

TRABALHO DE GRADUAÇÃO

AUTOMAÇÃO PREDIAL SEM FIO UTILIZANDO BACNET/ZIGBEE COM FOCO EM ECONOMIA DE ENERGIA

Lucília Pereira Borges e Rodrigo de Carvalho Dores

Brasília, Setembro de 2010



**ENGENHARIA
MECATRÔNICA**
UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia
Curso de Graduação em Engenharia de Controle e Automação

TRABALHO DE GRADUAÇÃO

AUTOMAÇÃO PREDIAL SEM FIO UTILIZANDO BACNET/ZIGBEE COM FOCO EM ECONOMIA DE ENERGIA

**Lucília Pereira Borges
Rodrigo de Carvalho Dores**

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

Banca Examinadora

Prof. Adolfo Bauchpiess, UnB/ ENE (Orientador) _____

Prof. Lélío R. Soares Júnior, UnB/ ENE _____

Prof. Marco A. Egito Coelho, UnB/ENE _____

Brasília, Setembro de 2010

FICHA CATALOGRÁFICA

BORGES, LUCÍLIA PEREIRA &
DORES, RODRIGO DE CARVALHO
Automação predial sem fio utilizando BACnet/ZigBee com foco em economia de energia,
[Distrito Federal] 2010.

xi, 88p., 297 mm (FT/UnB, Engenheiro, Controle e Automação, 2010). Trabalho de
Graduação – Universidade de Brasília.Faculdade de Tecnologia.

1.Automação

2. *Wireless*

3.BACnet

4.Controle

I. Mecatrônica/FT/UnB

REFERÊNCIA BIBLIOGRÁFICA

BORGES, L.P. & DORES, R.C., (2010). Automação predial sem fio utilizando BACnet/ZigBee com foco em economia de energia. Trabalho de Graduação em Engenharia de Controle e Automação, Publicação FT.TG-nº 06, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 88p.

CESSÃO DE DIREITOS

AUTOR: Lucília Pereira Borges e Rodrigo de Carvalho Dores.

TÍTULO DO TRABALHO DE GRADUAÇÃO:Automação predial sem fio utilizando BACnet/ZigBee com foco em economia de energia.

GRAU: Engenheiro

ANO: 2010

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.

AGRADECIMENTOS

*A toda minha família que sempre me apoiou
em todos os momentos dessa minha jornada.*

*Aos meus amigos que, com alegria e
companheirismo, fizeram com que essa
fase se tornasse menos árdua.*

Lucília Pereira Borges.

*A meus amigos que tornaram
essa jornada inesquecível.
A minha família por estar presente a todo momento,
especialmente a minha mãe
por toda sua ajuda nesse trabalho.
A minha namorada pelo seu apoio.
A todos meus colegas e professores
por todos momentos que nos trouxeram até aqui.*

Rodrigo

RESUMO

Neste trabalho foi implementada uma rede wireless composta por sensores e atuadores, com o propósito de diminuir o consumo de energia com sistemas de ar condicionado. Utilizaram-se sensores magnéticos sem fio para detecção do estado, aberto ou fechado, de janelas e portas. Tendo essa informação pode-se decidir ligar ou desligar o ar condicionado. Ao fazer isso, economiza-se energia que seria dispendida para refrigerar um ambiente com influência direta do meio externo. Para a monitoração e controle do sistema foi utilizado o software ActionView. Esse supervisor se comunica com um roteador através do protocolo BACnet MS/TP. A comunicação desse roteador com os módulos que realizam o controle e o sensoreamento é feita utilizando o protocolo BACnet/ZigBee .

ABSTRACT

In this project a wireless network of sensors and actuators was developed with the purpose of lessening energy consumption by air conditioning systems. Wireless magnetic sensors were used for the detection of the states, open or closed, of windows and doors. With this information at hand it is possible to decide whether to turn on or off the air conditioning. By doing this the energy wasted with the cooling of an environment with direct influence of an extern environment may be saved. For the supervision and control of the system it was used the software ActionView. This SCADA communicates with a router using the BACnet MS/TP protocol. This router communicates using the protocol BACnet/ZigBee with the modules that do the control and the sensing.

SUMÁRIO

FICHA CATALOGRÁFICA	III
REFERÊNCIA BIBLIOGRÁFICA	III
CESSÃO DE DIREITOS	III
CAPÍTULO 1 – INTRODUÇÃO E MOTIVAÇÃO.....	1
CAPÍTULO 2 – AUTOMAÇÃO PREDIAL	2
2.1 CONTROLE DE ACESSO	4
2.2 SENSORES	5
2.2.1 Sensores de contato físico	6
2.2.2 Sensores de Aproximação.....	6
CAPÍTULO 3 ECONOMIA ENERGÉTICA	9
CAPÍTULO 4 SISTEMAS SUPERVISÓRIOS	11
4.1 ARQUITETURA DO SISTEMA	11
4.2 SOFTWARES SCADA.....	12
4.3 SOFTWARE ACTIONVIEW	13
CAPÍTULO 5 PROTOCOLOS	14
5.1 ZIGBEE	16
5.1.1 Rede ZigBee	16
5.1.2 Topologia de redes ZigBee.....	18
5.1.3 Clusters.....	19
5.2 BACNET	20
5.2.1 Arquitetura	21
5.2.2 Camada de rede e roteamento.....	21
5.2.3 Serviços, objetos e propriedades	23
5.2.4 BACnet MS/TP.....	24
5.2.5 BACnet/ZigBee	25
5.3 BITCLOUD ZIGBEE® PRO STACK	26
5.3.1 Arquitetura	26
5.3.2 <i>Callback</i> , <i>task handler</i> e o agendador de tarefas	27
CAPÍTULO 6 AMBIENTE DE TESTES	29
6.1 AR CONDICIONADO HÍBRIDO.....	30
6.2 CENTRAL DE ALARMES PPA.....	30
6.3 SENSORES DE ABERTURA.....	30
6.3.1 Sensor magnético sem fio	31
6.3.2 Sistema com radiofrequência	33
6.3.3 Instalação dos sensores	33
6.4 ZIGBIT	37
6.4.1 Meshbean2 <i>board</i>	38
6.5 GRAVADOR.....	39

6.6	ATUADOR.....	40
6.6.2	Relé de Estado Sólido	41
6.7	REDE	41
6.7.1	ZigBee.....	42
6.7.2	BACnet/ZigBee	42
6.7.3	BACnet MS/TP.....	43
6.7.4	Roteamento	44
6.8	SOFTWARE EMBARCADO.....	45
6.8.1	Aplicação	46
6.8.2	Coordenador	48
6.8.3	End device	50
6.8.3.1	Módulo Sensor.....	51
6.8.3.2	Módulo Atuador.....	52
6.9	SUPERVISÓRIO ACTIONVIEW.....	52
6.9.1	Configuração do canal.....	53
6.9.2	Tela	54
6.9.3	Rotinas	55
CAPÍTULO 7 ANÁLISE DE RESULTADOS.....		56
7.1	ANÁLISE DA REDE	56
7.1.2	ZigBee.....	57
7.1.3	MS/TP	57
7.1.4	Análise Geral	58
7.2	ANÁLISE DA ECONOMIA DE ENERGIA	59
7.2.2	Testes	61
7.2.3	Análise dos testes.....	62
CAPÍTULO 8 CONCLUSÃO E TRABALHOS FUTUROS.....		68
REFERÊNCIAS BIBLIOGRÁFICAS		69
ANEXO 1 VB SCRIPT ON TIMER		71
ANEXO 2 VB SCRIPT ON VALUE CHANGE		73
ANEXO 3 VB SCRIPT ON MOUSE CLICK.....		74
ANEXO 4 LOG		75

LISTA DE FIGURAS

Figura 2.1. Ilustração cabeamento estruturado	3
Figura 2.2. Diagrama de bloco de um transdutor.....	5
Figura 2.3. Tipos de chaves de contato, [1].....	6
Figura 2.4. Esquema construtivo de um sensor indutivo, [1].....	7
Figura 2.5. Esquema construtivo de um sensor capacitivo, [1].	7
Figura 4.1. Um típico sistema SCADA,[5].....	12
Figura 5.1. Modelo ISO OSI com adaptação, [8].	14
Figura 5.2. Arquitetura ZigBee, [13].....	17
Figura 5.3. Topologia ZigBee, [16]	18
Figura 5.4. Funcionamento rede Mesh.[14]	19
Figura 5.5. Crescimento do número de vendedores Ids,[19].....	20
Figura 5.6. Camadas BACnet.....	21
Figura 5.7. Camadas BACnet/ZigBee [21].....	25
Figura 5.8. Arquitetura Bitcloud.	26
Figura 5.9. Comparação da execução Síncrona x Assíncrona, [24].....	27
Figura 6.1. Diagrama de Blocos do Sistema.....	29
Figura 6.2. Central de alarmes PPA	30
Figura 6.3. Reed-Switch, [26]	31
Figura 6.4. Sensor magnético.....	32
Figura 6.5. Foto dos resistores da central com a adaptação realizada.	34
Figura 6.6. Circuito acrescentado ao sensor magnético	35
Figura 6.7. Circuito optoacoplador.....	36
Figura 6.8. Ligações central, circuito optoacoplador, placa meshbean2.	37
Figura 6.9. Zigbit.	38
Figura 6.10. Placa Meshbean.....	39
Figura 6.11. Módulo Gravador, [28].....	40
Figura 6.12. Esquema elétrico do Módulo Gravador, [28].....	40
Figura 6.13. Módulo Atuador [28].	41
Figura 6.14. Esquema elétrico – SSR Teletronic	41
Figura 6.15. Camadas de aplicação.	46
Figura 6.16. Máquina de estados da aplicação	47
Figura 6.17. Máquina de estados do coordenador.....	48
Figura 6.18. Máquina de estados do end device.	50
Figura 6.19. Arvore de Sistema e Endereços das Variáveis	54
Figura 6.20. Tela gráfica ActionView.	55

Figura 7.1. Trecho do log de comunicação do ActionView	56
Figura 7.2. Medidor de energia trifásico	59
Figura 7.3. Controlador MT531Ri plus	60
Figura 7.4. Conversor EIA-485 para USB.....	60
Figura 7.5. Esquema de conexões PC, CONV32 e controlador	61
Figura 7.6. Variação de temperatura e umidade com funcionamento normal da sala.....	62
Figura 7.7. Variação de temperatura e umidade com a porta e janelas abertas.	63
Figura 7.8. Variação de temperatura e umidade em situações distintas.	64
Figura 7.9. Interferência de uma janela aberta no consumo.	65

LISTA DE TABELAS

Tabela 2.1. Tecnologia <i>wireless</i>	3
Tabela 6.1. Características elétricas do sensor PPA	32
Tabela 6.2. Características elétricas da central PPA	33
Tabela 6.3. Tabela de objetos BACnet do módulo Sensor	52
Tabela 6.4. Tabela de objetos BACnet do módulo Sensor	52
Tabela 7.1. Dados Coletados	67

LISTA DE SÍMBOLOS, NOMECLATURA E ABREVIACÕES

ASHRAE	<i>American Society of Heating, Refrigerating and Air-Conditioning</i>
BACnet	<i>Building Automation and Control Network</i>
CFTV	Circuito Fechado de Televisão
IEA	<i>International Energy Agency</i>
LEED	<i>Leadership in Energy and Environmental Design</i>
GBC	<i>Green Building Council</i>
WBCSD	<i>World Business Council for Sustainable Development</i>
SCADA	<i>Supervisory Control and Data Acquisition</i>
IEEE	<i>Institute of Electrical and Electronics Engineer</i>
UTR	Unidade Terminal Remota
CPU	<i>Central Processing Unit</i>
CLP	Circuito Logicamente Programável
IHM	Interface Homem-Máquina
UTM	Unidade Terminal Mestre
EIA	<i>Electronic Industries Alliance</i>
IED	<i>Intelligent Eletronic Device</i>
UCL	Unidades de Controle Local
ISO	<i>International Organization for Standardization</i>
OSI	<i>Open System Interconnection</i>
ISM	<i>Industrial, Scientific, Medical</i>
FFD	<i>Full Function Device</i>
RFD	<i>Reduced Funtion Device</i>
ZC	<i>ZigBee Coordinator</i>
ZR	<i>ZigBee Router</i>
ZED	<i>ZigBee End Device</i>
HVAC	<i>Heating, Ventilating, and Air Conditioning</i>

NPDU	Network layer Protocol Data Unit
APDU	Application layer Protocol Data Unit
DNET	<i>Destination Network</i>
DLEN	<i>Destination Length</i>
DADR	<i>Destination Address</i>
SNET	<i>Source Network</i>
SLEN	<i>Source Length</i>
SADR	<i>Source Address</i>
OID	<i>Object_Identifier</i>
RF	Radiofrequência
GPRS	<i>General Packet Radio Service</i>
LARA	Laboratório de Automação e Robótica
LED	Light Emiting Device
BZLL	BACnet ZigBee Link Layer
API	<i>Application Programming Interface</i>
ANSI	<i>American National Standard Institute</i>
ARCNET	<i>Attached Resource Computer Network</i>
MS/TP	<i>Master Slave / Token Pass</i>
PTP	<i>Picture Transfer Protocol</i>
VBS	<i>Visual Basic Script</i>

CAPÍTULO 1 – INTRODUÇÃO E MOTIVAÇÃO

Sistemas de automação predial tem como objetivo aumentar conforto, segurança e economia de energia. Sistemas que são mais comumente automatizados são os sistemas de ar condicionado. Para controlar esses sistemas é preciso levar em conta a circulação de ar devido a portas e janelas abertas. Em alguns casos, como nos sistemas de ar condicionado evaporativo, a troca de ar com o ambiente externo é necessária, enquanto que nos sistemas “split” não há essa troca. Para obter eficiência energética é necessário saber como janelas e portas abertas influenciam o sistema e quais são as melhores formas de realizar o controle.

Para implementar sistemas de automação em edifícios em construção pode-se utilizar cabeamento estruturado, porém cabear um edifício após sua construção tem um custo elevado. Por isso, nesses casos é recomendado utilizar uma rede sem fio para a comunicação entre os dispositivos desse sistema. Um protocolo sem fio que vem crescendo no mercado da automação predial é o ZigBee. Um outro protocolo que vem tendo muito destaque neste mercado é o BACnet. E nos últimos anos muitos estudos têm sido feitos com o intuito de juntar esses dois protocolos complementares. Este trabalho tem o objetivo de desenvolver dispositivos capazes de se comunicar por uma rede BACnet/ZigBee.

Este trabalho visa à redução do consumo energético em ambientes em que há um grande dispêndio de energia com aparelhos de ar condicionado. Isso foi feito por meio da utilização de sensores magnéticos que, em tempo real, enviaram o estado de abertura ou fechamento de janelas e porta, para que dessa forma seja desligado ou ligado o ar condicionado

Um sistema de alarme comercial da empresa PPA foi adaptado neste projeto para monitorar e fornecer o estado de portas e janelas a um sistema supervisor ActionView. A integração foi feita por uma rede BACnet/ZigBee. Um comando de desligamento do ar foi implementado caso existam portas ou janelas abertas. Ou seja, tivemos por meta utilizar ferramentas do controle de acesso e integrá-las à automação para alcançarmos nosso objetivo que é a economia de energia.

CAPÍTULO 2 – AUTOMAÇÃO PREDIAL

Com a abertura dos mercados de informática e de telecomunicações no Brasil, a partir da década de 90, possibilitou-se a popularização de diversas tecnologias de controle e serviços de automação. Nesse sentido, a automação predial se desenvolveu tendo como principal objetivo a melhoria do estilo de vida dos ocupantes de uma edificação através do aumento do conforto ambiental, da segurança física e do aumento da eficiência energética, tornando o ambiente mais confortável, seguro e eficiente.

O uso racional de energia elétrica, a segurança e o conforto dos usuários e a preocupação com o meio ambiente são fatores relevantes para o uso de tecnologias aplicadas em edifícios. Os edifícios inteligentes são concebidos para causarem o mínimo de impacto ambiental, com o menor consumo de energia possível e melhor aproveitamento de recursos, além de proporcionar maior segurança e conforto para seus usuários. A automação é bastante necessária e está sendo uma exigência da maioria das novas construções. E os prédios que ainda não foram automatizados, com certeza, o serão no futuro, gerando um conflito com relação à infra-estrutura exigida. Assim, deve-se tomar uma decisão: modificar a estrutura do prédio ou construir outro com tais exigências.

Na construção de uma nova estrutura torna-se mais viável a automação em que a integração é feita por meio de cabeamento estruturado. Este é um sistema flexível que comporta em uma única infraestrutura de cabos e componentes, diversos serviços de comunicação como voz, dados, imagem, automação e outros. Dessa maneira, os projetos interagem entre si, a arquitetura com o sistema de refrigeração, a parte elétrica, hidráulica e segurança, o que reduz os custos no gerenciamento da infraestrutura e os custos relacionados a esta. A Figura 2.1 demonstra essa integração.

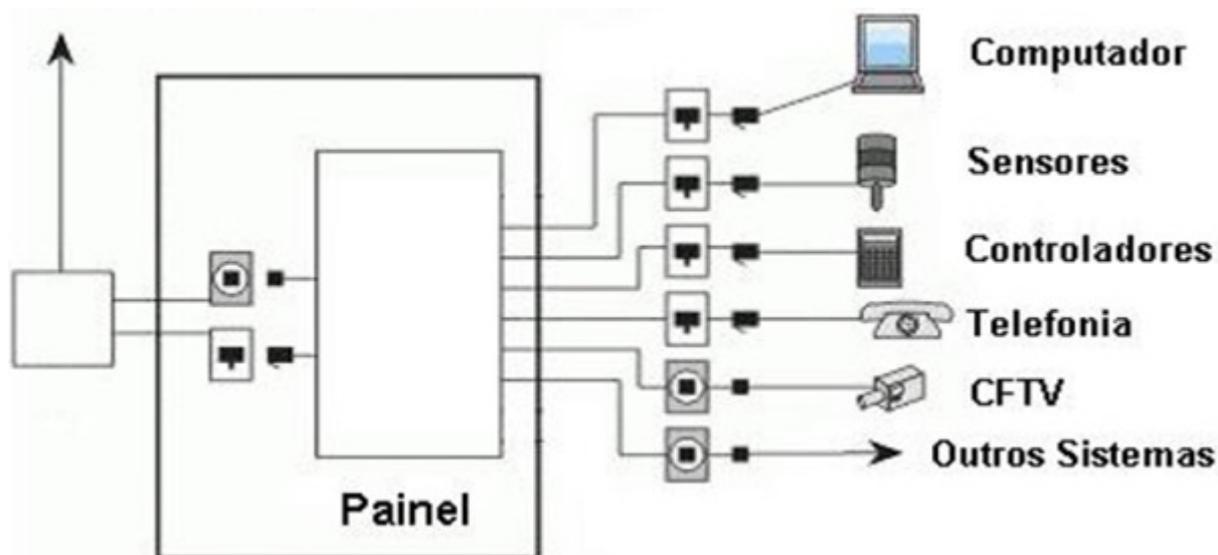


Figura 2.1. Ilustração cabeamento estruturado.

Esses são denominados prédios inteligentes, pois já nascem com o conceito de automação e integração no projeto. Prédios já construídos possuem obstáculos que podem comprometer a implementação da automação e das vantagens que esta proporciona. Geralmente, nesses prédios não há local para a inserção de mais cabos e fios e não existe a possibilidade de se modificar a infraestrutura que a automação exige. É nesse sentido que o emprego de dispositivos sem fio na automação predial vem se tornando mais viável.

Algumas tecnologias podem auxiliar nesse processo denominado *retrofit*, como, por exemplo, tecnologias wireless. Essas tecnologias estão disponíveis em sensores, equipamentos de controle e computadores e podem formar redes sem estrutura física permitindo que construções antigas, ou pouco automatizadas, tenham seus domínios interligados sem a necessidade de se modificar a estrutura do prédio.

As redes sem fio podem ser empregadas em diversas localidades dentro de um edifício. Podendo abranger desde a conexão de um computador a uma impressora até a implementação de sensores e atuadores. A Tabela 2.1 mostra algumas aplicações da tecnologia wireless em um edifício

Tabela 2.1. Tecnologia *wireless*.

Elementos a serem automatizados	Tecnologia
Ar condicionado	ZigBee ou Proprietária
Iluminação	ZigBee ou Proprietária
Bombas de água	ZigBee ou Proprietária
Alarme de Incêndio	ZigBee ou Proprietária
Segurança e controle de acesso	ZigBee Bluetooth ou Proprietária

As tecnologias *wireless* utilizam o ar como meio físico de transmissão sendo que a comunicação entre os dispositivos pode ser feita por luz infravermelha ou radiofrequência. As tecnologias por radiofrequência são as mais utilizadas na elaboração de redes sem fio e as tecnologias por luz infravermelha são utilizadas em conexões entre dois dispositivos próximos.

O uso de uma determinada tecnologia sem fio depende da aplicação, do número de elementos a serem colocados na rede, do custo, da quantidade de dados a serem transmitidos.

Com o objetivo de resolver a questão de incompatibilidade entre dispositivos de diversos fabricantes, o *Institute of Electrical and Electronic Engineers* (IEEE) e o *European Telecommunications Standardisation Institute* (ETSI) desenvolveram vários padrões para redes locais sem fio. Equipamentos que utilizam padrões definidos pelo IEEE são os recomendados para a elaboração de projetos de automação sem fio. Os padrões definidos pelo IEEE possuem a vantagem de serem mais conhecidos, com maior número de fabricantes e menor preço. Dentre os padrões mais conhecidos encontram-se o Bluetooth, o IEEE 802.11b e o *ZigBee*.

2.1 CONTROLE DE ACESSO

Constantemente surgem novas necessidades de segurança, identificação e economia, fazendo com que as atuais tecnologias ofereçam tranquilidade e simplicidade. O aumento do sentimento de insegurança nas grandes cidades somado à necessidade de identificação de funcionários e visitantes de estabelecimentos e o monitoramento de ambientes e pessoas impulsionam uma demanda por sistemas de controle de acesso e pontos automatizados.

Inserido no contexto da automação predial, o controle de acesso é um dos principais motivos para se optar em automatizar. Esse é muito utilizado em áreas em que a necessidade de ter o conhecimento exato da passagem da pessoa é essencial.

O controle de acesso de qualquer ambiente pode ser feito através de uma central ou pela internet. Dentre as tecnologias disponíveis nesse segmento encontram-se:

- Autenticação/identificação de pessoas através de tecnologia biométrica (cartões, leitores biométricos);
- Leitoras de cartão por proximidade;
- Cancelas;
- Catracas;

- Sensores de presença;
- Sensores de contato com e sem fio;
- Barreiras horizontais, entre outras.

Na década de 80, os sistemas de controle de acesso começaram a se popularizar devido ao barateamento dos computadores de pequeno porte. Desde então, passaram a ter processamento suficiente para controlar as transações em tempo real.

Hoje em dia, os sistemas de controle de acesso tornam viável a integração com leitores biométricos, controle de estacionamento, interligação com alarme de incêndio, sensores de portas e janelas, integração com CFTV, etc.

2.2 SENSORES

Sensores são dispositivos que mudam seu comportamento sob a ação de uma grandeza física, podendo fornecer diretamente ou indiretamente um sinal que indica esta grandeza. Quando operam diretamente, convertendo uma forma de energia em outra, são chamados de transdutores. Os de operação indireta alteram suas propriedades, como a resistência, a capacitância ou a indutância, sob ação de uma grandeza, de forma mais ou menos proporcional. O sinal de um sensor pode ser usado para detectar e corrigir desvios em sistemas de controle, e nos instrumentos de medição, que frequentemente estão associados aos sistemas de controle de malha aberta (não automáticos), orientando o usuário. Portanto, para tal definição, nos referimos àqueles dispositivos que transformam uma grandeza física em uma elétrica, com o mesmo significado de sensores. O diagrama de bloco genérico de um transdutor é mostrado na Figura 2.2.

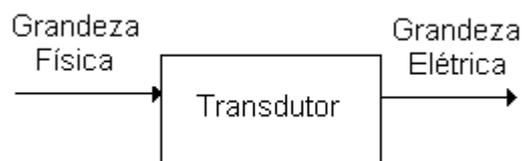


Figura 2.2. Diagrama de bloco de um transdutor.

A grandeza elétrica de saída de um transdutor pode ser uma tensão, uma corrente, uma resistência, etc. Dependendo da natureza da grandeza elétrica de saída, os transdutores são subdivididos em analógicos e digitais: “Para uma natureza física contínua na entrada, um transdutor analógico faz corresponder uma grandeza elétrica contínua na saída e proporcional à grandeza física medida, no entanto um transdutor digital faz corresponder uma sucessão de sinais digitais.” Resumidamente podemos dizer que os sensores podem ser de dois tipos:

- Digital: é aquele cuja saída assume apenas dois estados, ON/Off.

- Analógico: é aquele cuja saída varia proporcionalmente com a variação física percebida pelo sensor. Os sinais elétricos de saída mais comum são:

Corrente - 4 a 20mA

Tensão - 0 a 10v

2.2.1 Sensores de contato físico

São sensores que necessitam estar em contato com a grandeza a ser monitorada. Um contato mecânico e uma força resultante entre o sensor e o objeto são necessários para efetuar a detecção. Nesse tipo de sensor, há a presença de uma chave de contato que consiste em um atuador mecanicamente ligado a um conjunto de contatos. Quando um objeto entra em contato físico com o atuador, o dispositivo opera os contatos para abrir ou fechar uma conexão elétrica. As chaves de contato podem apresentar as configurações detalhadas na Figura 2.3.

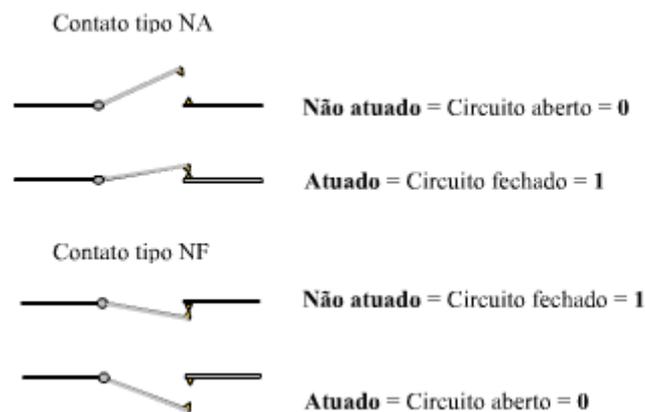


Figura 2.3. Tipos de chaves de contato, [1].

2.2.2 Sensores de Aproximação

São sensores que detectam o evento ou a grandeza sem que haja necessidade de estar em contato físico com a mesma, além do que, são blindados, à prova de vibração, etc. Podem ser indutivos, capacitivos, ópticos dentre outros.

Os sensores indutivos executam uma comutação eletrônica, quando um objeto metálico entra dentro de um campo eletromagnético de alta frequência, produzido por um oscilador eletrônico. Sua instalação se dá em máquinas ferramentas, máquinas operatrizes, de embalagens, têxteis, correias transportadoras e na indústria automobilística, para resolver problemas gerais de automação. Na Figura 2.4 pode ser observada sua composição em blocos.

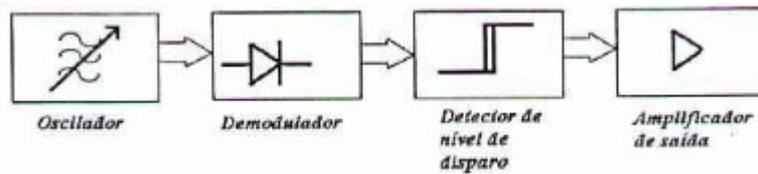


Figura 2.4. Esquema construtivo de um sensor indutivo, [1].

Em que:

Oscilador: diminui a freqüência de oscilação quando um evento for detectado.

Demodulador: converte o sinal do oscilador em nível de tensão cc.

Detecor de nível de disparo: dispara quando o oscilador diminui a freqüência.

Amplificador de saída: amplifica o sinal gerado pelo sensor e entrega-o a carga.

Funcionamento:

O oscilador com auxílio de uma bobina gera um campo magnético de alta freqüência. Este campo é direcionado para fora do elemento ativo, formando uma região de sensibilidade denominada de face sensível, chamada de distância de comutação. Quando um corpo metálico está distante da face sensível e, dentro da distância de comutação, este metal amortece a oscilação, provocando a comutação eletrônica, ou seja, faz o sensor mudar de estado. Com a retirada do corpo metálico da distância de comutação, o oscilador volta a trabalhar normalmente e o sensor volta a seu estado normal.

Os sensores capacitivos, assim como os sensores indutivos, também podem efetuar um chaveamento eletrônico sem qualquer contato físico. Estes sensores foram desenvolvidos para atuarem na presença de materiais orgânicos, plásticos, vidro, líquido, além de metais. Sua aplicação se dá em detectores de nível em tanques, contagem de garrafas (cheias ou vazias), contagem de embalagens plásticas, limitadores de carretéis, etc. Na Figura 2.5 pode ser vista sua composição em blocos.

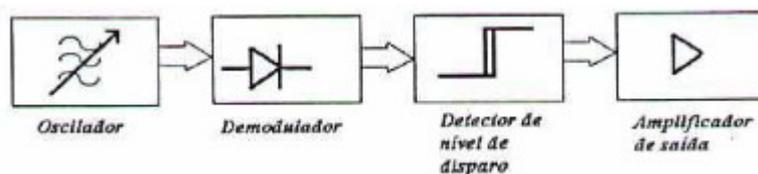


Figura 2.5. Esquema construtivo de um sensor capacitivo, [1].

Em que:

Oscilador: diminui a freqüência de oscilação quando um evento for detectado.

Demodulador: converte o sinal do oscilador em nível de tensão cc.

Detector de nível de disparo: dispara quando o oscilador diminui a frequência.

Amplificador de saída: amplifica o sinal gerado pelo sensor e entrega-o a carga.

Funcionamento:

Conforme pode ser notado na Figura 2.5 o esquema em blocos de um oscilador capacitivo é igual ao do indutivo. A diferença entre eles reside no fato de que no sensor capacitivo o princípio de funcionamento está baseado na variação do dielétrico no meio em que o sensor está inserido. Quando nesta região penetrar algum objeto, este provoca a variação do dielétrico e, conseqüentemente a variação da frequência do oscilador, variação esta que é detectada e transformada em um nível de tensão cc. Com a retirada do objeto da distância de comutação, o oscilador volta a trabalhar normalmente e o sensor volta ao seu estado normal.

Os sensores ópticos são fabricados segundo a tecnologia da emissão de irradiação infravermelha modulada, sendo divididos em três sistemas. Independente do sistema que um sensor óptico é construído, ele é totalmente imune à iluminação ambiente, quer ele seja manual ou artificial, devido o sensor estar sintonizado na mesma frequência de modulação do emissor.

No Sistema por Barreira (Unidirecionais), o elemento emissor da irradiação infravermelha, é alinhado frontalmente a um receptor de infravermelho, a uma distância pré-determinada e específica para cada tipo de sensor. Qualquer interrupção desta irradiação deixará de atingir o receptor, o que ocasionará um chaveamento eletrônico.

No Sistema por Difusão (Retroreflexivo), os elementos de emissão e recepção infravermelha, estão montados justapostos em um conjunto óptico, direcionados para a face sensível do sensor. Os raios infravermelhos emitidos refletem sobre a superfície de um objeto e retornam em direção de um receptor, a uma distância determinada como distância de comutação, que provoca o chaveamento eletrônico, desde que o objeto possua uma superfície não totalmente fosca.

O Sistema por Reflexão possui características semelhantes ao sistema por difusão, diferindo no sistema óptico. Os raios infravermelhos emitidos, somente refletem em um espelho prismático especial, colocado a uma certa distância, dentro da distância de comutação, frontalmente à face sensível do sensor, e retornam em direção ao receptor, formando uma barreira óptica. A comutação ocorre quando se retira o espelho ou quando interrompe o feixe de raios infravermelho entre o sensor e o espelho com algum objeto de qualquer natureza.

CAPÍTULO 3 ECONOMIA ENERGÉTICA

No Brasil, observa-se um grande desperdício de energia. Muitas pessoas não a usam com responsabilidade, o que colabora para uma perspectiva de crise energética futura.

Investir na economia energética colabora não só para tornar o ambiente mais sustentável como também para se gastar menos. De acordo com pesquisa divulgada pela Agência Internacional de Energia (IEA), o brasileiro paga mais pela energia elétrica do que consumidores de alguns países desenvolvidos com renda per capita superior.

Quanto menos energia utilizarmos, menor será a necessidade de produzi-la, o que é uma grande vantagem para o planeta, já que grande parte da eletricidade gerada atualmente é proveniente da queima de combustíveis fósseis. Esses combustíveis emitem gases de efeito estufa, como o dióxido de carbono (CO₂), um dos principais causadores do aquecimento global. Outras formas de se obter energia elétrica são por meio de hidrelétricas, que incorre no alagamento de grandes áreas, e de usinas nucleares que geram lixo atômico [2].

Constata-se que, em prédios comerciais, os maiores gastos energéticos devem-se à utilização de ar condicionado. Caso na fase de construção do edifício seja feito um projeto arquitetônico com materiais adequados, canais de ventilação, equipamentos de baixo consumo, com certeza, esse prédio será viável do ponto de vista de se consumir pouca energia. Posteriormente, procedimentos como diminuição do uso de aparelhos que consomem energia, levam a uma economia ainda maior.

Os primeiros edifícios verdes (green buildings, em inglês) surgiram na década de 1970 na Holanda, na Alemanha e nos países nórdicos. A sede do Parlamento alemão (Reichstag) tem um gerador que não só produz a própria energia com base em combustíveis renováveis como envia o excedente para as construções vizinhas. Na Noruega, foi inaugurada a primeira “cadeia ecológica” do mundo. O presídio da Ilha Bastoey, destinado a 115 detentos, usa painéis solares e recicla quase todo o material consumido [3]. Estima-se que há nos Estados Unidos cerca de 6 mil edifícios verdes e a tendência é aumentar para 100 mil. Esses edifícios são certificados pelo selo LEED (Liderança em Energia e Projetos Ambientais, na sigla em inglês), da organização Green Building Council (GBC) e representam grandes empresas, bancos, agências governamentais e prédios de escritórios.

Nos Estados Unidos, os edifícios comerciais consomem 65% da energia elétrica distribuída no país, sendo que apenas ar-condicionado, gasta mais da metade da energia de um prédio. O uso do ar-condicionado também é responsável por 50% dos gastos com energia no Brasil. Esse é apenas um dos exemplos dos problemas relacionados às construções que

a nova onda de prédios verdes pretende enfrentar. Atualmente, a tecnologia já existente pode aumentar a eficiência energética em 35% e reduzir os custos com aquecimento, resfriamento, ventilação e água. De acordo com pesquisa realizada pelo Conselho Mundial de Negócios para o Desenvolvimento Sustentável (WBCSD), combinar os materiais certos também pode significar economia de energia [3].

CAPÍTULO 4 SISTEMAS SUPERVISÓRIOS

Sistemas supervisórios, ou SCADA (*supervisory control and data acquisition*), coletam informação, transferem para uma central, realizam qualquer análise ou controle e mostram a informação para o usuário. Os controles necessários são enviados de volta para o processo. Sistemas SCADA são definidos de diversas maneiras por diferentes autores. De acordo com a IEEE[4], define-se como um sistema operando com sinais codificados sobre canais de comunicação para controlar UTRs (Unidade Terminal Remota).

Estes sistemas fornecem dados em tempo real para a administração sobre as operações, implementam controle mais eficiente, aumentam a segurança da planta e reduzem custos de operação. Essas vantagens se tornaram possíveis com a combinação de *hardware* e *software* em um sistema SCADA com protocolos de comunicação e a ligação do sistema com redes externas incluindo a internet.

Sistemas supervisórios existem desde que existem sistemas de controle. Os primeiros SCADAs eram painéis que mostravam os dados para um operador através de mostradores analógicos e luzes e realizavam o controle utilizando diversos botões. Muitos dispositivos desses ainda são utilizados para controle e supervisão de diferentes locais. Com a criação da CPU e outros dispositivos eletrônicos, fabricantes incorporaram a eletrônica digital no controle de relés. CLPs (Controlador Logicamente Programável) são até hoje um dos sistemas mais utilizados na indústria. Conforme a necessidade de monitorar e controlar mais dispositivos cresceu, os CLPs se tornaram mais distribuídos e o sistema ficou mais inteligente e de menor tamanho.

4.1 ARQUITETURA DO SISTEMA

Sistemas SCADA seguem uma arquitetura que consiste de alguns elementos básicos [4][3]:

- Operador: homem que irá monitorar o sistema;
- Interface homem-máquina (IHM): apresenta os dados para o operador e fornece diferentes interfaces para facilitar o controle e supervisionamento;
- Unidade Terminal Mestre (UTM): equivalente à unidade mestre em uma arquitetura mestre/escravo. A UTM apresenta dados para o operador através da IHM, recebe dados do campo, e envia sinais de controle para as UTRs;
- Comunicação: pode ser através de cabos (como *Ethernet*, EIA-232 ou EIA-485), *wireless* (*Wi-Fi* ou *ZigBee*) ou rádio;

- Unidade Terminal Remota (UTR): Funciona como o escravo na arquitetura mestre/escravo. Envia sinais de controle para os dispositivos e recebe dados desses dispositivos e envia os dados para a UTM. Uma UTR pode ser um CLP ou uma IED (*intelligent electronic device*, Dispositivo Eletrônico Inteligente), por exemplo.

4.2 SOFTWARES SCADA

O *software* SCADA pode ser dividido em dois tipos, proprietário ou aberto. Fabricantes desenvolvem proprietário para que ele se comunique apenas com seu hardware. O problema é a alta dependência do fabricante. Sistemas abertos têm ganhado popularidade devido à interoperabilidade com diferentes equipamentos. O *ActionView*, *software* SCADA da Spin Engenharia de Automação Ltda, é um exemplo de sistema aberto. Um sistema típico está indicado na Figura 4.2.

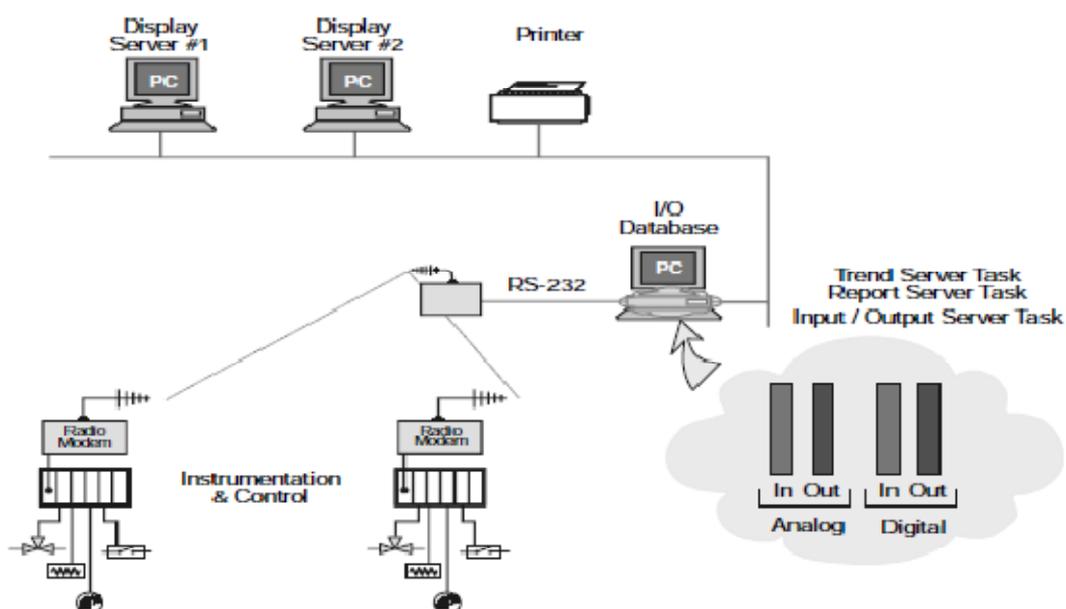


Figura 4.1. Um típico sistema SCADA,[5].

As características de um software SCADA são:

- Interface gráfica;
- Alarmes;
- Gráficos;
- Log de Eventos;

- Interface com UTRs;
- Banco de dados;
- Rede;
- Tolerância a falhas e redundâncias;
- Processo distribuído Cliente\Servidor;
- Monitoramento em tempo real e
- Drivers para controle de diferentes componentes.

4.3 SOFTWARE ACTIONVIEW

ActionView é um *software* SCADA desenvolvido pela Spin Engenharia de Automação Ltda. É um sistema aberto e configurável para qualquer sistema de supervisão e controle. Possui todas as características de um SCADA mencionadas anteriormente. Por ser um sistema aberto, possui a possibilidade de se comunicar utilizando diversos protocolos. Alguns deles são BACnet (MS/TP e IP), Modbus, IEC-61850, DNP30, OPC e outros inclusive alguns protocolos proprietários.

O sistema é separado em módulos. Existem 8 tipos de módulos, porém para este trabalho só serão abordados os utilizados. Os módulos utilizados foram AVMaster e o Módulo de Comunicação BACnet. No Manual *ActionView* Introdução [6], estes módulos foram definidos da seguinte forma:

AVStudio: licença para executar em um computador o módulo de desenvolvimento de aplicativos, responsável pela geração da base de dados de tempo real e das telas. Um módulo pode desenvolver um número ilimitado de aplicações, e independe do número de pontos

AVServer: licença de um módulo “*Run-Time*” servidor de comunicação e dados de tempo real. O *AVServer* é limitado por número de tags e vem acompanhado de um módulo de visualização.

AVMaster: licença para executar em um computador tanto o módulo *AVStudio*, como o módulo *AVServer*. Vem acompanhado também, de um módulo emulador de campo, que permite testar todas as funcionalidades das aplicações parametrizadas.

Módulos de Comunicação: o software inclui “módulos de comunicação” compatíveis com os protocolos dos principais fabricantes de UCLs (Unidades de Controle Local), UTRs (Unidades Terminais Remotas), CLPs (Controladores Lógicos Programáveis), relés e outros IEDs (Dispositivos Eletrônicos Inteligentes).

CAPÍTULO 5 PROTOCOLOS

Protocolos são uma parte fundamental da automação. Eles definem o formato e a ordem das mensagens trocadas entre dois ou mais dispositivos e também as ações tomadas na transmissão e/ou recebimento de uma mensagem [7]. Para que dois dispositivos se comuniquem é preciso que eles falem o mesmo protocolo.

Como sistemas SCADA precisam fornecer informações em tempo real é preciso que se conheça o tempo gasto em cada operação do sistema, principalmente na comunicação. Assim protocolos para automação se diferenciam de outras aplicações, porque além de precisarem entregar a informação de forma confiável é preciso que se faça isso num período de tempo determinado. Para garantir isso, diversos fabricantes foram desenvolvendo seus protocolos proprietários. O próximo passo na evolução dos protocolos foram os protocolos abertos. Com isso, qualquer fabricante poderia utilizar estes protocolos facilitando a integração.

Diversos modelos foram criados para definir o uso e a organização dos protocolos para facilitar seus projetos. Os modelos por camada foram bem aceitos e um dos mais aceitos é o modelo da ISO (*International Organization for Standardization*) OSI (*open systems interconnection*)[4]. Esse modelo é dividido em sete camadas como mostrado na Figura 5.1.



Figura 5.1. Modelo ISO OSI com adaptação, [8].

A camada física é a conexão entre os dispositivos e o sinal eletrônico que transmite a informação.

A camada de enlace (*Data Link*) organiza a mensagem em frames, regula o acesso à camada física e controla o endereçamento. Pode ser responsável por tratamento de erros e controle de fluxo.

A camada de rede é responsável pela tradução do endereço global para local, roteamento, sequenciamento, controle de fluxo, controle de erro e multiplexação.

A camada de transporte garante a entrega da mensagem ao destino final, trata da segmentação, sequenciamento, controle de fluxo e tratamento de erros. Apesar de ter funções similares à camada de enlace seu escopo é diferente. Seu tratamento na mensagem é apenas para o destino final enquanto a de enlace é ponto a ponto sendo tratada em cada nó que a mensagem passar.

A camada de sessão é usada para estabelecer a conexão e manter longos diálogos entre os dispositivos.

A camada de apresentação é para encriptar/descriptar mensagens e tradução das mensagens.

A camada de aplicação fornece os serviços de comunicação que a aplicação precisar.

Cada camada se comunica com a próxima e com a anterior. O pacote vai passando de camada em camada até chegar à sua aplicação de destino. Estas camadas servem para serem diretivas de como projetar o protocolo e não para o restringir. Inclusive, na maioria das vezes nem todas são utilizadas [9]. Em aplicações de tempo real onde há restrições de tempo, este modelo não é recomendado. Porém, muitas vezes é utilizado um subconjunto destas camadas onde algumas camadas ou são removidas ou unidas a outras.

O projeto de protocolos para Sistemas SCADA precisa levar diversos fatores em consideração, sendo que os principais são eficiência, segurança, flexibilidade e facilidade de implementação[10]. Segurança é a capacidade de detectar erros. Flexibilidade é permitir o tráfego de diversos tipos de informação. Eficiência é definida como a relação entre bits de informação transmitidos e o total de bits transmitidos.

Mensagens são divididas em três partes[10]: estabelecimento da mensagem que inclui sinais de sincronização, informações de transmissão da mensagem e o encerramento da mensagem. A informação é a parte que contém os dados que precisam ser transmitidos e o encerramento da mensagem inclui checagens de segurança e formas de determinar o fim da mensagem.

O desenvolvimento de protocolos precisa levar em conta tantos fatores que é preciso escolher qual é o objetivo deste novo protocolo. Protocolos projetados com o foco em automação predial são novos se comparados a protocolos da automação industrial. *ZigBee*

e *BACnet* são exemplos de protocolos que foram desenvolvidos principalmente para o setor predial.

5.1 ZIGBEE

A comunicação é um dos aspectos mais importantes de projetos de automação. Como já foi dito, a comunicação sem fio vem se tornando cada vez mais confiável e assim, sendo uma ferramenta importante para a automação.

Para este tipo de implementação, a tecnologia que melhor se adéqua é do grupo das LR-WPAN (*Low Rate - Wireless Personal Area Network*). Existem diversas tecnologias nesta área. Algumas das que possuem maior expressão no mercado mundial atualmente são: *ZigBee* e *WirelessHart*. Apesar de essas tecnologias terem muitas similaridades, cada uma, foca em um tipo de aplicação principal. O *ZigBee* foi desenvolvido para automação predial e residencial e vem expandindo seu uso. Foi considerado o protocolo mais adequado à aplicação do presente trabalho. O *WirelessHart* foi desenvolvido principalmente para aplicações industriais [11]. *Bluetooth* e *Wi-fi* são as tecnologias de comunicação wireless mais comuns, porém elas não se enquadram nos requisitos comumente necessários a automação. O comparativo entre algumas destas tecnologias já foram realizadas em trabalhos anteriores e não serão discutidos aqui [8], [12].

5.1.1 Rede ZigBee

O protocolo *ZigBee* foi idealizado pela *ZigBee Alliance* que é uma parceria de diversas empresas que visou criar um produto que reunisse baixo custo, baixo consumo de energia, comunicação wireless para aplicações prediais, residenciais e industriais.

O protocolo *ZigBee* foi desenvolvido a partir de uma combinação de protocolos para comunicação wireless. Utiliza o padrão IEEE 802.15.4. A Figura 5.2 apresenta a arquitetura do *ZigBee*, onde cada camada tem sua função e interfaces bem definidas e mostra quais foram definidas pelo padrão da IEE 802.15.4, pela *ZigBee Alliance* e o que é responsabilidade da aplicação.

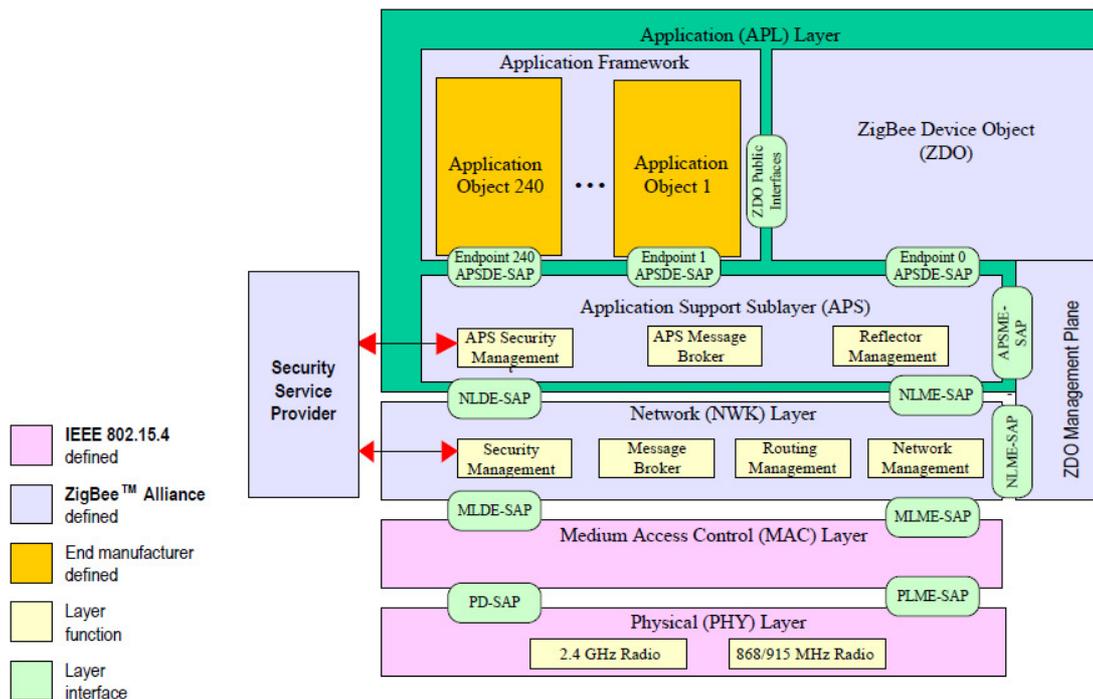


Figura 5.2. Arquitetura ZigBee, [13].

Esse protocolo funciona na frequência de ISM (*Industrial, Scientific, Medical*) que é de 868 MHz na Europa, 915 Mhz nos EUA e Austrália e 2.4 Ghz na maioria das outras jurisdições incluindo o Brasil. Possui taxas de transmissão entre 10 Kbps e 115 Kbps e um alcance de 10 a 100 metros. Uma de suas vantagens é poder comunicar cada membro entre si. Um dispositivo pode se comunicar com mais de 65000 [13].

Apesar de operar na mesma frequência que a tecnologia Wi-Fi, em testes realizados pela *Schneider Electric* [14], foi comprovada que a existência em situações reais de automação residencial não compromete a entrega dos pacotes. Somente em casos extremos, com a rede *Wi-Fi* sendo usada com um tráfego muito acima do que é normalmente utilizado, pode ocorrer a falha na entrega das mensagens, porém isso ocorre principalmente quando a rede *Wi-Fi* está operando no modo IEEE 802.11b, quando se usa uma rede IEEE 802.11g, ou melhor, a interferência diminui muito. Comprovaram ainda em seus experimentos que se for utilizada também a retransmissão por aplicação obtém-se 0% de falha na entrega.

5.1.2 Topologia de redes ZigBee

Uma rede ZigBee possui dois tipos de dispositivos: FFD e RFD [15].

FFD – *Full Function Device* (Dispositivos de Funções Completas) – São dispositivos mais complexos e por isso consomem mais energia. Numa topologia *ZigBee* eles podem ser Coordenador, Roteador ou um dispositivo final. FFDs podem se comunicar com quaisquer membros da rede.

RFD – *Reduced Function Device* (Dispositivo de Funções Reduzidas) – São dispositivos mais simples onde sua pilha de protocolo pode ser implementada usando mínimos recursos possíveis de *hardware*. Só podem se comunicar com FFDs. São os dispositivos finais.

Existem três classes de dispositivos [15]:

ZC – *ZigBee Coordinator* (coordenador *ZigBee*) – Só pode ser implementado como um FFD. É responsável pela inicialização, distribuição de endereços, manutenção da Rede, reconhecimento de todos os Nós, entre outras funções, podendo servir como ponte entre várias outras Redes *ZigBee*.

ZR - *ZigBee Router* (Roteador *ZigBee*) - Só pode ser implementado através de um dispositivo FFD. É um intermediário entre o dispositivo final e o coordenador. Com os roteadores é possível expandir o alcance de uma rede *ZigBee*.

ZED - *ZigBee End Device* (Dispositivo final *ZigBee*) – É onde os atuadores ou sensores serão hospedados. Pode ser um FFD ou um RFD. É o nó que consome menos energia, pois passa mais tempo no modo *Sleep* (dormindo).

O ZigBee pode utilizar três tipos de rede: árvore, estrela e malha (geralmente chamada de *mesh*). Na Figura 5.3 são apresentados exemplos de topologias *ZigBee*.

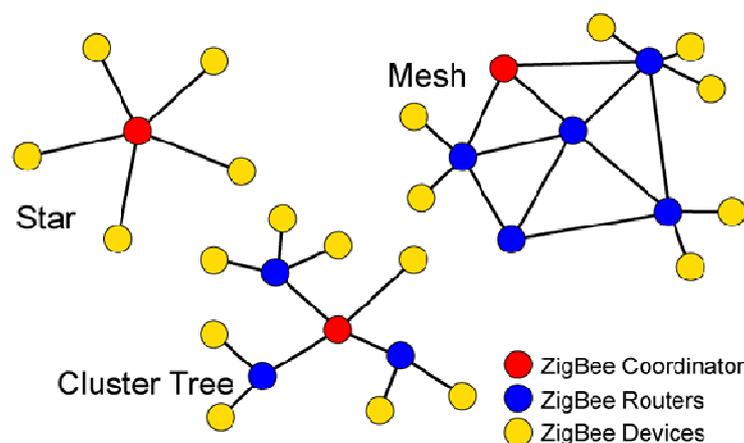


Figura 5.3. Topologia ZigBee, [16].

O ZigBee tem a capacidade de gerenciar uma rede inteligente do tipo *Mesh*. Com isso, o pacote percorre o melhor caminho entre os roteadores para chegar a seu destino. No caso de falha de algum roteador os pacotes adotam um outro caminho. Assim, quanto maior o número de dispositivos ZR na rede menor a chance de falha de entrega de pacotes. Um exemplo é mostrado na Figura 5.4, em que na ocorrência de uma interferência atrapalhando um caminho, é adotado um caminho melhor para garantir a entrega. Com isso quanto maior o número de ZR na rede mais segura ela se torna.

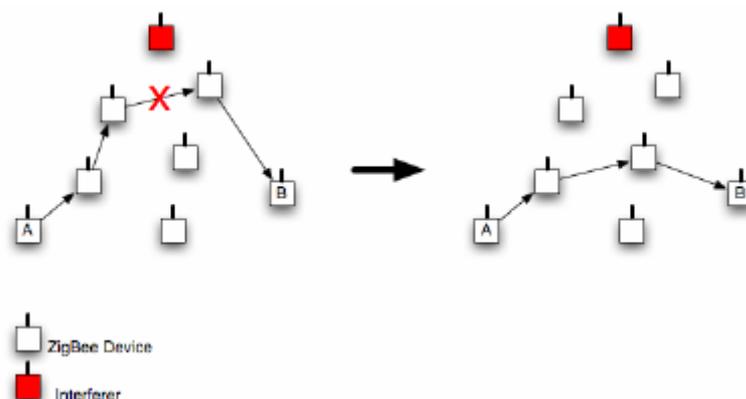


Figura 5.4. Funcionamento rede Mesh [14].

5.1.3 Clusters

O ZigBee possui uma gama muito grande de diferentes usos, sendo uma opção o tunelamento de protocolos, porém também é capaz de realizar a automação como o único protocolo na rede. Além disso, existem diversas funcionalidades diferentes que podem ser implementadas, específicas para cada caso. Para garantir a interoperabilidade sem necessitar que todos dispositivos tenham que implementar um grande número de funcionalidades, foram criados os *Clusters*.

Clusters são uma coleção de atributos e comandos que define a interface de comunicação. Os dispositivos devem implementar a interface de servidor e cliente respectivamente (referência ao *ZigBee Cluster Library*[17]). Assim, dispositivos com finalidades específicas como controle HVAC não precisam implementar funções de outros tipos de controle, como iluminação por exemplo.

5.2 BACNET

O BACnet (*Building Automation and Control network*) é um protocolo orientado a objeto desenvolvido com o apoio da ASHRAE (*American Society of Heating, Refrigerating and Air-Conditioning Engineers*) com o objetivo principal da automação de sistemas HVAC (*Heating, Ventilating, and Air Conditioning*), que já foi expandido para diversos outros tipos de automação e controle como iluminação, controle de acesso e sistemas de incêndio. Seu desenvolvimento começou em 1987 e seu uso vem crescendo muito desde então [18]. Cada fabricante pode pedir um *Vendor Id* para ASHRAE, e com isso seus equipamentos podem ter no cabeçalho das mensagens a informação de quem é o fabricante. A quantidade de números entregues pela ASHRAE vem crescendo muito conforme pode-se observar na Figura 5.5.

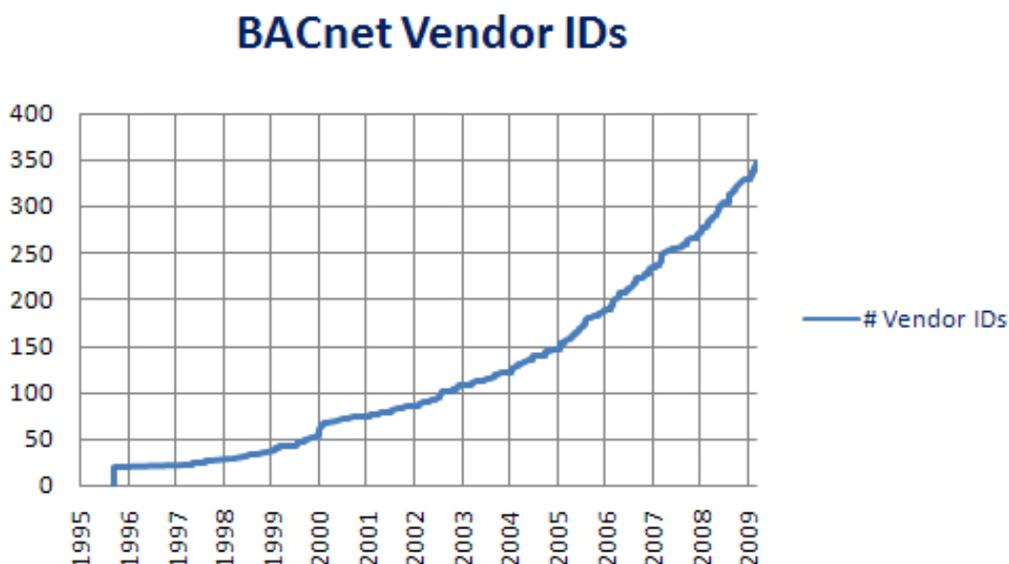


Figura 5.5. Crescimento do número de vendedores Ids,[19].

Este protocolo foi desenvolvido como um padrão americano, está se tornando padrão europeu e está próximo de entrar para os padrões ISO. Algumas de suas vantagens são [20]:

- Desenvolvido especificamente para automação e controle predial. Com isso, está voltado para suprir todas as necessidades específicas deste setor;
- Não depende de nenhuma tecnologia específica de rede. Além de não ficar preso a nenhuma tecnologia, o padrão vem expandindo o número de tecnologias que podem ser usadas;

- Não restringe o tamanho da aplicação. Pode ser completamente escalável da pequena até uma grande aplicação;
- Podem ser criadas funções específicas do fabricante sem alterar sua interoperabilidade e
- Está em constante desenvolvimento e acompanha as necessidades do mercado.

5.2.1 Arquitetura

A arquitetura do BACnet segue um modelo simplificado do modelo ISO OSI. Utiliza quatro das sete camadas, aplicação, rede, enlace e física. O BACnet pode ser usado em cima de diversas formas de rede, como MS/TP em EIA-485, *Point-to-Point* em EIA-232, ZigBee e outros que podem ser vistos na Figura 5.6. Nessa figura, pode-se ver também qual é a respectiva camada ISO OSI de cada modelo.

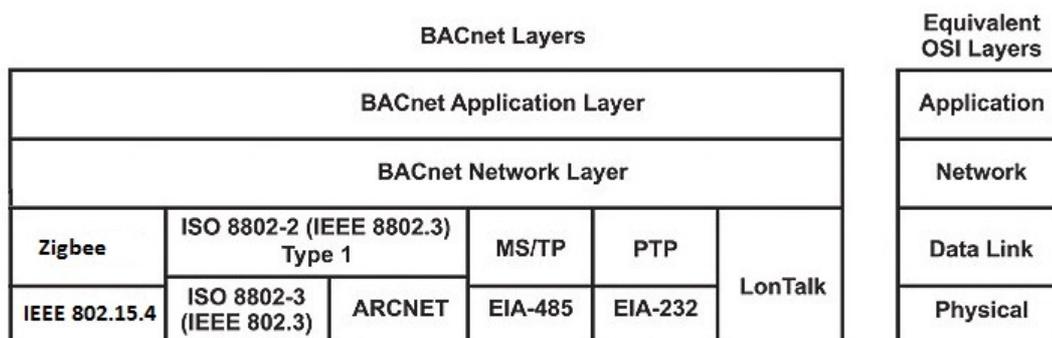


Figura 5.6. Camandas BACnet.

A norma BACnet não restringe a nenhuma topologia, ficando a critério do modelo acima que for utilizado. O uso de rede ZigBee como opção de camada de enlace só foi adicionada em janeiro de 2009 [21].

5.2.2 Camada de rede e roteamento

A camada de rede no BACnet tem o objetivo de fazer com que mensagens possam ser enviadas de uma rede para outra independente da tecnologia utilizada na camada de enlace. Esta camada permite que sejam enviadas mensagens para um dispositivo específico, *broadcast* em uma determinada rede ou *broadcast* para todas as redes. Cada dispositivo BACnet é identificado por um número de rede e seu endereço MAC. Dispositivos que conectam duas redes BACnet são chamados de BACnet router, ou roteador BACnet.

O datagrama da camada de rede possui tamanho variável e é chamado de NPDU (Network layer Protocol Data Unit). O datagrama APDU (Application layer Protocol Data Unit) está contido dentro do NPDU e ele contém a mensagem para a aplicação. O NPDU contém informações obrigatórias e outras opcionais. As informações obrigatórias são:

- Versão: Atualmente o protocolo BACnet está na versão 1;
- Controle: É um byte de controle que contém informação de quais são os dados contidos na NPDU. O bit7 informa se a mensagem é uma mensagem de rede ou se possui APDU. O bit6 e o bit4 são reservados. O bit5 determina se há a informação de destino e *Hop Count*, utilizados para roteamento. O bit3 determina se há a informação de fonte, também utilizado para roteamento. O bit2 determina se a mensagem espera resposta. Os bit1 e bit0 determinam a prioridade da mensagem de rede e
- Vendor ID: O número que identifica o fabricante do equipamento, caso não possua é 0.

As informações opcionais são:

- DNET, DLEN e DADR: São as informações a respeito do endereço de destino da mensagem quando o destino se encontra em outra rede diferente da origem;
- *Hop Count*: Número máximo de roteadores pelo qual a mensagem pode passar, para garantir que ela não fique em loop indefinidamente;
- SNET, SLEN e SADR: São informações a respeito do endereço do dispositivo que originou a mensagem que vem de uma outra rede diferente do destino;
- *Message Type*: Tipo de mensagem de rede, quando este campo está presente não há APDU e
- APDU: Mensagem que será passada para a camada de aplicação.

Roteadores BACnet possuem diversos serviços que podem ou precisam ser implementados para garantir o roteamento adequado. Estes serviços não serão abordados e podem ser encontrados na documentação oficial da norma BACnet[22]. Para um dispositivo ser um roteador BACnet não basta ele poder receber mensagens de redes diferentes e enviar ao dispositivo final, ele tem que ser capaz de encontrar outros roteadores na rede e poder perguntar a cada um quais são os dispositivos que aquele roteador alcança e com isso manter sua tabela de endereços de roteamento atualizada. Além desses serviços, ainda exige outros auxiliares para inicialização, reinicialização e tratamento de erros na rede.

O roteamento entre duas redes BACnet onde existe apenas um único roteador, e este já tenha sua tabela de endereços de roteamento atualizada, segue os seguintes

procedimentos. O dispositivo que irá enviar a mensagem coloca no NPDU a informação de destino, conforme foi mostrado acima, e envia a mensagem ao roteador. Este irá adicionar as informações de fonte e remover informação de destino do NPDU e enviar a mensagem ao seu destino final. Este dispositivo recebe a mensagem com a informação de quem foi a fonte da mensagem e, caso precise enviar uma resposta, irá seguir os mesmos passos usados no envio da mensagem original.

5.2.3 Serviços, objetos e propriedades

Como dito anteriormente, BACnet é orientado a objeto. Todo dispositivo BACnet possui um conjunto de objetos e cada um destes possuem um conjunto de propriedades. Toda informação que pode ser requisitada através de uma rede BACnet está contida dentro dos objetos. Desde seu desenvolvimento, o número de objetos vem crescendo. Na norma BACnet 2008 existem 31 objetos, desde então foram lançados vários *addenda* [23] que propõem novos objetos. Os objetos não são restritos a estes, sendo que, caso haja a necessidade, pode-se criar novos objetos proprietários. Estes não terão interoperabilidade, porém o dispositivo pode continuar operando em qualquer rede BACnet e apenas os objetos proprietários não serão reconhecidos.

Todo dispositivo BACnet é obrigado a possuir um objeto chamado *BACnet Device*, e não pode possuir outros objetos desse mesmo tipo. Esse objeto possui diversas configurações a respeito do dispositivo. Esse objeto tem um código que o identifica OID (*Object_Identifier*), e esse código é único na rede.

Todo objeto possui seu conjunto de propriedades. Ainda mais do que o número de objetos, o de propriedades vem crescendo desde sua primeira lista. Na primeira versão oficial da norma BACnet em 1995, existiam 123 propriedades e atualmente este número mais que dobrou. Cada objeto possui seu conjunto de propriedades, sendo que algumas são optativas e outras são obrigatórias. Também podem existir propriedades proprietárias porém elas podem não ser entendidas por outros fabricantes. Um dos objetos mais comuns é o *Analog Input*. Algumas de suas principais propriedades obrigatórias são:

- *Object_Identifier*: Código identificador que identifica o objeto. Cada objeto é único dentro de uma rede com as três informações, *Object_Identifier*, *Object_type* e OID do objeto device;
- *Object_Name*: Identifica o nome do objeto, é uma string de no mínimo um caractere. Ex. "Sensor Externo 2";
- *Object_Type*: Tipo do Objeto. Ex. *Analog_Input*;
- *Present_Value*: Valor atual, em unidades de engenharia. Ex. 2,63154 e

- Reliability: Indica o confiabilidade do valor em *Present_Value*. Os valores que esta propriedade aceita são: *NO_FAULT_DETECTED*, *NO_SENSOR*, *OVER_RANGE*, *UNDER_RANGE*, *OPEN_LOOP*, *SHORTED_LOOP*, *UNRELIABLE_OTHER*.

Os serviços são os meios com que um dispositivo BACnet requisita informação de outro dispositivos, comanda outro dispositivo e anuncia o acontecimento de um evento. O pedido do serviço se chama *Service Request* e sua resposta se chama *Service Response*. Toda a comunicação é baseada nestes serviços. Aquele que requisita o serviço é chamado de cliente, e aquele que responde de servidor, caracterizando a comunicação cliente-servidor. Porém, os dispositivos BACnet não são exclusivamente clientes ou servidores (exceção no caso do MS/TP), pois todos os dispositivos podem requisitar e responder serviços. Por essa razão, cada dispositivo poderá ser o cliente ou o servidor em determinado momento da comunicação. Existem cinco categorias de serviços: alarme e eventos; acesso a arquivos acesso a objetos; gerenciamento de equipamentos; e terminal virtual.

Os serviços mais comuns são *Who-is*, *I-am*, *Write-property* e *Read-property*. *Who-is* é um serviço que não requer confirmação e pode ser enviado a um dispositivo ou em *broadcast*, que, ao receber este serviço, caso ele seja um dos que requisitados, pode enviar o *I-am*. Este serviço pode ser em *broadcast* ou em *unicast* para o dispositivo que tiver enviado o *Who-is* e também não requer confirmação. Nele contém informações a respeito do dispositivo. Esta conversa é uma parte importante do BACnet para que se conheça quem são os outros dispositivos na rede. *Write-property* e *Read-property* são serviços de leitura e escrita, respectivamente, de uma propriedade de um objeto. Ambos necessitam de confirmação, e na própria confirmação já está a resposta do pedido.

5.2.4 BACnet MS/TP

Também conhecido como *Master-Slave/Token-Passing*, descreve como deve ser a comunicação BACnet numa rede em cima de EIA-485. Nessas redes temos o problema de que apenas um dispositivo na rede pode estar falando por vez. Para organizar isso, foi utilizado a técnica de *Token-Passing*, onde existe um único token na rede e apenas o dispositivo que estiver com o token pode iniciar serviços. Devido ao fato do gerenciamento do *Token* pode ter um custo em processamento e em tráfego na rede, os dispositivos BACnet foram separados em dois, mestre (*master*) e escravo (*slave*). Assim o escravo é sempre servidor e o mestre poder ser cliente ou servidor caso haja outros mestres na rede. O token fica sendo passado apenas entre os mestres. Uma rede com um número grande de mestres em relação aos escravos corre o risco de ter mais mensagens de gerenciamento de token do que mensagens com informação trafegando na rede.

5.2.5 BACnet/ZigBee

O uso do protocolo BACnet sobre uma rede *wireless* estava sendo estudado pela ASHRAE e diversos pesquisadores pelo mundo para encontrar a melhor solução. Em janeiro de 2009, a ASHRAE lançou o *addendum q* [21]. Esse anexo sobre a norma BACnet define o ZigBee como camada de Enlace ou Data Link.

Dois *clusters* ZigBee precisam ser implementados para essa utilização. O ZigBee *Generic Tunnel cluster*, que possui funções para tunelamento de protocolo, e *BACnet Protocol Tunnel cluster*, que possui a função de envio da mensagem BACnet.

Para utilizar o ZigBee como uma camada de enlace foi definida uma camada de interface entre a camada de rede BACnet e o ZigBee, que foi denominada de BZLL (*BACnet ZigBee Link Layer*). A primeira responsabilidade dessa camada é criar uma tabela virtual de endereçamento (*VMAC Table*). O maior endereço MAC que o BACnet aceita é de 7 bytes, porém a IEEE 802.15.4 define a utilização de MAC de 64bits. Assim, ao invés de usar o MAC, utiliza-se o BACnet *Identifier*, que é único de cada dispositivo na rede. Essa camada faz a tradução e passa para a camada do ZigBee. Na Figura 5.7 pode-se observar como é a interface entre as camadas.

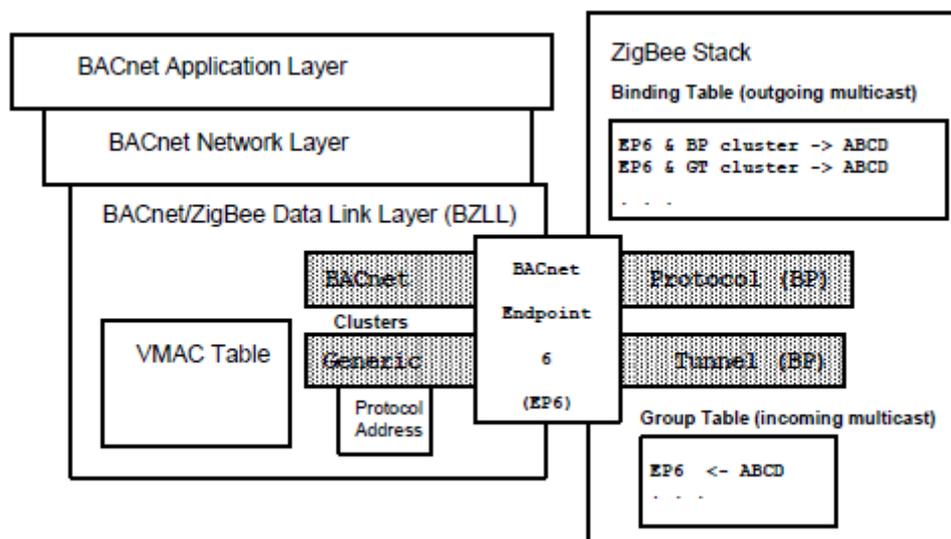


Figura 5.7. Camadas BACnet/ZigBee [21].

5.3 BITCLOUD ZIGBEE® PRO STACK

Bitcloud é uma pilha *software* para sistemas embarcados da Atmel. Foi desenvolvida para ser completamente compatível com ZigBee PRO e possui um conjunto de APIs para facilitar no desenvolvimento e gerenciamento de redes ZigBee.

5.3.1 Arquitetura

A arquitetura desta pilha segue a separação lógica encontrada na IEEE 802.15.4 e nas especificações *ZIGBEE* [13]. Ela é separada em 12 camadas como mostrado na Figura 5.8.

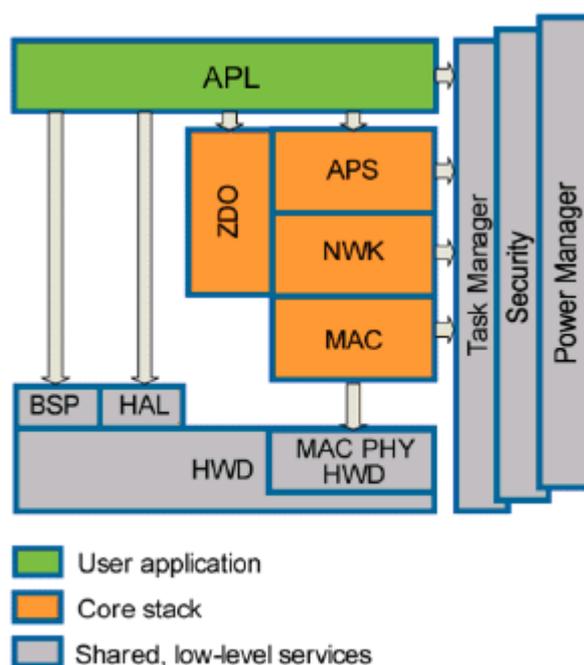


Figura 5.8. Arquitetura Bitcloud.

As camadas do núcleo da pilha são ZDO, APS, NWK e MAC. São as camadas mais ligadas ao gerenciamento da rede. APS (Sub-camada de suporte a aplicação) é a interface de mais alto nível com a camada de rede. ZDO (*ZigBee Device Object*) é responsável pelo gerenciamento da rede. NWK (*Network*) e MAC (*Medium Access Control*) são as camadas que realizam as funções de rede como definidas nas especificações *ZigBee* [13]. Estas devem ser transparentes ao usuário. HAL (*Hardware Abstraction Layer*) e BSP (*Board Support Layer*) são camadas de abstração de *hardware* e inclusive é recomendado que não se utilize nenhuma função de mais baixo nível para acesso ao *hardware* utilizando sempre que possível as funções destas camadas para isto. HWD e MAC PHY HWD são as camadas que fazem a interface de mais baixo nível com o *hardware*, também transparentes ao usuário. O *task manager* é o gerenciador de tarefa baseado em prioridades onde o tempo de execução de cada tarefa é crítico. A bitcloud também pode gerenciar a segurança da rede,

como criptografar as mensagens, e o consumo de energia podendo o dispositivo dormir e acordar para economizar energia.

Por convenção, todas funções da API possuem a camada a que elas pertencem em letras maiúsculas seguida de um sublinhado antes do seu nome, por exemplo, `ZDO_GetLqiRssi`.

O *Bitcloud* é orientado a eventos e sistemas desse tipo são comuns em sistemas embarcados, sendo assim chamadas de funções da API e são acompanhadas de funções de notificação assíncronas com o resultado da operação. No programa, quando uma camada superior fizer um pedido para uma camada inferior, deverá passar também um ponteiro para a função de *callback*, que será chamada quando a tarefa for completada. Apesar das principais funções da API serem assíncronas, não são todas as funções da API que seguem esta estrutura. Todas as funções da camada BAC são síncronas. Figura 5.9 temos uma comparação entre chamadas síncronas e assíncronas.

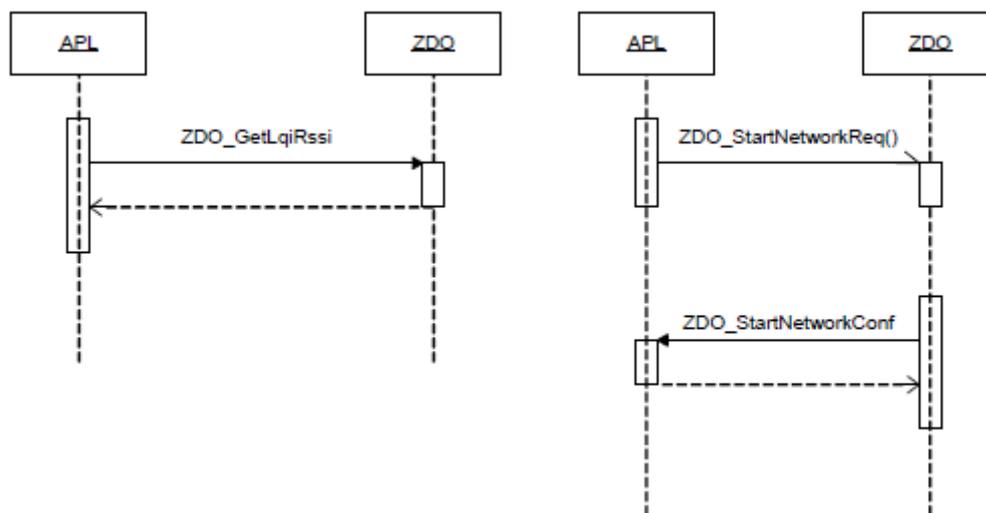


Figura 5.9. Comparação da execução Síncrona x Assíncrona, [24].

5.3.2 *Callback, task handler e o agendador de tarefas*

A aplicação é responsável por criar os *callbacks*, que são chamados pelas camadas inferiores, que irão tratar todos os eventos de interesse. Cada camada possui sua prioridade, sendo a APL a de menor prioridade, e *callbacks* são executados com a prioridade da camada que os invocou e não da camada onde eles residem. Utilizando estas prioridades, o agendador de tarefas decide quais são as próximas tarefas a serem executadas. É importante que *callbacks* da APL sejam executados em menos de 10 ms, caso contrário a pilha fica presa naquela tarefa e não pode executar outras de maior prioridade enquanto esta não acabar.

A camada de aplicação não é executada somente através de *callback*, também possui o *APL_TaskHandler*. O *task handler* de cada camada é executado com a prioridade daquela camada, assim sendo o *APL_TaskHandler* só é executado quando todas tarefas de maior prioridade já tiverem sido executadas, e por isso sua execução pode durar até 50 ms. Para postar uma tarefa para o agendador de tarefas, utiliza a função *SYS_PostTask* que recebe como parâmetro a camada do *task handler* a ser executado. Como o *APL_TaskHandler* pode ter uma execução mais longa que *callbacks*, pode-se separar a execução de um *callback* para que não ultrapasse o tempo limite. Para isso, basta salvar o estado da execução e postar a tarefa para o agendador. Assim, quando o *APL_TaskHandler* for executado a tarefa do *callback* pode ser terminada.

CAPÍTULO 6 AMBIENTE DE TESTES

O trabalho realizado se caracteriza numa rede *wireless* composta por quatro sensores magnéticos, uma central de alarmes, ar condicionado híbrido, duas placas meshbean, um módulo com chip Zigbit e um *desktop* com o *software Actionview*. Na figura X podemos ver como o sistema é interligado.

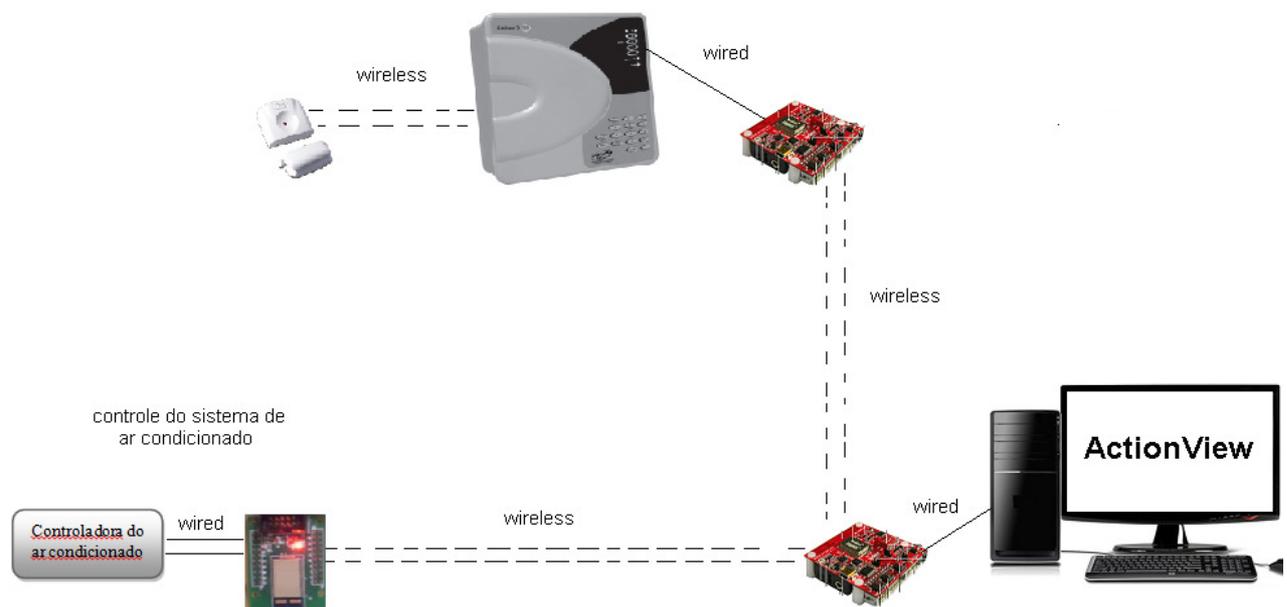


Figura 6.1. Diagrama de Blocos do Sistema.

Os sensores sem fio das janelas e porta se comunicam com a central de alarmes, que está conectada ao módulo sensor. Este módulo se comunica com o coordenador que está conectado ao computador com o ActionView. O coordenador também se comunica com o módulo atuador, e este está conectado à controladora do ar condicionado híbrido

A sala sendo monitorada é uma sala de reunião de 8 m x 3 m e fica dentro de um laboratório. O ar condicionado híbrido é responsável apenas pela refrigeração desta sala. O laboratório possui ar condicionados próprios.

6.1 AR CONDICIONADO HÍBRIDO

O ar condicionado híbrido foi projetado de forma a garantir conforto térmico e economizar energia. É um sistema híbrido, porque nesse contém um ar condicionado evaporativo e convencional. O evaporativo puxa o ar externo, umidifica e ventila para dentro da sala o ar resfriado e umidificado. Esse sistema utiliza menos energia, porém possui uma capacidade menor de resfriar um ambiente, se comparado ao convencional. O convencional não utiliza o ar externo, apenas refrigera e re-circula o ar ambiente, ele resfria o ambiente mais rápido e, geralmente, diminui a umidade do ar. O ar condicionado convencional gasta mais energia que o evaporativo. O ar condicionado híbrido fica alternando entre seus modos de operação e assim consegue controlar umidade e temperatura. Logo se economiza energia e manter conforto térmico[25].

6.2 CENTRAL DE ALARMES PPA

O modelo Códigos 5 plus da PPA alarmes foi o utilizado por ser viável economicamente e de muito fácil manipulação. Esse painel de alarmes é microcontrolado e possui 5 zonas. Esse modelo possui a limitação de não funcionar adequadamente se duas zonas estiverem abertas ao mesmo tempo, pois ela só processa um sinal de cada vez. Entretanto, isso não interferiu no andamento do projeto.



Figura 6.2. Central de alarmes PPA.

6.3 SENSORES DE ABERTURA

Comercialmente denominados “sensores magnéticos”. São dispositivos baseados em pequenos componentes chamados *reed-switch*. São instalados em portas, janelas e em

outros lugares onde se permita um movimento de abertura ou separação entre duas partes, pois a separação entre o *reed-switch* e um ímã que compõe o conjunto é que irá gerar a informação para a central de que houve uma abertura indevida do local monitorado [26].

Esses sensores serão os dispositivos que perceberão a informação e enviarão à central de alarmes. Eles podem estar ligados à central fisicamente através de fios (*hardwired*) ou transmitir seus sinais por meio de ondas de rádio, denominados de sensores sem-fio (*wireless*).

6.3.1 Sensor magnético sem fio

O componente principal desse sensor é o *reed-switch* (*reed* – palheta, *switch* – interruptor/chave), que se constitui de duas lâminas metálicas próximas uma da outra e inseridas em uma ampola de vidro. A Figura 6.3 representa esse dispositivo.

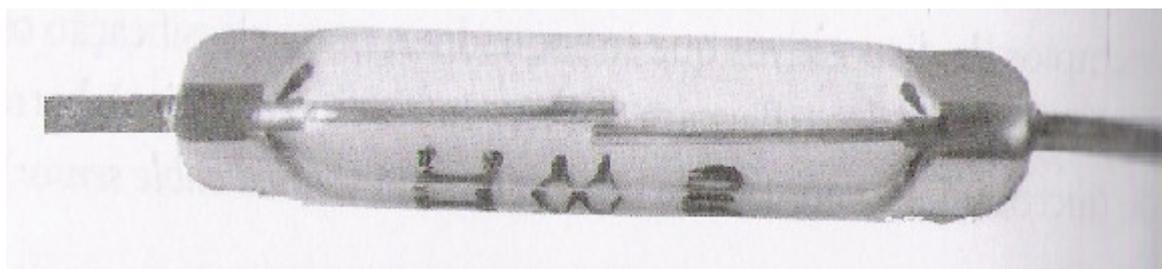


Figura 6.3. Reed-Switch, [26].

Cada lâmina está ligada a um terminal externo. Quando expostas a um campo magnético, as lâminas são atraídas uma sobre a outra, o que estabelece em contato elétrico entre as mesmas. Quando elas se encontram nesse estado, diz-se que estão com os “contatos fechados”. Quando se desencostam uma da outra, diz-se que estão com os “contatos abertos”. Alguns fabricantes estimam que um *reed-switch* pode cumprir 10 milhões de ciclos de abertura e fechamento [26].

A abertura dos contatos do *reed-switch* ativa o transmissor ao qual ele está ligado que, por sua vez, envia seu sinal ao circuito receptor de RF da central de alarmes. As frequências de transmissão variam conforme o modelo e fabricante: 315,00; 392,00; 433,00 e 868,00 MHz, por exemplo. Tais frequências estão dentro de bandas de frequências destinadas ao uso industrial, científico e médico, assim estabelecidas internacionalmente (*ISM Band* – *industrial, scientific and medical*). A Figura 6.4 apresenta o modelo utilizado no projeto.



Figura 6.4. Sensor magnético.

Este modelo é alimentado por uma bateria de 12 V tipo 23AE. O manual de instalação da central explica que comandos devem ser dados na central para que esta entre no modo de aprendizagem (modo em que a central poderá receber e memorizar os sinais dos transmissores).

A distância máxima do sensor à central varia de acordo com o modelo e a instalação. Quando o sensor transmite o seu sinal sem haver qualquer obstáculo entre ele e a central, pode-se ter uma distância aproximada de 30 metros[26]. Quando há a presença de obstáculos físicos essa distância pode diminuir, principalmente se forem metálicos.

Nas Tabela 6.1 e Tabela 6.2, encontram-se as configurações básicas, respectivamente, do sensor magnético e da central de alarmes utilizados no projeto.

Tabela 6.1. Características elétricas do sensor PPA.

Consumo em operação	6.5 mA
Frequência de Operação	433 MHz
Tensão Máxima de Operação	13 V
Consumo <i>Stand-by</i>	1 a 3 μ A

Tabela 6.2. Características elétricas da central PPA.

Modelo	Codigus 5 Plus
Entrada rede	127 ou 220 Volts AC 20W
Saída Auxiliar	12 – 14 Volts DC
Tensão de Carga da Bateria	12.5 Volts DC MAX 700 MA
Setores de Alarme	5 setores
Temperatura de operação	20° a 70°
Dimensões	215 x 195 x 80 mm
Peso	1.548 Kg sem bateria

6.3.2 Sistema com radiofreqüência

A comunicação via radiofreqüência (RF) ou ondas de radio de longo alcance (*Long Range Radio*) caracteriza-se por não usar um meio físico, parecido com a comunicação via rede de telefonia celular (voz e GPRS). A central de alarmes é ligada a um transmissor que envia os sinais de alarme e estado do sistema por intermédio de uma antena.

A central de alarmes utilizada possui um receptor de RF incorporado à placa e uma memória para 30 dispositivos sem fio.

6.3.3 Instalação dos sensores

Um dos sensores foi instalado na porta e os outros em cada uma das janelas existentes na sala de reunião do laboratório de automação e robótica (LARA).

Cada vez que uma janela ou a porta da sala é aberta, o sensor magnético comunica à central PPA. Na central existe um Led correspondente a cada sensor instalado e esse se apaga quando há essa comunicação. O normal é que os quatro leds utilizados estejam acesos e só se apaguem se algum setor for violado (ser aberto).

Para cada led há um resistor de proteção com uma tensão, em média, de 2.5 V se o led estiver aceso e 0 V quando apagado. Então, aproveitou-se desses resistores para obter o sinal que irá entrar na placa Meshbean2. Dessa forma, tem-se em tempo real a situação da porta e das janelas. Na Figura 6.5, pode-se observar essas conexões.

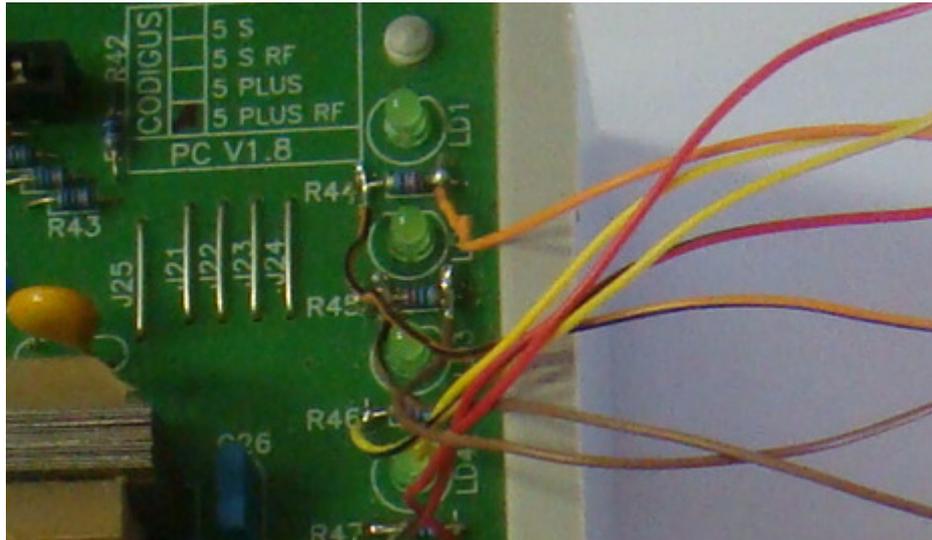


Figura 6.5. Foto dos resistores da central com a adaptação realizada.

Essa placa Meshbean enviará ao coordenador, toda vez que for solicitada por esse, o estado dos sensores. O coordenador também é uma placa Meshbean2 só que programado de forma diferente. Ele ficará conectado ao computador por cabo USB, e, por isso, o supervisor saberá o estado de todos os sensores.

Se algum sensor for aberto e permanecer nesse estado por mais de 2 minutos, o supervisor dará a mensagem para que o coordenador, por meio de comunicação sem fio, envie um comando ao módulo atuador, existente no quadro de relés, para que seja desligado o sistema de ar condicionado.

Inicialmente, encontrou-se um problema, os sensores apenas enviavam um pulso à central quando a região era violada e esta por sua vez disparava o alarme. O que condiz com o propósito do sistema disponível para o projeto, pois esse é utilizado comercialmente para gerar alarme de intrusão e não para monitorar o ambiente. Por isso, necessitou-se fazer alguns ajustes nos sensores, já que o interesse é que o sinal emitido pelo sensor seja binário, que esteja em um nível lógico quando estiver aberto e em outro nível lógico quando estiver fechado.

Para tanto, montou-se um circuito que foi acoplado ao do sensor. Esse é constituído por dois resistores de 4K7 ohms e um transistor BC 548. A bateria do sensor alimentará o circuito. Utilizou-se o *jumper* de comunicação existente no sensor para se obter um sinal binário. Quando esse *jumper* possui seus terminais em curto circuito fica a todo instante enviando sinal à central. A Figura 6.6 demonstra esse circuito.

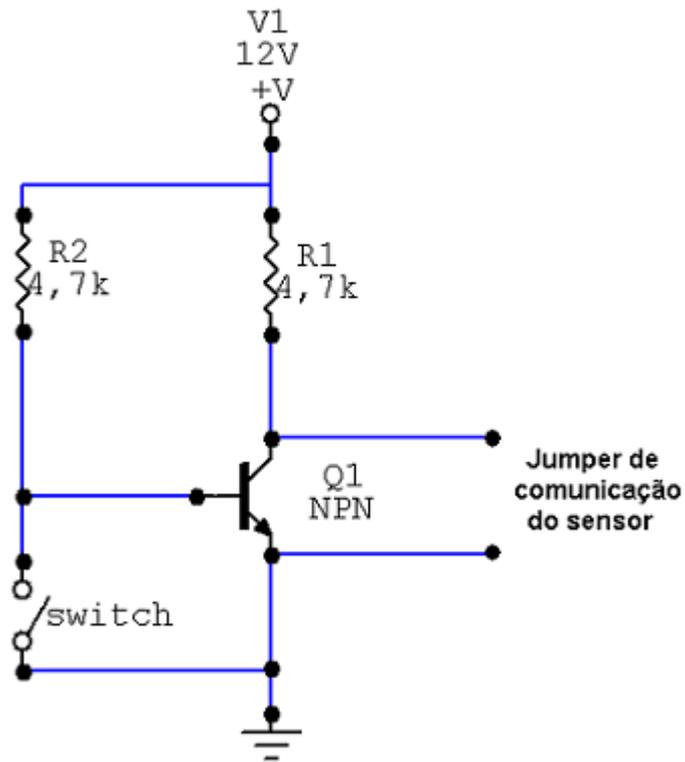


Figura 6.6. Circuito acrescentado ao sensor magnético.

Após a inserção desse circuito em cada um dos sensores o sinal passou a ser binário como necessário. Quando a área estiver aberta, há comunicação entre o sensor e a central. O próximo passo foi utilizar o sinal de cada resistor de proteção dos leds da central. Só que, para isso, montou-se outro circuito com a finalidade de colocar um terra comum para as tensões de saída dos resistores. Para tanto, foi necessário quatro optoacopladores e quatro resistores de 4k7 ohms para proteger a placa de possíveis picos de tensão. O desenho esquemático desse circuito pode ser visto na Figura 6.7 e a Figura 6.8 mostra uma fotografia de todas as ligações da central com o circuito optoacoplador e com a placa Meshbean2.

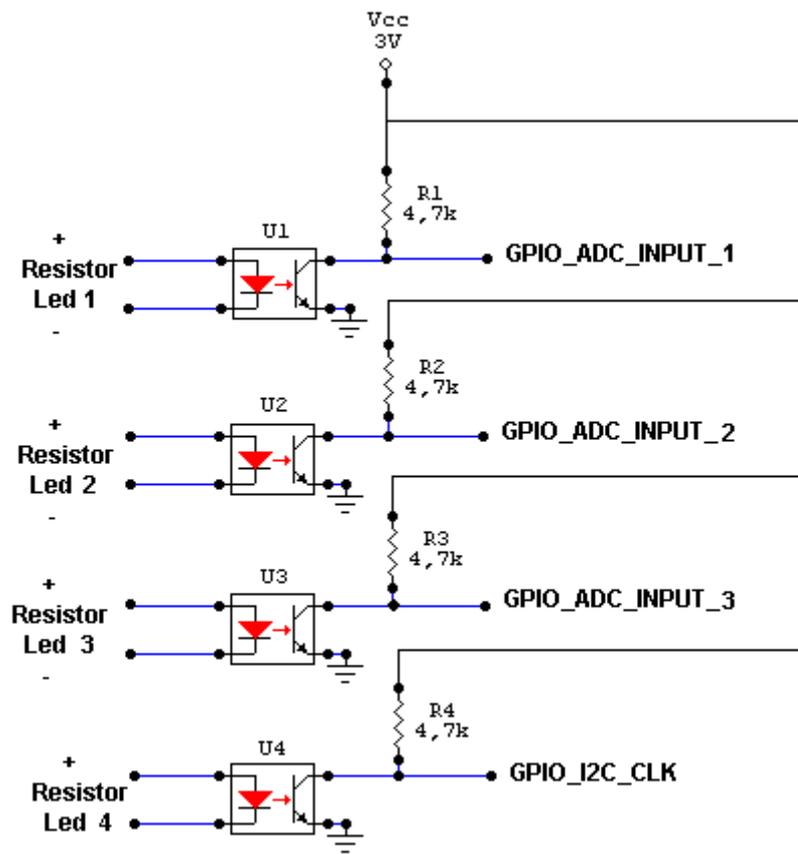


Figura 6.7. Circuito optoacoplador

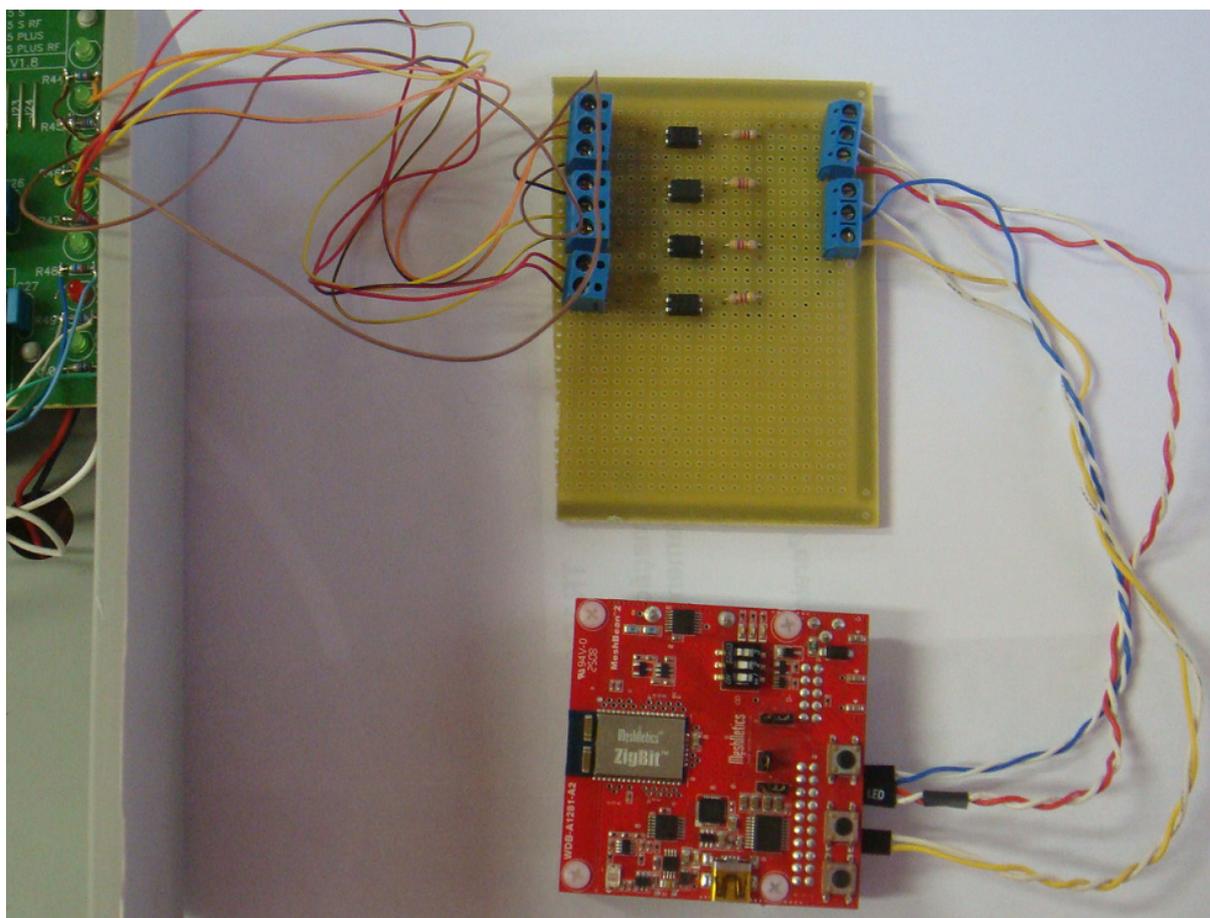


Figura 6.8. Ligações central, circuito optoacoplador, placa meshbean2.

Depois de todas essas modificações foi possível ver o estado de todos os setores monitorados através do supervisor Action View. Quando, por exemplo, abre-se uma janela observa-se pelo monitor do computador essa mudança. Posteriormente o desafio foi programar o Zigbit existente na placa atuadora para que esse recebesse o comando do coordenador de desligamento do ar condicionado.

6.4 ZIGBIT

Zigbit é um chip desenvolvido pela Meshnetics (atualmente faz parte da ATMEL). Ele é a junção de dois chips dentro de um só, um microcontrolador ATMEGA1281 e um *RF transceiver* AT86RF230 compatível com IEEE 802.15.4 com frequência de 2.4 GHz. Além desta versão, há também a versão que inclui um *dual-chip Antena*, eliminando assim a necessidade de uma antena externa. Existem outras versões além destas, porém não serão abordadas neste trabalho. A Figura 6.9 ilustra a interface entre estes componentes dentro do Zigbit.

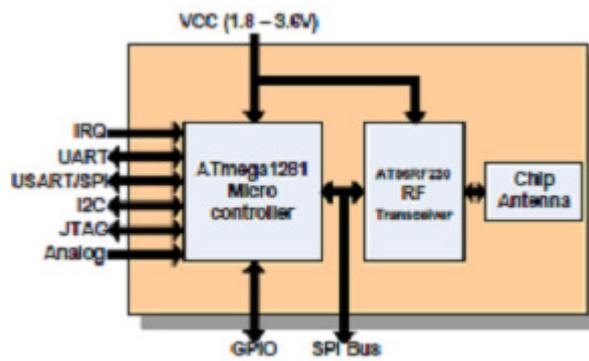


Figura 6.9. Zigbit.

O ATmega1281 é um microcontrolador de RISC de 8bits. Possui 128 kB de memória flash, 8 kB de RAM e 4 kB de EEPROM. Possui também interfaces ADC, I/O de uso geral, compatível com J-Tag, UART, USART, SPI, I2C, IRQ e *1-wire* [27].

Este chip é compatível com Bitcloud e foi escolhido como a melhor opção para este trabalho por já possuir uma biblioteca com a pilha ZigBee já implementada.

6.4.1 Meshbean2 board

A placa Meshbean2 faz parte de um kit de desenvolvimento da Meshnetics para facilitar o primeiro uso do Zigbit. Possui uma interface facilitada para diversos dos possíveis usos do módulo Zigbit. Esta placa possui sensor de temperatura, sensor de luz, três LEDs programáveis, três DIP switch, dois botões programáveis e um de reset e interface USB-UART para facilitar a comunicação serial com o computador. Também há a possibilidade de se acessar diversos pinos ADC, GPIO, I2C, SPI e J-Tag por uma interface na lateral da placa. Na Figura 6.10, observa-se uma imagem da placa.

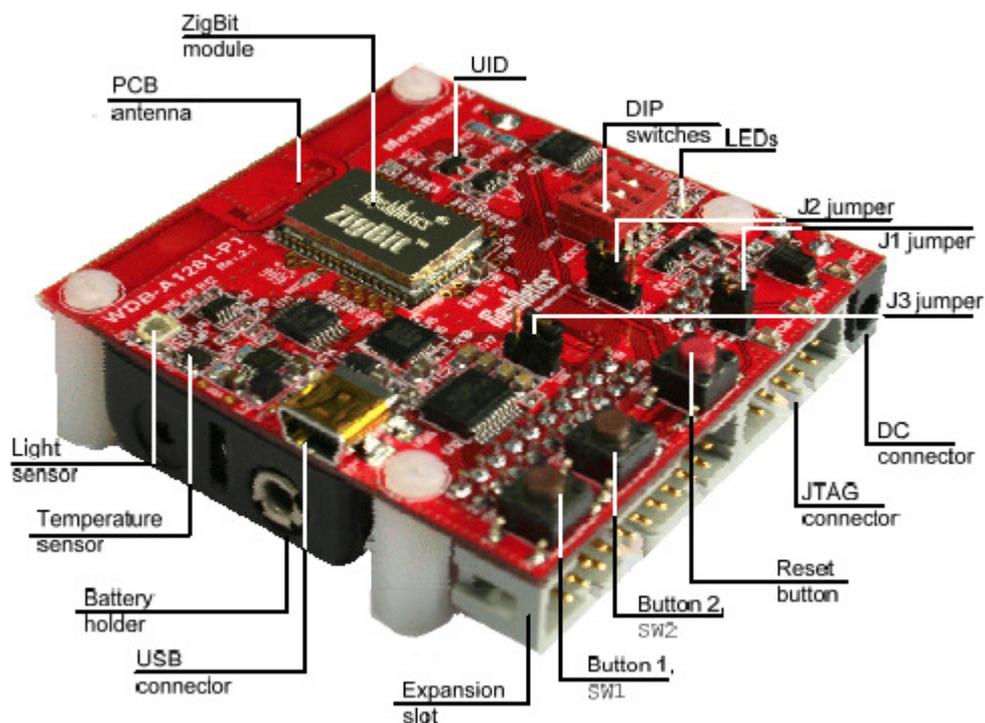


Figura 6.10. Placa Meshbean.

6.5 GRAVADOR

Esse módulo, desenvolvido por trabalhos anteriores [28], foi necessário para fazer a gravação dos programas no módulo Zigbit. Foi por meio dele, que o código responsável pelo controle liga-desliga do ar condicionado foi transferido ao Zigbit anexado ao quadro de relés.

As Figura 6.11 e Figura 6.12 apresentam, respectivamente, uma imagem do módulo gravador e seu esquema elétrico.

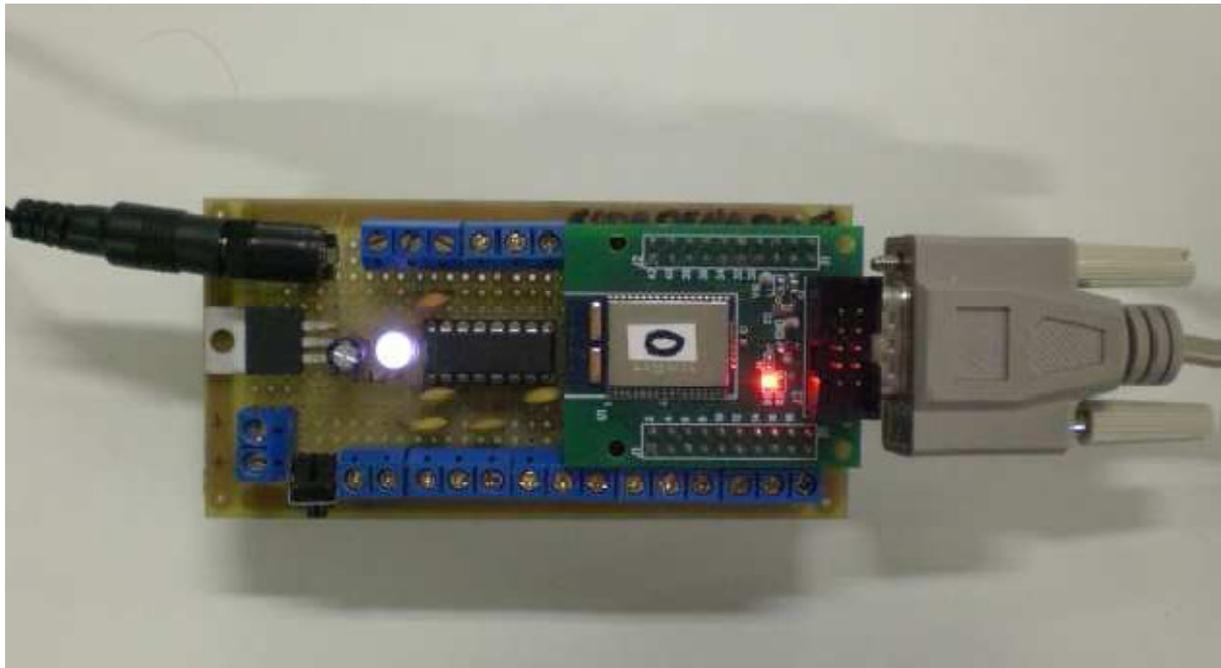


Figura 6.11. Módulo Gravador, [28].

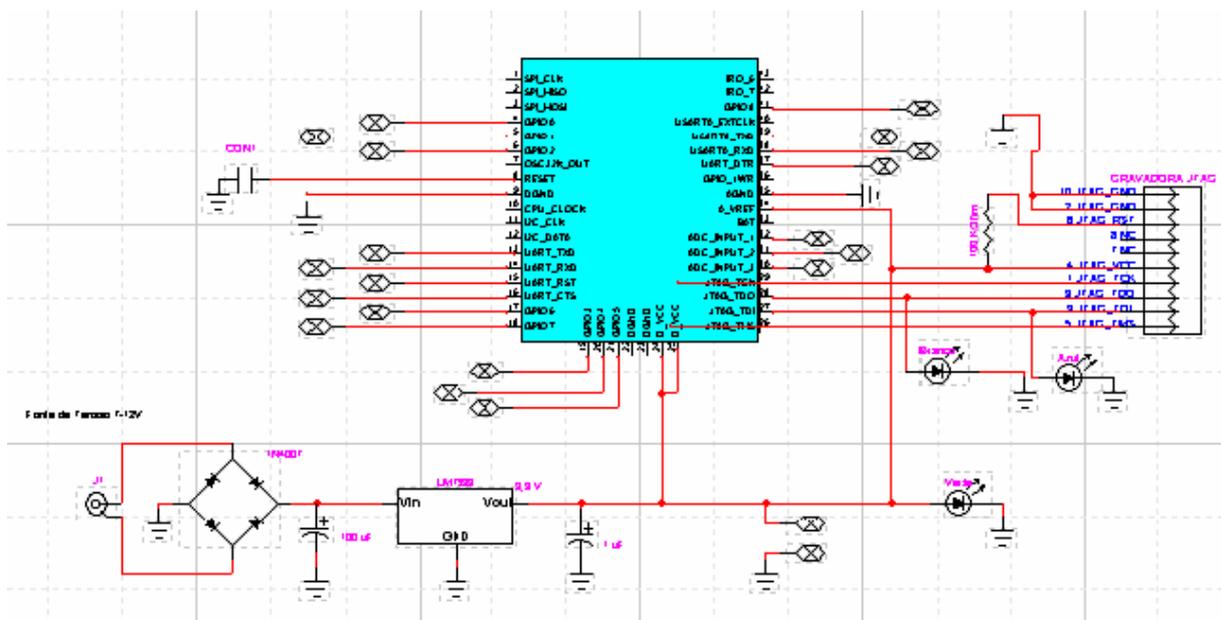


Figura 6.12. Esquema elétrico do Módulo Gravador, [28].

6.6 ATUADOR

O atuador (Figura 6.13) é o dispositivo que comanda o sistema de ar condicionado híbrido. No presente trabalho, ele terá a função de gerenciar apenas o relé que fará o liga-desliga do ar condicionado, além de enviar ao coordenador o *status* deste equipamento. Este relé está conectado ao ventilador do ar condicionado, assim caso o ventilador seja desligado todo o sistema é desligado.



Figura 6.13. Módulo Atuador [28].

6.6.2 Relé de Estado Sólido

Um dispositivo muito utilizado para acionamento de cargas de corrente alternada são as chaves estáticas. Elas promovem o chaveamento da potência fornecida pela carga, liga e desliga, mas não a modifica em nenhum outro aspecto. Chaves estáticas, disjuntores, contactores e relés de estado sólido são exemplos de dispositivos utilizados na área de chaveamento estático [29].

Utilizou-se o relé SSR T2405Z-M da Teletrom existente na sala de reunião do LARA para se fazer o controle liga-desliga do ar condicionado. O esquema elétrico desse relé pode ser visto na Figura 6.14.

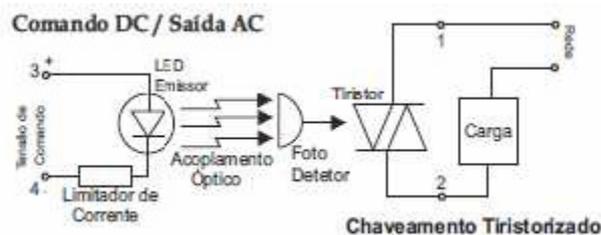


Figura 6.14. Esquema elétrico – SSR Teletronic

6.7 REDE

Ao todo, existem três redes. Uma rede ZigBee, composta pelos sistemas embarcados, uma rede BACnet/ZigBee, composta pelos mesmos dispositivos mas por onde passam serviços e

informações diferentes, e uma rede BACnet/MSTP, esta composta pelos sistemas embarcados e pelo supervisor.

6.7.1 ZigBee

Os dispositivos que compõe essa rede são o coordenador, o módulo sensor e o módulo atuador. Os dispositivos podem ser identificados por dois endereços, o MAC que possui 64 bits de acordo com a IEEE 802.15.4 e o *ShortAddress* que é um número de 16 bits que pode ser utilizado como um *alias* do MAC. É responsabilidade da aplicação garantir a unicidade do *ShortAddress* na rede. No caso do ZC (ZigBee Corrdinator), o *ShortAddress* é sempre 0. Todos os módulos foram configurados para utilizar a rede com o *PanId* de 0xAAAAAAAAAAAAAAC (número hexadecimal).

Os serviços definidos no *Generic Tunel Cluster* são utilizados apenas quando um novo dispositivo entra na rede ou quando há alguma alteração na rede, uma vez que quando a rede está estabelecida apenas serviços do BACnet *Protocol Tunnel cluster* circulam.

O *handshake* inicial tem o intuito de garantir que não haja dois dispositivos na rede com o mesmo *Protocol Address*, que é a propriedade que contém o número de identificação BACnet (*Device ID*) daquele dispositivo. Primeiramente, assim que um dispositivo entra na rede ZigBee, envia um *Advertise* com o *Protocol Address* como parâmetro. O ZC recebe e checka se existe algum dispositivo com esse parâmetro igual na rede. Caso não tenha, o ZC envia um *Write* no atributo *Protocol Address* com o mesmo valor que ele recebeu. E caso já tenha um outro dispositivo com o mesmo valor neste atributo, então o ZC envia um *Write* no atributo *Protocol Address* com um valor que esteja disponível. O dispositivo recebe o *Write* e checka se o valor que ele recebeu é diferente do que ele enviou. Se for igual ele já está nas tabelas de endereço do ZC e agora pode entrar na rede BACnet. Se for diferente, como sempre que há uma mudança no *Protocol Address* é preciso enviar um *Advertise* com esta informação, é enviado um novo *Advertise* com o valor novo. Assim que receber o *Write* com o mesmo valor como aconteceu na etapa anterior, o *handshake* está pronto. A partir deste ponto, este novo dispositivo na rede só utilizará serviços BACnet.

6.7.2 BACnet/ZigBee

Quando um dispositivo já está com a rede ZigBee estabelecida, ele pode então entrar na rede BACnet. O serviço para um dispositivo anunciar que entrou na rede é *I-am*. Este serviço é anunciado em *unicast* [21] para o coordenador, que recebe o serviço e com estas informações adiciona ou atualiza suas informações na tabela. Neste ponto, os dispositivos estão prontos para se comunicarem, porém como o coordenador não precisa realizar nenhum serviço de leitura ou escrita BACnet, a não ser que venham mensagens roteadas

de outra rede, não haverá nenhum serviço circulando nesta rede. Porém, caso haja necessidade, essa tarefa pode ser facilmente implementada no coordenador, pois os serviços já estão implementados e a tabela de endereços está pronta.

Apesar de todos os dispositivos BACnet poderem iniciar e responder serviços, nesta rede não há necessidade dos dispositivos comunicarem com nenhum dispositivo que não seja o coordenador. Por isso, apenas o coordenador possui uma tabela de endereços.

6.7.3 BACnet MS/TP

A rede BACnet MS/TP possui apenas dois dispositivos nela, o coordenador e o supervisor, e ambos são mestres MS/TP, porém, o supervisor percebe os dispositivos da rede BACnet/ZigBee como se fosse estivessem na rede MS/TP. Como fisicamente o computador possui apenas comunicação EIA-232 e a placa Meshbean2 também, não foi utilizada a rede EIA-485. Para isso seria necessário converter o sinal de EIA-232 para EIA-485 e depois converter de volta para EIA-232, o que apenas causaria um aumento no tempo de transmissão da mensagem e não modificaria nenhum parâmetro da comunicação. Como são apenas dois dispositivos conectados diretamente não há necessidade de fazer esta conversão.

Todos dos dispositivos ZigBee possuem um endereço MS/TP que está na tabela de endereços do coordenador. Esses dispositivos se comportam como escravos MS/TP nesta rede. Cada dispositivo dessa rede possui seu endereço MS/TP no objeto *AnalogValue* 0. Então, para descobrir os endereços, o coordenador envia um *ReadProperty* na propriedade *PresentValue* deste objeto. Na seção 6.8.2 há um maior detalhamento sobre este procedimento .

Como existem dois mestres na rede, há uma necessidade de gerenciar *tokens*. Então o *Token* é passado entre os dois mestres que só podem iniciar um serviço se estiverem com o *Token*. Como neste caso existem dois mestres e dois escravos, há um grande número de mestres para uma rede com tão poucos escravos e isso pode gerar um excesso de mensagens de gerenciamento de rede, se comparado com a quantidade de mensagens com informação sendo passadas, o que pode comprometer a eficiência da rede. O comportamento desta rede será analisado na seção 6.7.3. Todos dispositivos, mestres e escravos, seguem as máquinas de estados descritas na norma [22].

Apesar de se comportar como se todos dispositivos estivessem na mesma rede MS/TP, existem realmente duas redes BACnet e para mensagens circularem entre elas é preciso um roteamento, e quem realiza esta tarefa é o coordenador.

6.7.4 Roteamento

O roteamento realizado nesta rede não segue o padrão BACnet pois para implementar um roteador BACnet seria necessário utilizar muita memória do coordenador, e este já está com seus recursos muito limitados. Por essa razão, ao invés de implementar um roteamento completo entre as duas redes BACnet, foi feito de uma forma para mascarar a rede ZigBee para a rede MS/TP, isto é, dispositivos que estão na rede MS/TP não percebem que existe uma rede ZigBee. O roteamento implementado possui a limitação de a mensagem não poder passar por outro roteador para chegar ao seu destino de modo que, caso seja necessário comunicar com outra rede, será preciso utilizar o ActionView como um concentrador. Assim, a outra rede se comunicaria apenas com o supervisor e este realizaria toda a comunicação com esta rede.

Como na rede EIA-485 todos os dispositivos recebem todas as mensagens que passam na rede e fazem uma decodificação do cabeçalho da mensagem para ver se é destinada a ele, o coordenador checa se a mensagem é para si ou para algum outro dispositivo da rede ZigBee. É possível fazer isso, pois o coordenador possui uma tabela de endereços com todos os dispositivos da rede ZigBee. Caso a mensagem seja destinada a rede ZigBee, é enviada uma mensagem *Reply_Postponed* para o dispositivo que originou a mensagem. Esta mensagem significa que a resposta não virá agora, mas será enviada assim que a resposta estiver pronta. Feito isso, a mensagem pode ser enviada para o dispositivo ZigBee, porém o dispositivo que a recebe precisa saber que aquela mensagem foi roteada, do contrário ele não saberá responder para o destino certo. Então, o coordenador precisa adicionar as informações de origem na mensagem e na NPDU do BACnet já existem campos feitos para isso. A primeira coisa a ser feita é setar o bit5 do byte de controle para avisar que há informações de endereço de origem na NPDU. Com isso os campos SLEN, SNET e SADR precisam ser preenchidos. O SLEN é preenchido com 1, pois endereço MS/TP possui apenas um byte. O SNET é preenchido com 0 (rede local), pois como não é um roteamento completo BACnet é tratado como se fossem da mesma rede. SADR é preenchido com o endereço de origem da mensagem. Com estes campos preenchidos e adicionados na mensagem, esta é enviada ao dispositivo de destino.

Ao receber a mensagem, o dispositivo percebe que existe informação de endereço de origem na mensagem. Caso haja necessidade de resposta, o dispositivo irá preencher a NPDU da resposta com as informações de destino com os dados contidos na NPDU da mensagem recebida e envia esta mensagem de volta ao coordenador.

O coordenador, ao decodificar a NPDU, checa que a mensagem é destinada a rede MS/TP. Então, ele monta o cabeçalho MS/TP com as informações contidas na NPDU e remove as informações de roteamento da NPDU. Como não pode simplesmente enviar para rede

MS/TP, precisa receber o *Token* antes. Assim que ele recebe o *Token*, ele envia a mensagem para esta rede.

O coordenador não inicia nenhum serviço na rede MS/TP, porém foi preciso implementar o mestre MS/TP nele para que pudesse realizar o roteamento. Caso o roteamento fosse feito sem que o coordenador enviasse o *Reply_Postponed*, ou seja, ele apenas roteasse e deixasse o cliente daquele serviço esperando até que a resposta voltasse, poderia ocorrer *timeout* de silêncio na rede. *Timeout* de silêncio na rede é utilizado para garantir que caso o *Token* seja perdido, os mestres saibam que não está havendo comunicação e comece o procedimento de criar o *token* e colocá-lo de volta na rede [22]. Quando esse *timeout* estoura é considerado que o *Token* foi perdido o que pode causar colisão na rede na hora de enviar a resposta quando terminar o roteamento. Após o envio de um *Reply_Postponed*, os outros mestres na rede seguem a comunicação normalmente como se a resposta tivesse sido enviada. Para que a resposta seja enviada, após o roteamento, é preciso que o coordenador receba o *Token* e por essa razão, ele não pode ser apenas um escravo MS/TP.

6.8 SOFTWARE EMBARCADO

Softwares embarcados são aplicações desenvolvidas para rodar em sistemas microprocessados. O *software* embarcado deste projeto foi desenvolvido em linguagem C. O ambiente de desenvolvimento foi Windows Xp, utilizando a IDE da ATMEL, AVR Studio 4 e o WinAVR, versão 20100110, como compilador. Para fazer o download do software para o microcontrolador, foi utilizado o programa Bootloader da ATMEL.

O software que roda nos módulos Zigbit pode ser separado em cinco camadas: Bitcloud, Steve Kargs BACnet stack (de agora em diante referida como pilha S.K.), BZLL, BAC e aplicação. A Figura 6.15 mostra a hierarquia destas camadas. As setas indicam que a camada que aponta pede serviços para a camada apontada.

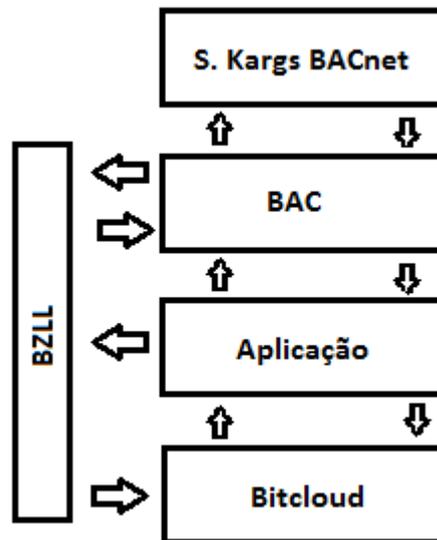


Figura 6.15. Camadas de aplicação.

As camadas BZLL e BAC foram desenvolvidas pela Spin Engenharia de Automação Ltda. Estas camadas foram desenvolvidas para incluir o ZigBee como uma camada de enlace à pilha S.K.. A BZLL foi idealizada de acordo com o *addendum q* da norma BACnet [21]. A camada BAC realiza interface entre a pilha BACnet e o resto da aplicação e realiza o roteamento entre as redes BACnet MS/TP e ZigBee. A pilha S.K. precisou sofrer algumas alterações para que se adequasse melhor a esta aplicação.

A camada de aplicação possui duas partes. Uma parte é comum a todos os dispositivos e uma parte que possui código específico para cada tipo de dispositivo. Existem três tipos: coordenador, módulo sensor e módulo atuador.

6.8.1 Aplicação

A aplicação possui funções comuns a todos dispositivos. Estas funções fazem a inicialização dos dispositivos e gerenciamento de rede. Estas funções são chamadas principalmente antes do dispositivo entrar na rede ZigBee. Esta camada também possui funções que são apenas uma interface para acessar as funções específicas de cada dispositivo.

É nesta camada que está definida a função *APL_TaskHandler*, que é a função que é sempre chamada pelo Bitcloud para executar as tarefas da aplicação. Dentro dessa função foi feita uma máquina de estados que pode ser vista na Figura 6.16.

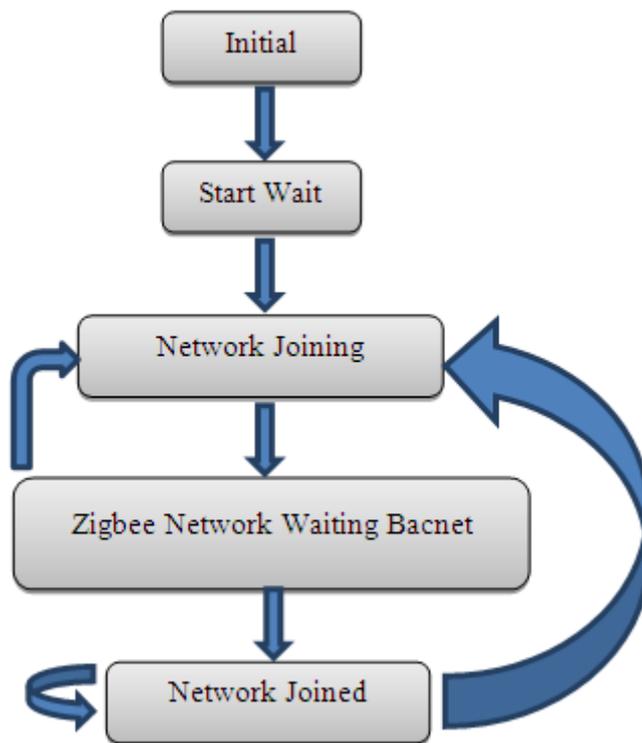


Figura 6.16. Máquina de estados da aplicação

O estado *Initial* é por onde a aplicação começa e realiza funções de inicialização de suas configurações. Assim que acabam as inicializações, é passado para o estado *Start Wait*.

Em *Start Wait*, caso o dispositivo esteja utilizando a placa Meshbean2, nada é feito, apenas espera que o botão programável seja apertado. Ao fazer isso, ou caso esteja utilizando a placa Atuadora que não possui botões, vai para o estado *Network Joining*.

No estado *Network Joining* há uma diferença se o dispositivo for ZC ou ZED. Caso seja ZC, a rede é criada. Caso contrário, o dispositivo tenta entrar na rede. Este estado é acionado caso haja algum problema de rede e seja preciso inicializar ou entrar em uma rede novamente. Enquanto não houver sucesso na criação ou para entrar na rede, o dispositivo ficará neste estado tentando. Quando tiver sucesso, passa para o estado *ZigBee Network Waiting BACnet*.

ZigBee Network Waiting BACnet é o estado em que, caso seja um ZED, o dispositivo envia uma mensagem de *Advertise* e assim inicializa o *handshake* descrito anteriormente. Uma vez feito o *handshake*, passa para o estado *Network Joined*. Caso o dispositivo seja o ZC, não faz nada neste estado e apenas passa para o próximo estado.

Em *Network Joined* a rede já está pronta para começar a comunicação BACnet. Neste estado, é chamado o *taskhandler* específico de cada dispositivo. A máquina de estado de cada um será explicada nas seções 6.8.2 e 6.8.3. O dispositivo fica neste estado enquanto não houver problemas com a rede.

6.8.2 Coordenador

O coordenador precisa ser único na rede e caso haja comunicação com o ActionView, este módulo fica ligado pela USB ao computador. Suas funções específicas se encontram no arquivo “coordinator.c”.

Esse módulo é um ZC e é responsável por criar a rede ZigBee em que os outros módulos vão entrar. Caso ocorra algum erro a este módulo, a rede inteira pode cair de modo que é necessário que ele seja robusto e capaz de tratar erros inesperados.

Além de gerenciar a rede, este módulo possui diversas responsabilidades. Na máquina de estados, detalhada na Figura 6.17, está indicada a sequência de ações que este módulo toma para garantir a confiabilidade da rede.

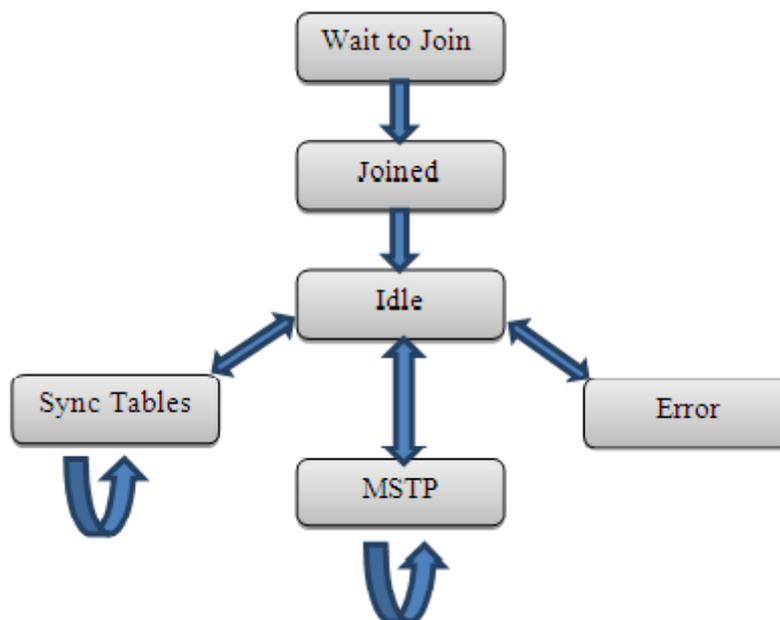


Figura 6.17. Máquina de estados do coordenador.

O estado *Wait to Join* é o primeiro estado quando o coordenador inicia. Ele fica neste estado até a rede ser criada. Neste estado, ele não toma nenhuma ação. Quando a rede é criada é acionado o estado *Joined*.

No estado *Joined* ele começa a criar a rede BACnet. Ele inicia todos os objetos que precisam inicialização e prepara suas tabelas de endereços para quando os dispositivos começarem a entrar na rede. Assim que termina a inicialização, é acionado o estado *Idle*.

No estado *Idle* checa-se qual foi o estado anterior e decide qual será o próximo. Ele pode ir para três estados *Sync Tables*, *MSTP* e *Error*. O estado *MSTP* é acionado a não ser que a condição de acionar os outros estados seja atendida. O estado *Sync Tables* é acionado a

cada 10000 vezes que passa pelo estado idle. Caso o coordenador detecte que tenha ocorrido algum erro durante a execução desta máquina de estados, ele direciona para o estado *Error*.

No estado MSTP ele roda a máquina de estados do Mestre de acordo com a norma BACnet [22]. Assim que executar essa tarefa retorna para o estado *Idle*.

No estado *Sync Tables*, é feita uma sincronização entre a tabela de endereços BACnet e a tabela de endereços ZigBee, garantindo a integridade da rede. Caso exista algum dispositivo que esteja na rede ZigBee e na BACnet porém o coordenador desconheça o endereço MSTP deste dispositivo, é enviado um *ReadProperty* no objeto *AnalogValue* 0 na propriedade *PresentValue*. O valor retornado é o endereço MSTP daquele dispositivo. A cada vez que entra neste estado é enviado no máximo um *ReadProperty*, assim garantindo que o buffer de saída do coordenador não fique lotado com estas mensagens. Assim que executar essa tarefa, retorna para o estado *Idle*.

No estado *Error* todas as *flags* de estado são zeradas e retorna para o estado *Idle*. Deste modo, a próxima vez que executar o estado *Idle* terá suas *flags* exatamente iguais àquelas quando foi executado pela primeira vez, porém suas tabelas de endereços continuarão preenchidas.

No final de cada estado, com exceção do estado *Wait to Join*, a execução é devolvida ao Bitcloud, para que ele execute as tarefas de alta prioridade. Porém, antes disso, é dado um *SYS_PostTask* passando o identificador da camada de aplicação como parâmetro. Assim, depois que o Bitcloud executar as suas tarefas, ele irá novamente chamar o *Task Handler* da aplicação. Como no final estado *Wait to Join* não é preciso voltar a executar o *Task Handler* da aplicação enquanto a rede não é criada, não é preciso chamar *SYS_PostTask* neste estado, porém é chamado no final da criação da rede.

O coordenador, que utiliza a placa MeshBean2, usa as seguinte interfaces da placa: um botão programável, o botão de *reset*, 3 LEDs para depuração e interface USB. Após um *reset*, o botão programável é responsável por disparar a criação da rede. Enquanto este botão não for apertado, a rede não será criada. Os três LEDs utilizados para depuração mostram se está ocorrendo comunicação na rede ZigBee. O LED verde mostra que a rede está funcionando, o vermelho que está enviando uma mensagem e o amarelo que está recebendo. A interface USB faz a alimentação da placa e a comunicação com o computador através de uma emulação de comunicação serial. Para essa comunicação funcionar é necessário instalar o driver CP210x USB to UART bridge da Silicon Labs que criará uma porta COM virtual. A configuração dessa porta é *Baud Rate*: 9600, *Data bits*: 8, *Parity bits*: N, *Stop bits*: 1 e *Hardware Flow Control*: None, forma esta definida como padrão na norma BACnet [22].

6.8.3 End device

Cada ZED possui um endereço único na rede e apesar de eles terem capacidade de comunicar entre si, toda comunicação é feita diretamente com o coordenador. Cada um desses dispositivos tem uma função. O Módulo Sensor lê os estados (aberto ou fechado) das janelas e porta da forma como foi mostrado na seção 6.8.3.1. O Módulo Atuador é responsável por ligar ou desligar o relé da forma como foi mostrado na seção 6.8.3.2. Apesar de ter este funcionamento diferente, ambos possuem a mesma máquina de estados, mostrada na Figura 6.18.



Figura 6.18. Máquina de estados do end device.

O estado *Wait to Join* é o primeiro estado quando o ZED inicia. Ele fica neste estado até entrar em uma rede. Neste estado, ele não toma nenhuma ação. Quando consegue entrar na rede é acionado o estado *Joined only in ZigBee*.

Em *Joined Only in ZigBee* o dispositivo já entrou na rede ZigBee porém ainda está acontecendo o *handshake*, que ocorre na máquina de estados da aplicação descrita na seção 6.8.1, para que ele entre na tabela BACnet do coordenador. Feito isso, ele passa para o estado *Joined*.

No estado *Joined*, ele começa a inicializar todos os objetos que precisam inicialização e envia um *I-am* para o coordenador. Assim que termina a inicialização é acionado o estado *Idle*.

Cada módulo realiza uma tarefa diferente em *Idle*, essa tarefa específica será explicada nas seções 6.8.3.1 e 6.8.3.2. Caso o dispositivo receba uma mensagem quando nesse estado, ele irá tratar a mensagem e enviar a resposta, se for o caso. Assim que ele receber a confirmação de que o coordenador recebeu a resposta, ele avança para o estado *Sleep Prepare*. Ambos iniciam um timer de 1 segundo para que caso eles não recebam nenhuma mensagem eles avancem para o estado *Sleep Prepare* quando o timer estourar.

No estado *Sleep Prepare*, é feito o teste para verificar se todas mensagens que deveriam ser enviadas já chegaram no seu destino ou se falharam. Caso todas já tenham sido enviadas, ele vai para o estado *Sleep*, caso contrário fica neste estado para tentar realizar o teste novamente.

Sleep é o estado em que se desliga a antena para economizar a bateria do dispositivo. Após desligar a antena, é iniciado um timer de 1 segundo, após esse tempo passa para o estado *Awakening*.

Em *Awakening* a antena é religada e volta para o estado *Idle*.

6.8.3.1 Módulo Sensor

O Módulo Sensor tem a função de ler os estados (ABERTO ou FECHADO) das janelas e porta. Para isso foi utilizada a placa Meshbean2 com alimentação por pilhas AA. As interfaces usadas foram um botão programável, o botão de *reset*, 3 LEDs para depuração e 6 pinos da interface lateral. Após um *reset*, o botão programável é responsável por fazer o dispositivo procurar a rede para entrar. Enquanto este botão não for apertado, o dispositivo ficará parado fora de qualquer rede. Os três LEDs utilizados para depuração mostram se está ocorrendo comunicação na rede ZigBee. O LED verde mostra que a rede está funcionando, o vermelho que está enviando uma mensagem e o amarelo que está recebendo. Os 6 pinos da interface utilizados foram +3,6V, GND, ADC_INPUT_1, ADC_INPUT_2, ADC_INPUT_3 e I2C_CLK. Os pinos +3,6V e GND foram utilizados como alimentação e terra, respectivamente, para o circuito optoacoplador. Os outros pinos foram utilizados como entrada digital para os estados das janelas e porta. Na tabela X pode-se ver quais os objetos BACnet desse módulo e o que cada um representa.

Tabela 6.3. Tabela de objetos BACnet do módulo Sensor

Objeto BACnet	Pino	Significado
Binary Input 0	I2C_CLK	Porta
Binary Input 1	ADC_INPUT_1	Janela 1
Binary Input 2	ADC_INPUT_2	Janela 2
Binary Input 3	ADC_INPUT_3	Janela 3
Analog Value 0	-	Endereço MS/TP
Analog Value 1	-	Numero de Aberturas da Porta

6.8.3.2 Módulo Atuador

O Módulo Atuador tem a função de comandar o relé que atua no ventilador do ar condicionado. Para isso, foi utilizada a placa atuadora com alimentação por uma fonte de 6V. Para atuar o relé foi utilizado o pino GPIO_1 como uma saída digital. Como esta placa não possui botão, este dispositivo não espera nenhum botão ser apertado para que comece a tentar entrar em alguma rede. Para ligar ou desligar o ventilador, basta dar um comando na variável BACnet *Binary Output 0*, com valor 1 ou 0 respectivamente.

Tabela 6.4. Tabela de objetos BACnet do módulo Sensor

Objeto BACnet	Pino	Significado
Binary Output 0	GPIO_1	Relé ligado ao Ventilador
Analog Value 0	-	Endereço MS/TP

6.9 SUPERVISÓRIO ACTIONVIEW

O Sistema Supervisório utilizado foi o ActionView 7.6. Os quatro principais aplicativos desse sistema que foram utilizados foram o AVStudio, ActionRU, Sppcom e Sppcomfg. O AVStudio é o módulo de configuração que trabalha offline, isto é, não há nenhuma comunicação com o campo nesse módulo. As configurações feitas nesse módulos são gravadas em duas bases de dados em Access e no arquivo de configuração do projeto que possui extensão “.ini”. O ActionRu é o módulo IHM do ActionView onde são vistos as telas e pontos criados no AVStudio. O Sppcom é o módulo de depuração da comunicação. Durante toda a comunicação, o status de cada ponto criado pode ser monitorado com este módulo. O

Sppcomfg é o aplicativo responsável pela comunicação, monitoramento e controle do sistema. Ele não possui nenhuma interface gráfica, os módulos ActionRU e Sppcom se comunicam com esse aplicativo e fazem a interface gráfica para o usuário.

6.9.1 Configuração do canal

O projeto possui apenas um canal de comunicação. Este canal utiliza BACnet Mestre MS/TP na porta COM 8 que está conectada pela USB ao coordenador. Sua configuração serial é 9600,8,n1. O *timeout* de aplicação desse canal precisa é de 3 segundos, o que garante o tempo máximo que a resposta de uma pergunta que foi roteada e respondida com sucesso pode demorar. Esse canal possui dois IED's (Dispositivos Eletrônicos Inteligentes), os módulos sensor e atuador. Como não será requisitado nenhum serviço de leitura ou escrita pelo supervisor ao coordenador, ele não precisa ser adicionado a lista de IED's.

O módulo sensor foi configurado com o primeiro endereço MS/TP permitido a escravos, 127. É enviado um *ReadPropertyMultiple* a cada 2 segundos. Neste serviço, é pedido a propriedade *Present_Value* dos objetos *Binary_Input* 0 a 3 e *Analog_Input* 1.

O módulo sensor foi configurado com o endereço MS/TP 128. É enviado um *ReadPropertyMultiple* a cada 5 segundos. Neste serviço, é pedida a propriedade *Present_Value* do objeto *Binary_Output* 0. Também foram configuradas duas variáveis utilizadas para enviar o comando de ligar e desligar o ventilador.

Além das Variáveis criadas para monitorar o ambiente, foram criadas variáveis internas utilizadas nas rotinas do sistema. As variáveis internas criadas foram para a programação do horário de ligar e desligar, indicar que há alguma janela aberta e para contar a quanto tempo tem janelas abertas no sistema. As rotinas que utilizam estas variáveis estão descritas na seção 6.9.3.

Todo projeto no ActionView possui uma árvore onde pode-se acessar as configurações do projeto. Nessa árvore pode-se acessar informações das regionais, dos sistemas, dos grupos, das variáveis e do canal e seus IED's (sigla para IED em inglês). Todas as variáveis que forem valores lidos do campo ou de comando precisam estar endereçadas corretamente. Variáveis internas não devem ser endereçadas. Na figura X temos à esquerda a árvore do sistema e à direita os endereços das variáveis que se comunicam com o campo.

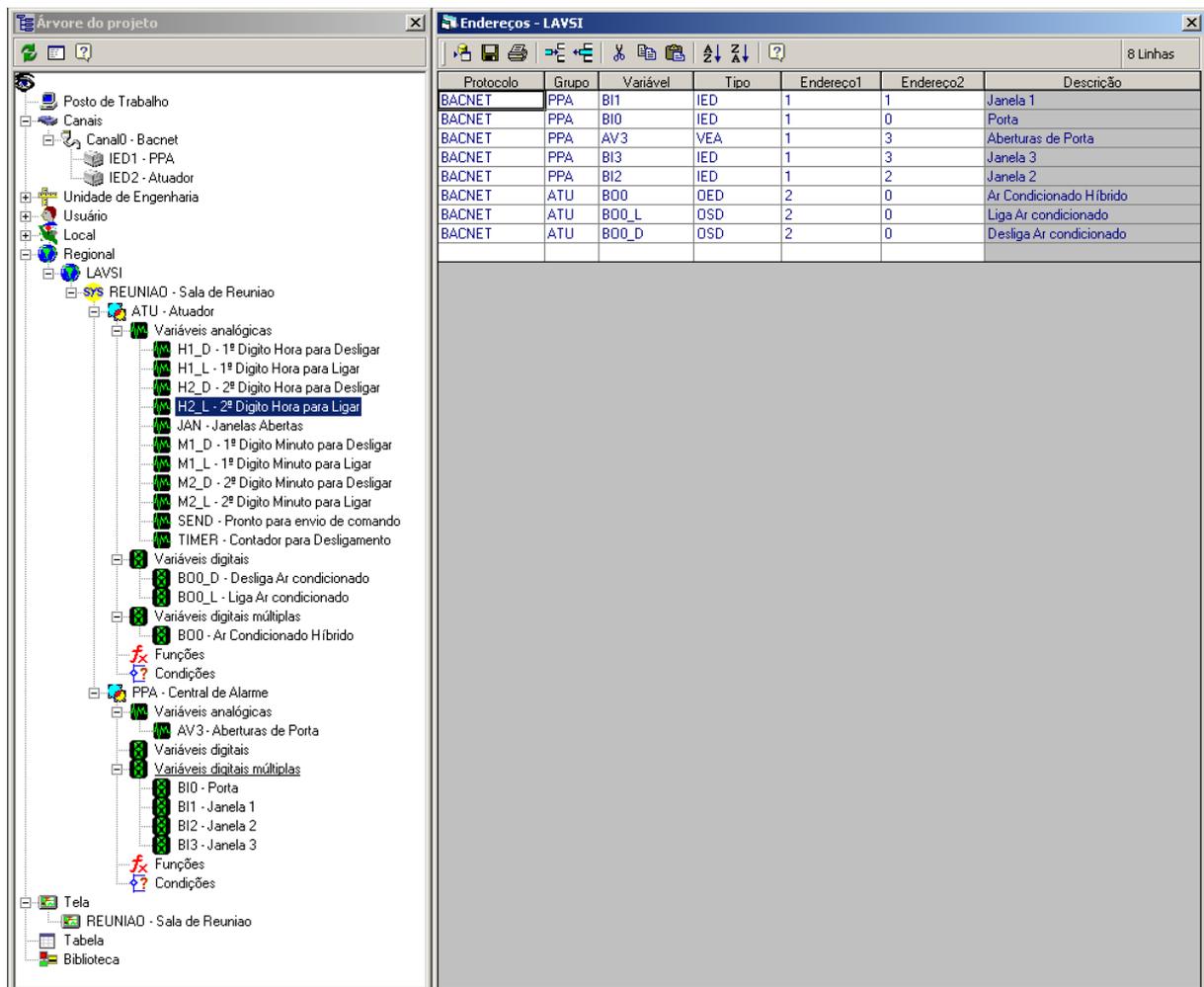


Figura 6.19. Árvore de Sistema e Endereços das Variáveis

6.9.2 Tela

Como este sistema monitora apenas um ambiente com um número pequeno de pontos, foi criado uma única tela onde é possível visualizar todos os pontos do sistema. Para fazer o desenho da tela foi utilizado o Microsoft Visio. Todos os textos da tela foram adicionados pelo AVStudio e não fazem parte da figura de fundo. A tela visualizada pelo ActionRu pode ser vista na Figura 6.20.

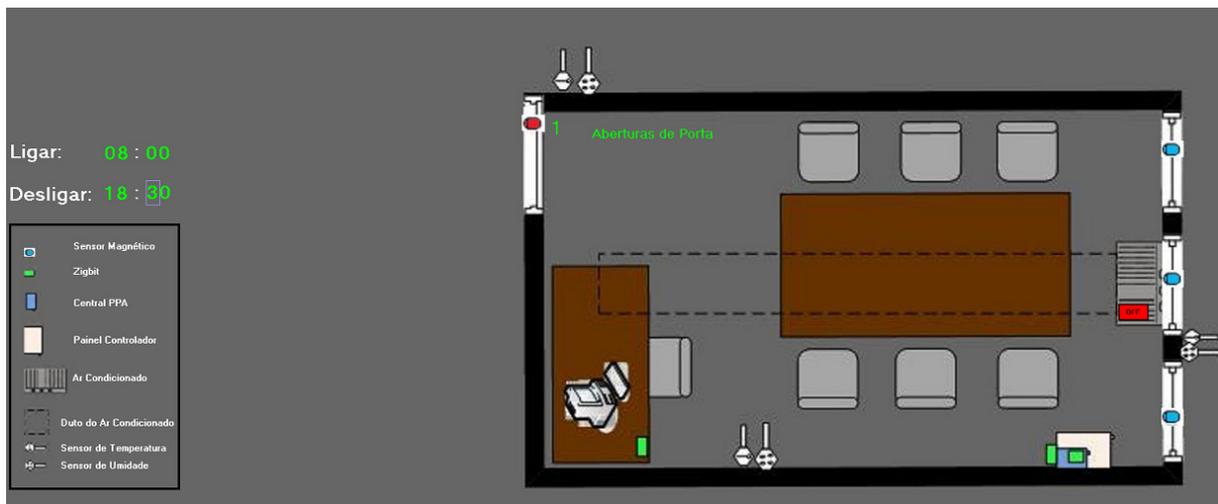


Figura 6.20. Tela gráfica ActionView.

O símbolo do sensor de abertura ao detectar abertura fica vermelho, caso contrário fica azul. Ao lado da porta, há um contador indicando quantas vezes ela foi aberta desde a inicialização do módulo sensor. No canto inferior esquerdo, possui uma legenda para a imagem da sala. Em cima das legendas, há um horário programável para ligar e desligar o ar condicionado.

6.9.3 Rotinas

O ActionView dá a possibilidade de programar em VBS rotinas específicas para seu projeto. Foram utilizadas duas rotinas neste projeto que podem ser vistas nos anexo 1, 2 e 3. A rotina OnValueChanged, que roda sempre que alguma das janelas troca de valor, passa a variável XXX para 1 sempre que houver uma janela aberta. Essa variável é utilizada por outra rotina. A rotina OnTimer roda a cada 3 segundos. Essa rotina primeiro checa se chegou o horário programado para ligar ou desligar o ar condicionado, caso tenha chegado envia o comando desejado. E também checa se alguma das janelas ou porta está aberta e caso tenha uma janela ou porta aberta por mais de 2 minutos é enviado um comando de desligar. Essa rotina não envia comando caso o ar condicionado já esteja no estado desejado. A rotina OnMouseClicked roda quando se clica no horário configuravel para ligar e desligar o ar condicionado pela tela do supervisor.

CAPÍTULO 7 ANÁLISE DE RESULTADOS

O sistema desenvolvido para este trabalho possui dois grandes objetivos. Um deles é montar uma rede BACnet sobre ZigBee e com isso ser capaz de monitorar o sistema pelo supervisor. O outro é com base nos estados, aberto ou fechado, das janelas e porta realizar um controle sobre o ar condicionado capaz de economizar energia. Por isso a análise do sistema foi dividida em duas: análise da rede e análise da economia de energia.

7.1 ANÁLISE DA REDE

A comunicação entre os dispositivos foi um dos processos mais complexos do projeto para conseguir robustez. Nos últimos testes não foram encontrados problema de comunicação e o sistema estava funcionando como um todo. Para analisar a comunicação, foi utilizado o log do ActionView. A Figura 7.1 apresenta parte deste log que será analisado e um log mais extenso pode ser visto no anexo 4,

```
[TP 11] 08/31 - 11:37:42,671 SND 00 ->55 FF 05 7F 01 00 33 A8 01 04 02 03 EF 0E 0C 00 C0 00 00 1E 09 55 1F 0C 00 C0 00 01 1E 09 55 1F 0C 00 C0 00 02 1E 09 55 1F 0C 00 C0 00 03 1E 09 55 1F 0C 00 80 00 03 1E 09 55 1F 3E C0 FF
[TP 12] 08/31 - 11:37:42,734 REC 00 ->55
[TP 12] 08/31 - 11:37:42,781 REC 00 ->FF 07 01 7F 00 00 21
[TP 11] 08/31 - 11:37:42,828 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:42,843 REC 00 ->55
[TP 12] 08/31 - 11:37:42,890 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:42,937 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:42,953 REC 00 ->55
[TP 12] 08/31 - 11:37:43,000 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:43,046 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:43,062 REC 00 ->55
[TP 12] 08/31 - 11:37:43,109 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:43,156 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:43,171 REC 00 ->55
[TP 12] 08/31 - 11:37:43,218 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:43,265 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:43,281 REC 00 ->55
[TP 12] 08/31 - 11:37:43,328 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:43,375 SND 00 ->55 FF 05 80 01 00 0F B3 01 04 02 03 97 0E 0C 01 00 00 00 1E 09 55 1F BD BC FF
[TP 12] 08/31 - 11:37:43,406 REC 00 ->55
[TP 12] 08/31 - 11:37:43,453 REC 00 ->FF 07 01 80 00 00 30
[TP 11] 08/31 - 11:37:43,484 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:43,500 REC 00 ->55 FF 06 01 80 00 12 B8 01
[TP 12] 08/31 - 11:37:43,546 REC 00 ->00 30 97 0E 0C 01 00 00 00 1E 29 55 4E 91 01 4F 1F 21 32 55 FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:43,593 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:43,609 REC 00 ->55 FF 00 01 00
[TP 12] 08/31 - 11:37:43,656 REC 00 ->00 00 73
[TP 11] 08/31 - 11:37:43,687 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:43,703 REC 00 ->55
[TP 12] 08/31 - 11:37:43,765 REC 00 ->FF 06 01 7F 00 49 61 01 00 30 EF 0E 0C 00 C0 00 00 1E 29 55 4E 91 01 4F 1F 0C 00 C0 00 01 1E 29 55 4E 91 00 4F 1F 0C 00 C0 00 02 1E 29 55 4E 91
[TP 12] 08/31 - 11:37:43,812 REC 00 ->00 4F 1F 0C 00 C0 00 03 1E 29 55 4E 91 00 4F 1F 0C 00 80 00 03 1E 29 55 4E 44 3F 80 00 00 4F 1F 8B 34 55 FF 00 01 00 00 00 73
```

Figura 7.1. Trecho do log de comunicação do ActionView

As informações que existem no log são a data com o horário (até milissegundos) em que a mensagem foi enviada ou recebida (se foi enviada é SND, se foi recebida é REC) e em qual canal de comunicação ela passou, sendo que nesse caso tem-se somente o canal 0. Depois disso há a mensagem em hexadecimal.

As mensagens que estão em preto foram enviadas pelo coordenador para o ActionView. Elas podem ser um envio de *Token*, caso o byte após o '55 FF' seja '00', ou *Reply_Postponed*, caso o byte após o '55 FF' seja '07'. As mensagens em vermelho são o Actionview enviando o *Token* para o coordenador. As mensagens em verde são um *Read_Property_Multiple* sendo enviado para o módulo sensor e sua resposta. Em azul temos o mesmo para o módulo atuador.

A análise deste log foi dividida em uma análise da rede ZigBee, outra da rede MS/TP e uma conclusão sobre o comportamento geral da rede.

7.1.2 ZigBee

Apesar de não haver um log próprio da comunicação pela rede Zigbee pode-se deduzir o comportamento da rede pela análise do log do ActionView. Uma vez que os dispositivos estão na rede e não há nenhuma mensagem na rede ZigBee a não ser que venha roteada do ActionView. E se a resposta chega de volta ao supervisor também sabe-se que não houve problemas na rede ZigBee para o envio da resposta. Assim, o log do ActionView é suficiente.

O log da Figura 7.1 mostra bem como esta comunicação pode funcionar pois temos uma pergunta para cada dispositivo e tempos de resposta diferentes. No envio do *Read_Property_Multiple* para o módulo sensor a resposta demora 1 segundo e 32 milissegundos. Esse tempo é uma demora aceitável pois o módulo pode dormir durante 1 segundo. Caso tente enviar pergunta durante esse tempo ela terá que esperar o módulo acordar para poder enviada. No envio deste mesmo serviço ao módulo atuador, obteve-se uma resposta quase imediata, apenas 125 milissegundos, pois assim que o coordenador recebeu o *Token* já enviou de volta a resposta. Isso significa que o módulo já estava acordado no momento em que o coordenador tentou enviar a mensagem, assim a demora da mensagem chegar foi apenas decorrente de tempo de processamento dela em cada um dos dispositivos e o tempo de transferência das mensagens.

7.1.3 MS/TP

Como numa rede EIA-485 todos os dispositivos recebem todas as mensagens trafegadas na rede, o log do ActionView é ideal para monitorar tudo o que acontece nesta rede. No log da

Figura 7.1 pode-se ver que a quantidade de mensagens sendo trocadas é muito alta pois o intervalo entre cada mensagem enviada pelo actionview é de até de 157 milissegundos.

Como o tempo entre o envio de uma mensagem e sua resposta desta rede é desprezível para esta aplicação, será analisada a eficiência, como definida no Capítulo 5. Para esse calculo será considerado o cabeçalho MS/TP e o *padding byte* (byte ' FF' opcional que é enviado pelo actionview no final de cada mensagem [22] como estabelecimento e encerramento da mensagem, e a NPDU que possui dados para camada de rede e a APDU que é a mensagem enviada para a camada de aplicação como a informação que deve ser transmitida. O trecho do log (anexo 4) analisado foi do dia 31/08/2010 das 11:37:38,359 às 11:37:41:859, sendo que nesse trecho há uma requisição de *Read_Property_Multiple* para cada módulo. A *Read_Property_Multiple* para o módulo atuador possui 15 bytes de informação de um total 26, sua resposta possui 18 de um total 28. Para o módulo sensor, a requisição possui 51 bytes de informação de um total 62, sua resposta possui 73 de um total 83. Há 31 envios e 31 recebimentos de mensagens sem informação (*Token* e *Reply_Postponed*), essas mensagens possuem 8 bytes e as enviadas pelo ActionView possuem o *padding byte* totalizando 9. Assim, temos 157 bytes de informação de um total de 726. Dividindo 157 por 726 temos que a eficiência é de 21,63%.

Com estes cálculos vemos que a eficiencia é baixa. Isso ocorre em redes MS/TP quando há muitos mestres proporcionalmente com o número de escravos, pois nesse caso temos muitas mensagens de gerenciamento de rede. Apesar dessa baixa eficiência não houve comprometimento da comunicação pois não houve nenhuma perda de mensagem por esse motivo.

7.1.4 Análise Geral

Na análise de comunicação não basta analisar se cada rede se comportou como deve, precisa também verificar se os tempos de respostas são aceitáveis e quando esse tempo pode comprometer alguma aplicação.

Neste trabalho foi monitorado o estado de portas e janelas para controle de um sistema de ar condicionado. Como a temperatura de um ambiente não muda muito em poucos segundos apenas com a abertura de uma janela ou porta, esse critério apenas limita que o sistema responda em menos de 30 segundos (um quarto do tempo estimado pela análise da seção 7.2 para que o sistema desligue o ar condicionado). Outro critério pode ser para controle de acesso, nesse caso precisa-se ter a informação de que uma porta ou janela foi aberta rapidamente. Uma pessoa passando por uma porta demora entre 2 e 4 segundos. Assim, é importante ter essa informação em pelo menos 2 segundos após a abertura para não perder o evento.

O ActionView envia uma solicitação dos estados dos sensores a cada 2 segundos para o módulo sensor e pelo log (anexo 4) vemos que no pior caso a resposta demora 1,3 segundos para chegar, de modo que, no pior caso, a atualização de um estado pode demorar 3,3 segundos. Esse tempo satisfaz primeiro critério, porém é maior do que o tempo mínimo que uma porta pode ficar aberta. Com esse tempo, pode-se perder alguma abertura de porta, porém em diversos testes de laboratório foi visto que este caso é bastante raro. E mesmo assim, caso o supervisor não receba a informação de que a porta está aberta, o número de aberturas da porta é atualizado pelo módulo sensor e o supervisor recebe essa informação através desta outra variável.

Deste modo, podemos concluir que os tempos do sistema satisfazem para controlar o ar condicionado e para controle de acesso. Esse tempo porém está alto para alguns sistemas que precisem de algum controle com resposta mais rápida ou de um relatório mais detalhado. Para esses sistemas é preciso que os ZED nunca durmam assim tem-se uma resposta bem mais rápida.

7.2 ANÁLISE DA ECONOMIA DE ENERGIA

Para se confirmar se a economia de energia foi alcançada, vários testes foram realizados. Para tanto, foi necessário utilizar alguns produtos comerciais que nos fornecessem as informações necessárias de forma confiável.

Para realizar a medição do consumo de energia, utilizou-se o medidor de energia trifásico existente na sala de reunião do LARA, da marca Landis Gyr, modelo zmd128amt-8131 (Figura 7.2) e o software de serviço Amazon v.3.0. Esse último foi necessário para ter os valores medidos no computador.



Figura 7.2. Medidor de energia trifásico

Outro dispositivo necessário à análise foi o controlador MT531Ri plus do fabricante Fullgauge. Esse possui uma saída auxiliar que permite sua operação como termostato ou umidostato. É um instrumento que indica e controla a temperatura e umidade ambientes. Essas medidas são obtidas a partir de sensores existentes dentro da sala de reunião, externo à sala, mas dentro do laboratório e externo ao prédio. São mostradas em um display ligado ao controlador além de poderem ser visualizadas pela tela do computador. Na Figura 7.3 pode ser visualizado esse equipamento.



Figura 7.3. Controlador MT531Ri plus

Foi necessário também o conversor CONV32, Figura 7.4, que se encarrega de transformar o padrão elétrico utilizado pelo PC (USB) para o padrão elétrico EIA-485 utilizado pelo controlador.



Figura 7.4. Conversor EIA-485 para USB.

O software utilizado para gerenciar o sistema de refrigeração também desenvolvido pela Full Gauge foi o Sitradi. Ele avalia, configura e armazena continuamente dados de temperatura, umidade, tempo, pressão e voltagem, permitindo a modificação on-line dos parâmetros de operação dos instrumentos. Foi instalado no computador em que está conectado o controlador através de uma Interface. O esquemático das ligação entre Controlador, CONV32 e PC é mostrado na Figura 7.5.

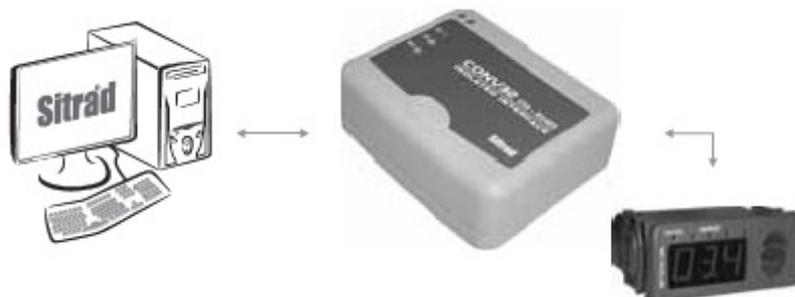


Figura 7.5. Esquema de conexões PC, CONV32 e controlador .

7.2.2 Testes

As seguintes leituras de consumo de energia, de temperatura e umidade foram realizadas na sala de reunião do LARA:

1. No dia 27 de julho de 2010, terça-feira, de 10:30 às 22:30 horas, com a sala de reunião em funcionamento normal;
2. No dia 31 de julho de 2010, sábado, de 10:30 às 22:30 horas, com todas as janelas e porta abertas e o ar condicionado ligado;
3. No dia 12 de agosto de 2010, quinta-feira, de 12:00 às 14:00 horas, com janelas e porta abertas e ar condicionado ligado, depois, fechou-se as janelas e a porta e seguiu-se com o teste até as 16:00 horas;
4. No dia 14 de agosto de 2010, sábado, de 10:30 às 22:30 horas, com todas as janelas e porta fechadas e o ar condicionado ligado;
5. No dia 30 de agosto de 2010, segunda-feira, quando a temperatura ambiente estava no *setpoint*, 22°C, deixou-se apenas uma janela aberta e o ar condicionado ligado por 20 minutos, de 15:53 até 16:13 horas. Depois, com a temperatura no *setpoint*, essa mesma janela foi fechada e o sistema refrigerador foi desligado e prosseguiu-se com o teste de 16:21 até 16:41 horas. Por fim, com todas as janelas fechadas, ligou-se o ar condicionado às 16:41 e fez-se as medições até a temperatura atingir o *setpoint* novamente;
6. No dia 31 de agosto de 2010, terça-feira, de 9:30 às 21:30 com as janelas fechadas, ar condicionado ligado e a porta aberta.

7.2.3 Análise dos testes

O primeiro teste efetuado, dia 27 de julho, revelou um consumo de 1865 watt-hora e o gráfico das variações de temperatura desse período pode ser visualizado na Figura 7.6. Nesse dia, a sala passou a maior parte do tempo com a porta fechada e as janelas sempre fechadas, funcionamento normal. O compressor foi ligado três vezes por um curto período de tempo. No resto do período desse teste, o evaporativo conseguiu manter a temperatura da sala próxima ao *setpoint*.

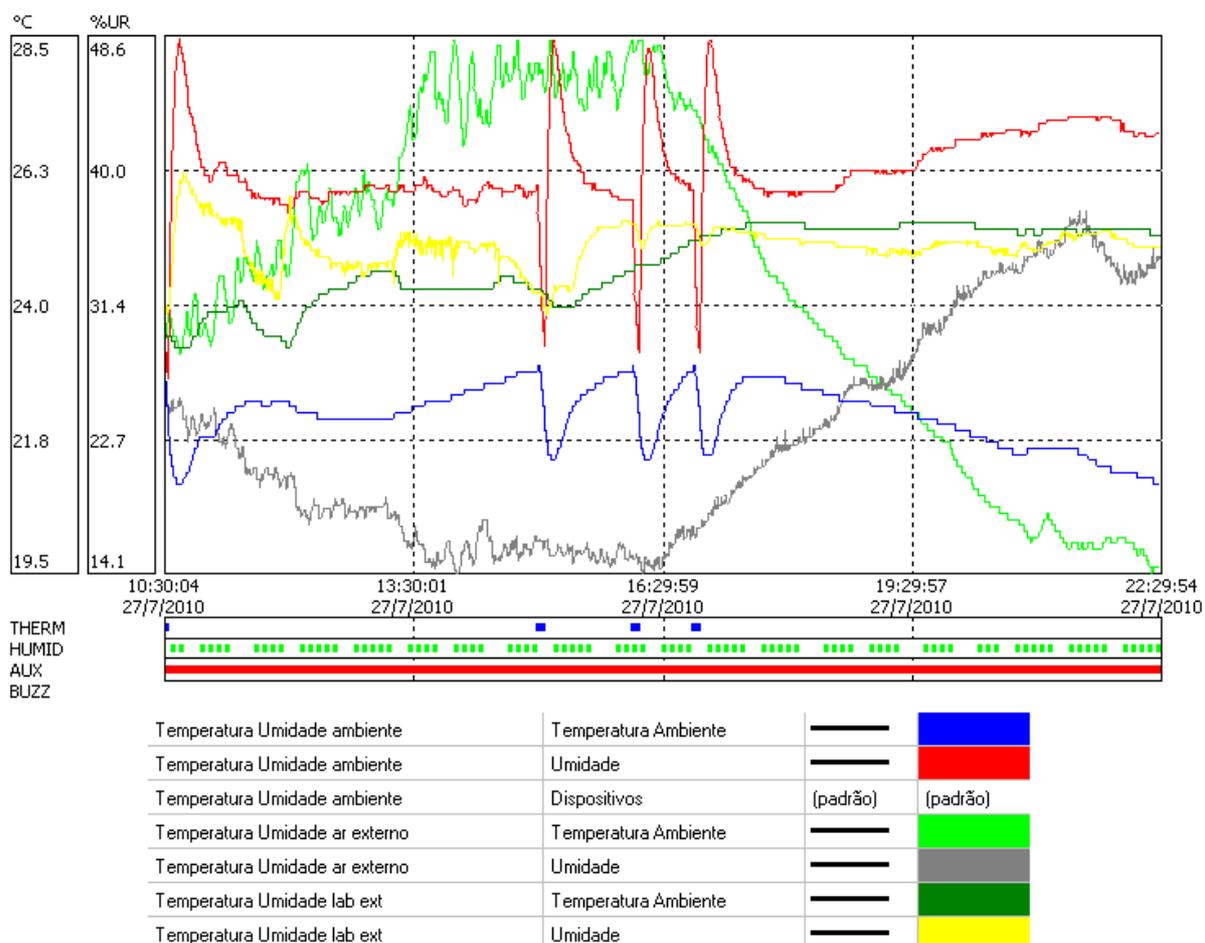


Figura 7.6. Variação de temperatura e umidade com funcionamento normal da sala.

O teste seguinte, dia 31 de julho, demonstrou um aumento de aproximadamente 42% do consumo de energia ao se comparar com o primeiro teste, totalizando um valor de 2644 watt-hora. Como esperado, já que o sistema de ar condicionado teve que manter a temperatura no *setpoint* (22°C) com as janelas e a porta da sala abertas. A Figura 7.7 representa a variação de temperatura e umidade desse dia. Esse valor poderia ser ainda maior, se o restante do laboratório não estivesse sendo refrigerado por outro sistema. E à

noite, como a temperatura externa caiu, o fato do sistema de refrigeração do laboratório ter sido desligado, não acarretou uma carga de trabalho alta para o compressor.

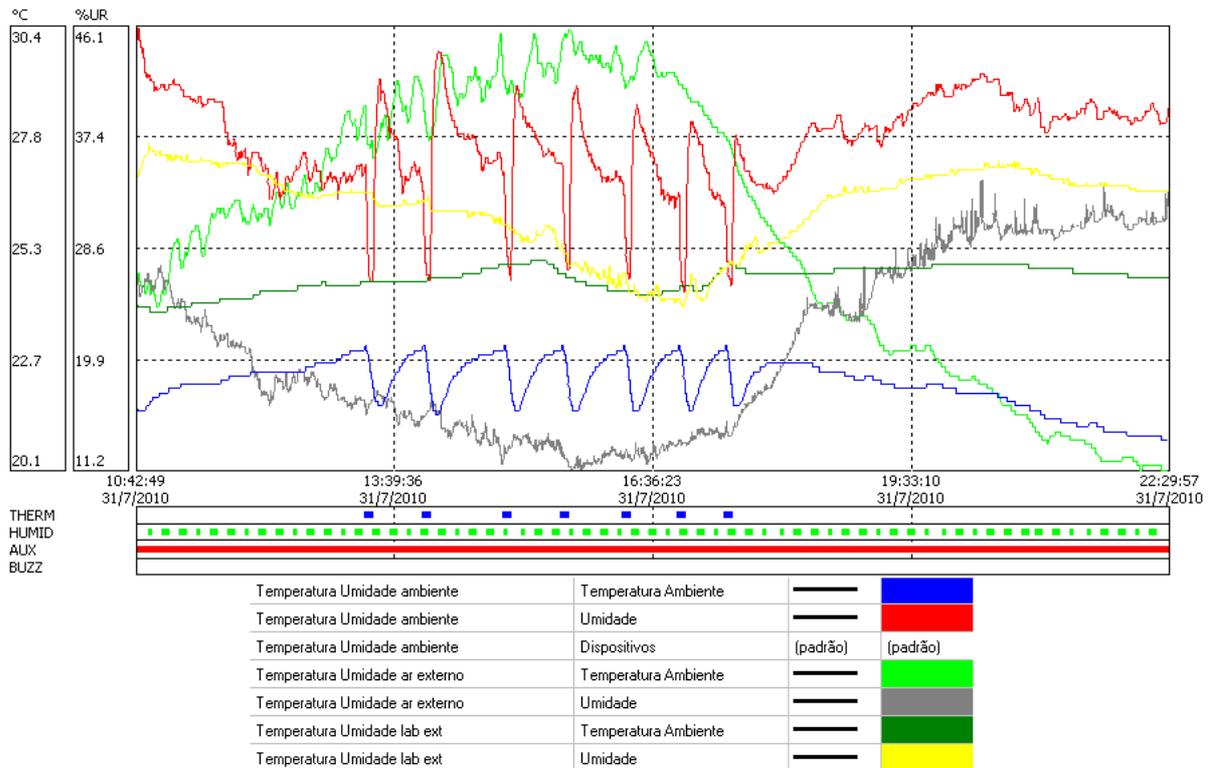


Figura 7.7. Variação de temperatura e umidade com a porta e janelas abertas.

O teste realizado no dia 14 de agosto não foi monitorado pelo software Sitrad, por isso não há gráfico da temperatura. Nesse experimento, a sala de reunião permaneceu todo o período com a porta e janelas e fechadas. O consumo desse período foi medido em 2210 watt-hora. Esse valor foi alto ao se comparar com o dia em que a sala reunião estava em funcionamento normal (teste 1). Isto aconteceu porque a temperatura externa do dia 14 de agosto entre às 10h e 22h variou entre 20° C e 29° C, sendo que de 12h às 17h a temperatura oscilava apenas entre 28° C e 29° C, temperatura esta mais elevada que a do dia 27 de julho (teste 1).

O próximo teste ocorreu no dia 12 de agosto e foram feitas duas medições. A primeira entre 11h57min e 14h06min, com porta e janelas abertas, revelou um consumo de 508 watt-hora. A segunda entre 14h06min e 16h02min, com porta e janelas fechadas, apresentou um consumo de 621 watt-hora. A Figura 7.8 representa o gráfico da variação de temperatura e umidade desses períodos. No primeiro caso, o laboratório apresentava uma temperatura próxima ao *setpoint* da sala de reunião e a temperatura externa estava próxima de 26° C. No segundo caso, o laboratório e a temperatura externa estão bem mais elevadas, e mesmo

que as janelas e portas estivessem fechadas, o compressor precisou ser ligado mais vezes para compensar esse aumento de temperatura.

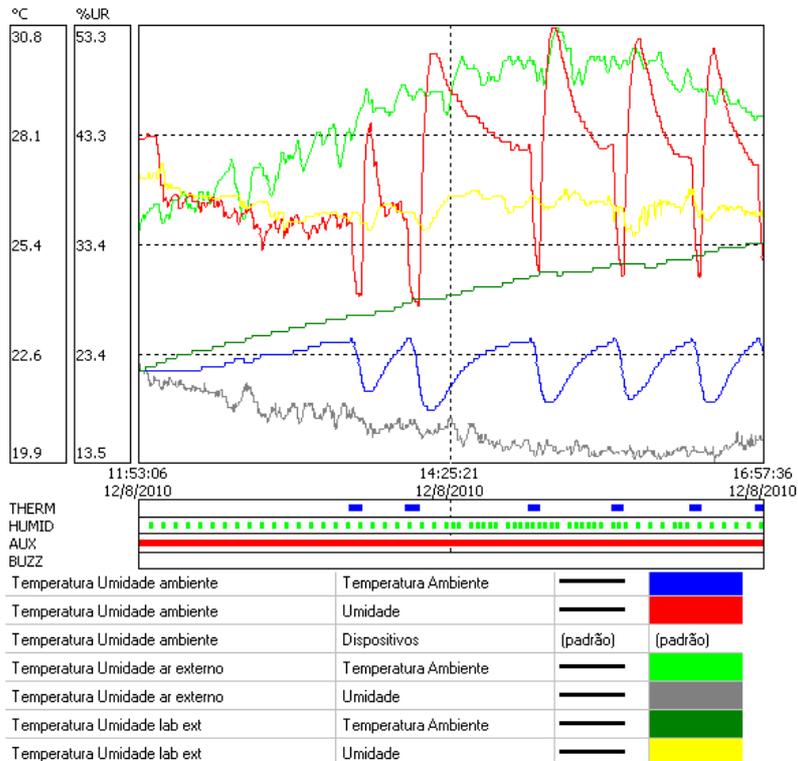


Figura 7.8. Variação de temperatura e umidade em situações distintas.

O teste efetuado dia 30 de agosto, no horário entre 15h54min e 16h14min, com apenas uma janela aberta, gerou um gasto de 517 watt-hora. Às 16h22min (temperatura da sala no *setpoint*) o sistema refrigerador foi desligado, permanecendo nesse estado até às 16h42min, o que fez com que a temperatura interna à sala chegasse a 27°C. A partir de 16h42min, o ar condicionado foi ligado, a janela que estava aberta foi fechada e a temperatura só atingiu novamente o *setpoint* após uma hora. Essas oscilações podem ser visualizadas na Figura 7.9. Nessa imagem, é notório o desempenho realizado pelo compressor. Logo que a janela é aberta, ele é ligado e começa a trabalhar com uma carga alta para manter a temperatura interna à sala no *setpoint*. Da mesma forma, quando o ar condicionado é ligado novamente, com a temperatura interna à sala bem alta, o compressor precisou ficar em funcionamento por um longo período para que fosse atingido o *setpoint*.

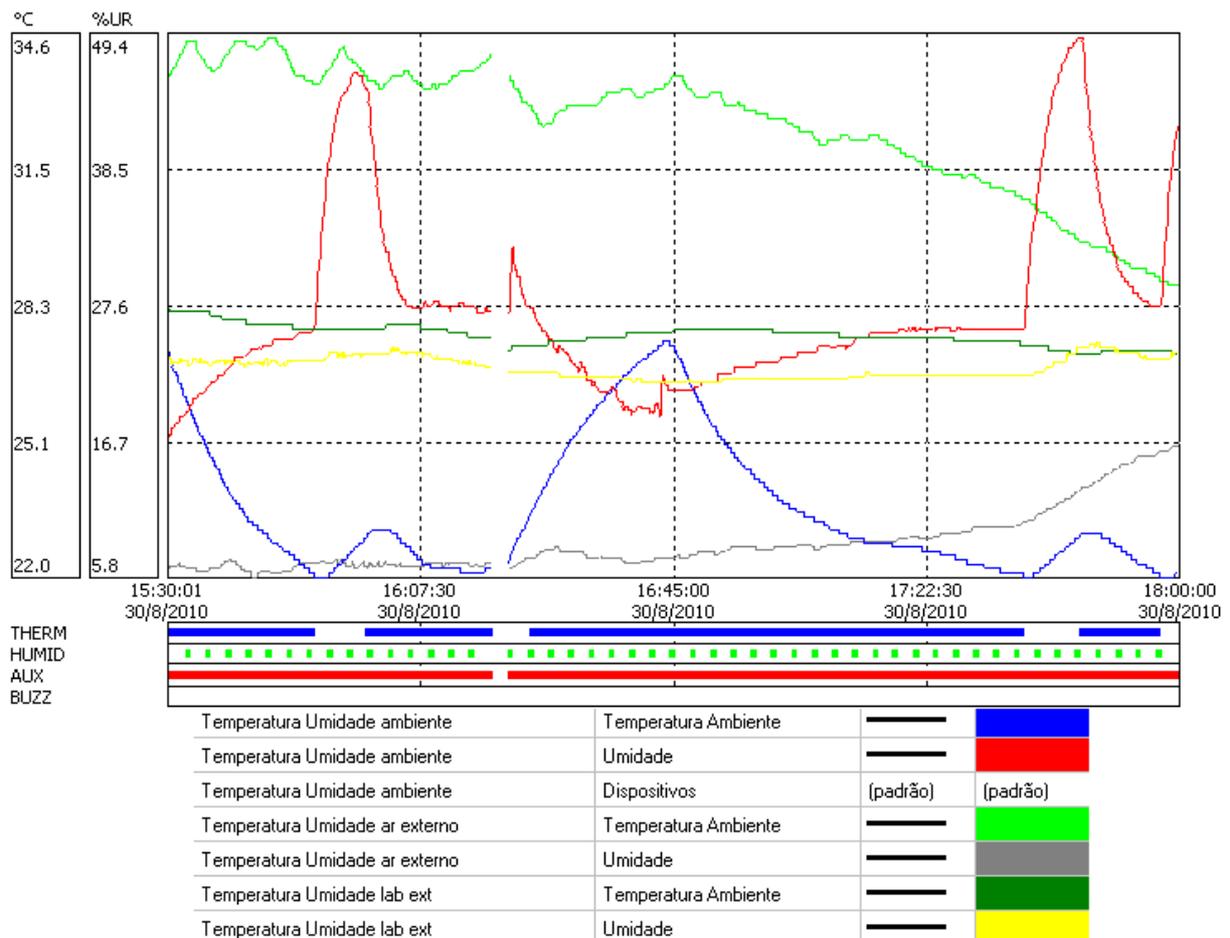


Figura 7.9. Interferência de uma janela aberta no consumo.

Por fim, o teste do dia 31 de agosto, com apenas a porta aberta, revelou um consumo de 7195 watt-hora. Houve um aumento de 172% no consumo ao se comparar ao experimento que possuía, além da porta, as janelas abertas (Figura 7.7). Dois fatores são responsáveis por esse aumento. Primeiro, o fato de que nesse dia a temperatura externa atingiu valores muito maiores que nos outros experimentos de 12 horas de duração. O outro fator foi o sistema refrigerador do restante do laboratório estar desligado, de modo que o fluxo de ar pela porta fez com que o ar condicionado da sala e reunião precisasse do compressor durante um período maior de tempo para manter a temperatura no *setpoint*. Essa análise pode ser observada na Figura 7.10.

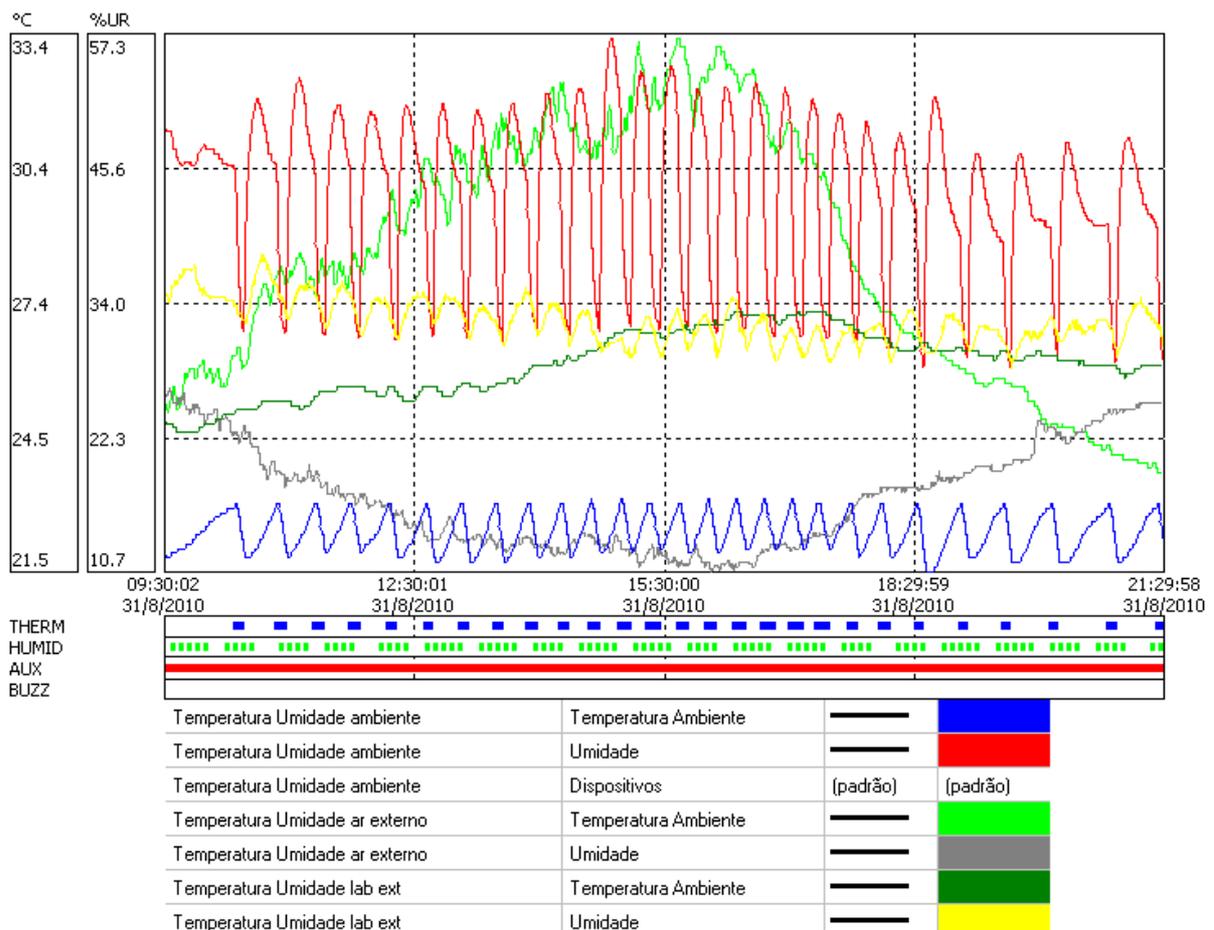


Figura 7.10. Influência da porta no consumo.

A partir dos testes realizados, constata-se que tanto a porta quanto as janelas influenciam consideravelmente no consumo de energia se estiverem abertas. Esse efeito é potencializado caso haja uma grande diferença entre a temperatura externa e o *setpoint*.

Com base na Figura 7.9, observa-se que para manter a temperatura interna próxima ao *setpoint*, no caso de ter uma janela aberta, é preciso que o compressor fique ligado durante todo esse instante. E essa análise se estende também para a porta como mostra a Figura 7.10. Mesmo que a temperatura interna se eleve muito quando o ar condicionado estiver desligado, o trabalho efetuado pelo compressor para resfriar esse ambiente após esse aumento é comparável ao trabalho desempenhado para manter a temperatura com a janela aberta. Ainda assim, é preferível desligar o ar condicionado após um certo tempo que uma porta ou janela tenha sido aberta, pois, dessa forma, a temperatura interna irá aumentar e a sensação térmica dessa elevação será percebida e mais rápido uma atitude será tomada. Quanto mais rápido a porta ou janela for fechada, menor será o dispêndio de energia. Por isso, convencionou-se em dois minutos para esperar, após uma abertura, para se desligar o ar condicionado, já que esse tempo não é tão alto e permite que haja ou uma passagem pela porta ou uma abertura rápida de uma das janelas.

A Tabela 7.1 resume os dados retirados dos testes efetuados.

Tabela 7.1. Dados Coletados

Data	Período	Número de horas	Condição das portas e janelas	Temperatura Ext (faixa)	Temp int (faixa)	Consumo de energia
27/07/2010	10h38min-22h35min	12h	Dia comum*	19,5°C-28,5°C	21 °C-23 °C	1865 Wh
31/07/2010	10h31min-22h31min	12h	Abertas	20°C-30°C	21 °C-23 °C	2644 Wh
12/08/2010	11h57min-14h06min	2h	Abertas	26°C-29°C	21 °C-23 °C	508 Wh
12/08/2010	14h06min-16h02min	2h	Fechadas	28,3°C-30,8°C	21 °C-23 °C	621 Wh
14/08/2010	10h31min-22h25min	12h	Fechadas	20°C-29°C**	21 °C-23 °C	2210 Wh
30/08/2010	15h54min-16h14min	20min	1 janela aberta	33°C-34°C	22°C-23°C	517 Wh
30/08/2010	16h22min-16h42min	20min	1 janela aberta	32°C-33°C	22°C-27°C	-
30/08/2010	16h42min-17h40min	1h	Fechadas	29°C-33°C	22°C-27°C	***
31/08/2010	9h34min-21h39min	12h	Porta aberta	24°C-33°C	21 °C-23 °C	7195 Wh

* Um dia de uso rotineiro da sala

** [30]

*** Consumo não registrado.

CAPÍTULO 8 CONCLUSÃO E TRABALHOS FUTUROS

O projeto teve como objetivo a implementação de uma rede sem fio de sensores e atuadores para se obter economia de energia com sistemas de ar condicionado.

A instalação dos sensores foi feita com algumas modificações para que o sinal emitido por esses fosse constante e que acusasse os estados dos setores (porta ou janela). Apesar da limitação do modelo da central de larmes de não funcionar adequadamente se duas zonas estiverem abertas ao mesmo tempo, pois ela só processa um sinal de cada vez, isso não interferiu nos resultados esperados. Isso porque, se ao menos um dos setores estiverem abertos, o sistema implementado tomará uma atitude.

A combinação dos protocolos BACnet e Zigbee se mostrou uma escolha adequada à automação predial. Conseguiu-se montar uma rede estável com os tempos de resposta adequados à aplicação. E uma vez que o supervisório ActionView é capaz de se comunicar através do protocolo BACnet MS/TP, um roteamento simples foi capaz de integrar com sucesso as redes cabeada e sem fio.

A integração entre controle de acesso e controle do ar condicionado foi satisfatória. Toda vez que há uma mudança de estado de algum setor, o supervisório registra o evento e toma a atitude de controle necessária. Optou-se por a cada 2 minutos que alguma zona tenha sido aberta a atitude seja o desligamento do ar condicionado, já que dessa forma, como a análise dos testes mostra, é possível obter economia de energia.

Como o ambiente utilizado para os testes, a sala de reunião do LARA, é pequeno e quase não há abertura das janelas, não se sabe até quanto pode se ganhar em termos de economia de energia. Assim, sugere-se fazer esses mesmos testes em ambientes maiores e com uma quantidade maior de ocupantes.

Outro projeto importante para dar continuidade ao estudo, seria integrar as medidas de temperatura e umidade ao sistema supervisório para que esse realize um controle mais apurado e preciso, uma vez que a temperatura externa influencia bastante na demanda do sistema de ar condicionado.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Apostila de introdução a sensores e automação e controle, Universidade Gama Filho. Disponível em: <<http://www.ebah.com.br/introducao-a-sensores-e-automacao-pdf-a20913.html>> Acesso em 15 de Agosto de 2010.
- [2] DIAS, RUBENS. A; MATTOS, CRISTIANO R.; BALESTIERI, J.A.P. Uso racional de energia: ensino e cidadania. São Paulo: Editora UNESP. 2006. 190 p.
- [3] KNAPP, LAURA. A onda verde chega às construções. Horizonte Geográfico, Editora Horizonte. Disponível em: <http://www.cbcs.org.br/noticias/cbcsimprensa/200806_aondaverdchegconstrucoes.php?> Acesso em 17 de Agosto de 2010.
- [4] KRUTZ, RONALD L. Securing SCADA systems. Indianapolis: Wiley Publishing. 2006.
- [5] CLARKE, GORDON; REYNDERS, DEON. Practical Modern SCADA Protocols: DNP3, 60870.5 and Related Systems. Oxford : IDC Technologies. 2004.
- [6] Manual Actionview Introdução. Spin Engenharia de Automação Ltda. 2008.
- [7] KUROSE, JAMES F.; ROSS, KEITH W. Computer Networking: A Top-Down Approach Featuring the Internet. Addison Wesley Longman. 2000.
- [8] BRITO, MARCOS VINÍCIUS T. de; SILVA, ROBSON PAULO F. da. Implementação de Bacnet sobre Zigbee para Rede de Automação Predial Wireless. 2009. Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF.
- [9] KOPETZ, H. Real-Time Systems: Design Principles for Distributed Embedded Applications. Boston: Kluwer Academic Publishers, 1997.
- [10] BAILEY, DAVID; WRIGHT, EDWIN. Practical SCADA for Industry. Oxford : IDC Technologies. 2003.
- [11] WirelessHart: Wireless Aplications. Disponível em: <http://www.hartcomm.org/protocol/wihart/wireless_applications.html>. Acesso em: 12 jul. 2009.
- [12] OLIVEIRA, DANIEL B. S. de; ZANATTA, MÁRCIO P. de B. F.; SANTOS, RODRIGO A.; Redes de Sensores e Atuadores Wireless para Automação Predial. 2007. Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF.
- [13] Zigbee Specification. Zigbee Alliance Inc. 2007.
- [14] THONET, GILLES; ALLARD-JACQUIN, PATRICK; COLLE, PIERRE. Zigbee Wi-Fi Coexistence. Schneider Eletric White Paper and Test Report. 2008.
- [15] Controle remoto e aquisição de dados via XBee/ZigBee (IEEE 802.15.4). Disponível em: <<http://www.rogercom.com/ZigBee/ZigBee.htm>>. Acesso em: 12 jul. 2009.
- [16] Disponível em: <<http://www.meshnetics.com/zigbee-faq/>>. Acesso em: 31 ago. 2010
- [17] Zigbee Cluster Library. Zigbee Alliance Inc. 2007.
- [18] GOLDSCHMIDT, IRA. The Development of BACnet. Strategic Planning for Energy and the Environment. Vol. 18, No. 2, pp. 16-24. Association of Energy Engineers, EUA.
- [19] Number of BACnet Vendor IDs Continues to Climb! (3/09) Ithaca, NY. Disponível em: <<http://www.bacnet.org/Oldnews/index.html>>. Acesso em: 21 ago.2010.
- [20] NEWMAN, H. MICHAEL. The Many Advantages of BACnet. Refrigeration Service Engineers Society Journal, 1999.

- [21] Addendum q to ANSI ASHRAE Standard 135-2008. 2009.
- [22] AMERICAN NATIONAL STANDARDS INSTITUTE/AMERICAN SOCIETY OF HEATING, Refrigerating, Air-Conditioning Engineers. Standard 135-2008, BACNet® -- A Data Communication Protocol for Building Automation and Control Network. Atlanta: ASHRAE, 2008.
- [23] Disponível em : <<http://www.bacnet.org/Addenda/index.html>> Acesso em: 21 ago.2010.
- [24] BitCloud User Guide. ATMEL. 2009.
- [25] OLMOS, JOSÉ L. Sistema Híbrido de Climatização Visando Conforto Térmico e Eficiência Energética. 2009. Dissertação de Mestrado em Engenharia Elétrica. Publicação PPGENE.DM-393/09. Departamento de Engenharia Elétrica. Universidade de Brasília. Brasília, DF.
- [26] MAGALDI, HÉLIO R. Alarmes: o livro do instalador. Brasil: Editora Novatec. 2008.
- [27] ZigBit OEM Modules ZDM-A1281-*. MeshNetics 2008 Datasheer.
- [28] AVILA, A.G.; SALOIO, B.H., Instrumentação e Controle de um sistema de ar condicionado híbrido. 2009. Trabalho de Graduação em Engenharia de Controle e Automação, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF.
- [29] FILHO, P.R.M.; DIAS, Y.F.G. Acionamento de Potência para rede de automação wireless. 2008. Trabalho de Graduação em Engenharia de Controle de Automação, Publicação FT.TG-n° 012, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF.
- [30] Disponível em:
<http://www.wunderground.com/history/airport/SBBR/2010/8/14/DailyHistory.html?req_city=NA&req_state=NA&req_statenname=NA> Acesso em: 03 set. 2010.

ANEXO 1 VB SCRIPT ON TIMER

```
Sub OnTimer()
```

```
Dim DIF ' diferenca entre a hora da agenda e a hora atual.
```

```
Dim GRUPO 'Grupo do Atuador
```

```
dim Comando 'Comando a executar 1 - Ligar 2 - Desligar 0 - Nada  
Comando = 0
```

```
GRUPO="ATU"
```

```
*****
```

```
' Rotina ligar ar condicionado *
```

```
*****
```

```
IF VAR.VALUE(GRUPO,"SEND") = 1 THEN
```

```
    'monta a hora do fancoil ser ligado
```

```
    HORA_ON = VAR.VALUE(GRUPO,"H1_L") & VAR.VALUE(GRUPO,"H2_L") & ":" &  
VAR.VALUE(GRUPO,"M1_L") & VAR.VALUE(GRUPO,"M2_L")
```

```
    HORA_ON2 = cdate(HORA_ON)
```

```
    DIF = DateDiff("n", HORA_ON2, TIME)
```

```
    IF DIF >= 0 AND DIF <= 1 THEN
```

```
        IF VAR.VALUE(GRUPO,"BO0") = 0 THEN
```

```
            Comando = 1
```

```
        END IF
```

```
    END IF
```

```
*****
```

```
' Rotina Desligar ar condicionado *
```

```
*****
```

```
    HORA_OFF = VAR.VALUE(GRUPO,"H1_D") & VAR.VALUE(GRUPO,"H2_D") & ":" &  
VAR.VALUE(GRUPO,"M1_D") & VAR.VALUE(GRUPO,"M2_D")
```

```
    HORA_OFF2 = cdate(HORA_OFF)
```

```
    DIF = DateDiff("n", HORA_OFF2, TIME)
```

```
    IF DIF >= 0 AND DIF <= 1 THEN
```

```
        IF VAR.VALUE(GRUPO,"BO0") = 1 THEN
```

```
            Comando = 2
```

```
            Call Command(GRUPO,"BO0_D",6,,0,,false,false)
```

```
        END IF
```

```
    END IF
```

```

*****
' Rotina Desligar ar condicionado *
' Janelas Abertas *
*****_*****

IF VAR.VALUE(GRUPO,"JAN") >= 1 THEN
    VAR.VALUE(GRUPO,"TIMER") = VAR.VALUE(GRUPO,"TIMER")+1
ELSE
    IF VAR.VALUE(PPA,"BIO") = 1 THEN
        VAR.VALUE(GRUPO,"TIMER") = VAR.VALUE(GRUPO,"TIMER")+1
    ELSE
        VAR.VALUE(GRUPO,"TIMER") = 0
    END IF
END IF

IF VAR.VALUE(GRUPO,"TIMER") >= 20 THEN
    Comando = 2
END IF

IF Comando = 1 THEN
    Call Command(GRUPO,"BOO_L",6,,1,,false,false)
    VAR.VALUE(GRUPO,"SEND") = 0
ELSE
    IF Comando = 2 THEN
        Call Command(GRUPO,"BOO_D",6,,,0,,false,false)
        VAR.VALUE(GRUPO,"SEND") = 0
    END IF
END IF

End sub
ELSE
    VAR.VALUE(GRUPO,"SEND") = 1
END IF

```

ANEXO 2 VB SCRIPT ON VALUE CHANGE

```
Sub OnValueChanged()  
    IF VAR.VALUE = 1 THEN  
        VAR.VALUE("ATU","JAN") = VAR.VALUE("ATU","JAN")+1  
    ELSE  
        VAR.VALUE("ATU","JAN") = VAR.VALUE("ATU","JAN")-1  
    END IF  
End sub
```

ANEXO 3 VB SCRIPT ON MOUSE CLICK

```
Sub OnMouseClicked()  
  
' Ajuste do relógio da agenda do ar condicionado  
  
Input = InputBox("Ajuste o Valor: ")  
  
if ISNUMERIC(Input) THEN ' não deixa usar letras  
if input = "" THEN EXIT SUB ' evita mensagem de erro ao cancelar  
    if len(input) > 1 then ' não deixa usar mais de um dígito  
        MsgBox "Somente um dígito"  
        EXIT SUB  
    else  
        VAR.VALUE = Input  
    end if  
else  
  
    MsgBox "Valor Inválido"  
    exit sub  
  
END IF  
End sub
```

ANEXO 4 LOG

[TP 11] 08/31 - 11:37:38,359 SND 00 ->55 FF 05 80 01 00 0F B3 01 04 02 03 96 0E 0C 01 00 00 00 1E 09 55 1F 2C E9 FF
[TP 12] 08/31 - 11:37:38,390 REC 00 ->55
[TP 12] 08/31 - 11:37:38,437 REC 00 ->FF 07 01 80 00 00 30
[TP 11] 08/31 - 11:37:38,468 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:38,484 REC 00 ->55 FF 06 01 80 00 12 B8 01 00
[TP 12] 08/31 - 11:37:38,531 REC 00 ->30 96 0E 0C 01 00 00 00 1E 29 55 4E 91 01 4F 1F 0B 7A 55 FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:38,562 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:38,593 REC 00 ->55 FF 00 01 00
[TP 12] 08/31 - 11:37:38,640 REC 00 ->00 00 73
[TP 11] 08/31 - 11:37:38,671 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:38,687 REC 00 ->55
[TP 12] 08/31 - 11:37:38,734 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:38,781 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:38,796 REC 00 ->55
[TP 12] 08/31 - 11:37:38,843 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:38,890 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:38,906 REC 00 ->55
[TP 12] 08/31 - 11:37:38,953 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:39,000 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:39,015 REC 00 ->55
[TP 12] 08/31 - 11:37:39,062 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:39,109 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:39,125 REC 00 ->55
[TP 12] 08/31 - 11:37:39,171 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:39,218 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:39,234 REC 00 ->55
[TP 12] 08/31 - 11:37:39,281 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:39,328 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:39,328 REC 00 ->55
[TP 12] 08/31 - 11:37:39,390 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:39,437 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:39,437 REC 00 ->55
[TP 12] 08/31 - 11:37:39,500 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:39,531 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:39,546 REC 00 ->55
[TP 12] 08/31 - 11:37:39,609 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:39,640 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:39,656 REC 00 ->55
[TP 12] 08/31 - 11:37:39,718 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:39,750 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:39,765 REC 00 ->55
[TP 12] 08/31 - 11:37:39,812 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:39,859 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:39,875 REC 00 ->55
[TP 12] 08/31 - 11:37:39,921 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:39,968 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:39,984 REC 00 ->55
[TP 12] 08/31 - 11:37:40,031 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:40,078 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:40,093 REC 00 ->55
[TP 12] 08/31 - 11:37:40,140 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:40,187 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:40,203 REC 00 ->55
[TP 12] 08/31 - 11:37:40,250 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:40,296 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:40,312 REC 00 ->55
[TP 12] 08/31 - 11:37:40,359 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:40,406 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:40,406 REC 00 ->55
[TP 12] 08/31 - 11:37:40,468 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:40,515 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:40,515 REC 00 ->55
[TP 12] 08/31 - 11:37:40,578 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:40,609 SND 00 ->55 FF 05 7F 01 00 33 A8 01 04 02 03 EE 0E 0C 00 C0 00 00 1E 09 55 1F 0C 00 C0 00 01 1E 09 55 1F 0C 00 C0 00 02 1E 09 55 1F 0C 00 C0 00 03 1E 09 55 1F 0C 00 80 00 03 1E 09 55 1F 81 C6 FF
[TP 12] 08/31 - 11:37:40,687 REC 00 ->55
[TP 12] 08/31 - 11:37:40,734 REC 00 ->FF 07 01 7F 00 00 21
[TP 11] 08/31 - 11:37:40,781 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:40,796 REC 00 ->55
[TP 12] 08/31 - 11:37:40,843 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:40,890 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:40,906 REC 00 ->55
[TP 12] 08/31 - 11:37:40,953 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:41,000 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:41,015 REC 00 ->55
[TP 12] 08/31 - 11:37:41,062 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:41,109 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:41,109 REC 00 ->55
[TP 12] 08/31 - 11:37:41,171 REC 00 ->FF 00 01 00 00 00 73

[TP 11] 08/31 - 11:37:41,203 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:41,218 REC 00 ->55
[TP 12] 08/31 - 11:37:41,281 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:41,312 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:41,328 REC 00 ->55
[TP 12] 08/31 - 11:37:41,390 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:41,421 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:41,437 REC 00 ->55
[TP 12] 08/31 - 11:37:41,484 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:41,531 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:41,546 REC 00 ->55
[TP 12] 08/31 - 11:37:41,593 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:41,640 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:41,656 REC 00 ->55
[TP 12] 08/31 - 11:37:41,703 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:41,750 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:41,765 REC 00 ->55
[TP 12] 08/31 - 11:37:41,812 REC 00 ->FF 06 01 7F 00 49 61 01 00 30 EE 0E 0C 00 C0 00 00 1E 29 55 4E 91 01 4F F 0C 00 C0 00 01 1E 29 55 4E 91 00 4F 1F 0C 00 C0 00 02 1E 29 55 4E 91
[TP 12] 08/31 - 11:37:41,859 REC 00 ->00 4F 1F 0C 00 C0 00 03 1E 29 55 4E 91 00 4F 1F 0C 00 80 00 03 1E 29 55 E 44 3F 80 00 00 4F 1F A9 AC 55 FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:41,906 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:41,921 REC 00 ->55
[TP 12] 08/31 - 11:37:41,984 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:42,015 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:42,031 REC 00 ->55
[TP 12] 08/31 - 11:37:42,078 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:42,125 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:42,140 REC 00 ->55
[TP 12] 08/31 - 11:37:42,187 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:42,234 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:42,250 REC 00 ->55
[TP 12] 08/31 - 11:37:42,296 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:42,343 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:42,359 REC 00 ->55
[TP 12] 08/31 - 11:37:42,406 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:42,453 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:42,468 REC 00 ->55
[TP 12] 08/31 - 11:37:42,515 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:42,562 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:42,578 REC 00 ->55
[TP 12] 08/31 - 11:37:42,625 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:42,671 SND 00 ->55 FF 05 7F 01 00 33 A8 01 04 02 03 EF 0E 0C 00 C0 00 00 1E 09 55 1F 0C 00 C0 00 01 1E 09 55 1F 0C 00 C0 00 02 1E 09 55 1F 0C 00 C0 00 03 1E 09 55 1F 0C 00 80 00 03 1E 09 55 1F 3E C0 FF
[TP 12] 08/31 - 11:37:42,734 REC 00 ->55
[TP 12] 08/31 - 11:37:42,781 REC 00 ->FF 07 01 7F 00 00 21
[TP 11] 08/31 - 11:37:42,828 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:42,843 REC 00 ->55
[TP 12] 08/31 - 11:37:42,890 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:42,937 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:42,953 REC 00 ->55
[TP 12] 08/31 - 11:37:43,000 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:43,046 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:43,062 REC 00 ->55
[TP 12] 08/31 - 11:37:43,109 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:43,156 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:43,171 REC 00 ->55
[TP 12] 08/31 - 11:37:43,218 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:43,265 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:43,281 REC 00 ->55
[TP 12] 08/31 - 11:37:43,328 REC 00 ->FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:43,375 SND 00 ->55 FF 05 80 01 00 0F B3 01 04 02 03 97 0E 0C 01 00 00 00 1E 09 55 1F BD BC FF
[TP 12] 08/31 - 11:37:43,406 REC 00 ->55
[TP 12] 08/31 - 11:37:43,453 REC 00 ->FF 07 01 80 00 00 30
[TP 11] 08/31 - 11:37:43,484 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:43,500 REC 00 ->55 FF 06 01 80 00 12 B8 01
[TP 12] 08/31 - 11:37:43,546 REC 00 ->00 30 97 0E 0C 01 00 00 00 1E 29 55 4E 91 01 4F 1F 21 32 55 FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:43,593 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:43,609 REC 00 ->55 FF 00 01 00
[TP 12] 08/31 - 11:37:43,656 REC 00 ->00 00 73
[TP 11] 08/31 - 11:37:43,687 SND 00 ->55 FF 00 00 01 00 00 62 FF
[TP 12] 08/31 - 11:37:43,703 REC 00 ->55
[TP 12] 08/31 - 11:37:43,765 REC 00 ->FF 06 01 7F 00 49 61 01 00 30 EF 0E 0C 00 C0 00 00 1E 29 55 4E 91 01 4F 1F 0C 00 C0 00 01 1E 29 55 4E 91 00 4F 1F 0C 00 C0 00 02 1E 29 55 4E 91
[TP 12] 08/31 - 11:37:43,812 REC 00 ->00 4F 1F 0C 00 C0 00 03 1E 29 55 4E 91 00 4F 1F 0C 00 80 00 03 1E 29 55 4E 44 3F 80 00 00 4F 1F 8B 34 55 FF 00 01 00 00 00 73
[TP 11] 08/31 - 11:37:43,859 SND 00 ->55 FF 00 00 01 00 00 62 FF