimina os valores de temperatura lido que destoa
das demais*/
if (fabs(temp1-temp2)<=1 && fabs(temp3-temp2)<=1 && fabs(temp1-temp3)<=1){

    temp_media = ((temp1+temp2+temp3)/3);

} else {

    if (fabs(temp1-temp2)<1) temp_media = ((temp1+temp2)/2);
    if (fabs(temp3-temp2)<1) temp_media = ((temp3+temp2)/2);
    if (fabs(temp1-temp3)<1) temp_media = ((temp1+temp3)/2);

}

/*Imprime status do atuador do presente setor*/
if (calculo_ON_OFF(t_ref,temp_media,status_atuador)=='H') printf(" ATUADOR
= ON");
else printf(" ATUADOR = OFF");

estado_atuador = calculo_ON_OFF(t_ref,temp_media,status_atuador);

if (endereco == 201){
    //tempmedia e temperatura do ambiente 1
    float temp_a1;
    temp_a1 = void* _stdcall GetVarRef( char* TEMPERATURA, char*
T_INT_1);
    short _stdcall PutDataValue(long temp_a1, float tempmedia);

    //estado_atuador e o estado do atuador do ambiente 1
}
if (endereco == 205){
    //tempmedia e temperatura do ambiente 2
    float temp_a2;
    temp_a2 = void* _stdcall GetVarRef( char* TEMPERATURA, char*
T_INT_2);
    short _stdcall PutDataValue(long temp_a2, float tempmedia);

    //estado_atuador e o estado do atuador do ambiente 2
}

```

```

        return temp_media;
    }

/*Funcao que limpa a porta serial e a entrada padrão*/
void limpa_Serial (FILE *Serial)
{
    fflush(stdin);
    rewind(Serial);
}

/*Funcao que imprime o cabecalho na tela*/
void cabecalho(){
    printf ("*****");
    printf ("\n\n      ACIONAMENTO DE POTENCIA EM REDE DE AUTOMACAO
WIRELESS      ");
    printf ("\n\n*****");
    printf ("\n\n_____ PARAMETROS DE CONTROLE
_____");
}

/*Fucao que realiza os calculos do controlador liga-desliga e retorna o status
do atuador*/
char calculo_ON_OFF (float t_ref, float t_media, char status_atuador){
    if (t_ref > t_media && (t_ref - t_media)>=0.5){
        return 'L';
    }

    if(t_ref < t_media && (t_ref - t_media)<=-0.5 ){
        return 'H';
    }

    return status_atuador;
}

void modo_API(FILE *Serial){
    int j = 0;
    unsigned char ch;
    unsigned char str[20];
    Sleep(80);
    limpa_Serial(Serial);
    fprintf(Serial, "+++");
    limpa_Serial(Serial);
    Sleep(80);

    ch = fgetc(Serial);

    for(j=0; ch != 255 && j<3 ; ch=fgetc(Serial))
        {
            str[j++]=ch;

```

```

    }
    str[j] = '\0';
// printf("\n%s",str);

limpa_Serial(Serial);
fprintf(Serial, "ATAP1");
fprintf(Serial, "\r");
limpa_Serial(Serial);
Sleep(80);
ch = fgetc(Serial);

    for(j =0; ch != 255 && j < 3; ch=fgetc(Serial))
        {
            str[j++]=ch;
        }
    str[j] = '\0';
// printf("\n%s",str);

limpa_Serial(Serial);
fprintf(Serial,"ATCN");
fprintf(Serial, "\r");

limpa_Serial(Serial);
Sleep(80);

ch = fgetc(Serial);
for(j=0; ch != 255 && j < 3; ch=fgetc(Serial))
    {
        str[j++]=ch;
    }
    str[j] = '\0';
//printf("\n%s",str);
}

//Função disponibilizada pela SPIN para comunicar com o ActionView
void SendPoint( void*      pVD,
                float      floValor,
                unsigned long  unsITime,
                short      shoMilisecs,
                short      shoError // intError
                )
{
    struIODataRec IOD;
    IOD.intErrors = shoError; //Monta estrutura que deverá ser passada para
AVRUNTIM
    if (unsITime == 0)
    {
        IOD.unITime = time(NULL);
        IOD.intMsecs = 1024; // 0x400;
    }
    else
    {
        IOD.unITime = unsITime;
        IOD.intMsecs = shoMilisecs;
    }
}

```

```
IOD.floValue = floValor;
if (pVD) // Obtem objeto ponto
{
    IOD.ptrVD = pVD;

    // This is function to set a value on tag object inside main SCADA DLL
    TRACE("PutDataRec = %f\n",IOD.floValue);
    PutDataRec(&IOD,NULL);
}
}
```

ANEXO IV

Este anexo apresenta uma padronização para trabalhos em conforto térmico, que foi seguida para execução deste projeto, sendo obtido através de [1]

ROTEIRO PARA O TRABALHO DE CONFORTO TÉRMICO

O trabalho referente a medições e avaliações de conforto térmico de um ambiente interno, deverá conter no mínimo os seguintes passos:

- 1) Todas as variáveis ambientais deverão ser medidas, ou seja, temperatura do ar, temperatura média radiante, velocidade do ar e umidade do ar, pelo menos em dois períodos do dia, um de manhã (no mínimo 1 hora e 20 minutos) e um à tarde (no mínimo 1 hora e 20 minutos), com registros de medições a cada 5 minutos.
2. O ambiente a ser escolhido pelo aluno ou grupo de no máximo 2 alunos, deverão conter no mínimo 5 pessoas em seu interior, para que se possa aplicar questionário de avaliação subjetiva.
3. O isolamento térmico das vestimentas das pessoas, deverão ser determinadas de acordo com a ASHRAE Fundamentals, cap. 8 – THERMAL COMFORT
4. O valor da taxa metabólica também deverá ser estimado de acordo com a ASHRAE, sendo que a atividade deve ser bem detalhada no relatório final.
5. As medições deverão seguir rigorosamente os preceitos da ISO/DIS 7726, no que diz respeito a todas as variáveis ambientais.
6. É imprescindível a verificação da homogeneidade do ambiente, cujos resultados deverão constar do relatório final;
7. O PMV e PPD deverão ser calculados segundo equação da ISO 7730 (1994), podendo também ser utilizado algum software que esteja disponível, como Analysis, Thermal Comfort, ou algum outro, sendo que a indicação da determinação deve constar do relatório;
8. Todos os resultados referentes às medições de variáveis ambientais e de PMV e PPD deverão ser apresentados de forma analítica (tabelas) e gráficas.
9. Para efeito de ser verificado que as medições não apresentaram erros, deverão ser apresentados os resultados de normalidade das variáveis medidas, exceto com relação à velocidade do ar;
10. O PMV e o PPD, também devem ser calculados, levando-se em consideração todas as imprecisões das medições, conforme norma ISO 7726;
11. Deverá ser comparado os resultados analíticos do conforto (PMV) com as sensações relatadas pelas pessoas, a fim de se verificar a aplicabilidade do modelo normalizado;
12. As características do ambiente analisado deverão constar do relatório final;

13. As sensações térmicas e preferências térmicas das pessoas pesquisadas deverão ser obtidas por questionários como o apresentado em anexo, sendo que deverão ser coletados os dados a cada 20 minutos após o início das medições;
14. Todas as conclusões a respeito das medições, análises e avaliação do ambiente, serão de responsabilidade do aluno ou grupo, devendo fazer parte integrante do relatório final a ser avaliado.

AVALIAÇÃO DE CONFORTO E ACEITABILIDADE TÉRMICA

Avaliação das condições, sensações e aceitabilidade dos ambientes:

O preenchimento dessas tabelas tem a função da avaliação da situação de conforto térmico através do julgamento subjetivo. As respectivas análises e comparações dessa avaliação com os dados ambientais coletados pelos equipamentos servirão de valiosos subsídios para a análise térmica dos ambientes da edificação.

Os quesitos nº 3, 4 e 5 devem ser devidamente anotados a cada horário marcado. Os quesitos de nº 1, 2, devem ser anotados apenas uma vez, quando do primeiro horário de anotação. No caso de ter havido mudanças nas vestimentas entre um horário de anotação e outro, favor apontar após a tabela do quesito 2.

Quesito 1 Dados do respondente:

Idade:..... Altura:..... Peso:..... Sexo:..... Data:.....

Quesito 2 Marque as vestimentas que está utilizando:

(Tabela conforme ISO 9920/95)

Roupas de baixo e Acessórios		Blusa leve fina, manga curta	
Sapato com sola fina		Camiseta	
Sapato com sola grossa		Calças	
Botinas		Calça curta (bermuda)	
Meia soquete fina		Calça tecido fino	
Meia soquete grossa		Calça jeans	
Meia até o Joelho		Calça grossa, de lã ou flanela	
Meia de nylon longa fina		Vestidos e Saias	
Meia calça com pernas longas		Saia leve, de verão	
Meia calça com pernas curtas		Saia pesada, de inverno	
Cueca		Vestido de verão, mangas curtas	
Calcinha		Vestido de inverno, manga longa	
Soutien		Vestido completo, fechado	
Camiseta de baixo		Casacos e Suéteres	
Camiseta de baixo manga longa		Colete sem mangas fino	
Gravata		Colete sem mangas grosso	
Camisas e Blusas		Suéter manga longa fino	
Camisa de manga curta		Suéter manga longa grosso	
Camisa manga longa tecido fino		Jaqueta leve	
Camisa manga longa normal		Jaqueta/japona, normal	
Camisa de flanela ou moleton		Paletó	
Blusa leve fina, manga longa		Paletó de verão, blazer	

Houve alguma mudança de vestimentas entre um horário e outro? Qual?

Quesito 3 Tabela de percepção (ISO 10551/95)

“Com relação a sua sensação térmica, como você está se sentindo nesse momento?”

Com muito calor				
Com calor				
Levemente com calor				
Neutro				
Levemente com frio				
Com frio				
Com muito frio				

Quesito 4 Tabela de avaliação (ISO 10551/95)

“De que maneira você se encontra nesse momento?”

Confortável				
Levem. desconfortável				
Inconfortável				
Muito desconfortável				

Quesito 5 Tabela de preferências térmicas (ISO 10551/95)

“Como você preferia estar se sentindo agora?”

Bem mais aquecido				
Mais aquecido				
Um pouco mais aquecido				
Assim mesmo				
Um pouco mais resfriado				
Mais resfriado				
Bem mais resfriado				

ANEXO V

Este anexo apresenta o conteúdo do CD com o material deste projeto. O CD contém 5 pastas e um arquivo do relatório do projeto de graduação no formato .pdf. As pastas são divididas em: Apresentação, Softwares, Figuras, Manuais, ActionView.

- Apresentação: contém a apresentação deste trabalho de graduação feita em 23/03/2010
- Softwares:
 - Supervisorio_Lavsi: pasta com o programa supervisorio desenvolvido em [11] e atualizado para este projeto, utilizando o Visual Studio 2010.
 - Driver_CON_USBBEE.zip: driver para utilização da placa CON-USBBEE
- Figuras: contém as figuras utilizadas no trabalho e o arquivo em vsd com a planta baixa do laboratório.
- Manuais: contém alguns datasheets e manuais utilizados neste trabalho.
- ActionView: contém todas as informações de configuração do ActionView, manuais do programa, inclusive de instalação.