

## **TRABALHO DE GRADUAÇÃO 2**

# **ACIONAMENTO DE POTÊNCIA PARA REDE DE AUTOMAÇÃO *WIRELESS***

**Pedro Ramos Mateus Filho  
Yuri Ferreira Gomes Dias**

**Brasília, Julho de 2008.**

**UNIVERSIDADE DE BRASÍLIA**

**FACULDADE DE TECNOLOGIA  
CURSO DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E  
AUTOMAÇÃO**

TRABALHO DE GRADUAÇÃO 2

**ACIONAMENTO DE POTÊNCIA PARA REDE DE  
AUTOMAÇÃO PREDIAL *WIRELESS***

**Pedro Ramos Mateus Filho  
Yuri Ferreira Gomes Dias**

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

**Banca Examinadora**

Prof. Adolfo Bauchspiess, UnB/ ENE (Orientador)

---

Prof. Geovany Araújo Borges, UnB/ ENE

---

Prof. Ricardo Zelenovsky, UnB/ ENE

---

Brasília, Julho de 2008.

## FICHA CATALOGRÁFICA

FILHO, PEDRO RAMOS MATEUS & DIAS, YURI FERREIRA GOMES Acionamento de potência para rede de automação predial <i>Wireless</i> , [Distrito Federal] 2008. xii, 67p., 297 mm (FT/UnB, Engenheiro, Controle e Automação, 2008). Trabalho de Graduação – Universidade de Brasília. Faculdade de Tecnologia.	
1. Automação	2. <i>Wireless</i>
3. Controle	4. Acionamento
I. Mecatrônica/FT/UnB	II. Título (série)

## REFERÊNCIA BIBLIOGRÁFICA

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

## CESSÃO DE DIREITOS

AUTORES: Pedro Ramos Mateus Filho e Yuri Ferreira Gomes Dias.

TÍTULO DO TRABALHO DE GRADUAÇÃO: Acionamento de potência para rede de automação *wireless*.

GRAU: Engenheiro de Controle e Automação

ANO: 2008

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.

---

Pedro Ramos Mateus Filho

SHCES 1409 Bloco J Apto 104 – Cruzeiro.  
70658-500 Brasília – DF – Brasil.

---

Yuri Ferreira Gomes Dias

SQS 404 Bloco G Apartamento 203.  
70238-070 Brasília – DF – Brasil.

**Dedicatória(s)**

*À Deus, aos meus irmãos, Fábio e Nayane, e aos meus pais, Giovanne e Ivone, meus mentores e heróis.*

*Yuri Ferreira Gomes Dias*

*Aos meus pais Pedro e Tianinha, minha noiva Fabiana, minha filha Julia e ao professor Adolfo.*

*Pedro Ramos Mateus Filho*

## **Agradecimentos**

*Aos meus pais Pedro e Tianinha, por todo carinho e apoio que me deram para que fosse possível alcançar este objetivo.*

*À minha noiva e companheira Fabiana, por compreender as horas dedicadas ao trabalho e pela força nos momentos difíceis.*

*À minha linda filha Julia, que me trouxe uma grande motivação no decorrer do curso.*

*Aos meus irmãos, amigos e familiares pelo incentivo.*

*Ao meu companheiro de projeto Yuri e ao professor Adolfo pelos ensinamentos e paciência no decorrer do trabalho.*

*Pedro Ramos Mateus Filho*

*Em primeiro lugar aos meus pais que me ensinaram a lidar com as dificuldades e sempre me dedicar cada vez mais para realizar os meus objetivos, é para eles e por eles que realizo os meus planos de vida.*

*Agradeço aos meus irmãos, por conseguirem superar as minhas loucuras, irritações e insensibilidade para com eles. Obrigado por toda a ajuda.*

*Aos meus familiares, que me incentivaram desde o começo com carinho e assistência imediata a qualquer problema que tivesse principalmente ao meu primo Jairo Junior, minha prima e madrinha Aline e minha tia e madrinha Eliva. E à criança que me incentivou a dar continuidade nesse projeto de graduação.*

*Gostaria, também, de citar aqueles que estudaram comigo nesses cinco longos anos de graduação e que fizeram a minha vida ser mais feliz. Nunca me esquecerei da ajuda que vocês me deram, desde as engraçadas caronas até o simples conselho de vida.*

*Aos funcionários da FIAT que me ensinou a lutar por um sonho que para os outros se mostra impossível de realizar.*

*Ao pessoal da TV Globo Brasília que me apoiava em horas difíceis e faziam a minha alegria diária nesta etapa final de projeto. Em especial ao Adonias, por sua ajuda em teorias desconhecidas por mim, até então. Ao Bruno por mostrar algumas das dificuldades de um projeto final. Ao Rodrigo Meira, que considero como um irmão mais velho que não tive, quebrando galhos quando passava por dificuldades no projeto e ao Fred por me ajudar a abrir a cabeça, mostrando que o mundo não é tão pequeno como eu pensava que fosse.*

*Agradeço ao professor Adolfo e ao professor Geovany, pessoas que considero e levo como um exemplo de vida. Obrigado por terem nos auxiliado, desde os momentos de dificuldade aos momentos de incerteza do projeto.*

*E, por fim, ao meu grande amigo e companheiro de projeto Pedro, que me ajudou a ser cada vez mais dedicado às atividades acadêmicas e a manter o meu sonho que não pára com a conclusão deste projeto, mas começa com o fim deste curso.*

*Yuri Ferreira Gomes Dias*

---

## RESUMO

O presente trabalho faz parte de uma linha de pesquisas em ambientes inteligentes do GRAV, e se propõe à realização do controle térmico de um ambiente utilizando conceitos de automação predial wireless. Neste sentido, foram projetados módulos sem fio para aquisição e processamento de dados, além de um sistema supervisor, a fim de realizar o controle térmico. A comunicação, sem fio, dos módulos utilizou a tecnologia ZigBee. O microcontrolador utilizado para a concepção dos módulos foi o ATmega8. Para a análise gráfica dos dados de temperatura, o software Matlab® mostrou-se necessário devido à sua robustez na construção de gráfico detalhados. O sistema de controle de temperatura sem fio foi construído a fim de iniciar uma linha de pesquisas de automação predial wireless utilizando a tecnologia ZigBee.

---

## ABSTRACT

This work is a part of a research in ambient intelligence of GRAV, and proposes to perform on thermal control in environment using concepts of wireless building automation. In this sense, there were projected wireless modules to data processing and acquisition, addition supervisory system to realize the thermal control. The module's wireless communication used the ZigBee technology. The microcontroller used to module's creation was ATmega8. To the graphic analyze using Matlab® was necessary to construction detailing graphics. The wireless control system of temperature was constructed to start researches of wireless building automation using ZigBee technology.

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>1</b>
1.1	CONTEXTUALIZAÇÃO	1
1.2	OBJETIVOS DO PROJETO	1
1.3	SISTEMA PROPOSTO	1
1.4	APRESENTAÇÃO DO MANUSCRITO	2
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA</b>	<b>3</b>
2.1	INTRODUÇÃO	3
2.2	SISTEMAS DE AUTOMAÇÃO PREDIAL	3
2.3	REDE DE SENSORES SEM FIO	4
2.4	ZIGBEE	4
2.4.1	TRANSCEIVER XBee	5
2.4.2	PLACA CON-USBEE	8
2.5	MICROCONTROLADORES	9
2.5.1	ATmega8	10
2.5.2	ATmega128	11
2.6	ALGORITMOS DE CONTROLE	12
2.6.1	CONTROLE LIGA-DESLIGA	12
2.6.2	AÇÃO PROPORCIONAL	14
2.6.3	AÇÃO INTEGRAL	14
2.6.4	AÇÃO DERIVATIVA	15
2.6.5	CONTROLE PROPORCIONAL-INTEGRAL-DERIVATIVO	15
2.7	ACIONAMENTO DE CARGAS AC	16
2.7.1	DISPOSITIVOS TIRISTORES	17
2.7.2	CHAVES ESTÁTICAS AC	19
2.7.3	RELÉ DE ESTADO SÓLIDO (SSR)	20
<b>3</b>	<b>DESENVOLVIMENTO</b>	<b>22</b>
3.1	SALA CONTROLADA	22
3.2	AR CONDICIONADO	22
3.3	MÓDULO DE CONTROLE	23
3.4	MÓDULO DE ACIONAMENTO	24
3.5	MÓDULO SENSOR	25
3.6	PLACA DE CONFIGURAÇÃO	25
3.7	REDE	25
3.8	SOFTWARE SUPERVISÓRIO	26
<b>4</b>	<b>RESULTADOS EXPERIMENTAIS E ANÁLISE</b>	<b>28</b>
4.1	EXPERIMENTOS	28
4.2	ANÁLISE	32
<b>5</b>	<b>CONCLUSÕES E EXPECTATIVAS FUTURAS</b>	<b>34</b>
	<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>	<b>35</b>
	<b>BIBLIOGRAFIA</b>	<b>36</b>
	<b>ANEXOS</b>	<b>37</b>
<b>I.</b>	<b>DIAGRAMAS ESQUEMÁTICOS</b>	<b>38</b>
<b>II.</b>	<b>SOFTWARES</b>	<b>38</b>
<b>III.</b>	<b>DESCRIÇÃO DO CONTEÚDO DO CD</b>	<b>55</b>

# LISTA DE FIGURAS

1.1	Sistema Proposto.....	2
2.1	Aspecto funcional de um sistema de automação .....	3
2.2	Sistema de automação ZigBee .....	5
2.3	Foto do módulo XBee .....	5
2.4	Modos de operação do módulo XBee .....	7
2.5	Estrutura de comando enviado ao XBee. O comando exemplo altera o endereço destino do módulo para 1F .....	7
2.6	Estrutura de um Frame API.....	8
2.7	Tipos de antenas para o XBee .....	8
2.8	Placa CON_USBBEE.....	9
2.9	ATmega8 .....	11
2.10	ATmega128 .....	12
2.11	Histerese em um controle on-off.....	13
2.12	Controle on-off.....	13
2.13	Controle Proporcional .....	14
2.14	Ação Integral .....	15
2.15	PID - $K_p=4$ ; $T_i=1.5$ ; $T_d=0.1$ (tracejado), $T_d=0.4$ (pontilhado), $T_d=2$ (contínuo) ....	16
2.16	SCR - a) estrutura PNP; b) símbolo elétrico.....	17
2.17	Curva característica de um SCR.....	18
2.18	O triac estrutura, símbolo elétrico e circuito SCR equivalente .....	18
2.19	Curva característica de um triac .....	19
2.20	Chave estática monofásica AC com dois SCR em ligação antiparalela .....	19
2.21	Chave estática monofásica AC monofásica .....	20
2.22	Relé de estado sólido .....	20
3.1	Sala Controlada.....	22
3.2	Foto do nó atuador com ATmega8 .....	23
3.3	Circuito impresso do circuito de referência para o ATMEGA128 .....	24
3.4	Foto do circuito de acionamento .....	24
3.5	Foto do nó sensor .....	25
3.6	Foto da placa de configuração .....	25
3.7	Rede configurada.....	26
3.8	Fluxo de funcionamento do software supervisorio .....	27
4.1	Experimento 1.....	28



4.2	Experimento 2.....	29
4.3	Experimento 3.....	30
4.4	Experimento 4.....	31
4.5	Experimento 5.....	32
I.1	Circuito do módulo de controle com ATmega8 .....	39
I.2	Circuito do módulo sensor .....	39
I.3	Circuito da placa de configuração.....	40
I.4	Circuito do módulo de acionamento .....	40
I.5	Circuito de referência para o ATmega128 .....	41

# LISTA DE TABELAS

2.1	Pinos do XBee .....	6
2.2	Comparação da qualidade do link para antena <i>whip</i> e <i>chip</i> .....	8
3.1	Configuração dos módulos da rede.....	26

# LISTA DE SÍMBOLOS

## Símbolos Latinos

G	Giga	$10^9$
m	mili	$10^{-3}$
M	Mega	$10^3$

## Símbolos Gregos

$\mu$	micro	$10^{-6}$
-------	-------	-----------

## Subscritos

<i>d</i>	derivativo
<i>i</i>	integral
<i>máx</i>	máximo
<i>mín.</i>	mínimo
<i>p</i>	proporcional

## Siglas

AC	<i>Alternate Current</i>
ADC	<i>Analog to Digital Converter</i>
API	<i>Application Programming Interface</i>
bps	bits por segundo
CI	Circuito Integrado
CISC	<i>Complex Instruction Set Computer</i>
CMOS	<i>Complementary Metal Oxide Semiconductor</i>
DIP	<i>Dual In-line Package</i>
EEPROM	Electrically-Erasable Programmable Read-Only Memory
FINEP	Financiadora de Estudos e Projetos
FT	Faculdade de Tecnologia
Hz	Hertz
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
JTAG	<i>Joint Test Action Group</i>
LARA	Laboratório de Robótica e Automação
LAVSI	Laboratório de Automação, Visão e Sistemas Inteligentes.

PC	<i>Personal Computer</i>
PI	Proporcional Integral
PID	Proporcional Integral Derivativo
PWM	<i>Pulse Width Modulation</i>
RISC	<i>Reduced Instruction Set Computer</i>
SCADA	<i>Supervisory Control and Data Acquisition</i>
SCR	<i>Silicon Controlled Rectifier</i>
SISC	<i>Specific Instruction Set Computer</i>
SMD	<i>Surface Mounting Device</i>
SRAM	<i>Static Random Access Memory</i>
SPI	<i>Serial Peripheral Interface</i>
TQFP	<i>Thin Quad Flat Pack</i>
TTL	<i>Transistor-Transistor Logic</i>
V	Volts
UART	<i>Universal Asynchronous Receiver Transmitter</i>
UnB	Universidade de Brasília
USART	<i>Universal Synchronous Asynchronous Receiver Transmitter</i>
W	Watt
ZigBee	Padrão IEEE 802.15.4 de comunicação sem fio

# 1 INTRODUÇÃO

## 1.1 CONTEXTUALIZAÇÃO

O conceito “*Ambient Intelligence*” é um novo paradigma que leva a uma densa rede de sensores e atuadores, que são controlados por um supervisor e que podem ou não executar tarefas locais, além de fornecer serviços ao usuário de forma praticamente invisível. A conexão de todos estes dispositivos pode ser feita por fio, como fiação telefônica (*Phoneline Networks*) ou via instalações elétricas (*Powerline Networks*), ou *wireless*. Normalmente utilizam-se nós de sensoriamento e atuação *wireless* que oferecem uma grande flexibilidade no desenvolvimento, em particular no retrofitting de prédios já em utilização.

A automação predial vem crescendo em importância e em particular redes sem fio ZigBee, que foram desenvolvidas especificamente para este fim, têm sido empregadas. O ZigBee é um padrão alternativo para redes de comunicação que não necessitam de muito controle e é projetado para oferecer flexibilidade quanto aos dispositivos que pode controlar.

Este projeto propõe a construção de um sistema experimental de automação *wireless* no LAVSI (Laboratório de Automação, Visão e Sistemas Inteligentes) e no LARA (Laboratório de Robótica e Automação). Este sistema deverá prover o conforto térmico aos usuários do laboratório, através da utilização de uma rede de sensores de temperatura, umidade e velocidade do vento e atuadores para acionamento dos aparelhos de ar condicionado.

## 1.2 OBJETIVOS DO PROJETO

O enfoque deste trabalho de graduação é o desenvolvimento do sistema de controle dos aparelhos de ar condicionado, o que inclui a implementação do controlador e o projeto do circuito de acionamento.

Foi desenvolvida toda a infra-estrutura necessária para a realização do controle de temperatura, como o projeto e construção dos nós sensores e atuadores e o desenvolvimento do software supervisor que coordena a comunicação na rede.

## 1.3 SISTEMA PROPOSTO

O trabalho propõe um sistema de controle de ar condicionado utilizando o padrão IEEE 802.15.4. Consiste em um módulo de acionamento, um módulo de controle, um supervisor e módulos sensores conforme a figura 1.1.

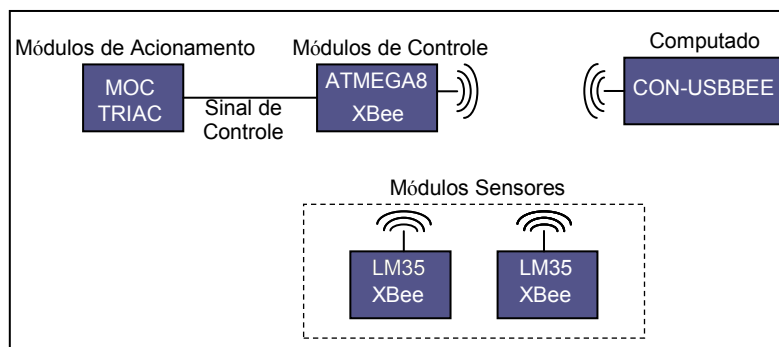


Figura 1.1 – Sistema Proposto

O módulo de controle tem a função de controlar a placa de acionamento dos aparelhos de ar condicionado, o módulo de acionamento aciona o aparelho, fazendo o chaveamento para o controle da temperatura, o módulo sensor é responsável pela aquisição da grandeza do experimento, no caso, a temperatura, e o supervisor coordena o fluxo de dados entre os módulos sensores e os módulos de controle, além da execução da rotina de controle, geração de sinais e transmissão de dados aos módulos sensores e de controle. Os dados são armazenados em um arquivo de texto, com a reprodução gráfica via Matlab®. A tecnologia ZigBee foi utilizada devido à sua grande facilidade de manipulação e seu baixo consumo de energia.

#### 1.4 APRESENTAÇÃO DO MANUSCRITO

O trabalho está dividido em 5 capítulos. No capítulo 2 busca-se fazer um apanhado de assuntos pertinentes à automação predial wireless e à fundamentação teórica necessária ao entendimento do desenvolvimento do trabalho realizado.

Dando continuidade, o capítulo 3 mostra o desenvolvimento do trabalho, ilustrando a parte de *hardware* e *software* realizados durante os experimentos. O Capítulo 4 apresenta uma análise dos dados obtidos no processo de controle térmico do ambiente.

Já o capítulo 5 busca chegar a uma conclusão sobre o método e automação predial utilizado. Determinando assim, a eficiência e o grau de satisfação do método. O capítulo 5 contempla, também, idéias a fim de serem implementadas futuramente.

## 2 REVISÃO BIBLIOGRÁFICA

### 2.1 INTRODUÇÃO

Este capítulo vislumbra os principais conceitos aplicados à automação predial e a forma de transmissão de dados por sensores dispostos em módulos sem fio. Noções sobre sistemas de automação predial e as redes de sensores sem fio serão abordados.

### 2.2 SISTEMAS DE AUTOMAÇÃO PREDIAL

De acordo com [1], sistemas de automação predial são usados tanto para alterar o clima interno dos edifícios como para reduzir despesas operacionais. Originalmente, consistia de aquecimento, ventilação e condicionamento de ar. Depois, o controle de iluminação, segurança, transporte e supervisão foram agregados a esses sistemas visando à redução dos custos em geral.

Existem, segundo [2], dois padrões abertos de comunicação para automação predial, um deles conhecido como BACnet que foi aprovado pela sociedade americana de Engenheiros de aquecimento, refrigeração e condicionamento de ar e o outro chamado de LonWorks, desenvolvido pela Echelon Corp. Por agora, os esforços para a realização da automação nos prédios estão concentrados no corte de custos, baseados na racionalização das operações dos edifícios, tais como ar condicionado e iluminação, ilustrados na figura 2.1.

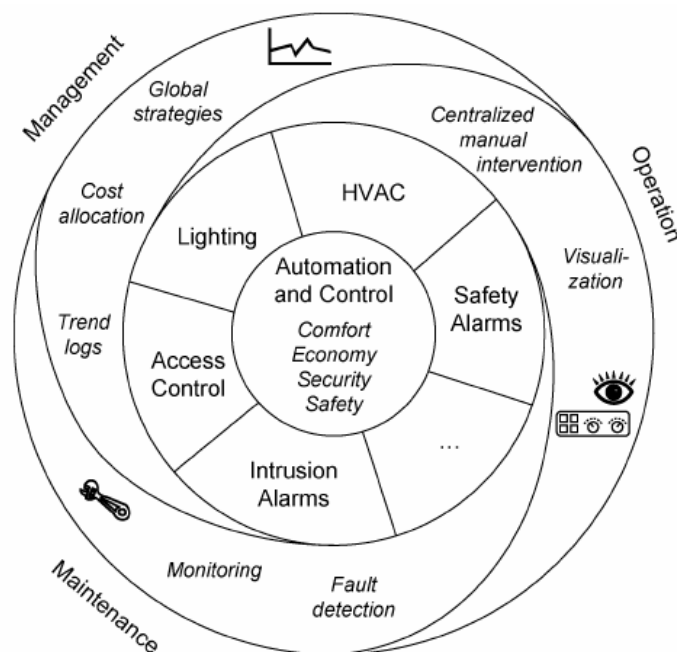


Figura 2.1 – Aspecto funcional de um sistema de automação [2]

Para ser eficaz, qualquer sistema de automação deve permitir que todos os sistemas mecânicos e elétricos trabalhem a partir de um único ponto de controle. Isso é um grande problema, pois cada sistema é feito com um *hardware* e um *software* proprietário, dificultando a integração de todos em uma unidade de controle.

## 2.3 REDE DE SENSORES SEM FIO

Sensores têm um papel fundamental em sistemas embarcados, de aparelhos autônomos ou “inteligentes”, até o controle e monitoramento de máquinas industriais. Sensores eletrônicos respondem a estímulos, como pressão, temperatura, umidade ou movimento e geram um sinal elétrico mensurável, transportando informações sobre o objeto em estudo. De acordo com [3], a convergência recente de sensoriamento e tecnologias de comunicação sem fio de baixa potência permitiram o advento de redes de sensores sem fio, onde é possível definir que a arquitetura básica de hardware de um nó sensor é composta por um microcontrolador e um transceptor sem fio. E suas principais características são:

- Ter dimensões físicas reduzidas;
- Capacidade de operação durante um longo tempo;
- Dispor de pouca quantidade de energia

[1] se refere às redes de sensores sem fio como um pequeno nó sensor que “sente” o ambiente, executa cálculos computacionais e comunica com outros nós usando um transmissor acoplado em seu sistema. Tipicamente, a transmissão dos dados medidos até a estação base é feita ponto a ponto. Os nós sensores são normalmente alimentados por baterias, dando maior portabilidade a estes elementos da rede.

## 2.4 ZIGBEE

De acordo com [4], é uma tecnologia de redes sem fio baseada no padrão IEEE 802.15.4 que ainda se encontra em fase de desenvolvimento e pretende realizar a interligação de pequenas unidades de comunicação de dados em áreas muito limitadas. Foi idealizada para interligar pequenas unidades telemétricas recorrendo a sinais de rádio frequência não licenciados. A figura 2.2 mostra uma visão geral do sistema de automação com ZigBee.



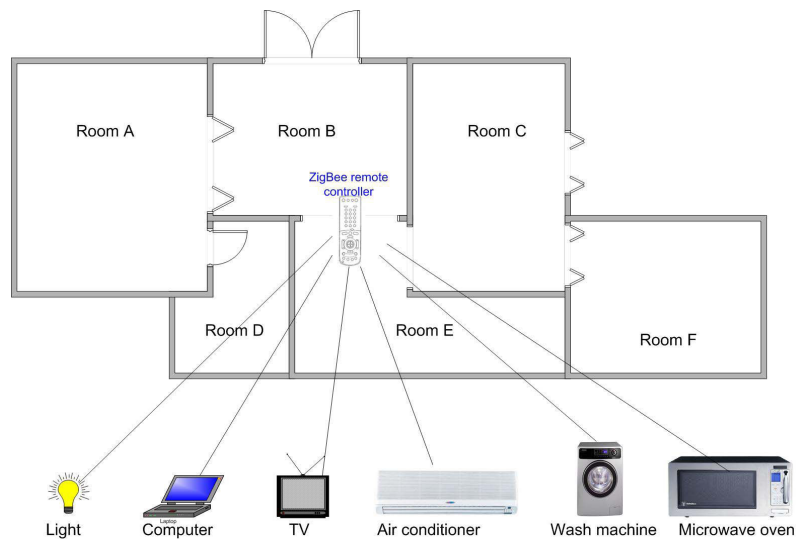


Figura 2.2 - Sistema de automação ZigBee [4]

ZigBee é comparável às tecnologias de rede Wi-Fi e Bluetooth, sendo que a tecnologia ZigBee funciona com um menor consumo de energia e seu alcance é reduzido, cerca de 30 metros *indoor*. As comunicações entre módulos ZigBee podem ser repetidas sucessivamente através de diversos módulos até o destino final do dado. Configurando os pontos da rede ZigBee como retransmissores de informação, formando uma malha, pode-se realizar uma extensão doméstica ou industrial sem necessidades de ligação elétrica entre os módulos.

Essa tecnologia atua com uma taxa de transmissão de até 250Kbps e uma rede pode atingir um alcance de no máximo 1500 metros funcionando em ambiente aberto.

#### 2.4.1 TRANSCIEVER XBee

O módulo Xbee, figura 2.3, é fabricado pela Digi (antiga MaxStream) e foi projetado para atender o padrão IEEE 802.15.4, operando na frequência de 2,4GHz.



Figura 2.3 – Foto do módulo XBee [5]

- Faixa de transmissão em área interna: 30 metros;
- Faixa de transmissão em área externa: 100 metros;
- Potência de transmissão: 100 mW;

- Sensibilidade de recepção: -92 dBm;
- Taxa de transmissão: 250.000 bps;
- Tensão de alimentação: 3,3 V.

Além de ser um transmissor, o módulo XBee possui várias outras funcionalidades que o posiciona como ótimo dispositivo para soluções de automação são elas:

- 5 entradas de conversão analógica digital de 10 bits;
- 8 pinos de entrada e saída de dados;
- Canais de PWM.

Todas as configurações do módulo são feitas através de uma lista de comandos e parâmetros pré-definidos, que são passados ao módulo via interface serial. A tabela 2.1 mostra a configuração dos pinos do módulo:

Tabela 2.1 – Pinos do XBee [5]

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

Como o módulo XBee possui uma UART, para realizar a comunicação serial com outro dispositivo, como um microcontrolador, basta conectar os pinos DOUT (*Data out*) e DIN (*Data in*) respectivamente aos pinos RX (*Receiver X*) e TX (*Transmitter X*), atentando-se para os níveis de tensão de operação dos dispositivos.

O Xbee possui modos de operação, ilustrados na figura 2.5, são eles: *Idle Mode*, *Transmit Mode*, *Receive Mode*, *Sleep Mode* e *API Mode*. Cada modo de operação possui uma determinada peculiaridade. A seguir, as funções de cada modo de operação.

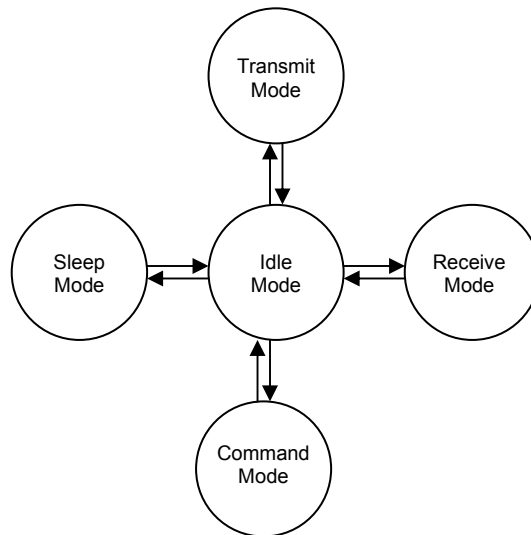


Figura 2.4 Modos de operação do módulo XBee [5]

- **Idle Mode**

Por *default* o módulo fica neste modo, sendo que na ocorrência de algum evento específico ele vai para um dos outros modos.

- **Transmit Mode**

Quando algum dado é recebido no buffer DIN da UART, o módulo entra neste modo e faz a transmissão pela antena, retornando ao modo *idle*.

- **Receiver Mode**

Quando algum dado é recebido através da antena, o módulo entra neste modo e manda os dados para a saída serial DOUT, em seguida retorna ao modo *idle*.

- **Sleep Mode**

O módulo pode ser configurado para entrar no modo *sleep*, no qual o consumo de energia é bem mais baixo do que no modo normal.

- **Command Mode**

É neste modo que o módulo tem todas as suas configurações realizadas. Para entrar no modo de comando deve-se enviar ao módulo uma sequência de caracteres “+++” e aproximadamente um segundo depois, o comando a ser dado com a seguinte estrutura:

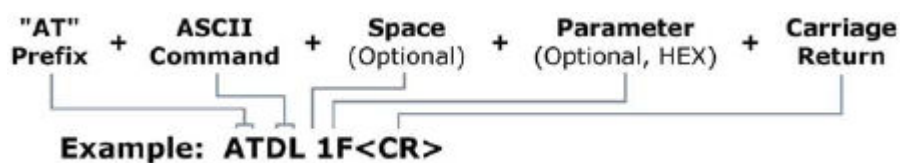


Figura 2.5 Estrutura de comando enviado ao XBee [5]  
O comando exemplo altera o endereço destino do módulo para 1F

A cada operação executada corretamente o módulo XBee retorna um “OK” pela serial.

- **API Operation**

A *API Operation* é uma forma de operação alternativa ao *Idle Mode*. A *Application Programming Interface* define uma estrutura específica cujos comandos, respostas à comandos e mensagens do estado do transceiver vão ser recebidas e transmitidas pelo módulo usando a UART. Quando no modo API, toda informação que entra ou sai do módulo está contida em frames que definem operações ou eventos. Esses frames são compostos, em seqüência, por uma parte que delimita o início do frame, outra que informa o seu tamanho, a parte que se encontra os dados e o seu *checksum* (onde se verifica a integridade do frame). A figura 2.6 mostra a organização dos frames do modo API.



Figura 2.6 Estrutura de um Frame API [5]

O módulo XBee possui conexão para três tipos de antena, *whip antenna*, *chip antenna* e um conector para antena externa. A figura 2.7 exemplifica, fora de escala, os três tipos de antena disponíveis. E a tabela 2.2 compara a qualidade do link para dois tipos de antenas: *chip* e *whip*.



Figura 2.7 Tipos de antenas para o XBee [5]

Tabela 2.2 – Comparação da qualidade do link para antena *whip* e *chip* [5]

Module	Antenna Type	Outdoor Distance (Visual Line-of-Sight)	Indoor Distance (Office Building)	Indoor Distance (Warehouse)
Xbee	<i>Chip</i>	100 m	30 m	-
	<i>Whip</i>	258 m	30 m	32 m

## 2.4.2 PLACA CON-USBBEE

Para a comunicação do XBee com o PC, foi utilizada a placa CON-USBBEE, ilustrada pela figura 2.8. Esta placa tornou-se muito útil por permitir a comunicação com os módulos pela porta USB, disponível em grande parte dos *notebooks* atualmente, ao invés da porta serial, praticamente indisponível nos computadores e *notebooks* atuais.

Os módulos XBee possuem um *firmware* instalado, o qual permite a utilização de algumas funcionalidades dos módulos em detrimento de outras. A placa CON-USBBEE disponibiliza a

interface para a atualização do *firmware*, habilitando, por exemplo, as funções de I/O e conversão analógico-digital.

A alimentação é fornecida pela própria conexão USB, sendo parecida com a conexão de um *pen-drive*. Após instalar o *driver* fornecido pelo fabricante e plugar a placa no computador, o sistema operacional cria uma porta de comunicação virtual, COMx (geralmente COM3 ou COM5), semelhante à uma porta serial padrão RS-232C.

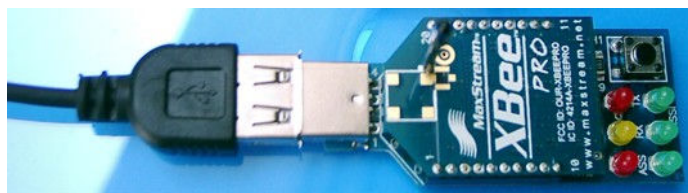


Figura 2.8 Placa CON-USBBEE [6]

Para informações adicionais sobre comandos, configurações de rede, endereçamento e API do módulo XBee consultar as referências [11] e [12] e manual do fabricante.

## 2.5 MICROCONTROLADORES

O termo controlador descreve os dispositivos que controlam um processo ou algum parâmetro do ambiente. Um controlador de temperatura de um ar condicionado pode ligar ou desligar o compressor a fim de regular a temperatura dentro de uma sala. Atualmente muitos controladores diminuíram muito de tamanho, possibilitando a criação de sistemas microprocessados.

Os microcontroladores são circuitos integrados com milhões de transistores que possibilitam realizar as funções de um autêntico controlador. Existem diversos tipos de microcontroladores, dentre eles, os mais conhecidos são: 8051, 8096, a família PIC de microcontroladores, ARM7 e os da família Atmel (com ênfase nos AVR).

Grande parte dos microcontroladores está baseada no conceito CISC, que normalmente tem mais de 100 instruções específicas para a realização de tarefas. Algumas instruções operam em certos endereços ou certos registradores o que dificulta a programação. São centrais de processamento de baixo desempenho, devido a maior parte de o circuito integrado estar relacionada à lógica de controle.

A tecnologia RISC, complementar ao conceito CISC, pertence em grande parte a sistemas dedicados. CI's com esse conceito possuem um conjunto restrito de instruções, permitindo que a unidade de controle seja reduzida e com isso ter um melhor desempenho na realização de tarefas. Geralmente, estes CI's consomem uma quantidade menor de energia.

Devido ao grande investimento na área da microeletrônica, verifica-se o surgimento do conceito SISC, que se baseia na idéia de especialização dos recursos em benefício de outras tarefas,

tais como interrupções e acesso à memória. Possui diversas instruções que trazem facilidades quanto às operações de controle.

Os microcontroladores utilizados neste projeto são os ATmega8 e o ATmega128 da Atmel. Esses controladores foram escolhidos partindo do princípio de fácil acesso e seu baixo custo quando comparados aos outros microcontroladores. O processo de inserção do código programável é facilitado pela confecção e utilização de uma gravadora.

### 2.5.1 ATmega8

Tem uma grande aceitação no mercado, por ser um microcontrolador de 8 bits com baixo custo e que exige pouca energia para o funcionamento. De acordo com [13], pode ser encontrado com encapsulamento DIP de 28 pinos e é fabricado com a tecnologia CMOS. É baseado na arquitetura AVR RISC.

Possui 32 registradores conectados diretamente com a unidade lógica aritmética (ULA). A forma em que os registradores foram projetados para funcionar faz com que o ATmega8 tenha uma velocidade de processamento maior que outros controladores com arquitetura RISC convencional, como 8051. Esse microcontrolador apresenta algumas características a serem citadas:

- 8 KB de memória Flash programável;
- 512 bytes de EEPROM;
- 1 KB de SRAM;
- 23 Portas de Entrada e Saída;
- Contadores/Temporizadores e PWM;
- USART, interface serial programável;
- Canais conversores Analógico/Digital;
- Porta SPI;
- Tensão de operação: 4,5 – 5,5 V.

A memória *flash* utilizada no ATmega8 tem a capacidade de ler enquanto escreve, isto é, faz a leitura dos dados sem precisar esperar os dados serem completamente escritos em sua memória, facilitando diversas aplicações e diminuindo o tempo de execução de um programa.

A programação das memórias pode ser feita utilizando a porta SPI. Segundo [14], este tipo de programação pode ser feita sempre que o *reset* for ativado, fazendo com que a porta SPI aceite a programação das memórias interna sob controle de um conjunto particular de instruções. Durante a programação a porta SPI aceita algumas instruções, são elas:

- Habilitar programação;

- Apagar memórias;
- Ler ou escrever na memória de códigos e na memória de dados;
- Programar *bits* de segurança, denominados *lock bits*;

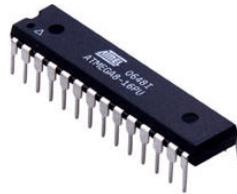


Figura 2.9 – ATmega8 [7]

### 2.5.2 ATmega128

Este microcontrolador possui características parecidas com o ATmega8, porém muito mais denso fisicamente. Essa maior densidade proporciona ao ATmega128 uma quantidade maior de portas e uma capacidade maior de armazenamento de dados.

Encontra-se a venda com encapsulamento SMD, do tipo TQFP e tecnologia CMOS de 8 bits. Tal configuração permite que este microcontrolador seja pequeno. Conforme [13] as características marcantes são:

- 128 KB de memória Flash;
- 4 KB de memória EEPROM;
- 4 KB de memória SRAM;
- 53 portas de Entrada/Saída;
- 32 Registradores;
- Contador de Tempo Real (RTC);
- Contadores/Temporizadores com modo de comparação e PWM;
- 2 USART, interface de programação serial;
- 8 canais de ADC;
- Padrão IEEE 1149.1 ou JTAG Interface de Teste.

O ATmega128 tem a capacidade de ler enquanto escreve em sua memória *flash*, fazendo com que os dados sejam lidos antes mesmo de estar completamente em sua memória, similar ao ATmega8.

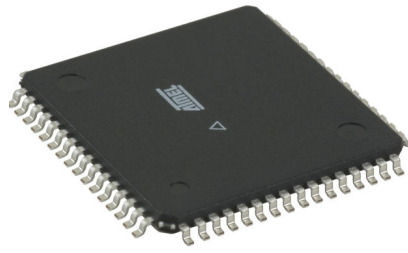


Figura 2.10 - ATmega128 [8]

## 2.6 ALGORITMOS DE CONTROLE

Controladores são dispositivos amplamente utilizados em sistemas industriais. Esses sistemas têm como função comparar sinais da entrada (valor de referência ou desejado) com sinais de saída da planta, fazer a verificação do desvio entre as medições e dispor sinais (sinais de controle) para diminuir ou eliminar o erro associado ao desvio dos valores mensurados inicialmente. Segundo [16], a maneira pela qual o controlador produz o sinal de controle se caracteriza como ação de controle.

Geralmente, o controlador verifica o sinal de erro atuante no sistema, cuja potência é baixa, e o amplifica. Este sinal amplificado é o sinal de controle disposto pelo controlador e enviado ao atuador, o qual [17] se refere como um dispositivo de potência que reduz o sinal de entrada na planta, de acordo com o sinal de controle, de tal modo que a saída se aproxime do sinal referência de entrada.

Alguns controladores podem ser classificados de acordo com o seu tipo de funcionamento no sistema de controle, sendo os de interesse para automação predial são:

- Controlador *On-Off*;
- Ação Proporcional;
- Ação Integral;
- Ação Derivativa;
- Controlador Proporcional - Integral - Derivativo (PID).

### 2.6.1 CONTROLE LIGA-DESLIGA

De acordo com [17], num sistema de controle do tipo *on-off* (Liga-desliga), também chamado de controle de duas posições, o elemento atuante funciona basicamente com dois níveis fixos: liga e desliga.

Neste tipo de controle, o sinal de saída do controlador assume uma determinada posição física, dependendo se o erro atuante, associado ao sistema, for positivo ou negativo. O sinal de realimentação é verificado pelo controlador que faz uma comparação com o seu padrão de referência, caso o valor da realimentação seja superior à referência, o controlador deve atuar de maneira a levar o sinal de saída ao seu valor mínimo. Caso contrário, a atuação deve elevar o sinal de saída para o valor máximo.



[18] menciona a fragilidade desse tipo de dispositivo com relação ao comportamento dinâmico e em regime permanente. Esse tipo de controle não consegue acompanhar de forma adequada o sinal de referência desejado. Porém, o controlador liga-desliga é bastante utilizado em equipamentos como geladeiras, uma vez que o controle de temperatura dentro da geladeira é um processo lento.

Este controlador tem a representação matemática da seguinte forma:

$$u(t) = \begin{cases} V_{\text{máx.}} & \text{para } e(t) > 0 \\ V_{\text{mín.}} & \text{para } e(t) < 0, \text{ onde } V_{\text{mín.}} \text{ geralmente é zero ou } -V_{\text{máx.}} \end{cases} \quad (2.1)$$

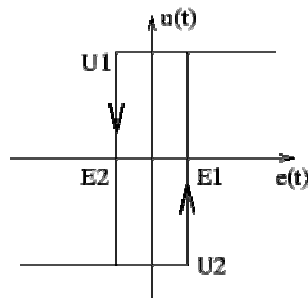


Figura 2.11 - Histerese em um controle Liga-Desliga [10]

Analisando  $u(t)$ , percebe-se uma inconsistência em zero, e na presença de ruídos, ocasionariam chaveamentos indesejados quando o sinal  $e(t)$  for próximo de zero. Para evitar esse problema, pratica-se o controle liga-desliga com histerese, representado pela figura 2.11.

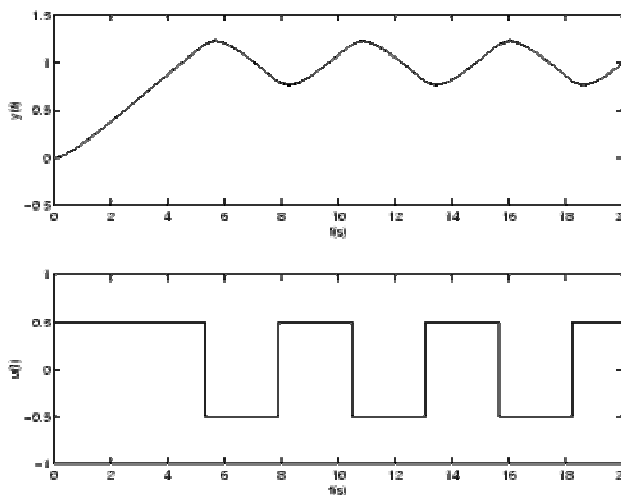


Figura 2.12 - Controle Liga-Desliga [9]

Nota-se que ao colocar ganhos muito altos em controladores do tipo proporcional, esses atuam como controladores do tipo liga-desliga. A figura 2.12 mostra uma curva característica de um liga-desliga, ilustrando o chaveamento de um suposto componente.

### 2.6.2 AÇÃO PROPORCIONAL

Um controlador com ação de controle proporcional é essencialmente um tipo de amplificador com ganho ajustável,  $K_p$ . Isso independe do tipo de mecanismo e do tipo de energia utilizada.

A relação entre a saída do controlador e o sinal de erro atuante é definida segundo a equação 2.2 a seguir:

$$u(t) = K_p e(t) \quad (2.2)$$

onde  $K_p$  é o ganho ajustável ou ganho proporcional do controlador.

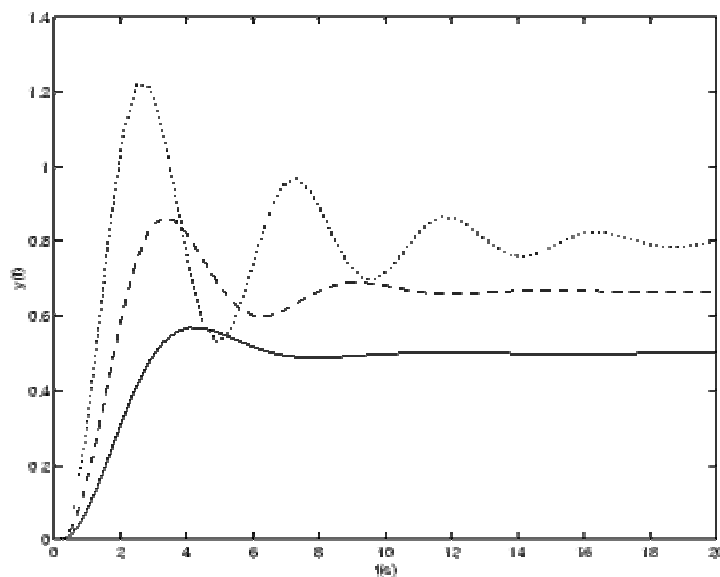


Figura2.13 – Controle Proporcional [9]

### 2.6.3 AÇÃO INTEGRAL

No controlador com ação de controle integral verifica-se que o valor de saída é modificado pela taxa de variação proporcional ao sinal de erro atuante. Ou seja, o controlador com ação integral tenta minimizar ou até eliminar o erro em regime permanente que o controle proporcional não consegue tratar.

A forma de controle integral pode ser caracterizada matematicamente de acordo com a equação 2.3 abaixo:

$$\frac{du(t)}{dt} = K_i e(t) \text{ ou} \quad (2.3)$$

$$u(t) = K_i \int_0^t e(t) dt \quad (2.4)$$

onde  $K_i$  é uma constante ajustável referente à integração.

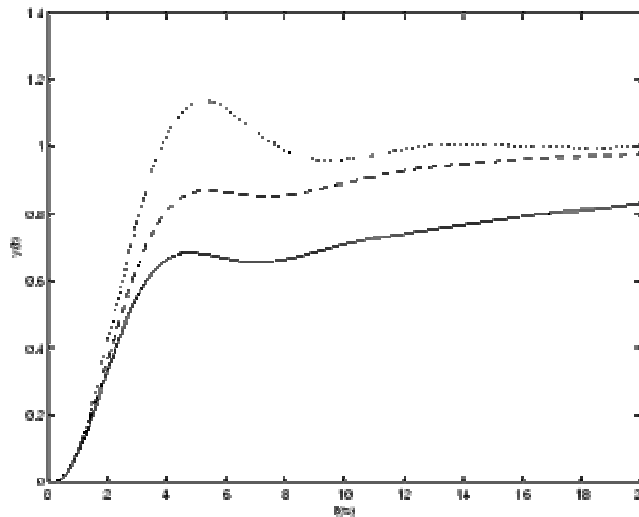


Figura 2.14 - Ação Integral [9]

#### 2.6.4 AÇÃO DERIVATIVA

A idéia de um controle derivativo está entrelaçada ao desejo de rapidez na resposta do sistema. Essa ação de controle é tratada com relação à velocidade de alteração do sinal de erro atuante com relação à referência.

É amplamente utilizada na concepção de controle do tipo PID, cuja idéia é colaborar com um ganho proporcional (amplificação), diminuir o erro intrínseco ao regime permanente e à prevenção de respostas rápidas para uma melhor adequação do sinal de saída em relação ao sinal de referência.

Este tipo de ação pode ser caracterizado matematicamente seguindo a seguinte equação:

$$u(t) = K_d \frac{de(t)}{dt} \quad (2.5)$$

#### 2.6.5 CONTROLE PROPORCIONAL-INTEGRAL-DERIVATIVO

A idéia do controle PID é a de realizar uma combinação do controle proporcional, do controle integral e do controle derivativo. Com essa combinação, este tipo de controle consegue obter o que há de melhor nas três ações de controle supracitados. A equação deste tipo de controlador pode ser verificada pela equação 2.6.

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt} \quad (2.6)$$

onde  $K_p$  é o ganho proporcional,  $T_i$  é o tempo integrativo e  $T_d$  é o tempo derivativo.

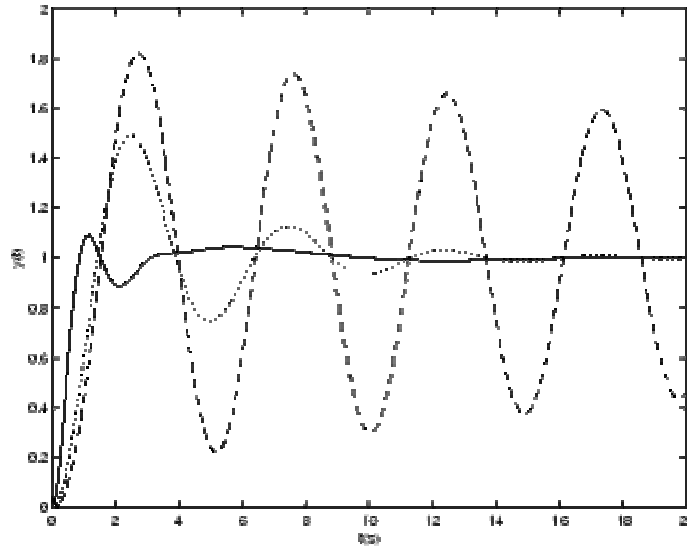


Figura 2.15 - PID -  $K_p=4$ ;  $T_i=1.5$ ;  $T_d=0.1$  (tracejado),  $T_d=0.4$ (pontilhado),  $T_d=2$ (contínuo) [9]

## 2.7 ACIONAMENTO DE CARGAS AC

Um dispositivo muito utilizado para acionamento de cargas de corrente alternada são as chaves estáticas. Elas promovem o chaveamento da potência fornecida para a 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.

De acordo com [19], as características dos dispositivos semicondutores denominados tiristores – que possuem dois estados estáveis (condução e não condução) – permitem que eles possam ser usados como chaves sem contatos. As chaves semicondutoras possuem várias vantagens em relação aos demais dispositivos de chaveamento, como:

- Velocidade de chaveamento muito alta;
- Como não há partes móveis, não há riscos de centelhas;
- Não necessita de manutenção de rotina, pois não possui partes móveis que se desgastam;
- É imune a vibrações e choques;
- É pequena, leve, confiável e de baixo custo;

- É controlada eletronicamente e além de ligar e desligar a carga, pode também ser usada para controlar a potência de 0 ao máximo;
- Isola-se facilmente o circuito de controle do circuito de potência e o controle à distância pode ser feito de maneira simples.

Elas apresentam também algumas desvantagens:

- Apresenta maiores perdas de potência no estado ligado, requerendo, portanto refrigeração;
- A queda de tensão no dispositivo no estado ligado pode não permitir seu uso em algumas aplicações;
- A capacidade de sobrecarga é limitada pela corrente máxima suportada pelo semicondutor.

O chaveamento AC pressupõe o controle bidirecional, sendo que nestas aplicações geralmente são utilizados um dispositivo denominado triac ou dois SCR's (retificadores controlados de silício). O uso de um ou outro é determinado geralmente pela frequência do chaveamento necessário, sendo em mais baixas frequências utiliza-se o triac e para frequências maiores usa-se os dois SCR's.

### 2.7.1 DISPOSITIVOS TIRISTORES

Os tiristores são semicondutores de potência com quatro camadas PNPN e são usados como chaves eletrônicas. Nesta família o mais utilizado é o SCR, presente em aplicações de fontes de alimentação reguladas de potência, *choppers*, inversores, etc.

- **SCR – Retificador Controlado de Silício**

O SCR possui um chaveamento muito rápido, pequeno porte e permite altos valores nominais de corrente e tensão.

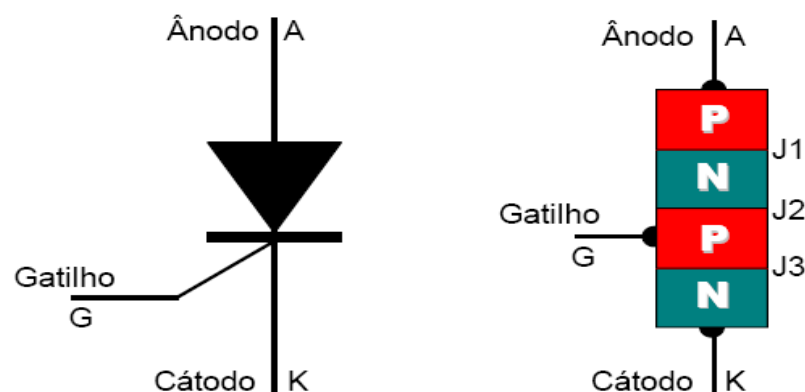


Figura 2.16 SCR - a) estrutura PNPN; b) símbolo elétrico [20]

Quando a tensão no ânodo (A) é positiva em relação ao cátodo (K) o SCR está diretamente polarizado, a partir deste momento uma tensão positiva na porta ou gatilho (G) em relação à K faz com que o SCR entre no estado ligado (condução). Uma vez no estado ligado a porta não tem mais controle sobre o dispositivo, de modo que ele só voltará para o estado desligado (não condução),

quando a tensão de alimentação for retirada ou quando a corrente no ânodo for menor que certo valor denominado corrente de sustentação ou de manutenção.

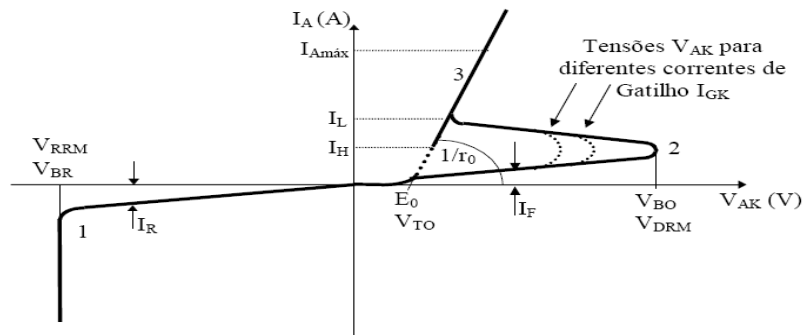


Figura 2.17 Curva característica de um SCR [19]

- **Triac – triodo de corrente alternada**

O triac, representado na figura 2.18a, é um dispositivo semiconductor capaz de conduzir em ambas as direções e pode ser controlado por um sinal na porta, o que o torna útil para controle de potência AC. Também é denominado um SCR bidirecional, sendo equivalente à ligação de dois SCR's em paralelo, de acordo com a figura 2.18c.

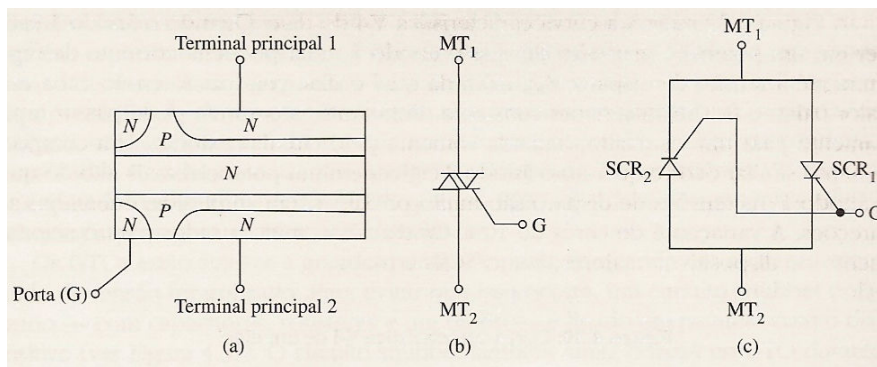


Figura 2.18 O triac estrutura, símbolo elétrico e circuito SCR equivalente [19]

Para qualquer tensão, positiva ou negativa, entre os terminais principais  $MT_1$  e  $MT_2$ , a aplicação de uma corrente entre a porta e  $MT_1$  faz com o dispositivo entre em condução. Assim como o SCR, o triac só retornará ao modo desligado com a tensão de alimentação for nula ou a corrente principal for menor do que a corrente de sustentação.

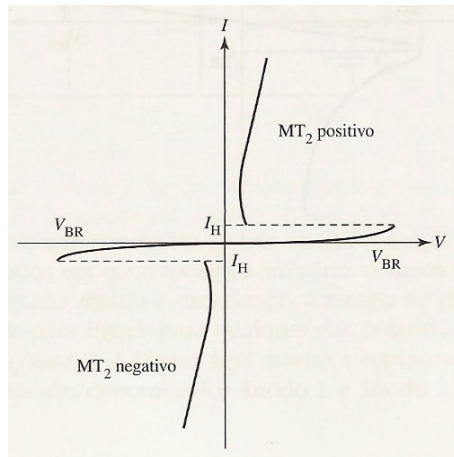


Figura 2.19 Curva característica de um triac [19]

A curva característica da figura 2.19 mostra a corrente através do triac, quando a tensão de ruptura ( $V_{BR}$ ) é aplicada entre os terminais  $MT_1$  e  $MT_2$ . A condução deste dispositivo acontece quando a tensão entre os terminais  $MT_1$  e  $MT_2$  é suficiente para o desenvolvimento de uma corrente interna elevada. O triac é considerado mais econômico que o SCR e fácil de controlar. Em circuitos de alta potência, dois SCRs costumam ser utilizados para realizar a função de um triac, uma vez que os mesmos possuem maior  $dv/dt$  e tempos de desligamento mais baixos. Devido a esta limitação de velocidade, os triacs são utilizados para acionamento com tensão AC de 60 Hz, em aplicações de iluminação, controle de velocidade de motores, aquecimento e em relés de estado sólido.

## 2.7.2 CHAVES ESTÁTICAS AC

Conforme citado anteriormente, a chave estática tem a função de chaveamento que liga ou desliga a carga da alimentação. Ele possui três circuitos principais, o de potência, o de disparo e o de proteção.

- **Ligação antiparalela com SCR**

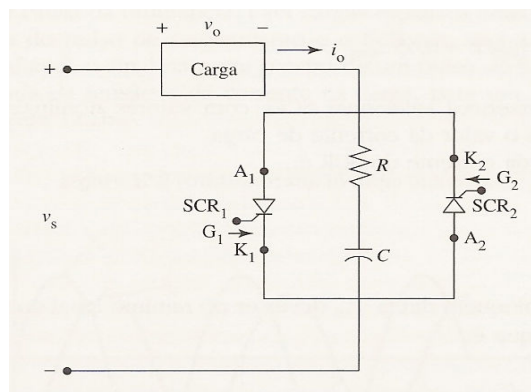


Figura 2.20 Chave estática monofásica AC com dois SCR em ligação antiparalela [19]

Na figura 2.20, para ligar a potência à carga, a chave  $SCR_1$  deve estar ligada no começo do semi-ciclo positivo, de modo análogo, no começo do semi-ciclo negativo a chave  $SCR_2$  deve ir para o estado ligado. Para ligar e manter ligadas as chaves  $SCR$  pulsos de disparo devem ser enviados à porta, pois a cada vez que a tensão cruza o zero o tiristor voltará para o estado desligado. Assim, para desligar a chave basta para de enviar os pulsos.

O circuito RC ligado em paralelo com as chaves é denominado *snubber RC*, ele tem a função de eliminar os transitórios de tensão que venham ocorrer no chaveamento, em especial nas cargas indutivas.

- **Ligação com triac**

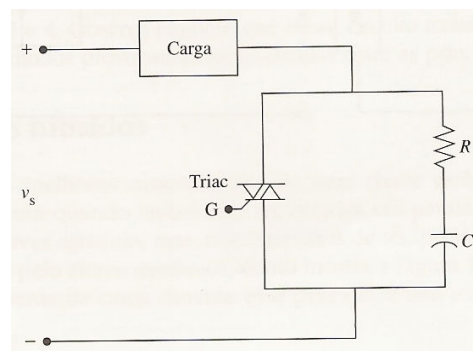


Figura 2.21 Chave estática monofásica AC monofásica [19]

A vantagem desta chave é que ela elimina um circuito de disparo, sendo também capaz de passar para o estado ligado com um pulso positivo ou negativo na porta além de conduzir correntes em ambas às direções.

### 2.7.3 RELÉ DE ESTADO SÓLIDO (SSR)

Por não conterem contatos e partes móveis, as chaves semicondutoras receberam o nome de relés de estado sólido (*solid-state relays – SSR's*). Estas chaves ficaram famosas porque substituíram os relés eletromagnéticos em algumas aplicações. A Figura 2.22 ilustra um relé de estado sólido.

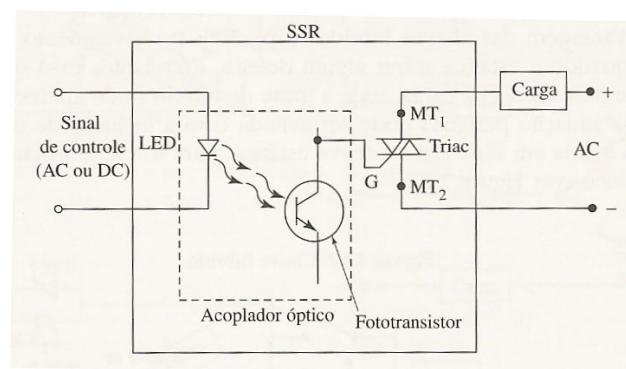


Figura 2.22: Relé de estado sólido [19]



Em aplicações AC o SSR utiliza um triac para ligar a carga à fonte. Eles também possuem, em sua estrutura, dispositivos denominados optoacopladores.

De acordo com [21], o optoacoplador combina um diodo emissor de luz (LED), e um dispositivo optoeletrônico, usualmente um fototransistor, que são montados num único invólucro. Além da função de isolar duas partes do circuito, este dispositivo possui um módulo interno com uma função denominada chaveamento com tensão zero. Isto significa que o triac será passado para o estado ligado no primeiro instante após a fonte de tensão passar pelo eixo do zero, o que evita a geração de faíscas e reduz a interferência eletromagnética.

### 3 DESENVOLVIMENTO

#### 3.1 SALA CONTROLADA

A figura 3.1 ilustra a disposição dos aparelhos de ar-condicionado e também a sua localização dentro do laboratório.

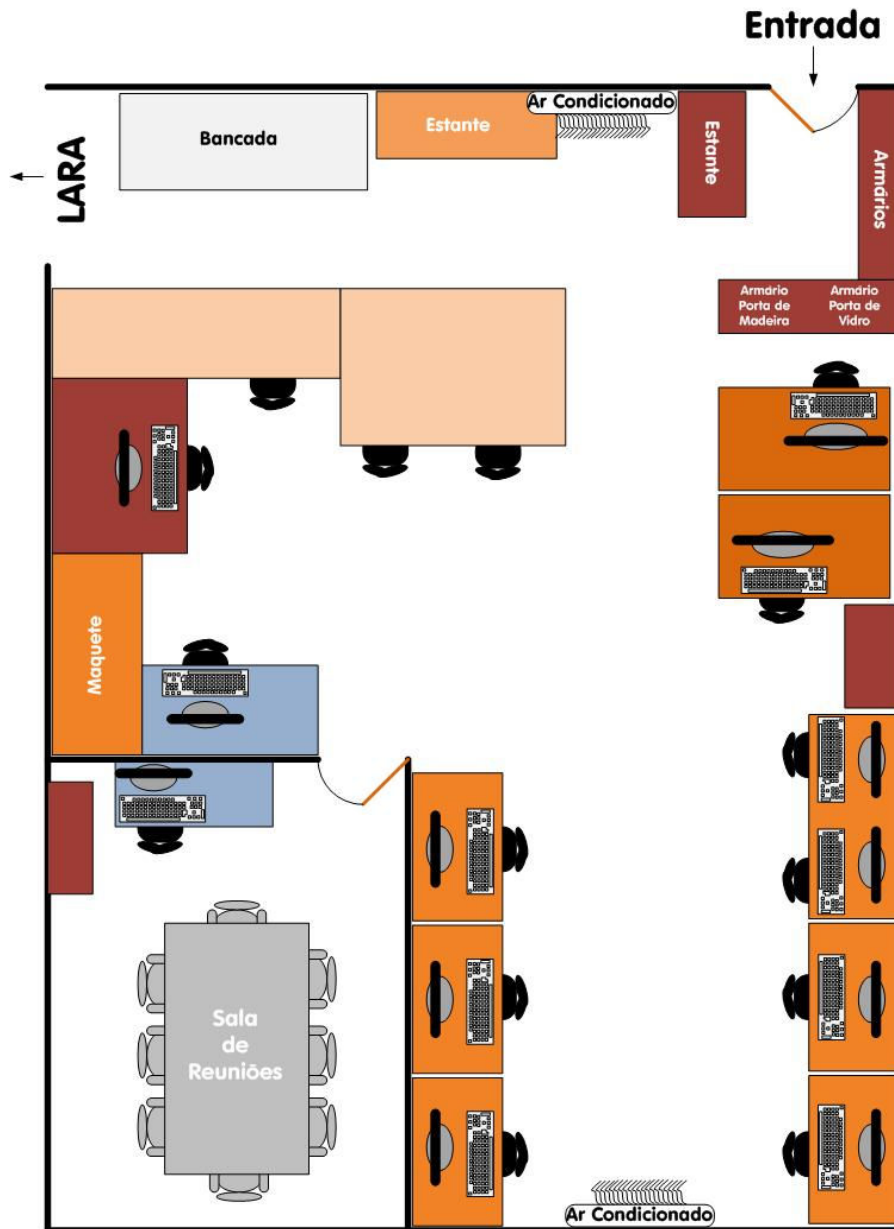


Figura 3.1 Sala controlada

#### 3.2 AR CONDICIONADO

Foram utilizados dois aparelhos de ar condicionado do tipo *split* com capacidade 22.000 BTU/h. Cada aparelho possui duas unidades, interna e externa. A unidade interna, também chamada de evaporadora, fica localizada dentro da sala e é responsável por insuflar ar frio na mesma. A unidade

externa, também chamada de condensadora, tem função de resfriar o gás que retorna da unidade evaporadora.

O acionamento da unidade condensadora é feito por uma placa eletrônica instalada na unidade evaporadora, que dependendo da temperatura solicitada pelo usuário e da temperatura do ambiente, envia um sinal de comando para ligar ou desligar a unidade externa. Trata-se de um sinal 220 V AC, por onde passa, no estado ligado, uma corrente de aproximadamente 0,8 A.

O controle foi realizado na unidade condensadora através do chaveamento deste sinal de comando proveniente da unidade evaporadora, fato que resultou em uma menor complexidade no projeto da placa de acionamento, uma vez que a corrente que passa pela mesma não é toda a corrente necessária para o funcionamento do ar condicionado (aproximadamente 12 A).

Para realização do controle liga-desliga, os aparelhos de ar condicionado são configurados para operarem em sua potência máxima, ou seja, selecionando-se a temperatura de 17 °C. Desta forma, o sinal proveniente da unidade evaporadora que liga a unidade condensadora sempre estará acionado, permitindo que ação de ligar e desligar tal unidade seja comandada pelo nó de controle através da placa de acionamento.

### 3.3 MÓDULO DE CONTROLE

O módulo, ilustrado pela figura 3.2, tem a função de controlar a placa de acionamento do ar condicionado. A placa é composta pelo ATmega8 e pelo XBee, com interface para gravação utilizando-se uma gravadora, e interface serial com a possibilidade de seleção através de *jumpers* entre a comunicação PC/XBee, XBee/ATmega8 e PC/ATmega8.

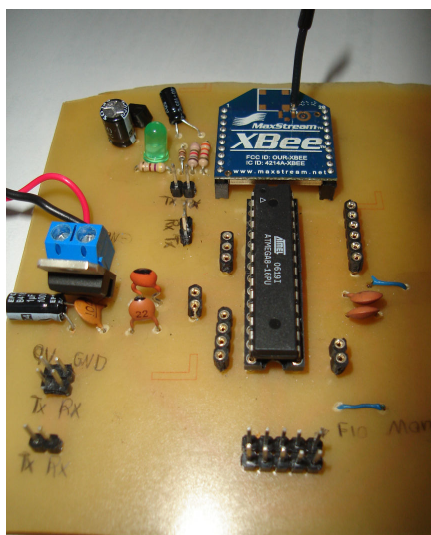


Figura 3.2 Foto do modulo de controle com ATmega8

O *firmware* gravado no microcontrolador recebe do supervisor (responsável pelo cálculo do controle liga/desliga) o estado do atuador, e através de uma porta de I/O comanda a placa de acionamento para que esta ligue ou desligue a unidade externa do ar condicionado.

A pouca memória do ATmega8 impede a implementação de controladores mais complexos, ou até mesmo a utilização de sistemas operacionais e protocolos de comunicação, assim a utilização do ATmega128 pode suprir essas deficiências.

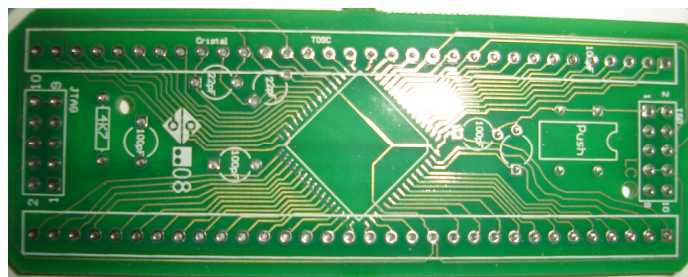


Figura 3.3 Circuito impresso do circuito de referência para o ATMEGA128

### 3.4 MÓDULO DE ACIONAMENTO

Como a carga a ser acionada é alimentada com tensão 220 V AC 60 Hz, foi montado um relé de estado sólido, conforme a figura 3.4, utilizando um triac como chave. O triac escolhido foi o TIC246D, que trabalha com corrente de até 16 A e suporta tensões de até 400 V. Deve ser observado que quanto maior a tensão e corrente a que ele for submetido, maior deverá ser o dissipador de calor a ser utilizado. O acoplador óptico foi utilizado o MOC3081, ideal para aplicações de 240 VAC. Além de isolar o circuito de controle do circuito de potência, ele possui internamente um triac com um circuito detector de zero.

No circuito da figura I.4 dos anexos, a resistência  $R_1$  tem a função de limitar a corrente no pino 1 do MOC em 15 mA e  $R_2$  limita a corrente no TRIAC interno, pino 6. O conjunto  $R_4, C_1$  é o chamado *snubber RC* e tem a função de reduzir as interferências eletromagnéticas geradas por cargas indutivas. A resistência  $R_3$ , denominado resistor de *gate*, impede que o triac seja acionado devido a alguma interferência gerada pelo próprio circuito. Este circuito é uma aplicação do manual do MOC3081.

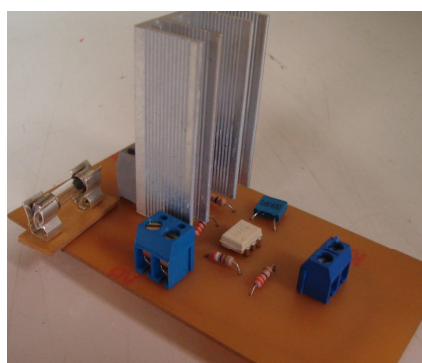


Figura 3.4 Foto do circuito de acionamento

### 3.5 MÓDULO SENSOR

No módulo sensor projetado, ilustrado na figura 3.5, foi utilizada a funcionalidade de conversão A/D do módulo XBee, permitindo assim que não fosse necessária a utilização de um microcontrolador nestes módulos.

O sensor de temperatura LM35, que possui ganho de 10 mV/°C e precisão de 0,5 °C, foi utilizado nos módulos. A placa possui ainda um amplificador operacional para amplificar o sinal do sensor em outro canal A/D do XBee e disponibiliza os pinos TX e RX do XBee para configuração.

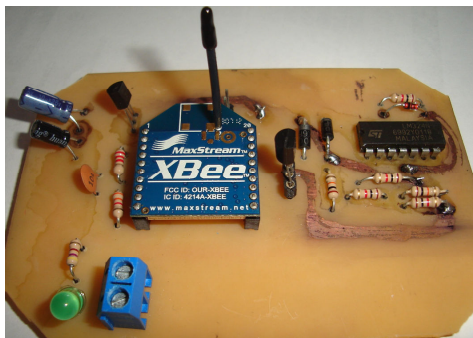


Figura 3.5 Foto do módulo sensor

Estes módulos foram construídos em conjunto com [11].

### 3.6 PLACA DE CONFIGURAÇÃO

A placa de configuração permite a comunicação do PC com os nós atuadores e sensores pela porta serial. Sua alimentação vem do próprio nó e a placa possui o CI MAX232 que converte os padrões TTL e CMOS para a EIA-232.



Figura 3.6 Foto da placa de configuração

### 3.7 REDE

O controle da rede é efetuado pelo software supervisor instalado em um PC no qual está conectado um módulo XBee. Ela é constituída por dois nós atuadores, dois nós sensores e o nó coordenador.

Os nós sensores e atuadores possuem endereçamento fixo, sendo que os sensores apenas enviam dados para o coordenador e nós atuadores enviam e recebem dados do coordenador. A comunicação é feita em modo *unicast*, e para isto o supervisor altera repetitivamente o endereço pessoal e de envio do nó coordenador, de acordo com a tabela 3.1.

Os nós sensores estão configurados para que a cada 7,5 s aproximadamente, sejam enviadas três amostras de temperatura. Os dados são enviados dentro de um pacote API com identificador 83. A figura 3.7 mostra um esquema da rede configurada.

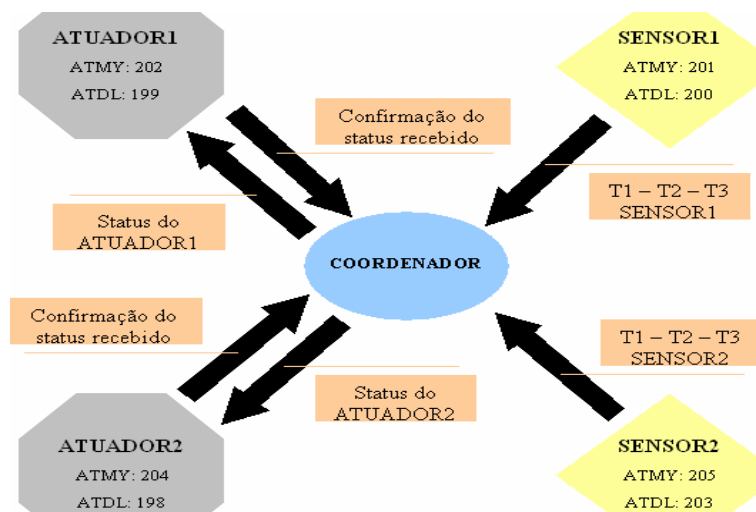


Figura 3.7 Rede configurada.

Tabela 3.1: Configuração dos módulos da rede

COORDENADOR:	REMOTE1:	REMOTE2:	ATUADOR1:	ATUADOR2:
ATMY: ~	ATMY: 201	ATMY: 205	ATMY: 199	ATMY: 204
ATDL: ~	ATDL: 200	ATDL: 203	ATDL: 202	ATDL: 198
ATIA: FFFF	ATIA: FFFF	ATIA: FFFF	ATIA: FFFF	ATIA: FFFF
ATIU: 1	ATIU: 1	ATIU: 1	ATIU: 1	ATIU: 1
	ATIR: 9C4	ATIR: 9C4		
	ATIT: 3	ATIT: 3		
	ATD0: 2	ATD0: 2		

### 3.8 SOFTWARE SUPERVISÓRIO

O supervisor foi desenvolvido na linguagem C e coordena todo o fluxo de dados da rede através do nó coordenador conectado ao PC. Ele trabalha alterando o endereço pessoal e destino do coordenador para que o mesmo receba/envie os dados para os nós indicados.

Ele grava em um arquivo os dados da temperatura de referência, temperatura dos nós, *status* dos atuadores e de tempo, permitindo a geração de gráficos com os dados coletados.

O supervisor é responsável pelo cálculo do controle liga-desliga e apenas envia o *status* do atuador para o nó determinado. O cálculo é realizado separadamente para cada setor da sala, sendo que

o status de cada atuador está vinculado aos dados de temperatura enviados pelo nó sensor localizado no mesmo setor. A referência de temperatura utilizada é única para toda a sala.

As operações de configuração são realizadas no modo de comando e a leitura de dados no modo *Idle* do módulo XBee. Cada ciclo de leitura dos dados de todos os sensores e atuadores dura aproximadamente 37 segundos, e após 18 segundos, aproximadamente, é feita a leitura de temperatura de um sensor.

As três amostras de temperatura obtidas dos sensores são tratadas pelo supervisor na tentativa de impedir que um valor lido incorretamente possa alterar erroneamente a média dos valores. Se a diferença par a par absoluta entre os três valores lidos for menor ou igual a 1 ou todas sejam maiores que 1, a temperatura considerada será a média dos três valores. Senão, a temperatura considerada será a média entre as duas amostras em que diferença seja menor ou igual a 1.

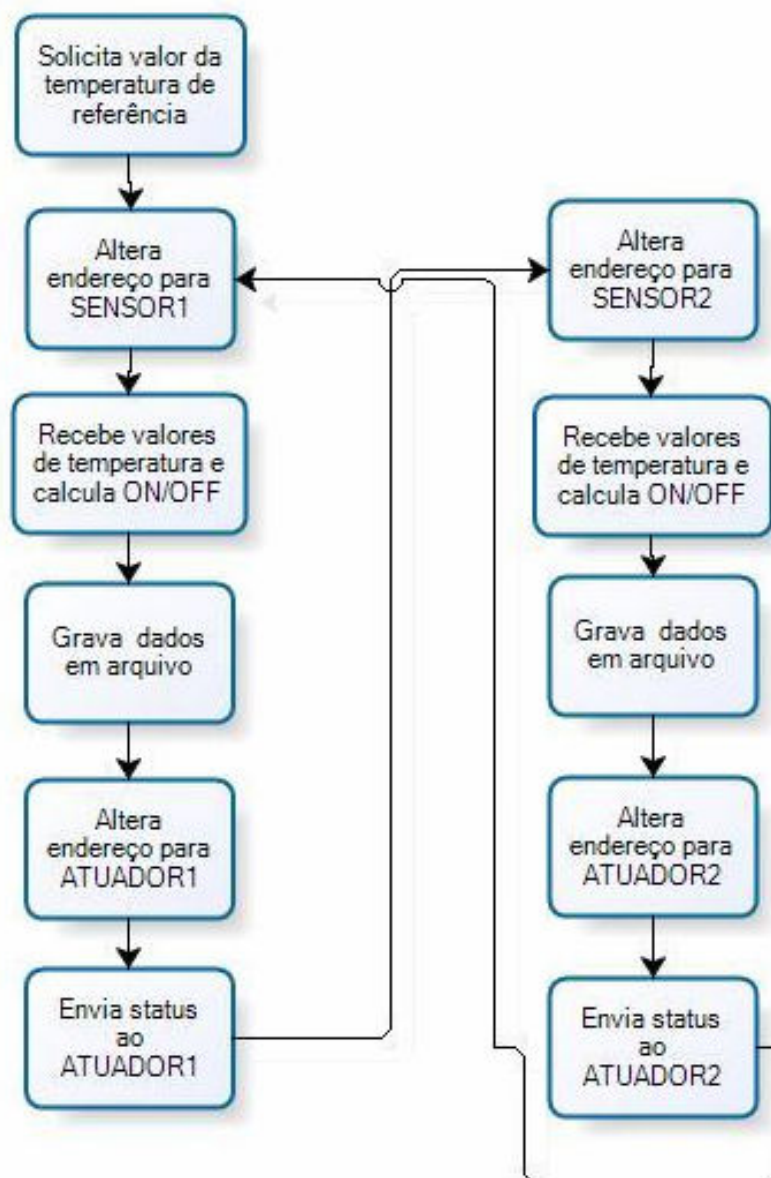


Figura 3.8 Fluxograma de funcionamento do software supervisor.



## 4 RESULTADOS EXPERIMENTAIS E ANÁLISE

### 4.1 EXPERIMENTOS

Foram realizados cinco experimentos com diferentes valores de referência. Em alguns deles foi solicitada a alteração da referência durante o experimento e em outros manteve-se o valor da referência constante. Os resultados foram separados por experimento com uma breve análise individual de cada gráfico, e ao final, uma análise geral dos resultados obtidos.

#### Experimento 1:

Temperatura de referência: 21,5 °C

Data: 24/06/2008

Hora: 08h20min

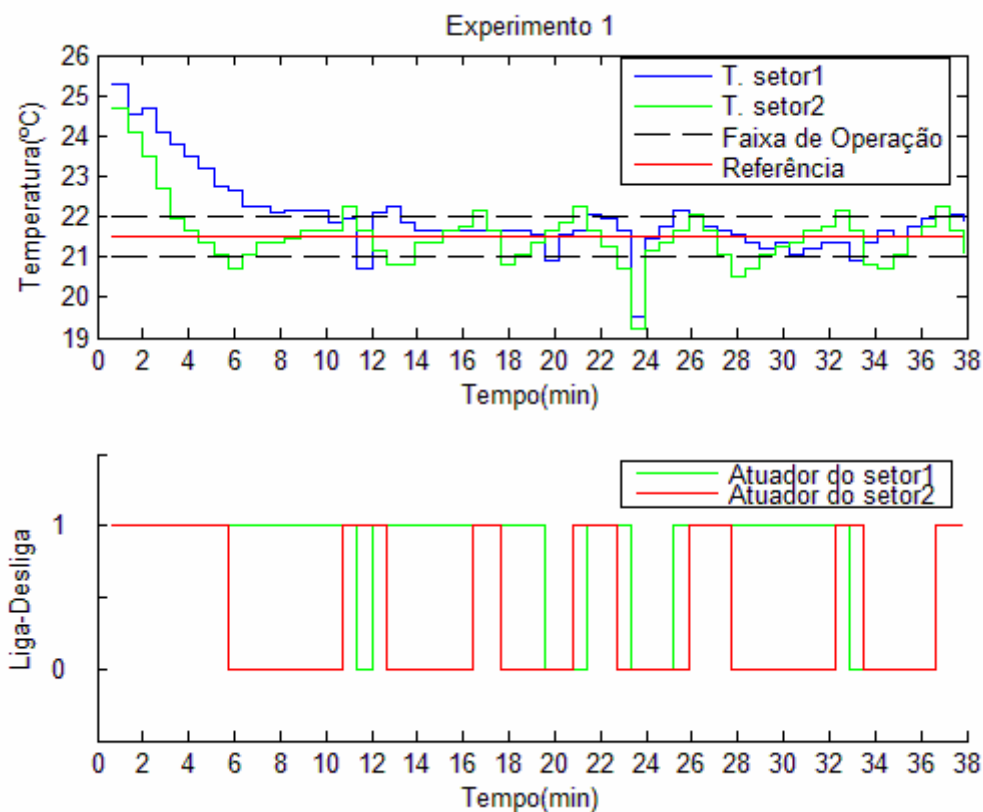


Figura 4.1 - Experimento 1

Como o experimento foi realizado no início do dia, o laboratório estava praticamente sem usuários e os aparelhos de ar condicionado estavam desligados. Assim, conforme mostrado no gráfico, a temperatura inicial nos dois setores estava por volta de 25,0 °C. Constata-se que o atuador do corredor consegue com menos tempo do que o atuador da entrada, colocar a temperatura abaixo da



faixa definida para o controlador liga-desliga, 0,5 °C. Tal fato pode ser explicado pelas dimensões dos dois setores, mesmo não havendo separação física entre eles.

### Experimento 2:

Temperatura de referência: 23,5 °C

Data: 24/06/2008

Hora: 09h00min

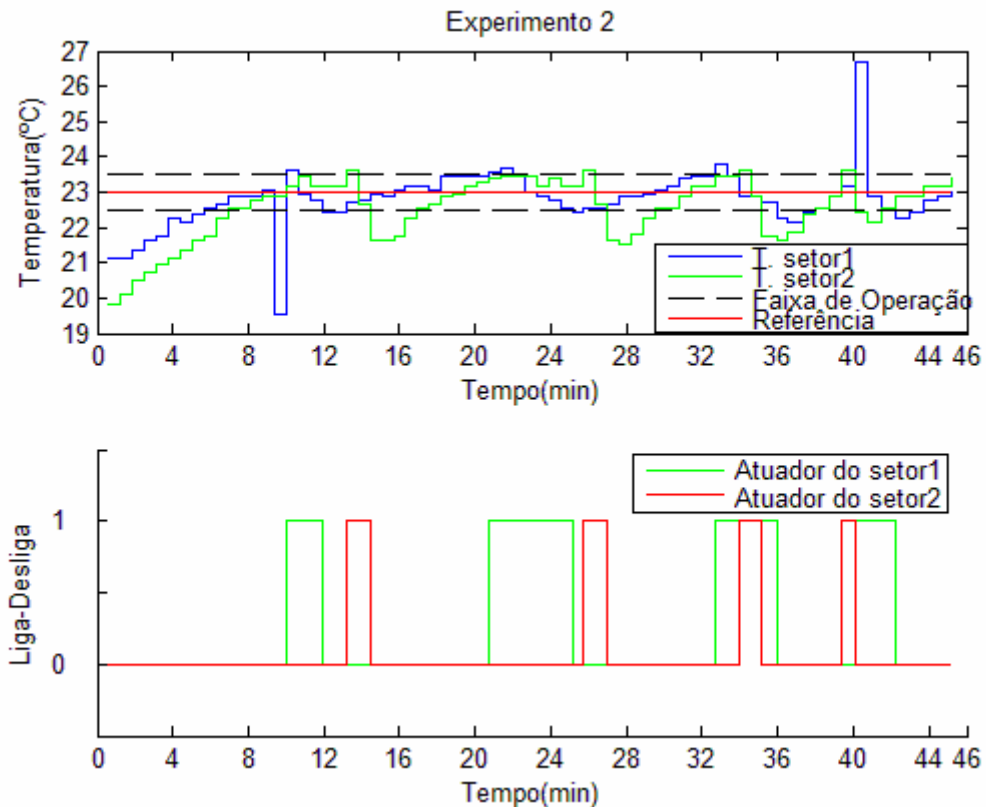


Figura 4.2 - Experimento 2

O experimento iniciou-se logo após o Experimento 1, assim, a temperatura inicial nos dois setores estava por volta de 21,5°. Nota-se que o atuador do corredor consegue em aproximadamente um minuto, colocar a temperatura no setor abaixo do limite inferior da faixa definida para o controlador. O ar frio insuflado por este atuador fica bastante concentrado devido a área do setor, o que faz com que a temperatura caia mais rápido.

Como no Experimento 1, o erro para o setor 1 não chegou a 1,0 °C, porém, devido ao relatado no parágrafo anterior, o erro abaixo da referência do setor 2 alcançou 1,5 °C.

### Experimento 3:

Temperatura de referência: 22,5 °C - 23,8 °C - 21,5 °C

Data: 24/06/2008

Hora: 20h10min

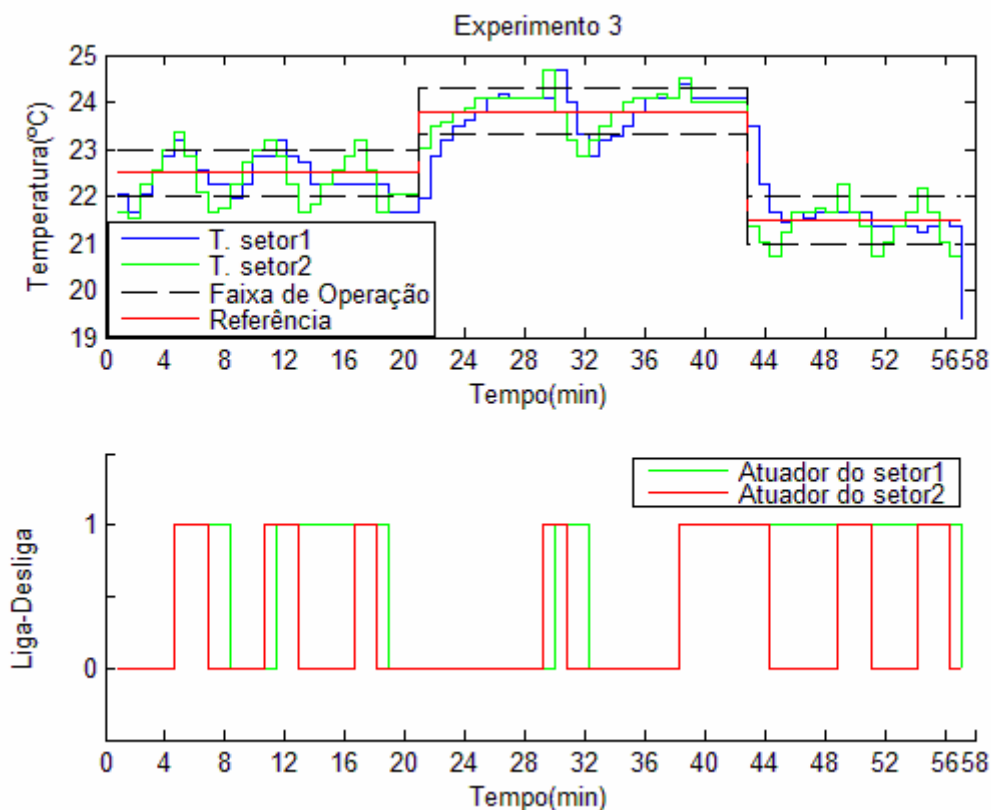


Figura 4.3 - Experimento 3

O experimento iniciou-se logo após um teste onde a referência era 21,5 °C, fato que explica a temperatura nos dois setores estarem próximas deste valor. Nota-se que os valores medidos acompanharam as referências, com um erro de aproximadamente 1,0 °C.

No momento em que se altera a referência, observa-se um intervalo de tempo diferente entre os dois pontos antes e depois da alteração em relação aos demais. Isso ocorre porque o supervisor solicita ao usuário, em intervalos de tempo pré-definidos, se o mesmo deseja alterar a referência de temperatura. No momento desta consulta o software deixa de coletar os dados de temperatura e os atuadores continuam nos seus estados anteriores, porém a variável tempo não para sua contagem.

#### Experimento 4:

Temperatura de referência: 24,0 °C - 21,5 °C - 22,5 °C

Data: 25/06/2008

Hora: 08h20min

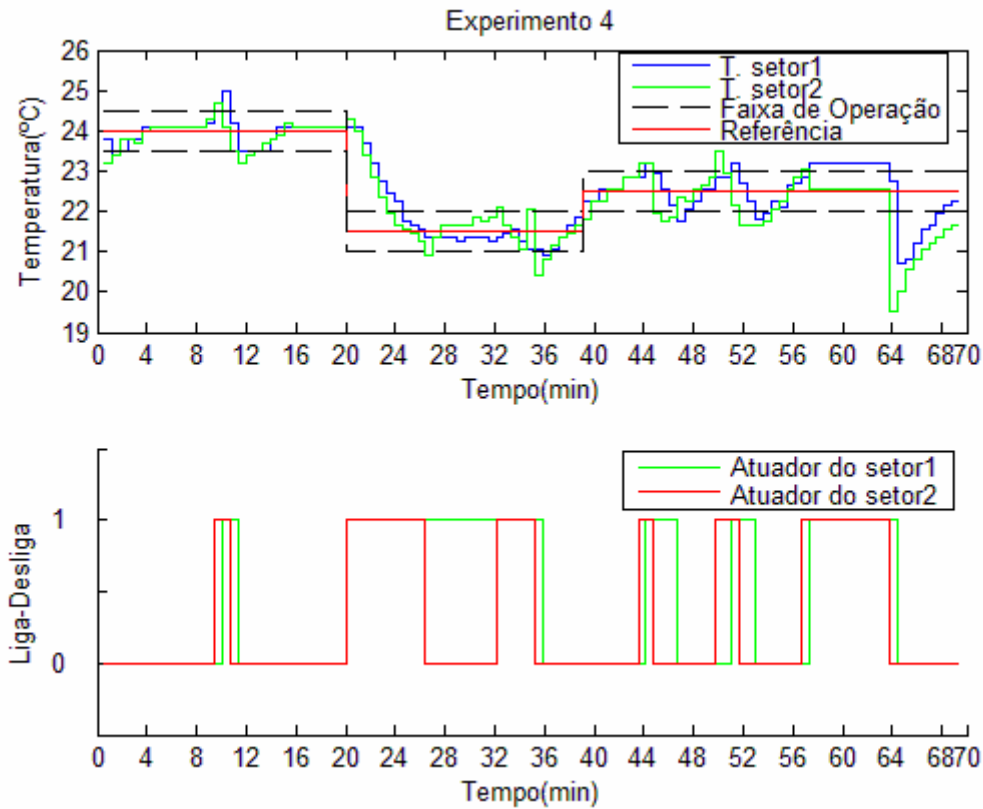


Figura 4.4 - Experimento 4

O laboratório estava fechado e sem usuários, e o valor da temperatura nos dois setores era de aproximadamente 23,5 °C, bem próxima da referência inicial. Os valores de temperatura seguiram as referências estabelecidas, com o erro de regime em cada referência não ultrapassando 1,0 °C.

Entre os instantes 58 e 64 minutos o supervisor ficou muito tempo aguardando a alteração ou não da referência, o que explica a o grande intervalo entre as duas amostras.

## Experimento 5:

Temperatura de referência: 23,5 °C - 21,5 °C

Data: 25/06/2008

Hora: 09h40min

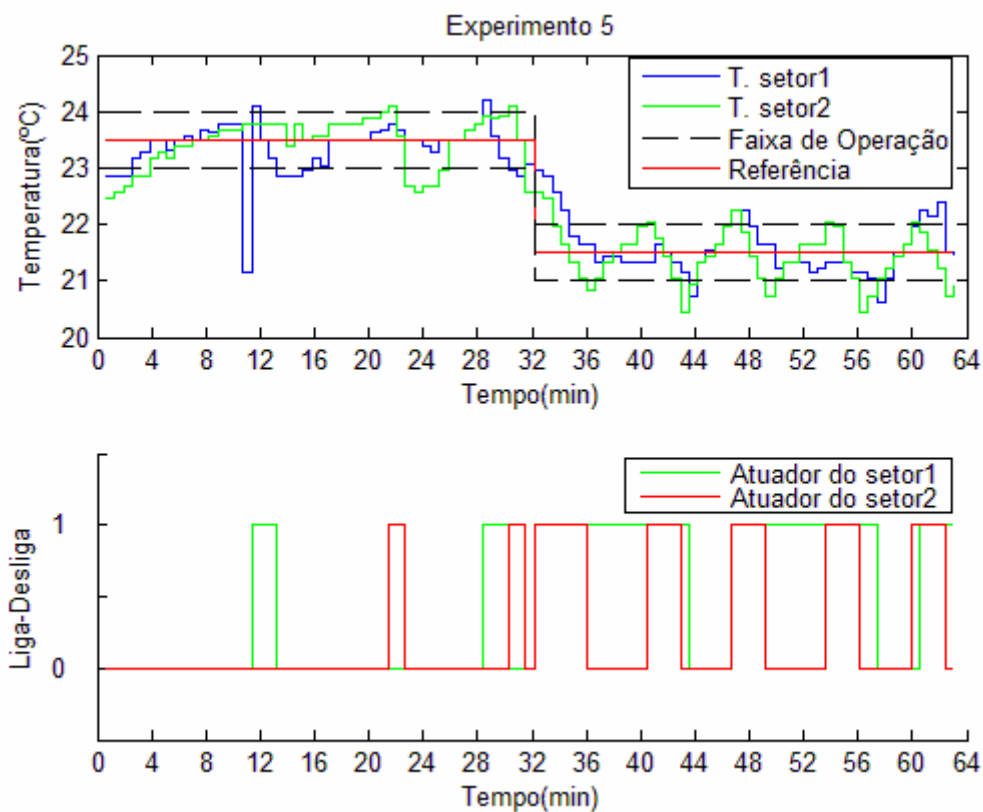


Figura 4.5 - Experimento 5

A baixa temperatura inicial se deve ao fato de o experimento ter sido realizado após o Experimento 4. Com o erro de regime em cada referência não ultrapassando 1,0 °C obteve-se mais uma vez um controle satisfatório.

Verifica-se que o atuador do corredor consegue abaixar a temperatura do referido setor com maior velocidade do que o atuador da entrada.

## 4.2 ANÁLISE

Em termos de controle de temperatura, o controlador liga-desliga implementado obteve resultados satisfatórios. Em todos os experimentos notou-se um erro de regime médio de aproximadamente 1,0 °C, o qual no contexto de controle de temperatura de um ambiente normal de trabalho pode ser considerado um bom resultado.

Tratando-se de conforto térmico, o fato de o aparelho de ar condicionado estar configurado para atuar em sua potência máxima quando ligado, provoca bastante desconforto nas pessoas que estiverem na direção do ar frio insuflado. Desta forma, devem-se considerar outras maneiras de se realizar o controle de temperatura, por exemplo, utilizando sensores de velocidade do vento e direcionando o ar frio para locais onde não existem pessoas.

Mesmo se tratando de um sistema lento, o aumento da taxa de amostragem irá melhorar a visualização das curvas e será de grande importância no momento em que forem realizados experimentos para obtenção da função transferência das salas do LAVSI. Como o supervisor implementado realiza as leituras de dados no modo *Idle* e as alterações de endereço no módulo XBee coordenador através do modo de comando, é necessário uma série de esperas (*sleeps*) no software para que as configurações possam ser realizadas corretamente e as leituras de dados sejam realizadas sem erros. A utilização dos *transceiver* XBee em modo API certamente será uma opção que deverá ser considerada, principalmente no momento em que forem inseridos na rede uma grande quantidade de nós sensores e atuadores.

O envio por parte dos nós sensores de 3 amostras de temperatura em cada pacote, permitiu a eliminação pelo supervisor de alguma temperatura destoante das demais. Porém, constata-se em quase todos os experimentos a existência de pontos fora dos padrões das curvas. Uma opção para a solução deste erro é o envio de mais amostras por parte do nó sensor, que permite uma maior seleção das temperaturas médias no software supervisor.

## 5 CONCLUSÕES E EXPECTATIVAS FUTURAS

O controle liga-desliga atingiu os resultados esperados para o conforto térmico do ambiente analisado, pois além atingir um erro de regime de aproximadamente 1,0 °C, em termos de futuro do projeto, os experimentos realizados possibilitaram uma primeira visualização do controle de temperatura no LAVSI, permitindo vislumbrar problemas e soluções nas futuras montagens de redes e controles mais complexos.

A utilização do ATmega8 nos nós atuadores foi possível devido à simplicidade do *firmware* utilizado. Como o controle liga-desliga é calculado pelo supervisor, o *firmware* ocupou apenas aproximadamente 1,2 Kbytes de memória do microcontrolador. Caso a opção futura seja uma rede com controle distribuído, é necessário a utilização de um microcontrolador com mais memória, como o ATmega128, permitindo assim a implementação de controladores mais complexos, de um protocolo de comunicação e até mesmo a utilização de um sistema operacional.

Um fator importante constatado na realização do trabalho foi a forma de fornecimento de energia aos nós. Como os nós atuadores estarão próximos da fonte de energia que alimenta os aparelhos de ar condicionado, é necessário que estes nós aproveitem esta disponibilidade para sua alimentação. Devido a necessidade de maior processamento e conseqüente consumo de energia, a utilização de baterias de 9 V mostrou-se inviável nos nós atuadores. Por outro lado, nos nós sensores é indispensável que a fonte de energia seja portátil, uma vez que os mesmos estarão localizados em diversos locais onde não estará disponível uma fonte de energia convencional. O uso de baterias de 9 V nestes nós é uma boa opção, porém, é extremamente importante que sejam utilizadas as funcionalidades do modo *sleep* do XBee, permitindo uma maior durabilidade das mesmas.

O levantamento da função de transferência do laboratório permitirá uso de controladores mais eficientes, e conseqüente melhora no controle de temperatura da sala. Uma compreensão mais detalhada do funcionamento dos aparelhos de ar condicionado poderá contribuir para que, havendo possibilidade, de acioná-lo de outras maneiras e como conseqüência obter a máxima economia de energia.

No prosseguimento do projeto de automação dos laboratórios, um grande esforço na área de desenvolvimento de software será necessário, com objetivo de criar funções específicas de interface com o nó coordenador e conseqüente controle eficiente do fluxo de dados na rede e promover a integração da rede ZigBee a um software SCADA.

## 6 REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Osterlind, F.; Pramsten, E.; Roberthson, D; Eriksson, J. *Integrating Building Automation Systems and Wireless Sensor Networks*. 2007. Swedish Institute of Computer Science. Disponível em: <http://ieeexplore.ieee.org/iel5/4416742/4416743/04416941.pdf?tp=&arnumber=4416941&isnumber=4416743>. Acesso em 25 de Junho de 2008.
- [2] Snoonian, D. Spectrum IEEE Volume 40 - *Smart Buildings*. 2003. Disponível em: <http://ieeexplore.ieee.org/iel5/6/27450/01222043.pdf?tp=&arnumber=1222043&isnumber=27450> Acesso em 25 de Junho de 2008.
- [3] Wannner, L. F. *Um Ambiente de Suporte à Execução de Aplicações em Redes de Sensores sem Fio*. Universidade Federal de Santa Catarina. Florianópolis. 2006. Disponível em: <http://www.lisha.ufsc.br/~guto/teaching/theses/lucas-msc.pdf> Acesso em 25 de Junho de 2008.
- [4] Tsang, K. F.; Lee, W. C.; Lam, K. L.; Tung, H. Y.; Xuan, K. Na *Integrated ZigBee Automation System: Na Energy Saving Solution*. Disponível em: <http://ieeexplore.ieee.org/iel5/4430702/4430703/04430752.pdf?tp=&arnumber=4430752&isnumber=4430703> Acesso em 25 de Junho de 2008.
- [5] MaxStream, *Product Manual v1.xAx - 802.15.4 Protocol For OEM RF Module Part Numbers: XB24-001, XBP24-001*, <http://www.maxstream.net>.
- [6] ROGERCOM – *Tutorial sobre ZigBee*. Tópico sobre o tranceiver wireless XBee/ZigBee IEEE 802.15.4. Disponível em <http://www.rogercom.com/>> Acesso em 23 de maio de 2008.
- [7] ThinkLABS, [triindia.co.in/shop/images/mega8.jpg](http://triindia.co.in/shop/images/mega8.jpg)> Acesso em 26 de junho de 2008.
- [8] DigiKey.com, [triindiarocky.digikey.com/.../313-64-TOFP.jpg](http://triindiarocky.digikey.com/.../313-64-TOFP.jpg)> Acesso em 26 de junho de 2008.
- [9] Silva J. M. G. Ações Básicas de Controle . Universidade Federal do Rio Grande do Sul. Disponível em: <http://www.ece.ufrgs.br/~jmgomes/pid/Apostila/apostila/node21.html> Acesso em 01 de Julho de 2008.
- [10] Silva J.M. G. Controle on-off. Universidade Federal do Rio Grande do Sul. Disponível em: <http://www.ece.ufrgs.br/~jmgomes/pid/Apostila/apostila/node20.html> Acesso em 01 de Julho de 2008.
- [11] Couto, F. L.; Figueiredo, L. F. da C. *Medição de conforto térmico para rede de automação predial wireless*. 2008. 98p. Trabalho de Graduação 2. Universidade de Brasília. Brasília.
- [12] Oliveira, D. B. S; Zanatta, M. P. B. F.; Santos, R. A. *Rede de sensores e atuadores wireless para automação predial*. 2007. 48 p. Trabalho de Graduação 2. Universidade de Brasília. Brasília.
- [13] ATMEL CORPORATION: *Microcontrolador ATmega*. Disponível em [http://www.Atmel.com/dyn/resources/prod\\_documents/doc2486.pdf](http://www.Atmel.com/dyn/resources/prod_documents/doc2486.pdf) Acesso em 22 de Junho de 2008.
- [14] Zelenovsky, Ricardo; Mendonça, Alexandre. *Programação e Projetos com a Família 8051*. Rio de Janeiro: MZ Editora LTDA, Agosto de 2005. 447 p.
- [15] ATMEL CORPORATION: *Microcontrolador ATmega128*. Disponível em [http://www.Atmel.com/dyn/resources/prod\\_documents/doc2467.pdf](http://www.Atmel.com/dyn/resources/prod_documents/doc2467.pdf) Acesso em 22 de Junho de 2008.
- [16] Ogata, Katsuhiko *Engenharia de controle moderno; tradução Paulo Álvaro Maya; revisão técnica Fabrizio Leonardi [et al.]*. – 4. ed. –São Paulo: Prentice Hall, 2003. 788p.
- [17] Nise, Norman S. *Engenharia de Sistemas de Controle*. Rio de Janeiro: LTC, 2002. 695 p.
- [18] SANTOS, R., J. *Controle Fuzzy para Racionalização de Energia em Processo de Condicionamento de Ar*. 2005. 134 p. Dissertação de Mestrado em Engenharia Elétrica. Universidade de Brasília. Brasília.
- [19] Ahmed, A. *Eletrônica de Potência*. São Paulo: Pearson Prentice Hall, 2000. 479 p.
- [20] Mussoi, F. L. R. *Tiristor SCR: Retificador Controlador de Silício*. 1.1. ed. Florianópolis: 2002, 43 p.
- [21] Soares, M. J. *Controle de cargas AC com conjunto TRIAC/MOC*. *Saber Eletrônica*, Tatuapé, jun. 2006. p.10-15.

## 7 BIBLIOGRAFIA

Borges, G. A., Bo, A. P. L., Martins, A. S., Cotta, L. C., Fernandes, M., Freitas, G., Beckman, E. *Desenvolvimento com microcontroladores Atmel AVR*. 2006. 34 p. Nota Técnica. Universidade de Brasília. Brasília.

BARRETT, Raymond; CALLAWAY, Edgar; GUTIERREZ, Jose. *IEEE 802.15.4 Low-Rate Wireless Personal Area networks: Enabling Wireless Sensor Networks*. Institute of Electrics & Electronics Engineering, 2003. Acessado em 1 de Junho de 2008.

PINHEIRO, José Maurício Santos Pinheiro. *As redes com ZigBee*. *Eletronica.org*, Julho de 2004. [http://www.projetoredes.com.br/artigos/artigo\\_zigbee.php](http://www.projetoredes.com.br/artigos/artigo_zigbee.php). Acessado em 1 de Junho de 2008.

GALEEV, Mikhail. *Home networking with ZigBee*. *Embedded.com*, Abril de 2004. <http://www.embedded.com/showArticle.jhtml?articleID=18902431>. Acessado em 1 de Junho de 2008.

ALMHPG. *Tiristor*. Antonio Almeida. Brasil, 2007. Disponível em [http://www.almhpg.com/apo\\_arquivos/tiristor/tiristor.htm](http://www.almhpg.com/apo_arquivos/tiristor/tiristor.htm)> Acesso em: 15 de Junho 2008.

ARDUINO. *Xbee*. Arduino. 15 de Agosto. 2007. Disponível em <http://www.arduino.cc/en/Main/ArduinoXbeeShield>> Acesso em: 2 de Junho 2008.

ATMEL – SOFTWARE AVRSTUDIO: *Software para interação com o AVR*. Disponível em [http://www.atmel.com/dyn/products/tools\\_card.asp?tool\\_id=2725](http://www.atmel.com/dyn/products/tools_card.asp?tool_id=2725)> Acesso em 25 de maio de 2008.

ELEKTRODA. *Protel 2004 Training Manual*. *PAfileDB*. Disponível em <http://www.elektroda.net/download/pafiledb.php?action=file&id=1321>> Acesso em 25 de Março de 2008.

ELECTRONICS DIY.com: *Projetos de comunicação serial e USB para microcontroladores AVR ATMEGA8 ATMEGA 128*. Disponível em [http://electronics-diy.com/electronic\\_schematic.php?id=761](http://electronics-diy.com/electronic_schematic.php?id=761)> Acesso em 23 de maio de 2008.

HARDWAREBOOK. *Porta Serial*. *Universidade de Monterrey-MEX*. Disponível em [http://www.hardwarebook.info/Serial\\_%28PC\\_9%29](http://www.hardwarebook.info/Serial_%28PC_9%29)> Acesso em 25 de Maio de 2008.

ITP. *XBee*. Rob Faludi. Disponível em <http://itp.nyu.edu/~raf275/meshnetworking/XBee/index.html>> Acesso em: 8 de Outubro de 2007.

MIKROE. *Optoacopladores*. *MikroElektronika*. Portugal, 2003. Disponível em <http://www.mikroe.com/pt/product/books/picbook/capitulo6.htm#Optoacopladores>> Acesso em: 1º de dezembro 2007.

MUNDO DO WIFI. *As redes com zigbee*. *Fórum*. São Paulo, 2003. Disponível em <http://www.mundowifi.com.br/artigos/177-as-redes-com-zigbee.html>> Acesso em: 30 de novembro 2007.

SQUIDBEE. *Zigbee*. *Libelium*. Disponível em [http://www.libelium.com/squidbee/index.php?title=Main\\_Page](http://www.libelium.com/squidbee/index.php?title=Main_Page)> Acesso em: 2 de Outubro de 2007.

ZIGBEE. *Site sobre o padrão ZigBee*. Disponível em <http://www.zigbee.org/>> Acesso em 1 de Junho de 2008.



## **8 ANEXOS**

## **9 I. DIAGRAMAS ESQUEMÁTICOS**

Este anexo contém os circuitos completos dos módulos: sensor, de acionamento e de controle. Na Figura I.1, tem-se o circuito do módulo de controle com ATmega8. Na Figura I.2, tem-se o circuito do módulo sensor. Na Figura I.3, tem-se o circuito da placa de configuração. Na Figura I.4, tem-se o circuito do módulo de acionamento. E na Figura I.5, tem-se o circuito de referência para o ATmega128.

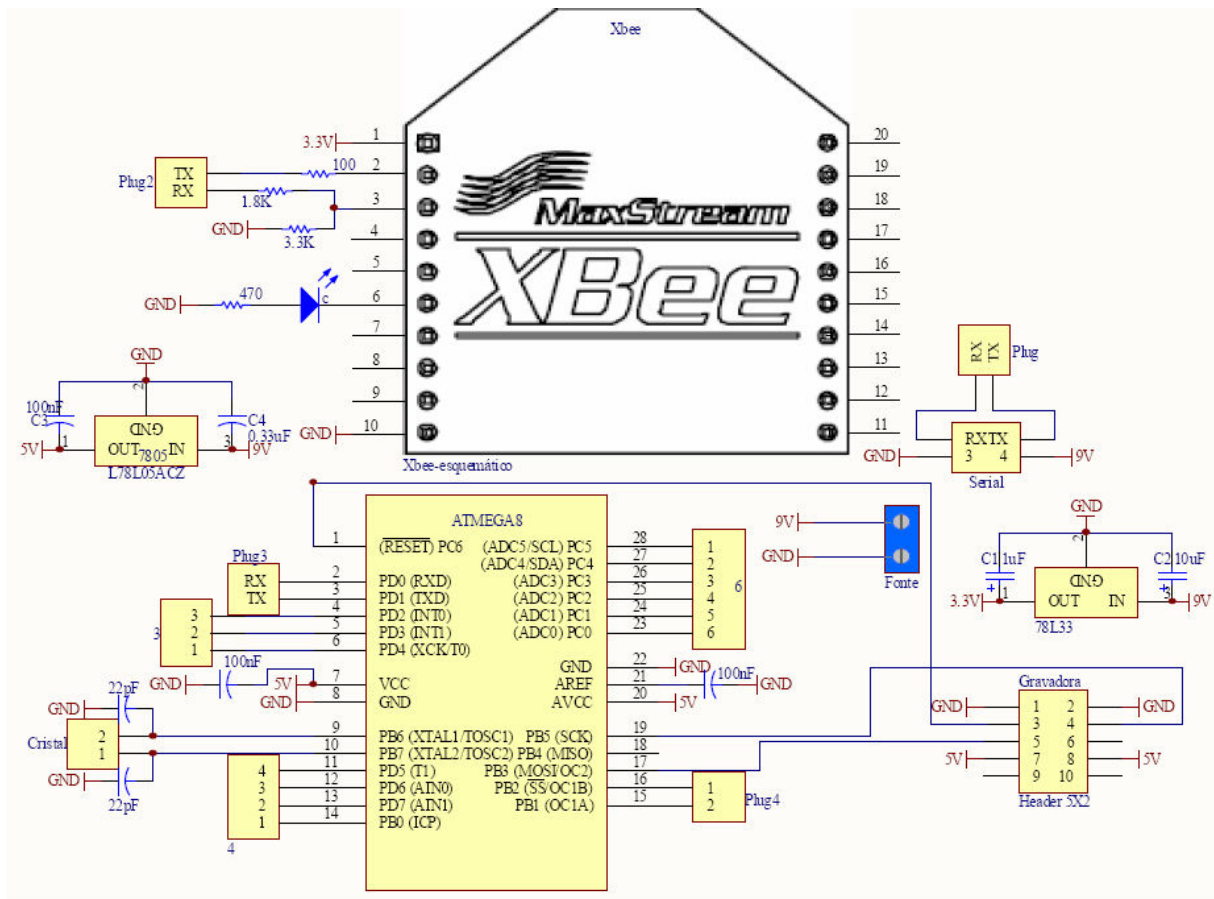


Figura I.1: Circuito do módulo de controle com ATmega8

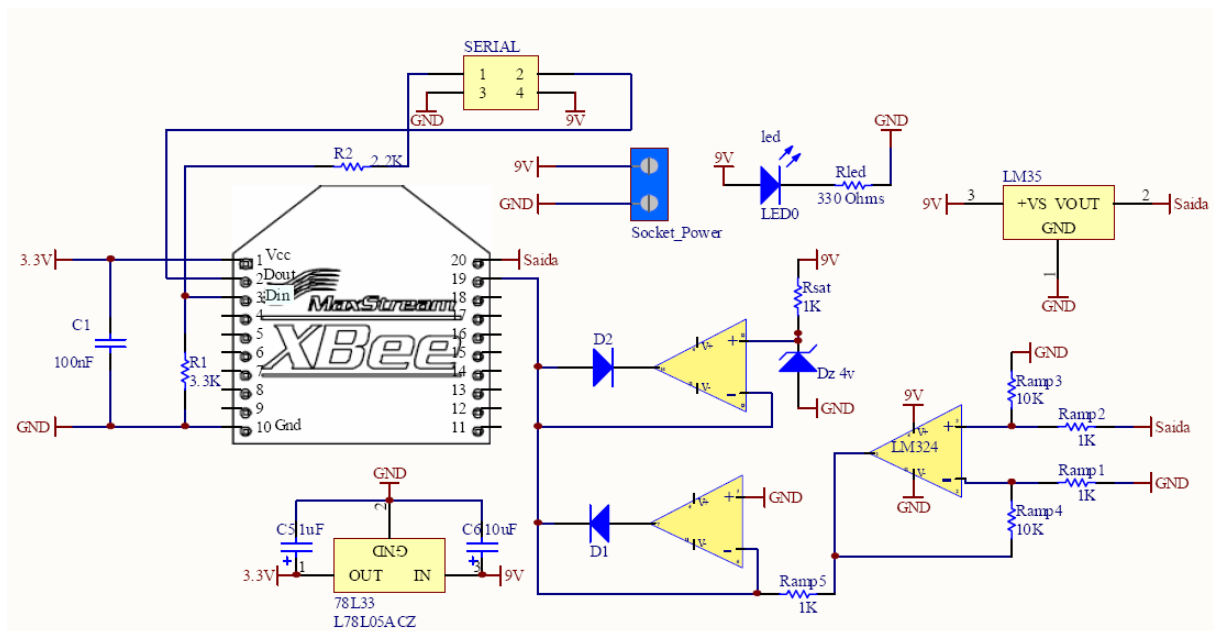


Figura I.2: Circuito do módulo sensor

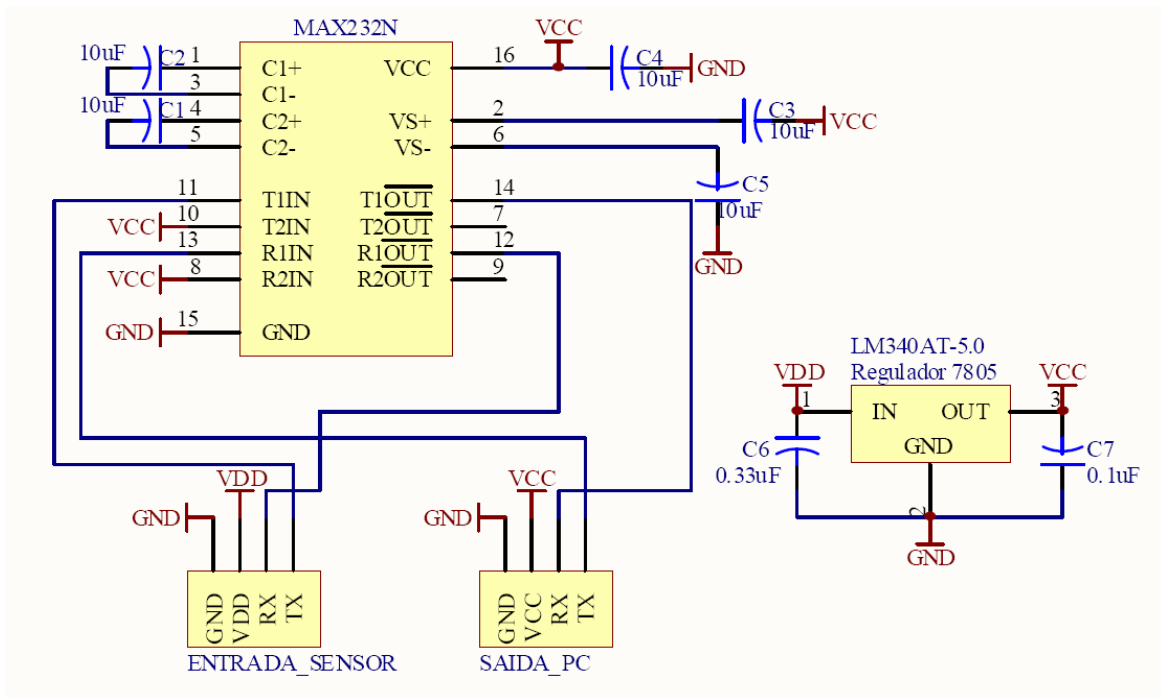


Figura I.3: Circuito da Placa de Configuração

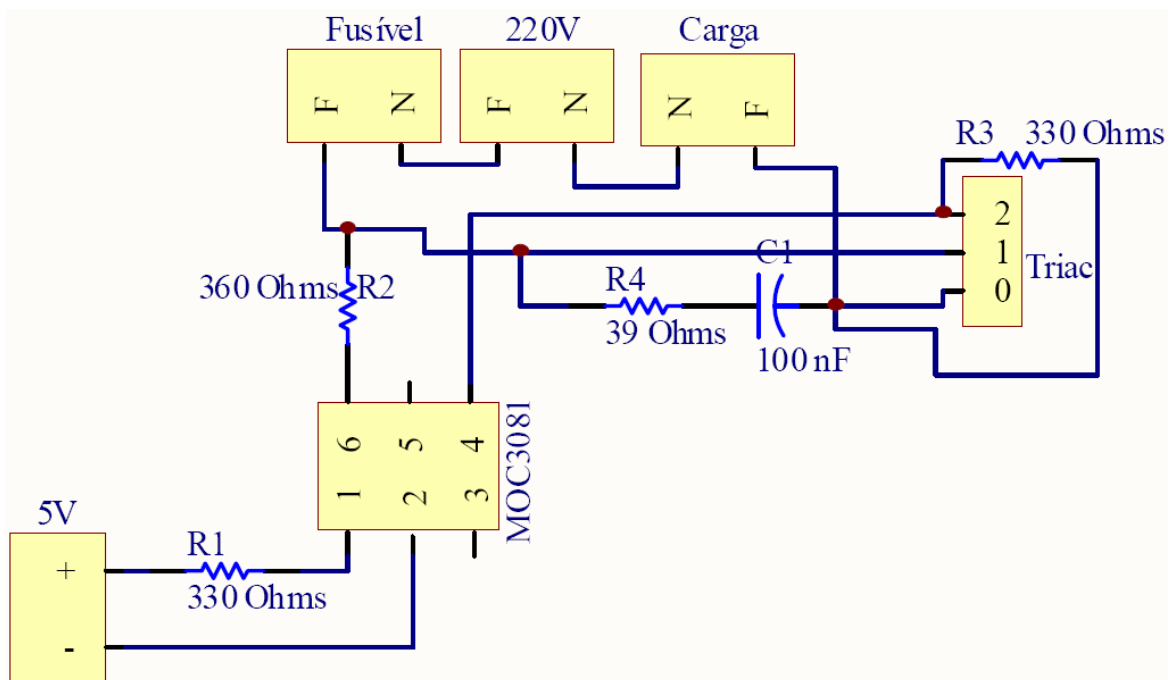


Figura I.4: Circuito do módulo de acionamento

Os pinos 3 e 5 do MOC 3081 são NC, portanto, eles estão em aberto no esquemático. O resistor R1 foi calculado, segundo o manual do dispositivo, considerando o valor da tensão (5V) e uma corrente de 15 mA, resultando em um  $R1=330$  Ohms.

As letras F e N que estão no esquemático, representam Fase e Neutro, respectivamente, uma vez que quando se trata de circuitos com corrente alternada.

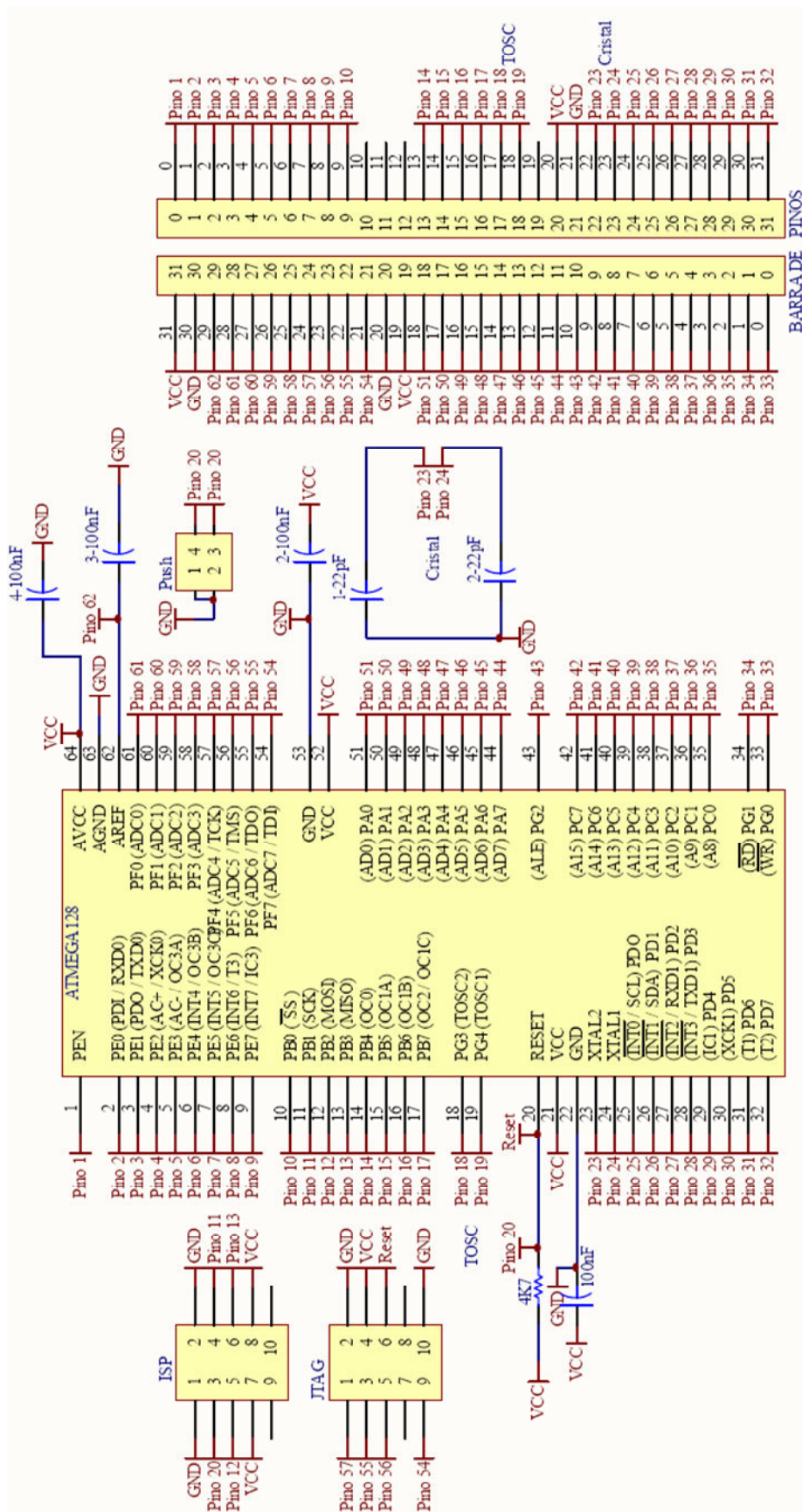


Figura I.5: Circuito de referência para o ATmega128

## **II. SOFTWARES**

Este anexo contém os programas: *Firmware* do módulo de controle e o *software* do supervisor, escritos em linguagem C.

## **Firmware do módulo de controle:**

```
/*Nome do arquivo: s1.c
   Autores: Pedro Ramos Mateus Filho
           Yuri Ferreira Gomes Dias
   Data: 21/05/2008
   Descrição: Firmware do ATmega8 para acionamento de carga num de controle
de temperatura em rede de automação wireless. Ele recebe do supervisor
o seu status definido e liga ou não uma porta de I/O que aciona um rele de
estado solido.

   Última modificação: 25/06/2008
*/

/*Bibliotecas incluídas no programa*/
#include <stdio.h>
#include <avr/io.h>
#include <stdlib.h>
#include <util/delay.h>

/*Funcao para configuracao e utilizacao da UART do microcontrolador*/
void USART_Init(void);
int USART_Transmit( char data);
int USART_Receive( void );

int main (){

char status_atuador;

USART_Init();

/*Configura os PORTB como saida e inicia PB0 como 0*/
DDRB = 0xFF;
PORTB = 0<<PB0;

fdevopen(USART_Transmit,USART_Receive);

/*LOOP que recebe os dados do XBee e retorna o valor lido. Dependendo
do valor ele liga ou desliga PB0.*/
while (1){

    fflush(stdin);
    status_atuador = fgetc(stdin);
    fflush(stdin);

    if (status_atuador == EOF){
        printf("%c", 'E');
    }else{
        printf("%c", status_atuador);
    }

    if (status_atuador == 'L'){
        PORTB = (0<<PB0);
    }

    if(status_atuador == 'H'){
        PORTB = (1<<PB0);
    }

    _delay_us(3000000);
}
```

```

return 0;
}

/*AS FUNCOES ABAIXO CONFIGURAM A UART DO MICRONTROLADOR:

    POR: Borges, G. A., Bo, A. P. L., Martins, A. S., Cotta, L. C.,
    Fernandes, M., Freitas, G., Beckman, E. Desenvolvimento com
    microcontroladores Atmel AVR. 2006. 34 p. Nota Técnica. Universidade de
    Brasília. Brasília.*/

void USART_Init(void) {

    /*zerando o conteúdo do registrador de dados da serial*/
    UDR=0;
    //Seta baud rate de 9600 para fosc= 1MHz
    UBRRH = 0;
    UBRL = 12;
    //Habilita transmissão e recepção serial
    UCSRB = _BV(TXEN) | _BV(RXEN);
    UCSRA = _BV(U2X);
    //Setar tamanho palavra
    //URSEL = 1 -> seleciona acesso para UCSRC (0 is UBRRH) reading and
writing
    //UMSEL = 0 -> Asynchronous Operation
    //UPM1 e UPM0 = 00 -> sem paridade
    //USBS = 1 -> seleciona 2 bits de parada a ser inserido pelo
transmissor
    //UCSZ 2 1 0 = 011 -> escolhe tamanho de caractere de 8 bits p/ dado
    UCSRC = _BV(URSEL) | _BV(UCSZ1) | _BV(UCSZ0);
}

/*
Função que faz a leitura de um byte da serial usando pooling
*/
int USART_Receive( void )
{
    /* Wait for data to be received */
    while ( !(UCSRA & (1<<RXC)) );

    /* Get and return received data from buffer */
    return UDR;
}

/*
Função que faz a transmissão de um byte pela serial.
*/
int USART_Transmit( char data )
{
    /* Wait for empty transmit buffer */
    while ( !( UCSRA & (1<<UDRE)) );

    /* Put data into buffer, sends the data */
    UDR = data;
    return 0;
}

```



## Software supervisorio:

```
/*Nome do arquivo: s1.c
  Autores: Pedro Ramos Mateus Filho
           Yuri Ferreira Gomes Dias
  Data: 21/05/2008
  Descrição: Software supervisorio de controle de temperatura em rede de
  automacao wireless. Ele coordena dois nós sensores e dois nós atuadores,
  processando valores de temperatura e realizando os dados do controlador
  liga-desliga. Este software foi utilizado no Trabalho de Graduação do
  autores.”

  Última modificação: 25/06/2008
*/

/*Bibliotecas incluídas no programa*/

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>

/*Funcoes criadas*/

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

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

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

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

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

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

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

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

/*Imprime cabecalho*/
void cabecalho();

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

int main(){

/*Variáveis utilizadas*/
FILE *Serial;
FILE *Dados;
unsigned char c;
  unsigned char *string=malloc(sizeof(char)*300);
  float temp1=0,temp2=0,temp3=0,t_ref=0,t_media=0;
```

```

int canal = 0,duracao = 0,i = 0,tamanho=0;
long int hora = 0;
char status_atuador1,status_atuador2;

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

/*Abre porta serial*/
Serial=fopen("COM5", "r+");
    if (Serial==NULL)
        {
            printf(" Falhou!");
        }
    else
        {
            printf(" Concluido!!\n\n");
        }

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

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

/*Solicita o valor da referencia e imprime cabecalho*/
printf("DIGITE O VALOR DA REFERENCIA DESEJADA:");
scanf("%f",&t_ref);
system("cls");
cabecalho();
limpa_Serial(Serial);

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

/*LOOP que coordena a chamada das funcoes de comunicacao com os modulos e
grava os dados de controle em arquivo*/
while (duracao < 100){

    printf("\n\nREFERENCIA = %0.2f",t_ref);
    printf("\nDURACAO = %d",duracao);

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

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

```

```

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

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

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

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

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

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

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

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

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

        /*Dependendo do valor selecionado, consulta sobre a
alteracao de
referencia*/
        if (duracao%50 == 0){
            printf("\n\nDESEJA ALTERAR A REFERENCIA? s/n: ");
            c = getchar();
            if (c == 's'){
                printf("\nDIGITE O VALOR DA REFERENCIA DESEJADA:");
                scanf("%f",&t_ref);
            }
        }
    }

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

/*Funcao que convete hexadecimal para inteiro: por Fellipe Lopes Couto*/
int ConvHEXDEC(char *Hexa)
{
    int k;

```

```

    k=strlen(Hexa);
    return (k);
}

/*Funcao que altera endereco para comunicao com atuador1. Ela entra no
modo de comando do XBee corrdenador e seta ATMY = 199 e ATDL = 202 e
apos encerra modo de comando. O retorno lido do XBee é recebido e pode
ser usado para debugar o programa.*/
void altera_atmy_atuador1(FILE *Serial){

    int j = 0;
    unsigned char ch;
    unsigned char str[20];
    limpa_Serial(Serial);
    fprintf(Serial, "+++");
    limpa_Serial(Serial);
    sleep(1500);

    ch = fgetc(Serial);

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

    limpa_Serial(Serial);
    fprintf(Serial, "ATMY199");
    fprintf(Serial, "\r");
    limpa_Serial(Serial);
    sleep(1000);
    ch = fgetc(Serial);

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

    limpa_Serial(Serial);
    fprintf(Serial, "ATDL202");
    fprintf(Serial, "\r");
    limpa_Serial(Serial);
    sleep(1000);
    ch = fgetc(Serial);

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

    limpa_Serial(Serial);
    fprintf(Serial, "ATCN");
    fprintf(Serial, "\r");
    sleep(1000);
}

```

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

```

void altera_atmy_atuador2(FILE *Serial){

    int j = 0;
    unsigned char ch;
    unsigned char str[20];
    limpa_Serial(Serial);
    fprintf(Serial, "+++");
    limpa_Serial(Serial);
    sleep(1500);

    ch = fgetc(Serial);

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

    limpa_Serial(Serial);
    fprintf(Serial, "ATMY198"); /*198*/
    fprintf(Serial, "\r");
    limpa_Serial(Serial);
    sleep(1000);
    ch = fgetc(Serial);

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

    limpa_Serial(Serial);
    fprintf(Serial, "ATDL204"); /*204*/
    fprintf(Serial, "\r");
    limpa_Serial(Serial);
    sleep(1000);
    ch = fgetc(Serial);

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

    limpa_Serial(Serial);
    fprintf(Serial, "ATCN");
    fprintf(Serial, "\r");
    sleep(1000);
}

```

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

```

void altera_atmy_remotel(FILE *Serial){

```

```

int j = 0;
unsigned char ch;
unsigned char str[20];
limpa_Serial(Serial);
sleep(2000);
limpa_Serial(Serial);
fprintf(Serial, "+++");
limpa_Serial(Serial);
sleep(1500);

ch = fgetc(Serial);

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

limpa_Serial(Serial);
fprintf(Serial, "ATMY200");
fprintf(Serial, "\r");
limpa_Serial(Serial);
sleep(1000);
ch= fgetc(Serial);

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

limpa_Serial(Serial);
fprintf(Serial, "ATCN");
fprintf(Serial, "\r");
sleep(1000);
}

/*Funcao que altera endereco para comunicacao com sensor2. Ela entra no
modo de comando do XBee corrdenador e seta ATMY = 203 e apos encerra modo
de comando. O retorno lido do XBee é recebido e pode ser usado para debugar
o programa.*/
void altera_atmy_remote2(FILE *Serial){

int j = 0;
unsigned char ch;
unsigned char str[20];
limpa_Serial(Serial);
sleep(2000);
limpa_Serial(Serial);
fprintf(Serial, "+++");
limpa_Serial(Serial);
sleep(1500);

ch = fgetc(Serial);

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

```

```

    /*printf("\n%s",str);*/

    limpa_Serial(Serial);
    fprintf(Serial,"ATMY203");
    fprintf(Serial, "\r");
    limpa_Serial(Serial);
    sleep(1000);
    ch = fgetc(Serial);

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

    limpa_Serial(Serial);
    fprintf(Serial,"ATCN");
    fprintf(Serial, "\r");
    sleep(1000);
}

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

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

    sleep(50);
    limpa_Serial(Serial);

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

    limpa_Serial(Serial);
    altera_atmy_remote2(Serial);
}

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

    int endereco;
    float temp_media = 0;
    char atuador;
    sleep(1000);
    limpa_Serial(Serial);
    c = fgetc(Serial);

    /*Aguarda até a chegada do FRAME API, que identificado pelo HEXA 7E*/
    while (c != 0x7E){
        limpa_Serial(Serial);
        c = fgetc(Serial);
    }

    if (c == 0x7E){

```

```

/*As linhas abaixo retira do frame apenas os dados que tem interesse,
endereco e temperatura*/
temp1=0,temp2=0,temp3=0;
sleep(100);
c = fgetc(Serial);
sleep(100);
c = fgetc(Serial);
ConvHEXDEC(&c);
sleep(100);
c = fgetc(Serial);
sleep(100);
c = fgetc(Serial);
ConvHEXDEC(&c);
endereco = c*100;
sleep(100);
c = fgetc(Serial);
ConvHEXDEC(&c);
endereco += c;
printf("\n\nEndereco do sensor: %d",endereco);
sleep(100);
c = fgetc(Serial);
sleep(100);
c = fgetc(Serial);
sleep(100);
c = fgetc(Serial);
sleep(100);
c = fgetc(Serial);
ConvHEXDEC(&c);
sleep(100);
c = fgetc(Serial);
sleep(100);
c = fgetc(Serial);
ConvHEXDEC(&c);
temp1 = c*1000;
sleep(100);
c = fgetc(Serial);
ConvHEXDEC(&c);
temp1 = c + temp1;
temp1 = temp1/3.28;
/*Imprime amostral*/
printf("    T1 = %0.2f",temp1);

sleep(100);
c = fgetc(Serial);
ConvHEXDEC(&c);
temp2 = c*1000;
sleep(100);
c = fgetc(Serial);
ConvHEXDEC(&c);
temp2 = c + temp2;
temp2 = temp2/3.28;
/*Imprime amostra2*/
printf("    T2 = %0.2f",temp2);

sleep(100);
c = fgetc(Serial);
ConvHEXDEC(&c);
temp3 = c*1000;
sleep(100);
c = fgetc(Serial);
ConvHEXDEC(&c);
temp3 = c + temp3;
temp3 = temp3/3.28;
/*Imprime amostra2*/

```



```

        printf("    T3 = %0.2f",temp3);

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

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

        }else{

            if (fabs(temp1-temp2)<1) temp_media =
((temp1+temp2)/2);
            if (fabs(temp3-temp2)<1) temp_media =
((temp3+temp2)/2);
            if (fabs(temp1-temp3)<1) temp_media =
((temp1+temp3)/2);
            if (fabs(temp1-temp2)>1 && fabs(temp3-temp2)>1 &&
fabs(temp1-temp3)>1) temp_media = ((temp1+temp2+temp3)/3);

        }

        /*Imprime status do atuador do presente setor*/
        if (calculo_ON_OFF(t_ref,temp_media,status_atuador)=='H')
printf("    ATUADOR = ON");
        else printf("    ATUADOR = OFF");
        sleep(100);
        c = fgetc(Serial);

    }

    return temp_media;
}

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

}

/*Funcao que imprime o cabecalho na tela*/
void cabecalho(){

    printf
("*****
");
    printf ("\n\n          ACIONAMENTO DE POTENCIA EM REDE DE AUTOMACAO
WIRELESS          ");
    printf
("\n\n*****
****");
    printf ("\n\n          PARAMETROS DE CONTROLE
          ");

}

/*Fucao que realiza os calculos do controlador liga-desliga e retorna o
status do atuador*/
char calculo_ON_OFF (float t_ref, float t_media, char status_atuador){

    if (t_ref > t_media && (t_ref - t_media)>0.5 ){

```

```
        return 'L';
    }

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

    return status_atuador;
}
```

### III. DESCRIÇÃO DO CONTEÚDO DO CD

O CD contém 6 pastas e um arquivo do relatório de projeto final em formato .pdf. As pastas estão divididas em: Apresentação, Softwares, Esquemáticos, Figuras do Trabalho, Arquivos do VISIO2007 e Datasheets. Os conteúdos destas pastas estão dispostos da seguinte forma:

- Apresentação
  - Apresentação Final do PG: Apresentação do dia 04 de Julho de 2008;
- Softwares
  - arv.lib: Biblioteca adicional contendo os componentes ATmega8 e ATmega128;
  - Firmware.c: Programa a ser gravado no microcontrolador;
  - Makefile: Arquivo de configuração para compilação;
  - s1.c: Programa supervisor em linguagem C;
  - s1.exe: Arquivo executável do supervisor;
  - Driver\_CON\_USBBEE.zip: Driver para utilização da placa CON USBBEE;
- Esquemáticos: contém os esquemáticos dos circuitos eletrônicos
- Figuras do Trabalho: contém as figuras utilizadas no trabalho e fotos dos módulos
- Arquivos do VISIO
  - Lavsi.vsd: Modelagem do Laboratório LAVSI;
  - Lavsi-Invertido: Modelagem de outro ângulo do Laboratório LAVSI;
- Datasheets: Alguns manuais utilizados para a construção dos circuitos eletrônicos.