

**FACULDADES DOCTUM DE CARATINGA
CURSO SUPERIOR DE ENGENHARIA ELÉTRICA**

**CONTROLE INTELIGENTE PARA MÁQUINAS ROTATIVAS ELÉTRICAS E
ILUMINAÇÃO**

GILMAR CORREA MORAES

Caratinga/MG

2017

GILMAR CORREA MORAES

**CONTROLE INTELIGENTE PARA MÁQUINAS ROTATIVAS ELÉTRICAS E
ILUMINAÇÃO**

Trabalho de Conclusão de Curso apresentado à Banca Examinadora do Curso Superior de Engenharia Elétrica das Faculdades DOCTUM de Caratinga como requisito parcial para obtenção do Grau de Bacharel em Engenharia Elétrica.
Professor Orientador: MSc. Robson Silva.

Caratinga/MG

2017

TERMO DE APROVAÇÃO

O Trabalho de Conclusão de Curso intitulado: CONTROLE INTELIGENTE PARA MÁQUINAS ROTATIVAS ELÉTRICAS E ILUMINAÇÃO, elaborado pelo(s) aluno(s) GILMAR CORREA MORAES foi aprovado por todos os membros da Banca Examinadora e aceito pelo curso de ENGENHARIA ELÉTRICA das FACULDADES DOCTUM CARATINGA, como requisito parcial da obtenção do título de

BACHAREL EM ENGENHARIA ELÉTRICA.

Caratinga 15 de Dezembro de 2017



Robson Silva

Prof. Orientador



Guilherme Casimiro

Prof. Avaliador 1



Reginaldo Eustáquio

Prof. Examinador 2

À Deus por todas as bênçãos desta vida.....

AGRADECIMENTOS

Agradeço sempre a Deus, por me conceder sabedoria e inteligência e por possibilitar a oportunidade de chegar a esse momento.

Aos familiares que sempre me deram o apoio moral, meu filho Gilmar Correa Moraes Júnior e a minha filha Mikaelly Miranda Moraes. A minha mãe em memória.

À minha esposa, Micheli Miranda da Rocha, pelo apoio e paciência durante todo o período desse trabalho. Pela ajuda e incentivos, mesmo quando os estudos apertavam muito.

Ao meu irmão, Manoel Aquino Correa Moraes, além do apoio, ajudou em alguns momentos com o custeio das mensalidades.

Ao meu orientador, Robson Silva, por toda a orientação, não somente nesse trabalho, mas também em vários outros aspectos relacionados à minha trajetória acadêmica.

Aos amigos e colegas que se separam ao fim desta jornada, ao Wemerson Carvalhos dos Santos que fez a revisão de texto deste TCC. Pelo companheirismo e trabalhos na companhia de tantos futuros engenheiros.

Aos professores que ajudaram neste trabalho ou em toda a minha vida acadêmica, por me proporcionarem o aprendizado e por serem exemplos de profissionais.

“Que os vossos esforços desafiem as impossibilidades, lembrai-vos de que as grandes coisas do homem foram conquistadas do que parecia impossível”

(CHARLES CHAPLIN)

CORREA MORAES, Gilmar. **CONTROLE INTELIGENTE DE MÁQUINAS ROTATIVAS ELÉTRICAS E ILUMINAÇÃO**. Caratinga, 2017. Trabalho de Conclusão de Curso de Engenharia Elétrica. Faculdades DOCTUM de Caratinga, Caratinga, 2017.

RESUMO

Sistemas automatizados otimizam processos e facilitam as funções diárias. Esse trabalho teve como objetivo propor, construir e testar um protótipo de automação. Neste contexto foi construída uma central de comando automático de máquinas elétricas (CAdME) que, ao ser implementada junto ao comando de acionamento de lâmpadas e ventiladores de uma sala, passou a controlar de forma automática as cargas dos ventiladores e o acionamento do conjunto de iluminação.

Palavras-chave: Automação. Controle. Estatística. Economia de energia.

CORREA MORAES, Gilmar. **CONTROLE INTELIGENTE DE MÁQUINAS ROTATIVAS ELÉTRICAS E ILUMINAÇÃO EM INSTALAÇÕES PÚBLICAS**. Caratinga, 2017. Trabalho de Conclusão de Curso de Engenharia Elétrica. Faculdades DOCTUM de Caratinga, Caratinga, 2017.

ABSTRACT

Automated systems streamline processes and make everyday functions easier. This work aimed to propose, construct and test a prototype of automation. In this context, an automatic control unit for electric machines (CAdME) was built. When it was implemented together with the control of the activation of lamps and fans in a room, it automatically controlled the loads on fans and the activation of the lighting assembly .

Key-words: Automation. Control. Statistics. Energy saving.

LISTA DE ILUSTRAÇÕES

| | | |
|------------------|--|----|
| Figura 1 | – Sistema de controle de malha aberta | 20 |
| Figura 2 | – Sistema de controle de malha fechada | 20 |
| Figura 3 | – Controle da modulação por largura de pulsos | 21 |
| Figura 4 | – Funcionamento do projeto | 22 |
| Figura 5 | – Microcontrolador ATmega 328P | 23 |
| Figura 6 | – Sensor de presença PIR | 23 |
| Figura 7 | – Detalhe da irradiação do sensor PIR | 24 |
| Figura 8 | – Funcionamento detecção de passagem | 24 |
| Figura 9 | – Sensor de temperatura DS18S20 | 25 |
| Figura 10 | – Sensor de Corrente SCT013-100A | 25 |
| Figura 11 | – Estrutura simplificada de um relé | 26 |
| Figura 12 | – Modelo de relé comercial | 26 |
| Figura 13 | – Triac BT 136 | 27 |
| Figura 14 | – Funcionamento de uma IHM | 28 |
| Figura 15 | – Display de Cristal Líquido 16x2 | 28 |
| Figura 16 | – Circuito Integrado PCF8574 | 29 |
| Figura 17 | – IDE de programação | 30 |
| Figura 18 | – Processo de compilação | 31 |
| Figura 19 | – Proteus | 31 |
| Figura 20 | – Placa PCI finalizada | 34 |
| Figura 21 | – Pinagens do ATmega 328P | 34 |
| Figura 22 | – Placa PCI montada | 35 |
| Figura 23 | – Placa PCI pontos de solda | 35 |
| Figura 24 | – Controle de malha aberta | 36 |
| Figura 25 | – Controle de malha fechada | 36 |
| Figura 26 | – Fluxograma do algoritmo | 37 |
| Figura 27 | – Fluxograma do funcionamento dos ventiladores | 37 |
| Figura 28 | – Painel da IHM | 38 |
| Figura 29 | – Circuito da IHM | 39 |
| Figura 30 | – Resposta ao controle por triac | 39 |
| Figura 31 | – Circuito de controle de carga | 40 |
| Figura 32 | – Controle da carga | 40 |
| Figura 33 | – Circuito detector de passagem de zero | 41 |
| Figura 34 | – Esquema parcial identificando os relés | 41 |
| Figura 35 | – Protótipo instalado | 42 |
| Figura 36 | – Telas de acesso da IHM | 43 |

| | |
|---|----|
| Figura 37 – Supervisório criado para o protótipo | 44 |
| Figura 38 – Correção do condutor da PCB | 49 |
| Figura 39 – Tensão em um período de tempo | 50 |
| Figura 40 – Aferindo a carga com o alicate amperímetro | 51 |
| Figura 41 – Decibelímetro da ABC Apps | 53 |
| Figura 42 – Aquisição manual de dados | 54 |
| Figura 43 – Aquisição de dados por supervisório | 54 |

LISTA DE TABELAS

| | | |
|------------------|--|----|
| Tabela 1 | – Ligação dos pinos LCD | 29 |
| Tabela 2 | – Lista de abreviaturas | 39 |
| Tabela 3 | – Coleta de dados | 44 |
| Tabela 4 | – Levantamento elétrico da sala 307 | 45 |
| Tabela 5 | – Levantamento elétrico da sala 311 | 45 |
| Tabela 6 | – Primeiros ensaios | 46 |
| Tabela 7 | – Aquisição manual de dados | 46 |
| Tabela 8 | – Dimensionamento da largura da trilha | 48 |
| Tabela 9 | – Carga máxima - figura 40 A | 51 |
| Tabela 10 | – Carga mínima - figura 40 B | 51 |
| Tabela 11 | – Relação de carga entregue aos ventiladores | 52 |
| Tabela 12 | – Nível de saída associado a temperatura | 52 |
| Tabela 13 | – Tabela comparativa de temperatura | 56 |

LISTA DE ABREVIATURAS E SIGLAS

| | |
|--------|--|
| EEPROM | <i>Electrically-Erasable Programmable Read-Only Memory</i> |
| IHM | Interface Homem Máquina |
| IDE | <i>Integrated Development Environment</i> |
| I/O | <i>Input/Output</i> |
| LCD | <i>Liquid Crystal Display</i> |
| PCI | Placa de Circuito Impresso |
| PWM | <i>Pulse Whidth Modulation</i> |
| PIR | <i>Passive Infrared Sensor</i> |
| RISC | <i>Reduced Instruction Set Computer</i> |
| RAM | <i>Random Access Memory</i> |
| SCR | <i>Silicon Controlled Rectifier</i> |
| TRIAC | <i>Triode for Alternating Current</i> |

LISTA DE SÍMBOLOS

| | |
|--------------|---------------------------------------|
| A | Àmpere |
| $I(t)$ | Corrente em função do tempo |
| \cos | Cosseno |
| σ | Desvio padrão de uma amostra |
| $^{\circ}C$ | Graus Célsius |
| KB | Kilobyte |
| Θ | Letra grega Theta |
| $\bar{\chi}$ | Média aritmética |
| MHz | Megahertz |
| m^2 | Metro quadrado |
| m^3 | Metro cúbico |
| ms | Milissegundo |
| P | Potência |
| π | Pi |
| rpm | Rotação por minuto |
| Σ | Somatório |
| V_{rms} | Tensão quadrática Média |
| V | Volt |
| $V(t)$ | Volt em função do tempo |
| VA | Volt-Àmpere |
| ωt | Velocidade angular em função do tempo |
| S^2 | Variância de uma amostra |
| sen | Seno |
| W | Watts |

SUMÁRIO

| | |
|---|-----------|
| 1 INTRODUÇÃO | 17 |
| 2 FUNDAMENTAÇÃO TEÓRICA | 18 |
| 2.1 Corrente alternada | 18 |
| 2.2 Potência média absorvida | 19 |
| 2.3 Máquinas elétricas (Assíncrono de Indução) | 19 |
| 2.4 Sistemas de controle | 19 |
| 2.4.1 Sistema de controle de malha aberta | 19 |
| 2.4.2 Sistema de controle de malha fechada | 20 |
| 2.5 Controle da modulação por largura de pulsos (PWM) | 20 |
| 2.6 Estatística | 21 |
| 2.6.1 Média aritmética simples | 21 |
| 2.6.2 Variância e Desvio padrão | 21 |
| 2.7 Protótipo | 22 |
| 2.7.1 Microcontrolador ATmega 328P | 22 |
| 2.7.2 Sensor de presença (PIR) | 23 |
| 2.7.3 Sensor de temperatura (DS18S20) | 24 |
| 2.7.4 Sensor de corrente (SCT013-100A) | 25 |
| 2.7.5 Relé | 26 |
| 2.7.6 Triac | 27 |
| 2.8 Interface Homem Máquina (IHM) | 27 |
| 2.8.1 Display de Cristal Líquido (LCD) | 28 |
| 2.8.2 PCF8574 (Expansor de Portas) | 29 |
| 2.9 Ventilador | 29 |
| 2.9.1 Ventilador de parede | 30 |
| 2.10 Programação do Microcontrolador | 30 |
| 2.11 Esquema eletrônico e placa de circuito impresso | 31 |
| 2.11.1 ISIS - Esquema eletrônico | 32 |
| 2.11.2 ARES - PCI | 32 |
| 3 PROCEDIMENTOS METODOLÓGICOS | 33 |
| 3.1 Protótipo | 33 |
| 3.1.1 Placa de circuito impresso | 33 |
| 3.1.2 Microcontrolador | 34 |
| 3.1.3 Placa de Circuito Impresso (PCI) | 35 |
| 3.1.4 Sensores | 36 |
| 3.2 Controle de malhas | 36 |

| | |
|--|-----------|
| 3.3 programação | 36 |
| 3.4 Controle | 38 |
| 3.4.1 Dimmer | 39 |
| 3.4.2 Detector de passagem do zero | 40 |
| 3.4.3 Controle por relés | 41 |
| 3.5 Instalação do protótipo | 41 |
| 3.5.1 Calibração | 42 |
| 3.5.2 Tempo de desligamento | 42 |
| 3.5.3 Temperatura padrão | 42 |
| 3.6 Operação | 43 |
| 3.7 Supervisório e coleta de dados | 43 |
| 3.8 Preparação do ambiente de coleta de dados | 44 |
| 3.8.1 Levantamento de dados do controle | 44 |
| 3.9 Coleta de dados via protótipo | 45 |
| 3.9.1 Aquisição de dados de forma manual | 46 |
| 3.9.2 Aquisição de dados por supervisório | 46 |
| 4 RESULTADOS E DISCUSSÕES | 47 |
| 4.1 Comparações entre as salas 307 e 311 | 47 |
| 4.1.1 Tensão incompatível com protótipo | 47 |
| 4.1.2 Correção da carga suportada pelo protótipo | 47 |
| 4.2 Inclusão e substituição de sensores no protótipo | 49 |
| 4.2.1 Inclusão do sensor PIR | 49 |
| 4.2.2 Substituição do sensor de temperatura | 49 |
| 4.3 Média da tensão | 49 |
| 4.3.1 Análise de carga dos ventiladores | 50 |
| 4.3.2 Resultados da calibração | 52 |
| 4.3.3 Controle automático | 52 |
| 4.4 Dados da temperatura | 53 |
| 4.4.1 Aquisição de dados por supervisório | 54 |
| 4.4.2 Comparativo entre cargas usuais e controladas | 55 |
| 4.5 Comparativos de temperaturas | 55 |
| 5 CONCLUSÃO | 57 |
| 6 REFERÊNCIAS BIBLIOGRÁFICAS | 58 |
| APÊNDICE A Planta baixa da sala 307 | 60 |
| APÊNDICE B Planta baixa da sala 311 | 61 |
| APÊNDICE C Esquema eletrônico do protótipo | 62 |
| APÊNDICE D Configuração ATmega 328P | 63 |

| | |
|--|-----------|
| APÊNDICE E Email - Ventiladores Delta | 64 |
| APÊNDICE F Código fonte em C, utilizado no microcontrolador | 65 |

1 INTRODUÇÃO

Após a revolução industrial em 1820 as máquinas possibilitaram que a humanidade desse um grande passo rumo ao desenvolvimento. Outro fator importante para o desenvolvimento industrial reside nos processos automatizados (TEIXEIRA, 2004).

Sistemas automatizados otimizam produtos e processos com eficiência e rapidez. Henry Ford revolucionou a indústria criando, entre outros, a chamada "Linha de produção" (STALLINGS, 2003). Seus carros eram fabricados de forma mais rápida diminuindo custos e consequentemente o valor de seus produtos, ganhando notoriedade e reconhecimento.

Engenheiros de vários segmentos, inspirados por ideias como a de Ford, buscam aperfeiçoar conceitos, tecnologias ou algo que possa facilitar ou meios de produção. Tarefas como acionamentos e ajustes, que são feitos de forma repetitiva, podem ser substituídos por um único equipamento de controle.

Em um sistema automatizado são empregado processos automáticos que comandam e controlam os mecanismos para seu funcionamento, mediante o uso de técnicas computadorizadas ou mecânicas (OGATA, 1982). Ao automatizar um processo cria-se mecanismos que sejam capazes de serem eficientes e de controle simplificado. O trabalho é executado de maneira ágil e eficaz necessitando de pouca ou de nenhuma interferência humana.

Neste contexto esse trabalho tem como objetivo propor, construir e testar um protótipo que permite controlar de forma automática máquinas rotativas elétricas e iluminação de ambientes com grande fluxo e permanência de pessoas. Para o campo de teste desse trabalho foram selecionado as salas 307 e 311 do bloco C das Faculdades Doctum de Caratinga e o trabalho dividido em cinco capítulos sendo:

Capítulo um a introdução onde é apresentado uma noção histórica sobre processo de automação.

Capítulo dois refere-se a todo material teórico utilizado na confecção do protótipo e desse trabalho.

Capítulo três apresenta a ordem cronológica de desenvolvimento e construção do protótipo, incluindo o processo de aquisição de dados.

Capítulo quatro se refere a apresentação e discussão dos principais dados que norteiam esse trabalho.

Capítulo cinco apresenta as principais conclusões com base na análise dos resultados.

2 FUNDAMENTAÇÃO TEÓRICA

Este estudo alinha-se a pesquisa experimental e bibliográfica, tendo como base livros, consultas na internet e análise de resultados obtidos por equipamentos de medição. Um protótipo foi desenvolvido para o auxílio e complemento como a parte de automação e coleta de dados.

Além da pesquisa bibliográfica, ensaios foram realizados em sala de aula durante os dias letivos. Neste caso, além do protótipo, utilizou-se também equipamentos homologados para aferir os sensores utilizados e medir o consumo.

Para melhor organização este estudo foi dividido em duas etapas:

- 1) – Construção do protótipo, de essencial importância para fundamentar este estudo.
- 2) – Levantamento bibliográfico, aquisição de dados, apuração e a conclusão.

Sobre a potência absorvida, não será levado em conta fatores inerentes aos estudos, como o fator de potência e os tipos de potências. O estudo analisará somente a carga instalada previamente, levantada e calculada no ambiente de estudo.

2.1 CORRENTE ALTERNADA

Segundo Halliday (2008) a corrente elétrica é o movimento ordenado das cargas elétricas em um condutor. Estas cargas são aceleradas pela diferença de potencial a que estão submetidas. As correntes elétricas alternadas em geral obtidas variam de forma senoidal com o tempo.

A forma de onda de tensão em CA pode ser descrita matematicamente pela fórmula:

$$v(t) = V_{max} \text{sen}(2\pi ft) \quad (2.1)$$

A forma de onda de corrente em CA pode ser descrita matematicamente pela fórmula:

$$i(t) = I_{max} \text{sen}(2\pi ft) \quad (2.2)$$

O Valor Eficaz RMS (*Root Mean Square*) de uma função periódica é definido como a raiz quadrada do valor médio da função ao quadrado (HALLIDAY, 2008), ou em forma algébrica:

$$V_{ef} = \frac{V_m}{\sqrt{2}} \quad (2.3)$$

2.2 POTÊNCIA MÉDIA ABSORVIDA

Segundo O'Malley (1983), a potência média consumida por um circuito CA de dois terminais pode ser obtida a partir da potência instantânea absorvida, expressada na equação 2.4.

$$P = VI.\cos\Theta - VI.\cos(2\omega t + \Theta) \quad (2.4)$$

O valor médio dessa potência é a soma dos valores médios dos dois termos. O segundo termo, sendo senoidal, possui um valor zero em um período. O primeiro termo, no entanto, é uma constante e deve ser a potência média absorvida pelo circuito em um período (O'MALLEY, 1983). A potência média é simplificada na equação 2.5.

$$P = VI.\cos\Theta \quad (2.5)$$

2.3 MÁQUINAS ELÉTRICAS (ASSÍNCRONO DE INDUÇÃO)

Segundo Fitzgerald (1975), motor assíncrono de indução é um motor elétrico de corrente trifásica, bifásica, ou monofásica, cujo rotor está excitado pelo estator e a velocidade de rotação não é proporcional à frequência da sua alimentação (a velocidade do rotor é menor que a do campo girante, devido ao escorregamento). O rotor assíncrono pode ser de dois tipos: gaiola de esquilo ou bobinado.

Ainda Fitzgerald (1975), diz que no caso do rotor "gaiola de esquilo" a magnetização do núcleo se dá pelo escorregamento do campo magnético (o campo magnético gira em torno do eixo central do motor) fazendo com que circule uma tensão e uma corrente induzida através das barras da gaiola.

2.4 SISTEMAS DE CONTROLE

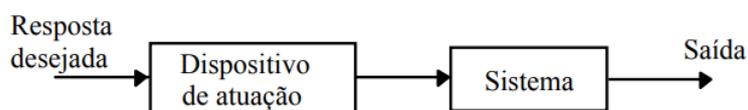
Ogata (1982) define sistema como a disposição, conjunto ou coleção de partes conectadas ou relacionadas de tal maneira a formarem um todo. Pode ser físico, biológico, econômico, etc.

Controle estuda como agir sobre um dado sistema de modo a obter um resultado arbitrariamente especificado. Controlador é o dispositivo utilizado para a obtenção do controle de um sistema. O sistema de Controle é um conjunto formado pelo sistema a ser controlado e o controlador.

2.4.1 Sistema de controle de malha aberta

O sistema de controle a malha aberta aquele em que a saída ou resposta não possui nenhuma influência sobre a entrada (OGATA,1982). A figura 1 mostra como o controle é realizado.

Figura 1: Sistema de controle de malha aberta

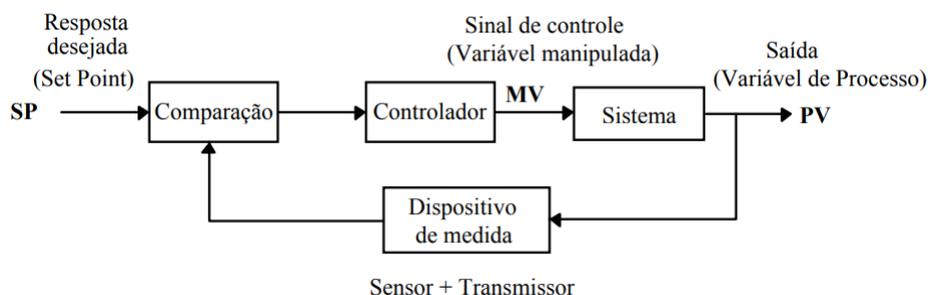


Fonte: (OGATA, 1982)

2.4.2 Sistema de controle de malha fechada

E ainda, o autor Ogata (1982) diz que o sistema de controle em malha fechada é aquele em que a saída ou resposta influencia na entrada do sistema. O sistema de malha fechada, pode ser observado na figura 2.

Figura 2: Sistema de controle de malha fechada



Fonte: (OGATA, 1982)

2.5 CONTROLE DA MODULAÇÃO POR LARGURA DE PULSOS (PWM)

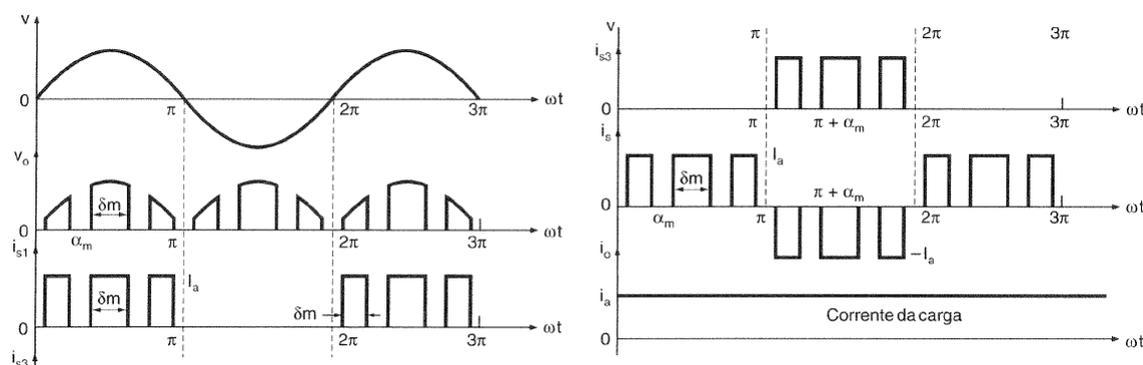
Rashid (1999) define que a modulação por largura de pulsos, o PWM (*Pulse Width Modulation*) são controles por chaveamento que ligam e desligam várias vezes durante um semiciclo e a tensão de saída é controlada pela variação da largura dos pulsos.

A tensão de saída e os parâmetros do conversor podem ser expressas na seguinte equação 2.6.

$$V_{medio} = \frac{V_m}{\pi} \sum_{m=1}^{\infty} [COS.\alpha m - COS(\alpha m - \delta m)] \quad (2.6)$$

A figura 3 mostra o estado de todas as entradas da equação 2.6.

Figura 3: Controle da modulação por largura de pulsos



Fonte: (RASHID, 1999)

2.6 ESTATÍSTICA

Estatística é um conjunto de métodos usados para se analisar dados (MONTGOMERY, 2012). Utiliza-se das teorias probabilísticas para explicar a frequência da ocorrência de eventos, tanto em estudos observacionais quanto em experimentos para modelar a aleatoriedade e a incerteza de forma a estimar ou possibilitar a previsão de fenômenos futuros.

2.6.1 Média aritmética simples

E ainda segundo Montgomery (2012), a média aritmética simples é obtida dividindo-se a soma das observações pelo número delas. É um quociente geralmente representado pelo símbolo \bar{x} . Uma série de "n" valores de uma variável "x", a média aritmética simples será determinada pela expressão 2.7.

$$\bar{x} = \frac{\sum X_i}{n} \quad (2.7)$$

2.6.2 Variância e Desvio padrão

A variância de uma variável aleatória ou processo estocástico é uma medida da sua dispersão estatística, indicando a diferença do valor esperado, como mostrado na expressão 2.8. A variância necessita das informações fornecidas por uma média aritmética.

$$S^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1} \quad (2.8)$$

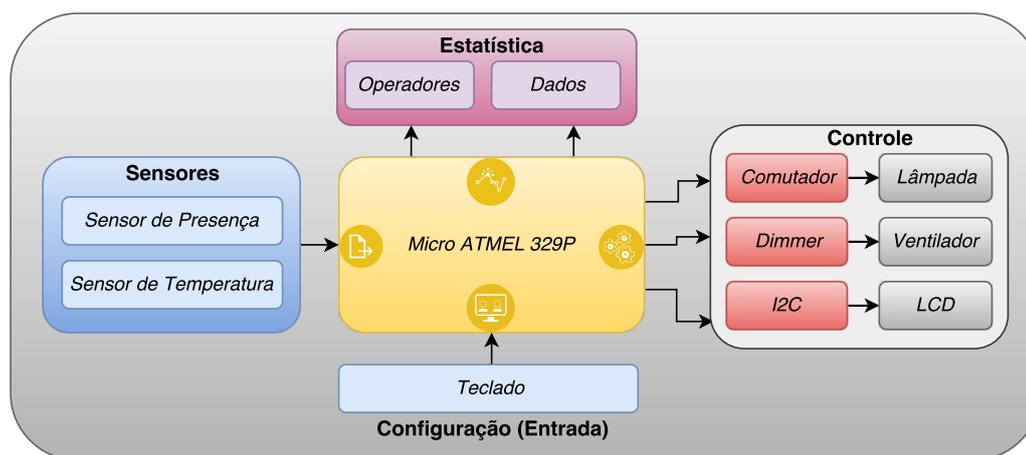
O desvio padrão encontra a variação dessa série em relação à média aritmética, similar ao que acontece com a variância, porém com mais exatidão. O desvio padrão é a raiz quadrada da variância, dado pela expressão 2.9.

$$\sigma = \sqrt{S^2} \quad (2.9)$$

2.7 PROTÓTIPO

A arquitetura computacional de Von Neumann segundo Stalins (2003), que é usada atualmente em computadores e sistemas microcontrolados, se aplica ao protótipo que foi usado nos ensaios. A figura 4 mostra como o projeto se comporta durante o funcionamento.

Figura 4: Funcionamento do projeto

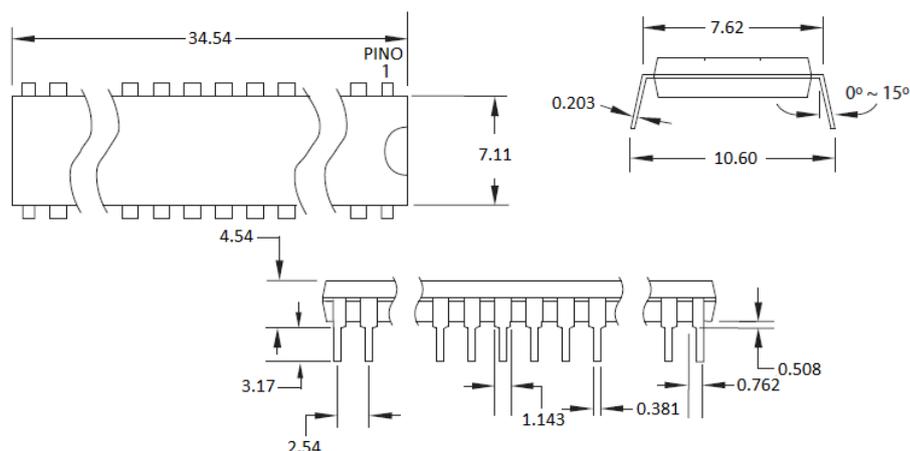


Fonte: Elaborada pelo autor

2.7.1 Microcontrolador ATmega 328P

Segundo a ATMEL, o microcontrolador ATmega328P é responsável pelas operações e cálculos. Trata-se de um dispositivo de 8 bits da família AVR, com arquitetura RISC (*Reduced Instruction Set Computer*) avançada. Ainda conta com 32 KB (*kilobyte*) de memória *flash* mas 512 Bytes são utilizados para o bootloader, 2 KB de RAM (*Random Access Memory*) e 1 KB de EEPROM (*Electrically-Erasable Programmable Read-Only Memory*). Possui 28 pinos, sendo que 23 desses podem ser utilizados como I/O (*Input/Output*). O nome AVR não é um acrônimo e não tem nenhum significado em especial. O encapsulamento e as medidas estão descritas na figura 5.

Figura 5: Microcontrolador ATmega 328P

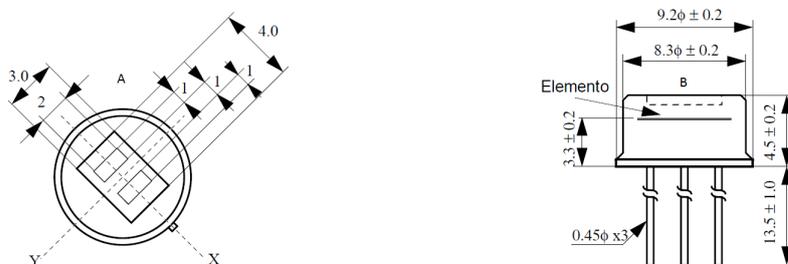


Fonte: Modificado do Datasheet ATmega 328P

2.7.2 Sensor de presença (PIR)

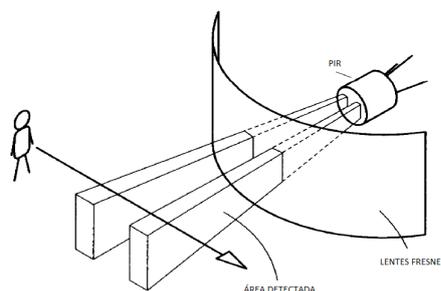
A fabricante Senba diz que o sensor PIR (*Passive textsl Infrared Sensor*), capta a variação térmica e é calibrado de acordo com a temperatura do corpo humano. A figura 6 (A) mostra as medidas do sensor visto de frente e a figura 6 (B) mostra o mesmo sensor, visto lateralmente.

Figura 6: Sensor de presença PIR



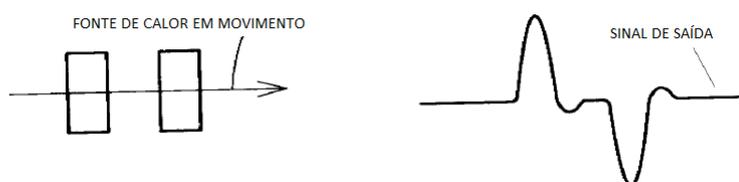
Fonte: Modificado do Datasheet Senba

Ainda segundo a Senba a temperatura do corpo humano provocará uma mudança repentina na luz infravermelha. Alguns modelos de sensores possui duas zonas específicas de leitura para detectar movimento em uma área maior. Neste caso, o sensor PIR é complementado por lentes especiais *fresnel*, que irradiam em várias direções, como mostrado na figura 7 .

Figura 7: Detalhe da irradiação do sensor PIR

Fonte: Modificado do Datasheet Senba

A Senba descreve que seus sensores PIR tem a particularidade de contar com dois (ou mais) detectores no seu interior, permitindo que mais zonas possam ser monitoradas, enquanto a outra serve de referência. A figura 8 mostra que quando o movimento passa para a outra zona, acontecerá o mesmo, de forma inversa.

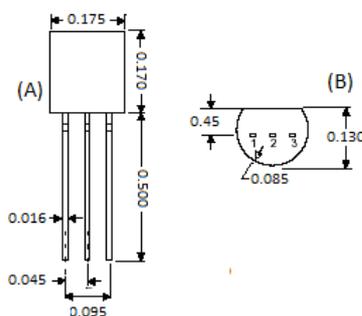
Figura 8: Funcionamento detecção de passagem

Fonte: Modificado do Datasheet Senba

2.7.3 Sensor de temperatura (DS18S20)

O DS18B20 da Maxim (2016), é um sensor de temperatura que opera na faixa de -55°C à $+125^{\circ}\text{C}$ e possui uma precisão de próximo aos 5°C ao longo do intervalo de -10°C a $+85^{\circ}\text{C}$, e não necessita de chips externos. Possui internamente um conversor A/D (Analógico/Digital), e uma memória na qual estão contidos os dados convertidos.

Este dispositivo leva no máximo 750ms para realizar uma conversão de temperatura e salvar no seu registrador correspondente. O encapsulamento do sensor é mostrado na figura 9.

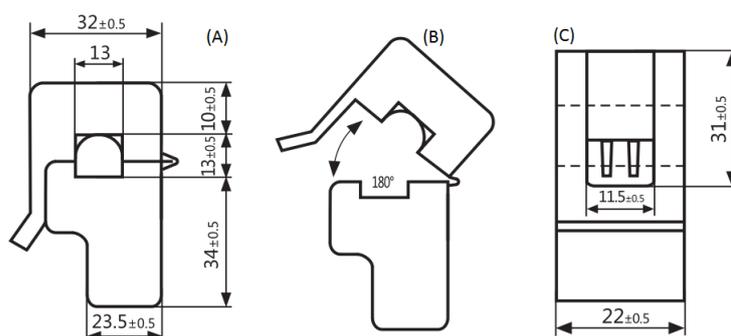
Figura 9: Sensor de temperatura DS18S20

Fonte: Modificado do Datasheet DS18B20 da Maxim Integrad

2.7.4 Sensor de corrente (SCT013-100A)

Segundo a YHDC (2016) o Sensor de corrente SCT013 é baseado em um transformador de corrente que pode transformar sinais CA de grande corrente em sinais de pequena amplitude. A corrente máxima que pode ser detectada pode atingir 100A (Âmpere) e o presente sinal de corrente pode ser lido por via analógica em uma porta I/O.

Ainda segundo YHDC (2016) o transformador de corrente é constituído de um enrolamento primário, um núcleo magnético e um enrolamento secundário. Este sensor possui uma trava manipulável, como mostrado na figura 10 (B). Esta abertura do núcleo permite colocar de forma não invasiva uma fase de um condutor. A figura 10 (A) mostra as medidas laterais e a figura 10 (C) mostra as medidas frontais.

Figura 10: Sensor de Corrente SCT013-100A

Fonte: Modificado do Datasheet SCT013-100A da YHDC

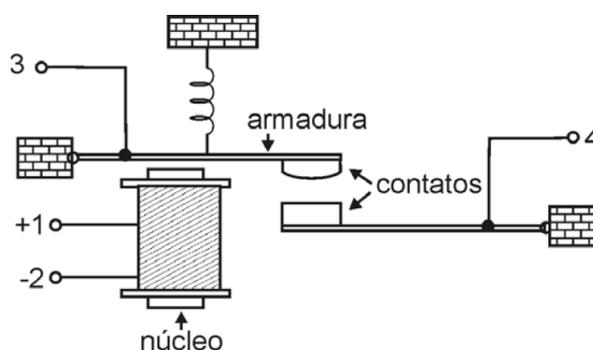
Segundo Fitzgerald (1975), a corrente alternada que flui no condutor do primário produz um campo magnético no núcleo que, em seguida, induz uma corrente no circuito do enrolamento secundário. Dessa forma a corrente no enrolamento secundário é proporcional à corrente que flui no enrolamento primário.

2.7.5 Relé

Segundo Braga (2012) o relé é um dispositivo eletromecânico com inúmeras aplicações possíveis em comutação de contatos elétricos, servindo para ligar ou desligar dispositivos.

Seu funcionamento consiste em um eletroímã posicionado em uma armadura móvel de metal ferroso que tem por finalidade controlar um jogo de contatos. Quando a bobina é percorrida por uma corrente elétrica um campo magnético é criado, atuando sobre a armadura, que provoca sua atração. Com esta atração, a armadura e consequentemente os contatos se movimentam, o que faz com que o contato móvel se encoste no contato fixo inferior (BRAGA, 2012). A estrutura simplificada é mostrada na figura 11.

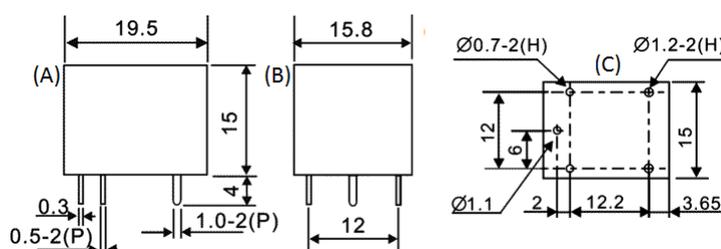
Figura 11: Estrutura simplificada de um relé



Fonte: (Braga, 2012)

Ainda segundo Braga (2012), existem vários modelos de relés e o que determina a utilização de um relé numa aplicação prática são suas características. O entendimento dessas características é fundamental para a escolha do tipo ideal, que possa atender as aplicações específicas nos quais foram desenvolvidos. A figura 12 (A), mostra as medidas laterais, a figura 12 (B), mostra as medidas frontais e a figura 12 (C), mostra as medidas dos terminais de soldagem.

Figura 12: Modelo de relé comercial



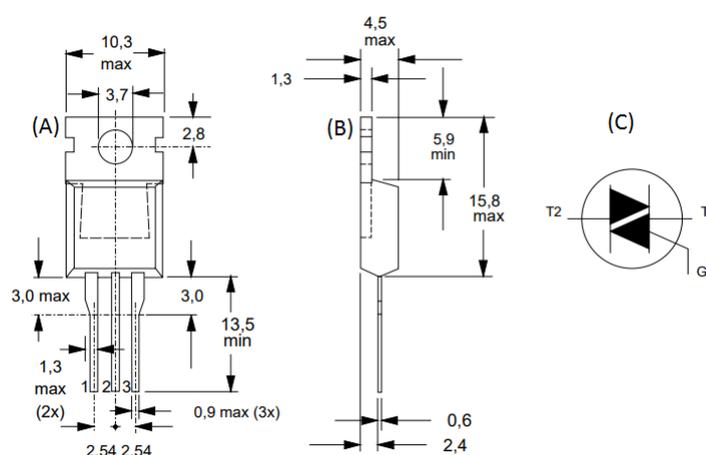
Fonte: Modificado do Datasheet relé da CPNET

2.7.6 Triac

Um TRIAC (*Triode for Alternating Current*) é um componente eletrônico equivalente a dois retificadores controlados ligados em antiparalelo com o terminal de disparo (ou gatilho) ligados juntos. Este tipo de ligação resulta em uma chave eletrônica bidirecional que pode conduzir a corrente elétrica nos dois sentidos (BOYLESTAD, 2005).

Segundo Boylestad (2005) o TRIAC, uma vez disparado, o dispositivo continua a conduzir até que a corrente elétrica caia abaixo do valor de corte, com o valor da tensão final da metade do ciclo de uma corrente alternada. Isto torna o TRIAC um conveniente dispositivo de controle para circuitos de corrente alternada, permitindo controlar grandes potências utilizando circuitos acionados por correntes da ordem de miliamperes. A figura 13 (A) mostra as medidas frontais, a figura 13 (B) mostra as medidas laterais e a figura 13 (C), mostra o símbolo de sua representação elétrica.

Figura 13: Triac BT 136



Fonte: Modificado do Datasheet BT136 da Philips Semiconductors

2.8 INTERFACE HOMEM MÁQUINA (IHM)

É a parte de um programa que se comunica com o usuário. Para Benyon (2011), o termo interface de usuário é definido como todas as partes de um sistema interativo (de *software* ou *hardware*) que fornecem informações e controle necessários para que o usuário realize uma determinada tarefa com o sistema interativo. A interface de usuário ou interface homem-máquina (IHM) é o ponto de ação no qual o ser humano está em contato com a máquina. A figura 14 mostra como o fluxo da informação é trafegada.

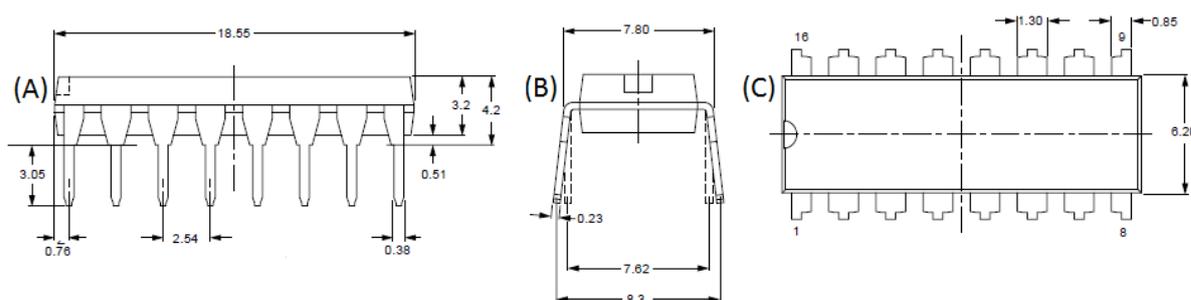
Tabela 1: Ligação dos pinos LCD

| Pontos de instalação do LCD 16x2 | | |
|----------------------------------|--------|---------------------|
| Pino LCD | Função | Ligação |
| 1 | VSS | GND |
| 2 | VDD | 5V |
| 3 | V0 | POTÊNCIOMETRO |
| 4 | RS | H - DADOS |
| 5 | RW | GND |
| 6 | E | HABILITAR SINAL |
| 7...14 | D0..D7 | BARRAMENTO DE DADOS |
| 15 | A | 5V |
| 16 | K | GND |

Fonte: Modificado do Datasheet LCD da Xiamen

2.8.2 PCF8574 (Expansor de Portas)

A NXP (2013) afirma que o PCF8574 é um expansor de portas que utiliza barramento I2C para fazer a comunicação. Em um LCD pode-se utilizar apenas duas saídas do microcontrolador expandindo-a para 10 portas. Os endereços dos barramentos vão de 0x20 a 0x27, operando a 5V. A figura 16 (A) mostra as medidas laterais, a figura 16 (B) mostra as medidas frontais e a figura 16 (C) mostra a medida superior.

Figura 16: Circuito Integrado PCF8574

Fonte: Modificado do Datasheet PCF8574 da NXP Semiconductors

2.9 VENTILADOR

Os ventiladores se enquadram em máquinas rotativas elétricas (FITZGERALD, 1975). Para definir o melhor equipamento para um determinado espaço é necessário um estudo que envolve dados como curvas de vazão, potência e a referência para qual a aplicação é destinada. Serão usadas as informações que o fabricante disponibiliza ao consumidor, por entender que todo o estudo já foi feito.

2.9.1 Ventilador de parede

Segundo a VENTI-DELTA os ventiladores de parede da linha Premium, fabricados por ela, são eficientes e com ventilação forte e potente, podendo ser usados em grandes espaços, dentro dos limites de suas especificações.

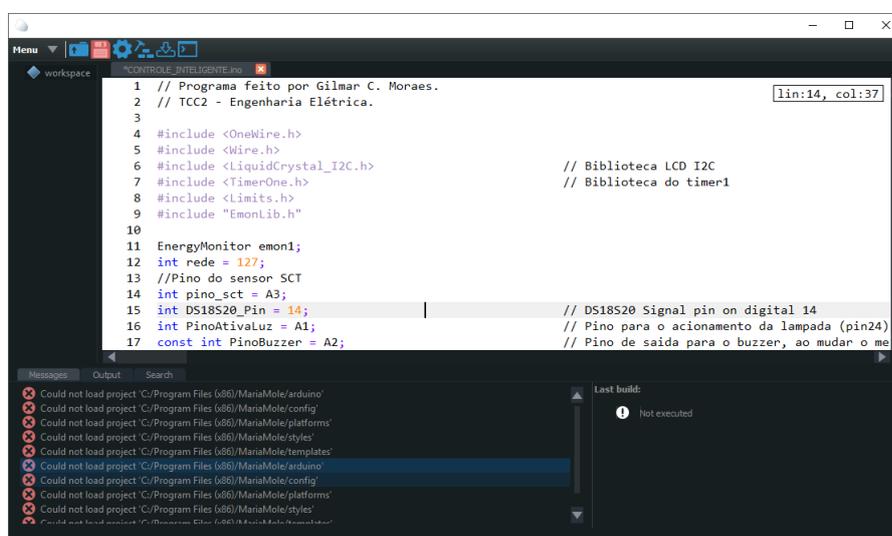
Especificações do ventilador de parede:

- Potência: 170W;
- Fator de Potência: 0,95;
- Rotação Máxima: 1.430 rpm;
- Hélice: 3 pás;
- Área de Ventilação: $40m^2$;
- Vazão: $230m^3/min$;
- Voltagem: Bivolt.

2.10 PROGRAMAÇÃO DO MICROCONTROLADOR

O microcontrolador precisa ser alimentado com o programa devidamente compilado em uma IDE (*Integrated Development Environment*) compatível (KLANDER, 1999). Neste caso a linguagem de programação usada será a C. Será usado uma interface de programação disponível em sistemas didáticos de estudo como mostrado na figura 17.

Figura 17: IDE de programação



Segundo Klander (1999), compilação é o processo de conversão de um código escrito normalmente em uma linguagem de alto nível, utilizado pelo programador (IDE) para uma linguagem de baixo nível (microcontrolador). A figura 18 ilustra como este processo é feito.

Figura 18: Processo de compilação

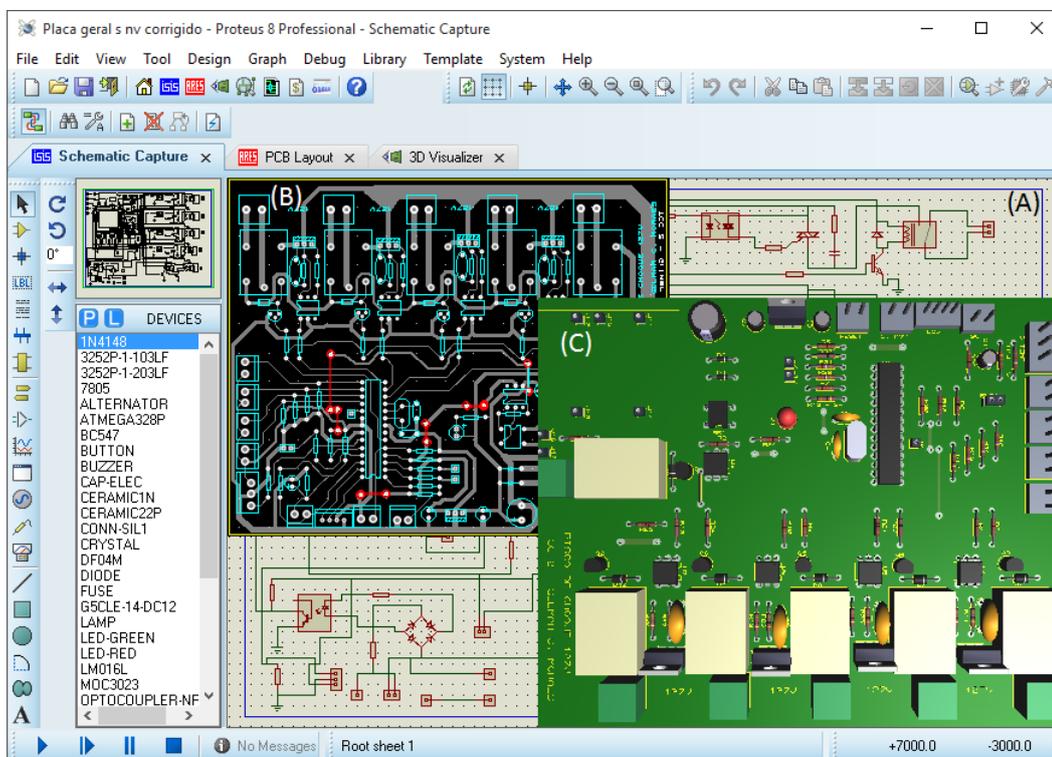


Fonte: Elaborado pelo autor

2.11 ESQUEMA ELETRÔNICO E PLACA DE CIRCUITO IMPRESSO

A LABCENTER, é a desenvolvedora do programa Proteus, é uma suíte que cria esquemas e simula circuitos eletrônicos e ainda é possível fazer o desenho para confeccionar placas de Circuito impresso (PCI). A figura 19, mostra a área de trabalho.

Figura 19: Proteus



Fonte: Elaborado pelo autor

2.11.1 ISIS - Esquema eletrônico

É necessário transcrever o experimento para um esquema eletrônico. A LABCENTER afirma que o esquema, pode ser feito de forma simples em um dos módulos chamado ISIS, nele existe uma vasta biblioteca de componentes.

E ainda, segundo a LABCENTER, o ISIS simula o funcionamento do circuito, podendo ainda, utilizar instrumentos virtuais como osciloscópio, voltímetros, amperímetros entre outros.

2.11.2 ARES - PCI

Segundo a LABCENTER é necessário que um esquema tenha sido feito no ISIS, para desenhar a PCI, e o módulo utilizado é o ARES.

3 PROCEDIMENTOS METODOLÓGICOS

Neste capítulo será descrito os componentes que compõe interface de controle de máquinas rotativas e iluminação, bem como o procedimento para obtenção dos dados que norteiam esse trabalho.

3.1 PROTÓTIPO

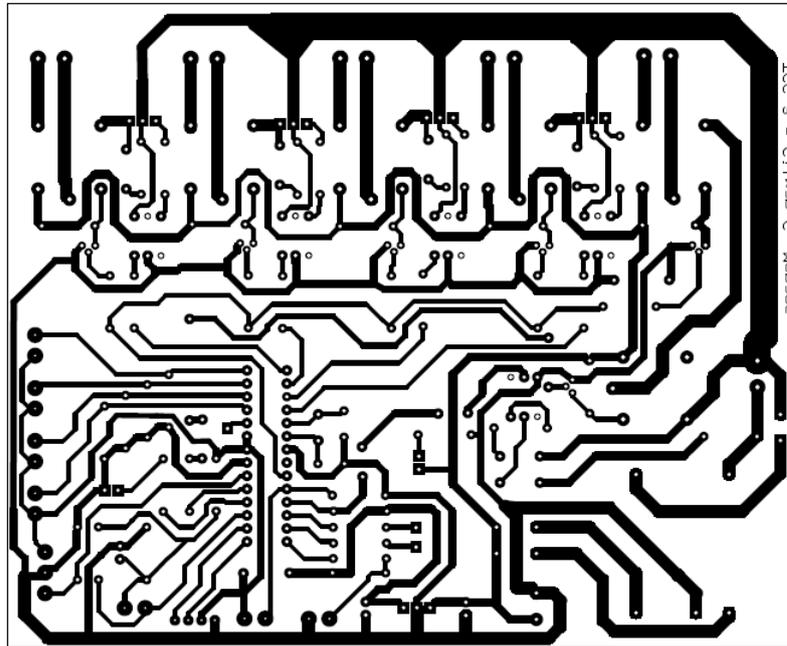
Para a confecção da pci (placa de circuito impresso) do protótipo utilizado nos ensaios, foi feito simulações no programa Proteus para obtenção dos parâmetros de funcionamento da interface de controle. Está é composta de placa de circuito impresso, microcontrolador, relés, transistores, reguladores de tensão, sensores, resistores, capacitores, tela de cristal líquido, chave sem retenção, fonte de alimentação, optacopladores e triacs. Os módulos dos principais componentes são descritos a seguir.

3.1.1 Placa de circuito impresso

Segundo Bomfin (2013), Oz é a espessura do cobre do laminado que cobre a placa (fenolite, fibra de vidro etc.).

A placa utilizada no protótipo é feita de fenolite, com uma cobertura de cobre de 1 oz, o tamanho da placa é 18cm (centímetros) de altura por 15cm de largura. O desenho da placa do circuito finalizado é apresentado na figura 20.

Figura 20: Placa PCI finalizada



Fonte: Elaborada pelo autor

3.1.2 Microcontrolador

Segundo a ATMEL, o microcontrolador ATmega 328P funciona a uma velocidade de 16MHz. Um cristal e dois capacitores são componentes auxiliares que compõe o oscilador externo. Na figura 21, são destacados os pinos 9 e 10, onde o cristal e capacitores são instalados.

Figura 21: Pinagens do ATmega 328P

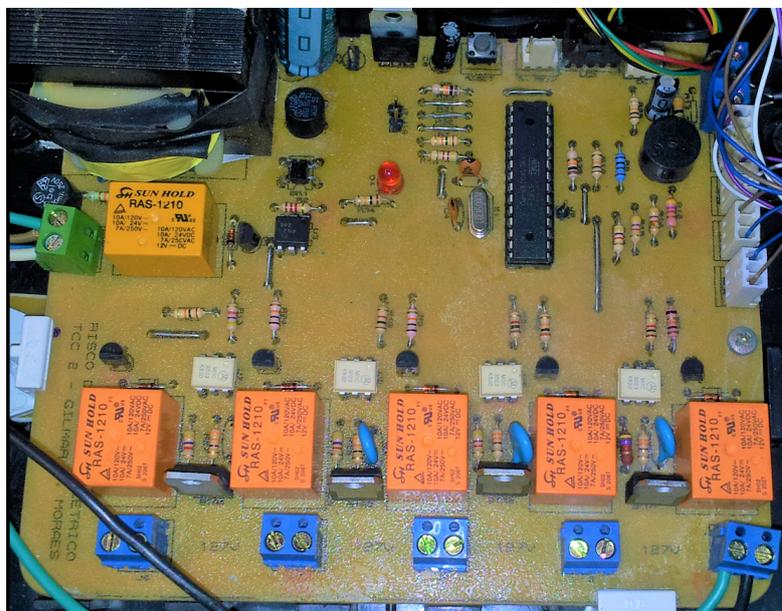
| | | | |
|--------------------------|----|----|------------------------|
| (PCINT14/RESET) PC6 | 1 | 28 | PC5 (ADC5/SCL/PCINT13) |
| (PCINT16/RXD) PD0 | 2 | 27 | PC4 (ADC4/SDA/PCINT12) |
| (PCINT17/TXD) PD1 | 3 | 26 | PC3 (ADC3/PCINT11) |
| (PCINT18/INT0) PD2 | 4 | 25 | PC2 (ADC2/PCINT10) |
| (PCINT19/OC2B/INT1) PD3 | 5 | 24 | PC1 (ADC1/PCINT9) |
| (PCINT20/XCK/T0) PD4 | 6 | 23 | PC0 (ADC0/PCINT8) |
| VCC | 7 | 22 | GND |
| GND | 8 | 21 | AREF |
| (PCINT6/XTAL1/TOSC1) PB6 | 9 | 20 | AVCC |
| (PCINT7/XTAL2/TOSC2) PB7 | 10 | 19 | PB5 (SCK/PCINT5) |
| (PCINT21/OC0B/T1) PD5 | 11 | 18 | PB4 (MISO/PCINT4) |
| (PCINT22/OC0A/AIN0) PD6 | 12 | 17 | PB3 (MOSI/OC2A/PCINT3) |
| (PCINT23/AIN1) PD7 | 13 | 16 | PB2 (SS/OC1B/PCINT2) |
| (PCINT0/CLKO/ICP1) PB0 | 14 | 15 | PB1 (OC1A/PCINT1) |

Fonte: Modificado do Datasheet ATmega 328P

3.1.3 Placa de Circuito Impresso (PCI)

A figura 22 mostra todos os componentes agrupado na pci.

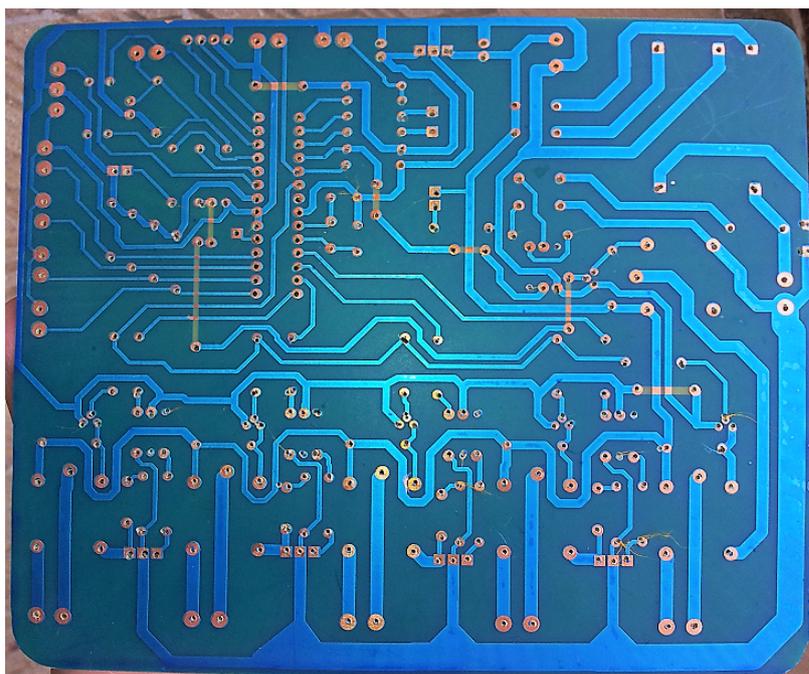
Figura 22: Placa PCI montada



Fonte: Elaborada pelo autor

Na figura 23 é mostrado os pontos de soldagem dos componentes eletrônico

Figura 23: Placa PCI pontos de solda



Fonte: Elaborada pelo autor

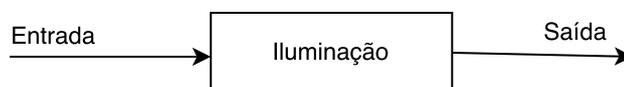
3.1.4 Sensores

Os sensores de monitoramento utilizado neste trabalho, estão descritos no capítulo 2, seção 2.7.2, seção 2.7.3 e seção 2.7.4.

3.2 CONTROLE DE MALHAS

Para o acionamento da iluminação não existem sensores instalados no protótipo, para determinar a luminância ou algo que faça realimentar a malha. Pode-se dizer que é um controle de malha aberta. Este controle é representado pela figura 24.

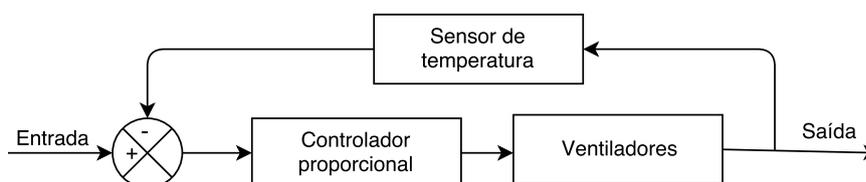
Figura 24: Controle de malha aberta



Fonte: Elaborada pelo autor

O controle dos ventiladores são feitos de forma proporcional, e este terá influência constante na velocidade dos motores. Verifica-se então o controle de malha fechada proporcional. Este controle é representado pela figura 25.

Figura 25: Controle de malha fechada



Fonte: Elaborada pelo autor

3.3 PROGRAMAÇÃO

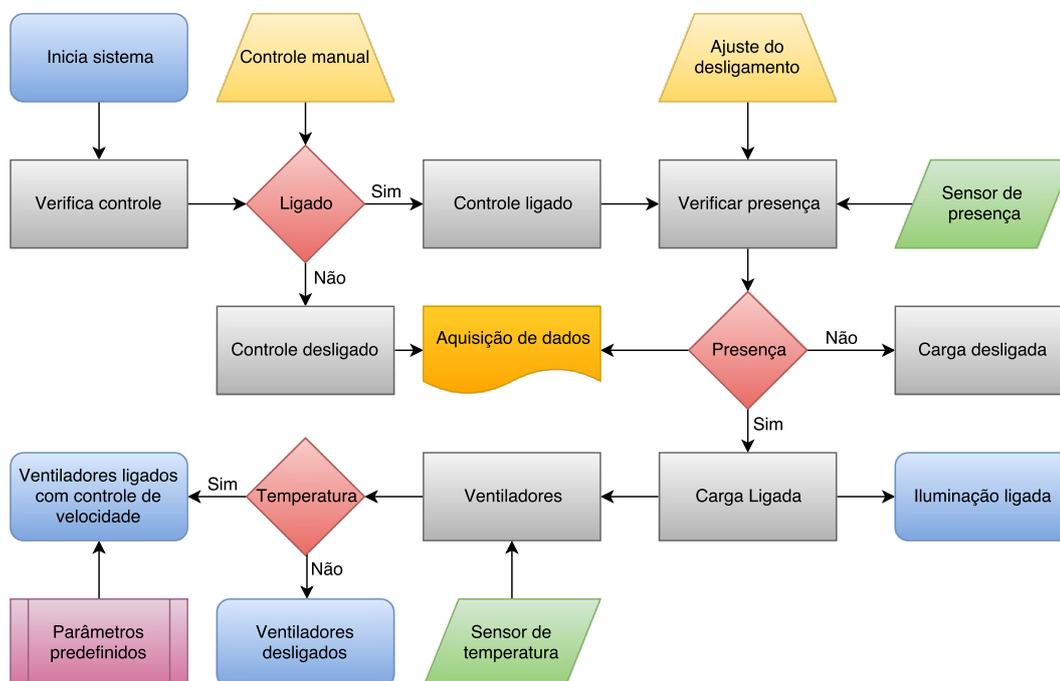
Para o funcionamento da interface de controle, foi necessário a criação de um algoritmo. O fluxograma é mostrado na figura 26. O código para o microcontrolador programado em C se encontra no Apêndice F.

Inicialmente o código verifica o estado do "CONTROLE". Estando "DESLIGADO" todas as automações no controle das cargas ficam desligadas. No estado "LIGADO", que é sua condição original, o programa segue uma rotina predefinida. Este estado pode ser alterado manualmente.

Outro ponto no fluxograma é a verificação de presença de pessoa(s) no ambiente de estudo. A verificação é feita pelo sensor PIR. Um contador interno vai decrementando um valor

inicial predefinido, até que a variável seja igual 0, nesta condição, toda a carga é desligada. A contagem regressiva sempre se reinicia se o sensor de presença mudar seu estado de verificação.

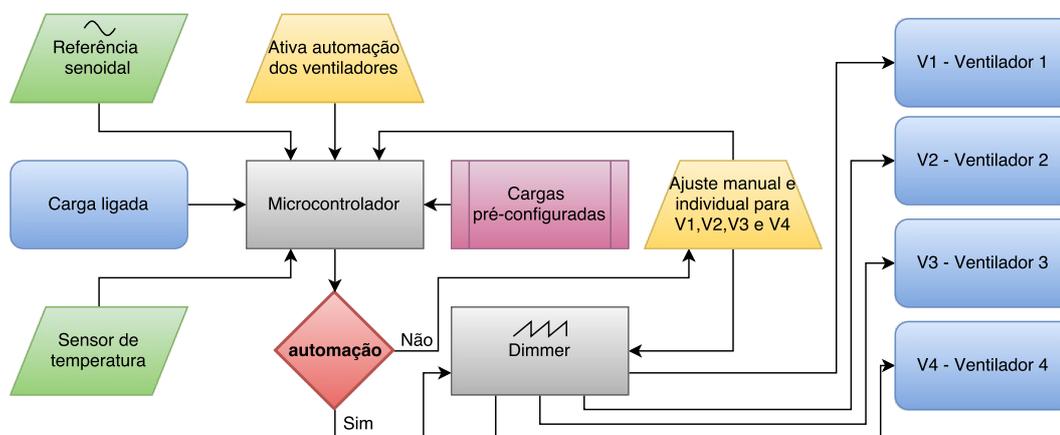
Figura 26: Fluxograma do algoritmo



Fonte: Elaborada pelo autor

A iluminação tem o acionamento feito de forma direta que é o ligado ou desligado. Os ventiladores são acionados de forma proporcional. O fluxograma da figura 27, mostra o esquema de funcionamento para os ventiladores.

Figura 27: Fluxograma do funcionamento dos ventiladores



Fonte: Elaborada pelo autor

O controle de cargas dos ventiladores é subordinado às verificações iniciais feitas pelo sensor de presença e pelo sensor de temperatura. Caso atendidas as condições iniciais, uma nova rotina é executada.

Rotina 1, Controle automático ativo:

Microcontrolador verifica inicialmente se o estado de controle automático dos ventiladores está ativo. Neste caso todo o controle da carga dos ventiladores respeitará as regras parametrizadas e associadas ao sensor de temperatura.

Rotina 2, Controle automático inativo:

Caso esteja desativado, o controle das cargas nos ventiladores é feito de forma manual e individual e não são consideradas as informações do sensor de temperatura.

3.4 CONTROLE

A figura 28 apresenta uma visão panorâmica da interface da IHM.

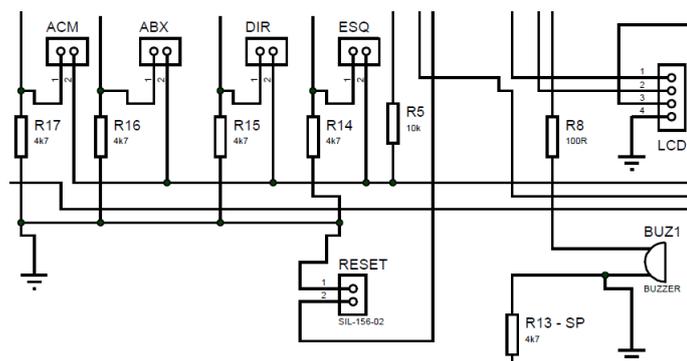
Figura 28: Painel da IHM



Fonte: Elaborada pelo autor

Na figura 29 mostra o diagrama esquemático da IHM e a tabela 6 descreve a lista de abreviaturas usadas no circuito.

Figura 29: Circuito da IHM



Fonte: Elaborada pelo autor

Tabela 2: Lista de abreviaturas

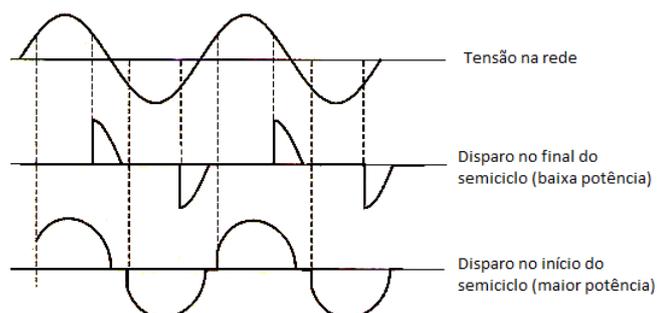
| ABREVIATURA | DESCRIÇÃO |
|-------------|-----------------|
| ACM | BOTÃO ACIMA |
| ABX | BOTÃO ABAIXO |
| DIR | BOTÃO DIREITO |
| ESQ | BOTÃO ESQUERDO |
| RST | BOTÃO REINICIAR |
| LCD | CONECTOR LCD |
| BUZZER | AUTO FALANTE |

Fonte: Elaborada pelo autor

3.4.1 Dimmer

O dimmer intercalado com a carga permite controlar a potência. Esse controle é feito variando-se o ponto de disparo do triac em cada semiciclo da alimentação de corrente alternada. Este disparo é controlado segundo regras definidas pelo microcontrolador. A figura 30 mostra o estado da onda senoidal em sua condição de mínima e máxima potência.

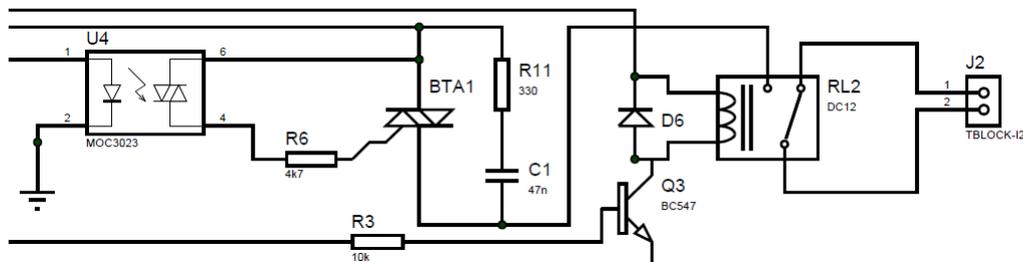
Figura 30: Resposta ao controle por triac



Fonte: Modificado de BRAGA, 2017

O circuito CA que comanda os ventiladores é isolado do circuito eletrônico de comando por componentes optoacopladores. O sinal vem do microcontrolador e alimenta o optoacoplador MOC 3023 que dispara o triac BTA1, O esquema eletrônico onde ocorre este procedimento é mostrado na figura 31.

Figura 31: Circuito de controle de carga

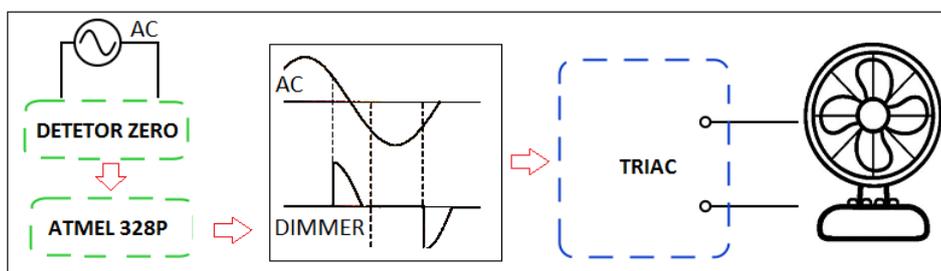


Fonte: Elaborada pelo autor

3.4.2 Detector de passagem do zero

O circuito identifica a passagem de zero da corrente alternada, definindo exatamente o momento do disparo do triac. O esquema de funcionamento é mostrado na figura 32.

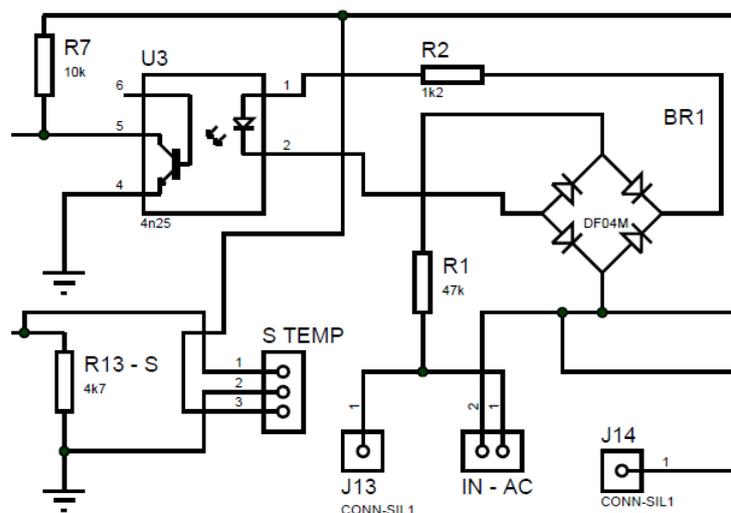
Figura 32: Controle da carga



Fonte: Elaborado pelo autor

O Circuito descrito na figura 33, mostra uma ponte retificadora definida por BR1 que retificará a tensão. O componente U3 é um opto acoplador que recebe a tensão retificada de BR1 e aciona U3, até que a onda senoidal da corrente alternada passe pelo ponto zero mudando o estado de leitura do microcontrolador.

Figura 33: Circuito detector de passagem de zero

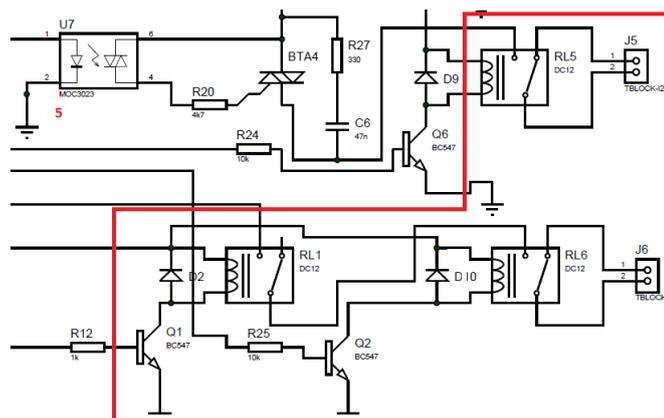


Fonte: Elaborado pelo autor

3.4.3 Controle por relés

O acionamento das lâmpadas e do *bypass*, são feitos utilizando relés. A figura 34 mostra o acionamento da iluminação feito pelo relé RL1. O *bypass* tem o controle feito pelos relés RL2, RL3, RL4, RL5 e RL6 (Apêndice C) e na figura 34, representado por RL5 e RL6.

Figura 34: Esquema parcial identificando os relés



Fonte: Elaborado pelo autor

3.5 INSTALAÇÃO DO PROTÓTIPO

O protótipo foi instalado na sala de 311 e pode ser visto na figura 35, afixado acima do conjunto de controles das cargas dos ventiladores e iluminação. O protótipo compartilha a

mesma malha de acionamento utilizada pelo conjunto de controles originais em uma ligação em série. Esse processo permite controlar as cargas de duas formas distintas, automático ou manual.

Figura 35: Protótipo instalado



Fonte: Elaborado pelo autor

3.5.1 Calibração

Para definir os parâmetros de controle, foi necessário criar uma correlação de tensão, corrente e nível de saída do protótipo, os resultados são apresentados no capítulo 4.

3.5.2 Tempo de desligamento

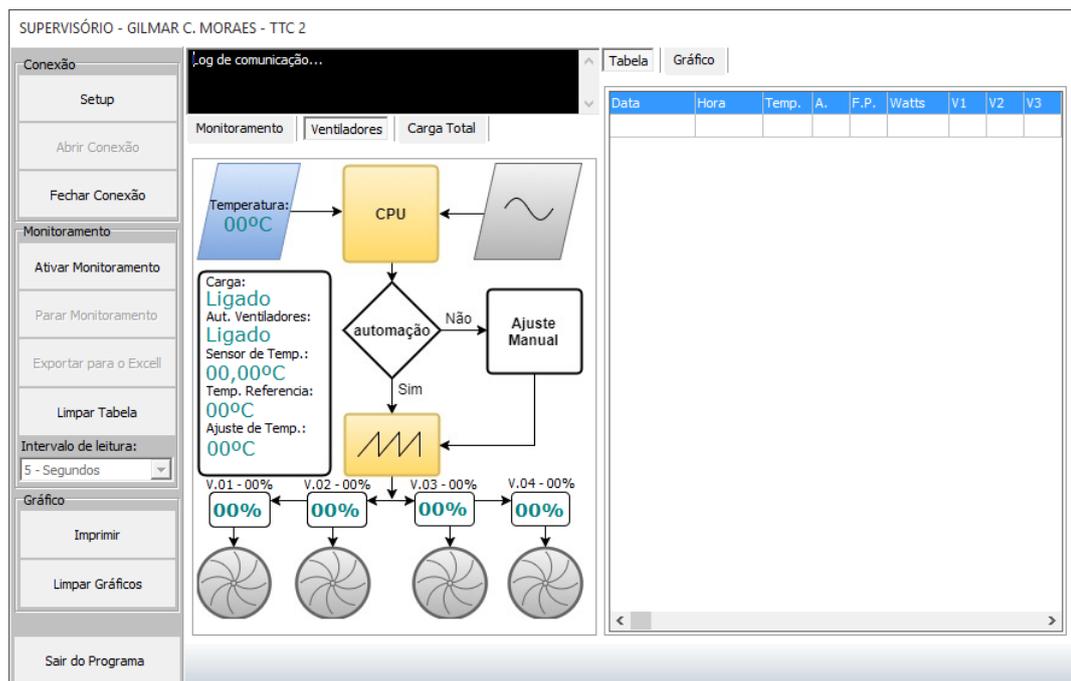
O protótipo possui um contador interno decrescente, que igualado a zero, desliga as cargas totalmente. O sensor PIR é o responsável por reiniciar o contador. O tempo definido como padrão para o desligamento, ficou em 240 segundos (4 minutos). Sempre que o sensor PIR, identifica movimento na sala, o contador retorna a variável ao valor de 240 segundos.

Pode-se alterar o valor de desligamento de forma temporária pela a IHM, ao reiniciar o sistema ou desliga-lo, as modificações realizadas são reiniciadas.

3.5.3 Temperatura padrão

A temperatura no ambiente é o que define a qual carga os ventiladores vão funcionar. Não é função protótipo controlar a temperatura no ambiente de estudo, e sim definir a melhor

Figura 37: Supervisório criado para o protótipo



Fonte: Elaborado pelo autor

3.8 PREPARAÇÃO DO AMBIENTE DE COLETA DE DADOS

Inicialmente foi necessário correlacionar os parâmetros de funcionamento do protótipo com as variáveis do ambiente. Começamos por homologar os sensores do protótipo (temperatura e corrente). Para aferir o sensor de temperatura, foi usando termômetro a base de mercúrio e um multímetro digital, modelo ET - 2082C. Para aferir o sensor de corrente foi usado um alicate amperímetro, modelo ET - 4055

3.8.1 Levantamento de dados do controle

Nesta etapa da pesquisa foi utilizado os dados colhidos no ambiente da sala 307 situada no bloco B, da Faculdades DOCTUM de Caratinga, entre os dias 11 e 18 de maio de 2017. A tabela 3 correlaciona a temperatura inicial e temperatura final juntamente com o número de ventiladores ligados.

Tabela 3: Coleta de dados

| DATA | TEMP. INICIAL | TEMP. FINAL | ESTADO DO VENTILADOR |
|------------|---------------|-------------|-----------------------|
| 11/05/2017 | 26°C | 29°C | 2 VENTILADORES ATIVOS |
| 15/05/2017 | 27°C | 29°C | 2 VENTILADORES ATIVOS |
| 16/05/2017 | 27°C | 28°C | 1 VENTILADOR ATIVO |
| 17/05/2017 | 25°C | 26°C | VENTILADOR DESLIGADO |
| 18/05/2017 | 24°C | 25°C | VENTILADOR DESLIGADO |

Fonte: Elaborado pelo autor

Para o levantamento de carga da sala 307, foi necessário a confecção da planta baixa (Apêndice A). O resultado é apresentado na tabela 4.

Tabela 4: Levantamento elétrico da sala 307

| CÓDIGO | DESCRIÇÃO | POTÊNCIA |
|--------|-------------------------|----------|
| V1 | VENTILADOR | 170W |
| V2 | VENTILADOR | 170W |
| L1 | 2X LÂMPADA FLUORESCENTE | 30W X 2 |
| L2 | 2X LÂMPADA FLUORESCENTE | 30W X 2 |
| L3 | 2X LÂMPADA FLUORESCENTE | 30W X 2 |
| L4 | 2X LÂMPADA FLUORESCENTE | 30W X 2 |
| | CARGA TOTAL | 580W |

Fonte: Elaborado pelo autor

O protótipo de automação foi instalado na sala 311. O levantamento de carga da sala 311 foi obtido por meio da planta baixa (Apêndice B). O resultado apresentado na tabela 5.

Tabela 5: Levantamento elétrico da sala 311

| CÓDIGO | DESCRIÇÃO | POTÊNCIA |
|--------|-------------------------|----------|
| V1 | VENTILADOR | 170W |
| V2 | VENTILADOR | 170W |
| V3 | VENTILADOR | 170W |
| L1 | 2X LÂMPADA FLUORESCENTE | 30W X 2 |
| L2 | 2X LÂMPADA FLUORESCENTE | 30W X 2 |
| L3 | 2X LÂMPADA FLUORESCENTE | 30W X 2 |
| L4 | 2X LÂMPADA FLUORESCENTE | 30W X 2 |
| L5 | 2X LÂMPADA FLUORESCENTE | 30W X 2 |
| L6 | 2X LÂMPADA FLUORESCENTE | 30W X 2 |
| L7 | 2X LÂMPADA FLUORESCENTE | 30W X 2 |
| L8 | 2X LÂMPADA FLUORESCENTE | 30W X 2 |
| | CARGA TOTAL | 990W |

Fonte: Elaborado pelo autor

3.9 COLETA DE DADOS VIA PROTÓTIPO

A coleta de dados ocorreu em momentos distintos:

3.9.1 Aquisição de dados de forma manual

O protótipo foi instalado no dia 03/10/2017 e o monitoramento de suas funções realizado pelo operador e se estendeu até o dia 25/10/2017. Nesta fase do experimento foi coletado diariamente a leitura de temperatura exibida no visor da interface, sempre ocorrendo do início ao final das aulas.

Os ensaios do período de 09/09/2017 a 16/10/2017 foram para ajustes, modificações e coletas de dados para a parametrização do protótipo. A tabela 6, discrimina as ocorrências e modificações neste período.

Tabela 6: Primeiros ensaios

| Testes com o protótipo em bypass | | |
|---|--|---------------------------------------|
| Data | Problema | Solução |
| 29/09/2017 | Instalação do protótipo, tensão incompatível | Troca de fonte |
| 03/10/2017 | Valor errado no sensor de temperatura | Reprogramação do microcontrolador |
| 04/10/2017 | Sensor PIR com pontos cegos | Ajuste de posição |
| 05/10/2017 | Sensor PIR com pontos cegos | Novo ajuste de posição |
| 06/10/2017 | Sensor PIR com pontos cegos | Desligamento de 60s para 240s |
| Teste normal | | |
| 09/10/2017 | Ventiladores com problema de desligamento | Estudar o código (feriado prolongado) |
| 16/10/2017 | Sensor PIR insuficiente | Adicionado outro sensor PIR |
| 16/10/2017 | Sensor de temperatura instável | Substituído por um eficiente |

Fonte: Elaborada pelo autor

A coleta de temperatura pelo protótipo ocorrida entre os dias 23 a 25 de outubro, estão representados na tabela 7.

Tabela 7: Aquisição manual de dados

| Data | Dado 1 | Dado 2 | Dado 3 | Dado 4 |
|------------|--------|--------|--------|--------|
| 23/10/2017 | 28°C | 29°C | 0 | 0 |
| 24/10/2017 | 26°C | 27°C | 27°C | 28°C |
| 25/10/2017 | 27°C | 28°C | 27°C | 27°C |

Os dados 3 e 4 do dia 23/10/2017 se refere ao dia da semana com apenas um período de aula.

3.9.2 Aquisição de dados por supervísório

A partir do dia 06/11/2017 a coleta dados passou a ser feita pelo supervísório.

Os dados obtidos nesta seção definiram os parâmetros de funcionamento do protótipo. Os resultados de controle e automação serão discutidos no capítulo 4.

4 RESULTADOS E DISCUSSÕES

Este capítulo será discutido os principais resultados obtidos pelo capítulo 3.

Inicialmente será relatado todos os atropelos ocorridos durante a instalação e o processo de coleta de dados.

4.1 COMPARAÇÕES ENTRE AS SALAS 307 E 311

O protótipo foi desenvolvido inicialmente para atender a carga instalada da sala 307, a troca do ambiente de estudo para a sala 311, ocorreu após a construção do equipamento, dois problemas foram encontrados e tiveram de ser corrigidos.

4.1.1 Tensão incompatível com protótipo

O protótipo foi desenvolvido inicialmente para trabalhar a uma tensão de 127V. Ao instalar o equipamento na sala 311 verificou-se que no ponto de instalação havia somente tensão de 220V. A solução para o problema, foi substituir a fonte por um modelo compatível. O nova fonte instalada possui entrada automática de tensão de 90V a 220V e saída de 12V com uma corrente de 5A.

4.1.2 Correção da carga suportada pelo protótipo

Para a automação das cargas, os acionamentos das lâmpadas e ventiladores são redirecionados para o protótipo. Sobre as cargas a Venti-Delta (Apêndice E) afirma que seu fator de potência é de 0.95 na carga máxima em 127V. A GE, fabricantes dos reatores e lâmpadas afirma que seu fator de potência é de 0.97 operando em 127V.

Para a carga da sala 307 temos: Para os ventiladores $P = 340W$, $FP = 0.95$ e a iluminação $P = 240W$, $FP = 0.97$ em uma tensão de 127V. Os cálculos é mostrado na equação 4.1.

$$I = \frac{P}{V.FP} \quad (4.1)$$

$$I_{vent.} = \frac{340W}{127V \times 0.95} = 2.82A$$

$$I_{ilum.} = \frac{240W}{127V \times 0.97} = 1.95A$$

$$I_{ilum.} + I_{vent.} = 4.77A$$

Por se tratar de equipamentos do mesmo fabricante serão consideradas as mesmas especificações para cálculo da carga em ambas as salas. Para a sala 311, os ventiladores $P = 510W$, $FP = 0.95$ e a iluminação $P = 480W$, $FP = 0.97$ em uma tensão de $127V$. O cálculo é mostrado na equação 4.2.

$$I = \frac{P}{V.FP} \quad (4.2)$$

$$I_{vent.} = \frac{510W}{127V \times 0.95} = 4.23A$$

$$I_{ilum.} = \frac{480W}{127V \times 0.97} = 3.89A$$

$$I_{ilum.} + I_{vent.} = 8.13A$$

Os relés utilizados no circuito suportam correntes de $10A$ em $127V$. Os triacs suportam corrente de $12A$ com tensão máxima no coletor de $600V$. como as cargas estão divididas, os componentes estão dentro das especificações.

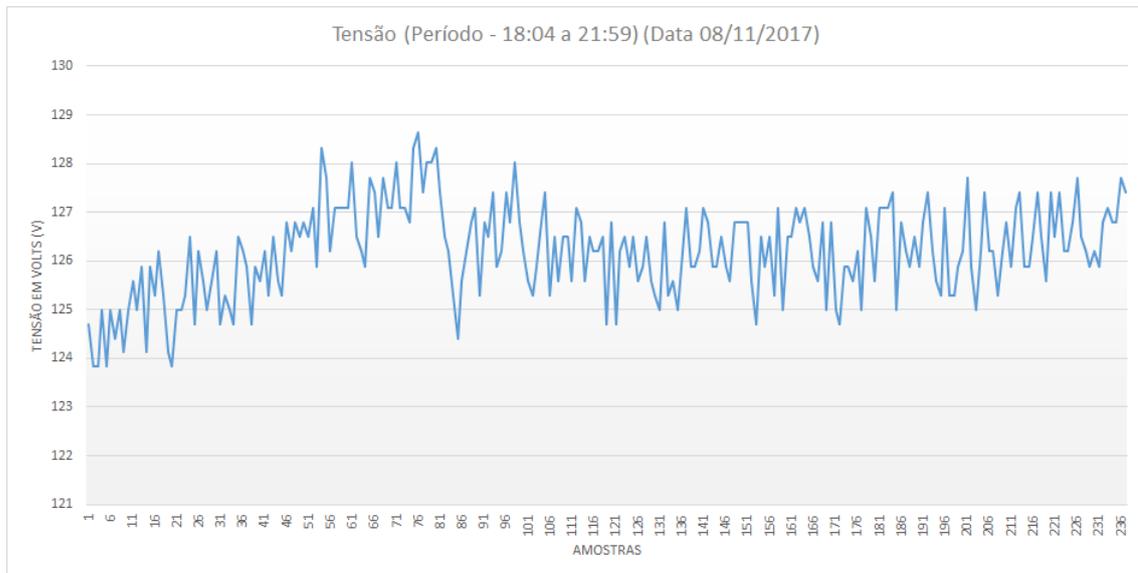
A espessura do condutor do protótipo que recebe toda a carga ficou subdimensionada. A tabela 8 mostra o correspondente da linha de cobre à corrente suportada.

Tabela 8: Dimensionamento da largura da trilha

| Corrente | Largura da trilha (mil/th) para 1 oz | Largura da trilha (mil/th) para 2 oz |
|----------|---|---|
| 1 | 10 | 5 |
| 2 | 30 | 15 |
| 3 | 50 | 25 |
| 4 | 80 | 40 |
| 5 | 110 | 55 |
| 6 | 150 | 75 |
| 7 | 180 | 90 |
| 8 | 220 | 110 |
| 9 | 260 | 130 |
| 10 | 300 | 150 |

Fonte: Bomfin (2013)

O modelo de placa usado no protótipo é 1 oz e a espessura usada foi 220mils (milésimo de polegada) que equivale a 5,588 milímetros e a corrente suporta de $8A$, neste caso a corrente para a sala 307 está dimensionado corretamente más subdimensionado para a sala 311.

Figura 39: Tensão em um período de tempo

Fonte: Elaborada pelo autor

As informações obtidas no gráfico da figura 39, serviram para determinar a tensão média no qual o equipamento foi submetido, o valor obtido está expresso na equação 4.3.

Para efeito de cálculo será usada a tensão obtida da média aritmética expressa na equação 4.3.

$$\bar{x} = \frac{\sum X_i}{n} \quad (4.3)$$

$$\bar{x} = \frac{124.7059 + 123.8318 + \dots + 127.4038}{237}$$

$$\bar{x} = 126.1963V$$

4.3.1 Análise de carga dos ventiladores

A figura 40 A, mostra o funcionamento em carga total, pode ser notado pela posição de ajuste do dimmer em "MÁX". A tabela 9 descreve os valores mensurados.

Figura 40: Aferindo a carga com o alicate amperímetro



Fonte: Elaborada pelo autor

Tabela 9: Carga máxima - figura 40 A

| Número | Valor |
|--------|--------------------|
| 1 | FP = 0.954 |
| 2 | Tensão = 125.5V |
| 3 | Corrente = 1.2A |
| - | Potência = 143.07W |

Fonte: Elaborada pelo autor

A figura 40 B, mostra o funcionamento em carga mínima, pode ser notado pela posição de ajuste do dimmer em "MIN". A tabela 10 descreve os valores mensurados.

Tabela 10: Carga mínima - figura 40 B

| Número | Valor |
|--------|--------------------|
| 1 | FP = 0.790 |
| 2 | Tensão = 125.9V |
| 3 | Corrente = 1.0A |
| - | Potência = 99.461W |

Fonte: Elaborada pelo autor

Neste ensaio verificou-se que a diferença da carga máxima para a carga mínima é de 43.609W. Outra informação a ser verificada é o fator de potência na carga mínima que ficou em 0.790.

4.3.2 Resultados da calibração

A tabela 11 apresenta os resultados obtidos para os vários níveis de calibração do protótipo em função da saída do controle e corrente. Será subtraído da corrente total o valor 3.5A referente a carga da iluminação.

Tabela 11: Relação de carga entregue aos ventiladores

| Controle (%) | Corrente (A) |
|--------------|--------------|
| 100% | 3.6A |
| 95% | 3.5A |
| 90% | 3.4A |
| 85% | 3.3A |
| 80% | 3.2A |
| 75% | 3.1A |
| 70% | 3.0A |
| 65% | 2.9A |
| 60% | 2.8A |
| 55% | 2.7A |
| 50% | 2.6A |

Fonte: Elaborada pelo autor

Valores abaixo de 50% não serão ativados pelo controle automatizado do protótipo e não será listado na tabela 11. O valor de referência da tensão utilizado nos cálculos está descrito na equação 4.3

4.3.3 Controle automático

Com base nos resultados da calibração, definiu-se os parâmetros de controle automático dos ventiladores, utilizando a temperatura do ambiente como regulador. A tabela 12 mostra a temperatura e seu respectivo controle.

Tabela 12: Nível de saída associado a temperatura

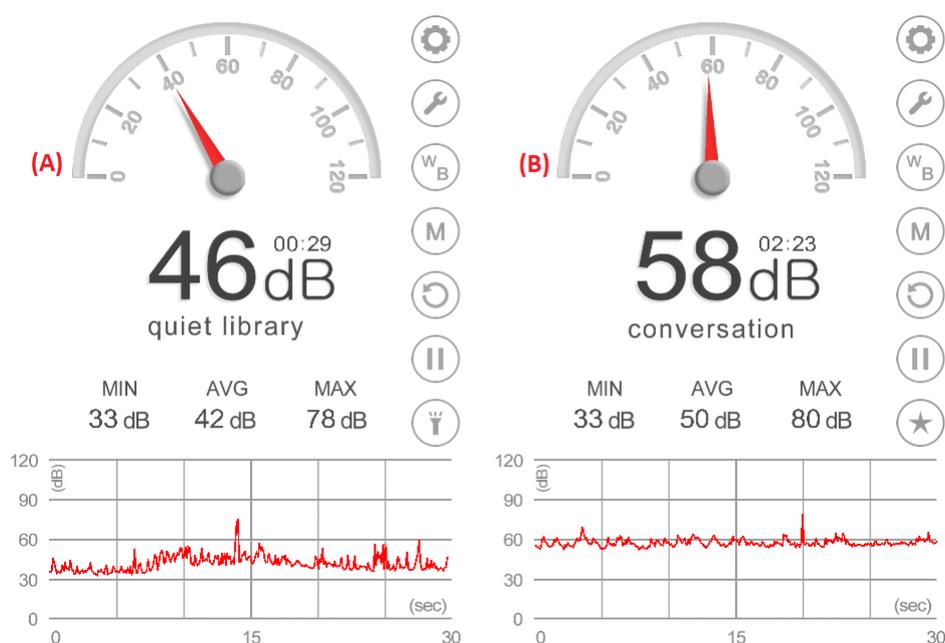
| Temperatura °C | Carga % |
|----------------|---------|
| <26° | 0 |
| 26° | 50% |
| 27° | 60% |
| 28° | 70% |
| 29° | 80% |
| >=30° | 90% |

Fonte: Elaborada pelo autor

A maior carga definida pelo controle automático é 90%. A carga em 100% gerava uma vibração na grade metálica dos ventiladores.

Paralelo ao monitoramento de nível de tensão, foi feito um acompanhamento com decibelímetro (ABC, 2017), e verificou-se que acima de 90% de saída, gerava um ruído considerável. A figura 41 A, mostra o valor de 46db (decibel) com os ventiladores desligados e a figura 41 B mostra o valor de 58bd com os ventiladores ligados (100%).

Figura 41: Decibelímetro da ABC Apps



Fonte: Elaborada pelo autor

4.4 DADOS DA TEMPERATURA

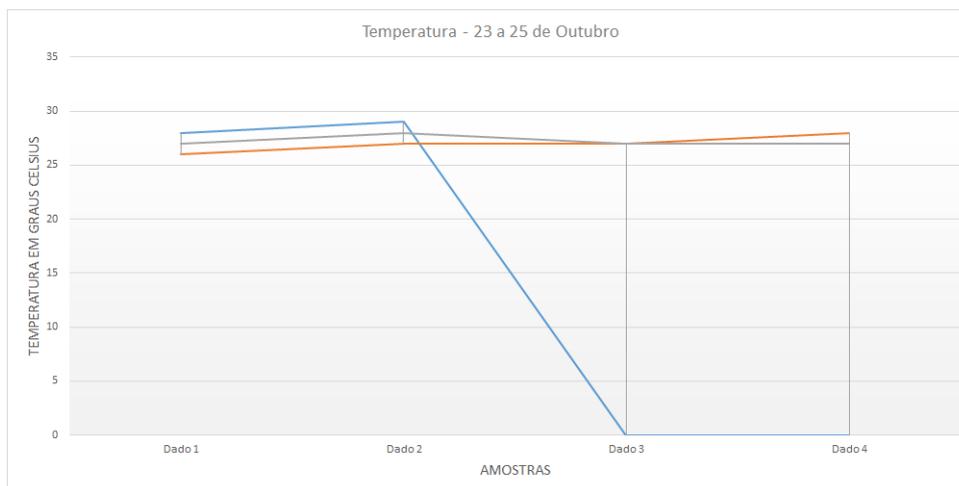
Os dados obtidos entre os dias 23 ao dia 25 de outubro apresentados na tabela 7, capítulo 3.9.1, foram utilizados para a confecção do gráfico da figura 42.

A curva azul representa os dados obtidos no dia 23/10/2017, que no início do primeiro período, foi aferido a temperatura da sala em 28°C (Celsius) e ao final do segundo período a temperatura aferida em 29°C, houve uma elevação de 1°C devido a presença dos estudantes na sala.

A queda abrupta apresentada no gráfico é devido a falta da coleta de dados da temperatura dos terceiro e quarto horários. É notório também o aumento em 1°C nas curvas do dia 24 e 25, mostrando um padrão de elevação temperatura em função da presença dos estudantes.

As temperaturas entre o terceiro e o quarto períodos não se mostraram relevantes para a pesquisa.

Figura 42: Aquisição manual de dados



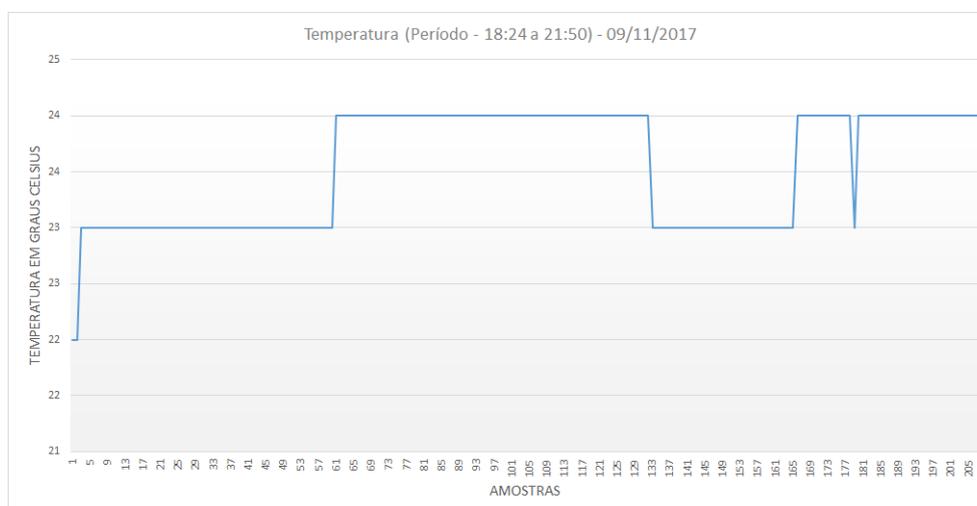
Fonte: Elaborada pelo autor

Devido as lacunas existentes na coleta dos dados realizadas de forma manual, foi necessário criar um supervisor. A coleta passou a ser feita de forma automática, e um dos dados obtidos, está representado no gráfico da figura 43.

Observa-se que no início do primeiro período a temperatura da sala era 22°C, com a chegada dos estudantes a temperatura subiu para 24°C e se mantendo até o final do segundo período. Com a saída de parte dos estudantes, houve uma diminuição da temperatura sala para 23°C, ao final do intervalo a temperatura retornou a 24°C. A queda abrupta da temperatura apresentada no gráfico é devido a abertura da janela.

4.4.1 Aquisição de dados por supervisor

Figura 43: Aquisição de dados por supervisor



Fonte: Elaborada pelo autor

4.4.2 Comparativo entre cargas usuais e controladas

Para este comparativo será usada a amostra do dia 24/10/2017. Será usado a média aritmética para definir uma temperatura aproximada. O cálculo é dado pela expressão 4.4.

$$\bar{\chi} = \frac{\sum X_i}{n} \quad (4.4)$$

$$\bar{\chi} = \frac{26^\circ + 27^\circ + 27^\circ + 28^\circ}{4}$$

$$\bar{\chi} = 27^\circ$$

Com base na tabela 12 o controle atribuído à temperatura de 27°C é de 60%, e na tabela 11, uma corrente de 2.8A. A potência é calculada e definida pela expressão 4.5. A tensão usada para o cálculo será de 126.1963V, calculada na equação 4.3 .

$$P = V.I.FP \quad (4.5)$$

$$P = 126.1963V \times 2.8A \times 0.95$$

$$P = 335.68W$$

A carga utilizada como referência está descrita na tabela 11 em 100% de controle, por entender que esta é a corrente absorvida pelo circuito. Para calcular a potência será utilizada a expressão 4.6.

$$P = V.I.FP \quad (4.6)$$

$$P = 126.1963V \times 3.6A \times 0.95$$

$$P = 431.59W$$

Levando-se em consideração a temperatura de 27°C e que o estudante sempre usa capacidade máxima dos ventiladores, pode-se concluir que existe uma economia de 95.31W ou 22.08% com o controle automatizado ativo.

Considerando-se o controle na temperatura mínima que é 26°C a 50%, temos uma potência consumida de 311.704W e neste caso a economia pode chegar a 27.77% em relação a carga máxima.

4.5 COMPARATIVOS DE TEMPERATURAS

A comparação de temperatura vista na tabela 13 verifica o comportamento em um período de análise no desvio padrão. Este comportamento mostra o quanto a temperatura está variando no período. A variância está expressada na fórmula 4.7 e o desvio padrão na fórmula 4.8.

$$S^2 = \frac{\sum_{i=1}^n (xi - \bar{x})^2}{n - 1} \quad (4.7)$$

$$\sigma = \sqrt{S^2} \quad (4.8)$$

Com esta análise pode-se verificar se os ventiladores estão ou não interferindo na temperatura ambiente na sala. Para se obter o desvio padrão é necessário primeiro calcular a variância.

Tabela 13: Tabela comparativa de temperatura

| Data | Amostras | Média | Variância | Desv. Padrão |
|------------|----------|--------------|-------------|--------------|
| 23/10/2017 | 2 | 28.5 | 0.5 | 0,707106781 |
| 24/10/2017 | 4 | 27 | 0,666666667 | 0,816496581 |
| 25/10/2017 | 4 | 27.25 | 0.25 | 0.5 |
| 06/11/2017 | 107 | 24,666666667 | 0,326573797 | 0,571466356 |
| 07/11/2017 | 225 | 23 | 0,221111111 | 0,470224533 |
| 08/11/2017 | 237 | 23 | 0,159650694 | 0,402192769 |
| 09/11/2017 | 207 | 24 | 0,269311946 | 0,51895274 |
| 10/11/2017 | 107 | 23 | 0 | 0 |

Fonte: Elaborada pelo autor

5 CONCLUSÃO

Nesse trabalho foi monitorado o padrão de funcionamento de iluminação e máquinas rotativas durante o horário de aula do terceiro turno da sala 311 do bloco C das Faculdades Doctum de Caratinga, antes e depois da implantação do protótipo de controle.

Com este trabalho foi possível concluir que os ventiladores e lâmpadas funcionavam na maioria das vezes em capacidade máxima antes, durante e depois do horário de aula, o que gera grande desperdício de recursos, visto que o mesmo ocorre em quase todas as salas de aula.

Verificamos que a temperatura da sala, entre o início e o final das aulas se elevava na média em um grau Celsius após a entrada de todos os estudantes, mesmo em dias mais frios. Essa conclusão foi verificada tanto pela análise dos dados colhidos de forma manual quanto pelos dados fornecidos pelo supervisor.

Com o controle das máquinas rotativas feito pelo protótipo foi possível concluir que o melhor ajuste quando a sala esta cheia seria de 90% da capacidade máxima, essa configuração além de gerar economia de energia também reduz os ruídos de vibração em 10 db sobre o funcionamento dos ventiladores, sendo que nenhum desconforto foi relatado pelos estudantes durante a fase de testes.

Salienta-se que o propósito desse protótipo não é o de controlar a temperatura nem a luminosidade da sala, e sim tornar racional o uso dos recursos disponíveis para o conforto dos estudantes. Isso foi executado com grande eficácia pelo experimento, visto que o controle automático, quando em operação, torna impossível o acionamento manual dos ventiladores. Outra função importante do controle automático é o seu acionamento que só é possível se houver alguém presente na sala.

Para trabalhos futuros a plataforma de controle (CAdME) poderá ser utilizada para implantar o uso racional de máquinas refrigeradoras de ambiente e implementar um sistema de coleta de dados via wi-fi, o que em minha opinião tornaria o emprego do CAdME mais eficiente.

6 REFERÊNCIAS BIBLIOGRÁFICAS

- ABC, Apps. *Decibelímetro (Sound Meter)*. Disponível em: <https://play.google.com/store/apps/details?id=com.gamebasic.decibel&hl=pt_BR>. Acesso em: 22 de maio. 2017.
- ATMEL. Datasheet: *ATmega328/P*. Publicação de material eletrônico, 2016. Disponível em: <http://www.atmel.com/Images/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf>. Acesso em: 22 de novembro. 2017.
- BENYON, David. *Interação Humano-Computador*. 2a ed. Rio de Janeiro: Editora Pearson, 2011.
- BOYLESTAD, Robert L.; NASHELSKY, Lois. *Dispositivos Eletrônicos e Teoria de Circuitos*. 8a ed. São Paulo: Editora Pearson, 2004.
- BRAGA, Newton C. *Dimmers e controles de potência (ART071)*. Disponível em: <<http://www.newtoncbraga.com.br/index.php/como-funciona/619-dimmers-e-controles-de-potencia-art071>>. Acesso em: 22 de maio. 2017.
- BRAGA, Newton C. *Relés - Conceitos e Aplicações*. 1a ed. São Paulo: Editora NCB, 2012.
- FITZGERALD, A.E.; JR. Charles Kingsley; KUSKO, Alexander. *Máquinas Elétricas: Conversão Eletromecânica da Energia, Processos, Dispositivos e Sistemas*. 1a ed. São Paulo: Editora McGraw-Hill do Brasil, 1975.
- GE, Lighting. Datasheet: *Reatores Eletrônicos*. Publicação de material eletrônico, 2014. Disponível em: <http://www.gelighting.com/LightingWeb/br/images/Reatores%20T8%20T10_tcm388-78559.pdf>. Acesso em: 22 de novembro. 2017.
- HALLIDAY, David *Eletromagnetismo*. 8a ed. Rio de Janeiro: Editora LTC, 2009.
- KLANDER, Kris; JAMSA, Lars. *Programando Em C/C++: A Bíblia - O Melhor Guia Para Programacao Em C/*. 1a ed. Rio de Janeiro: Editora Pearson, 1999.
- MAXIN Integrated. Datasheet: *Sensor de Temperatura DS18S20*. Publicação de material eletrônico, 2017. Disponível em: <<https://www.maximintegrated.com/en/products/analog/sensors-and-sensor-interface/DS18S20.html>>. Acesso em: 22 de novembro. 2017.
- MINIPA. Datasheet: *Alicate Wattímetro ET-4055*. Publicação de material eletrônico, 2012. Disponível em: <<http://www.minipa.com.br/images/Manual/ET-4055-1101-BR.pdf>>. Acesso em: 22 de novembro. 2017.
- MONTGOMERY, Douglas C.; RUNGER, George C. *Estatística Aplicada e Probabilidade para Engenheiros*. 5a ed. Rio de Janeiro: Editora LTC, 2012.
- NATIONAL. *Comunicação Serial*. Publicação de material eletrônico, 2015. Disponível em:

<<http://digital.ni.com/public.nsf/allkb/32679C566F4B9700862576A20051FE8F>>. Acesso em: 25 de novembro. 2017.

NXP Semiconductors. Datasheet: *Expansor de portas PCF8574*. Publicação de material eletrônico, 2013. Disponível em: <https://www.nxp.com/docs/en/data-sheet/PCF8574_PCF8574A.pdf>. Acesso em: 22 de novembro. 2017.

NYSENBA. Datasheet: *PIR Sensor D204B*. Publicação de material eletrônico, 2016. Disponível em: <<http://en.nysenba.com/index.php/page-5-5.html>>. Acesso em: 22 de novembro. 2017.

OGATA, Katsuhiko. *Engenharia de Controle Moderno*. 5a ed. Rio de Janeiro: Editora Pearson, 2011.

O'MALLEY, Jhon. *Análise de Circuitos*. Trad. Moema Sant'Anna. 2a ed. São Paulo: Editora Makron Books, 1993.

PHILIPS Semiconductors. Datasheet: *Triac BT136*. Publicação de material eletrônico, 1997. Disponível em: <<http://www.micropik.com/PDF/BT136-600.pdf>>. Acesso em: 22 de novembro. 2017.

LABCENTER. *Proteus*. Disponível em: <<https://www.labcenter.com/>>. acesso em: 25 de maio. 2017.

RASHID, Muhammad H. *Eletrônica de Potência: Circuitos, Dispositivos e Aplicações*. 1a ed. São Paulo: Makron Books Ltda, 2011.

STALLINGS, William. *Arquitetura e Organização de Computadores*. 5a ed. São Paulo: Editora Prentice Hall, 2003.

TEIXEIRA, Francisco M. P. *Revolução Industrial: Coleção O Cotidiano da História*. 12a ed. São Paulo: Editora Ática, 2004.

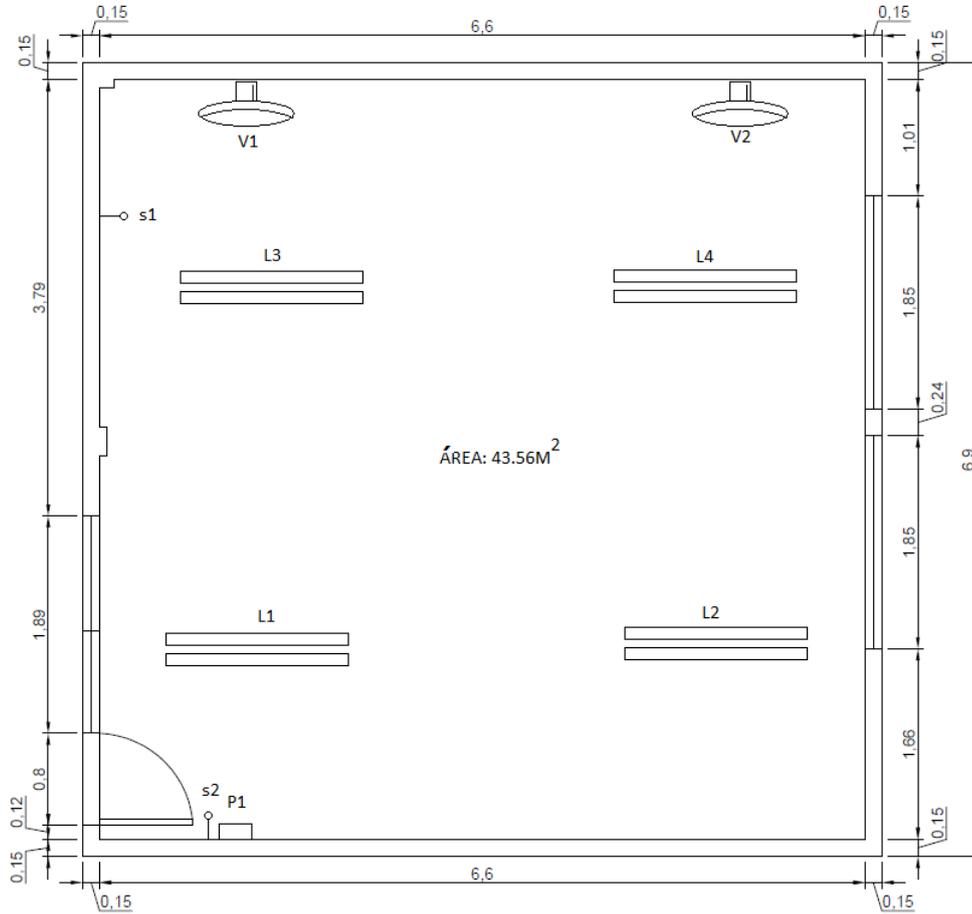
VENTI-DELTA. *Parede Premium 60CM - Grade de Aço*. Disponível em: <<http://ventidelta.com.br/conteudo/produtocatalogo/58/parede-premium-60cm-grade-de-aco>>. Acesso em: 22 de maio. 2017.

VILLATE, Jaime E. *Física 2 - Eletricidade e Magnetismo*. Porto, 2012.

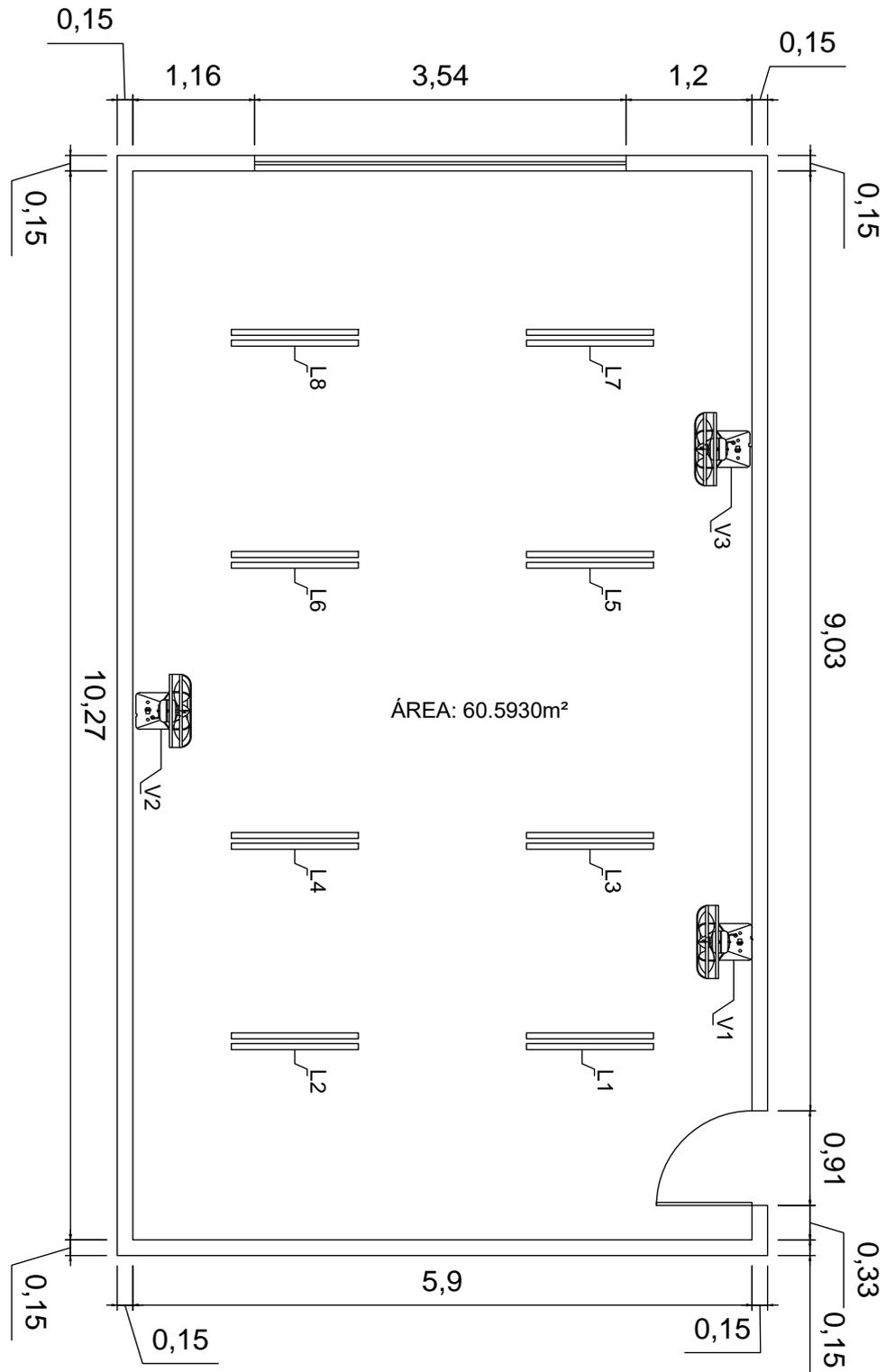
XIAMEN. Datasheet: *Tela de LCD 16x2*. Publicação de material eletrônico, 2015. Disponível em: <<https://www.sparkfun.com/datasheets/LCD/ADM1602K-NSW-FBS-3.3v.pdf>>. Acesso em: 22 de novembro. 2017.

YHDC. Datasheet: *Sensor de Corrente SCT013*. Publicação de material eletrônico, 2016 Disponível em: <<http://en.yhdc.com/product1311.html?productId=401>>. Acesso em: 22 de novembro. 2017.

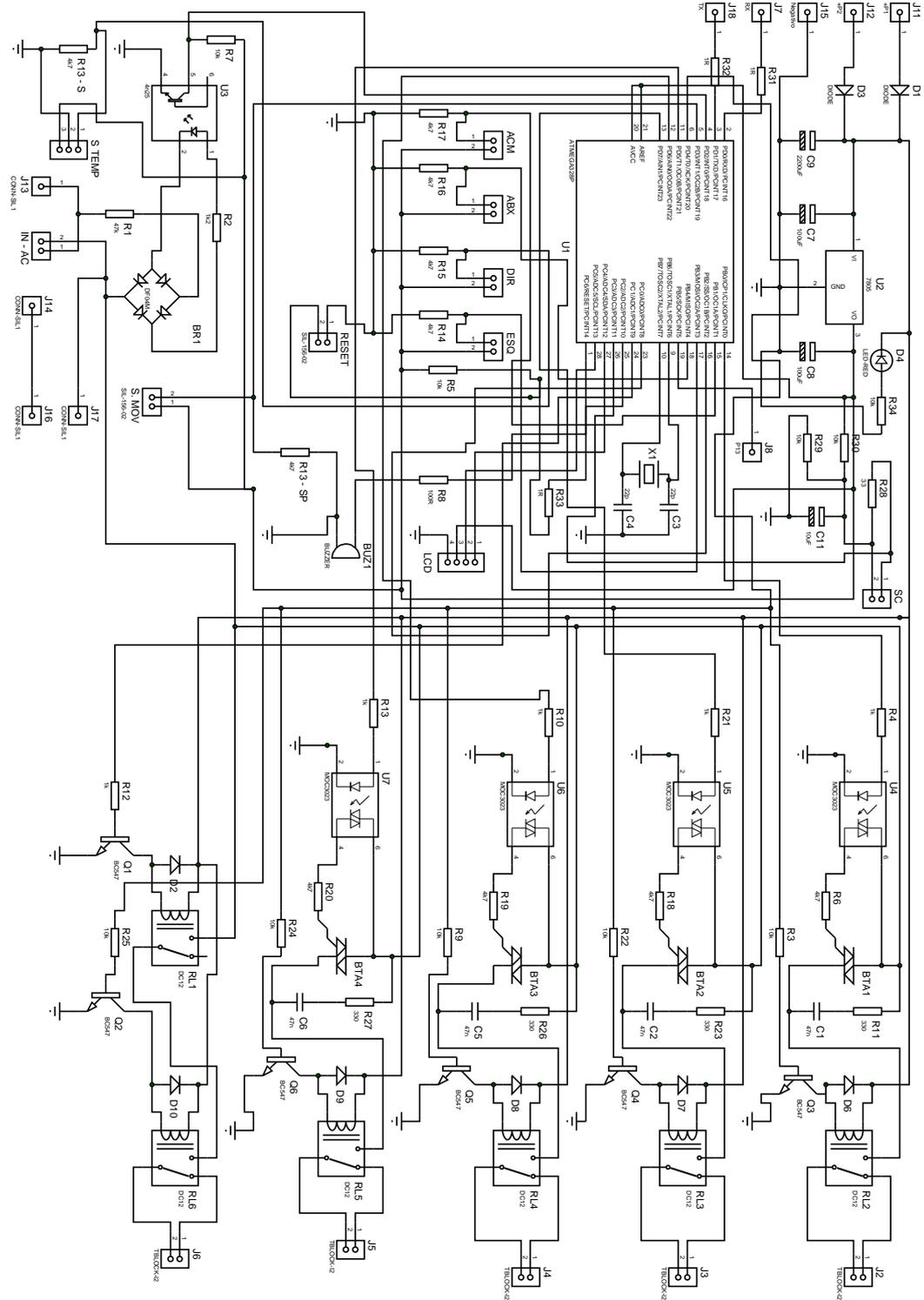
APÊNDICE A PLANTA BAIXA DA SALA 307



APÊNDICE B PLANTA BAIXA DA SALA 311



APÊNDICE C ESQUEMA ELETRÔNICO DO PROTÓTIPO



APÊNDICE D CONFIGURAÇÃO ATMEGA 328P

| PONTO FÍSICO | PONTO ARDUINO | FUNÇÃO ATRIBUIDA |
|--------------|---------------|--------------------|
| 01 | RESET | RESET |
| 02 | RX | RX |
| 03 | TX | TX |
| 04 | P. DIGITAL 02 | ATTACH INTERRUPT |
| 05 | P.DIGITAL 03 | SENSOR DE PRESENÇA |
| 06 | P. DIGITAL 04 | SISTEMA L/D |
| 07 | VCC | VCC |
| 08 | GND | GND |
| 09 | CRISTAL | CRISTAL |
| 10 | CRISTAL | CRISTAL |
| 11 | P. DIGITAL 05 | DIMMER 2 |
| 12 | P. DIGITAL 06 | DIMMER 3 |
| 13 | P. DIGITAL 07 | DIMMER 4 |
| 14 | P. DIGITAL 08 | DIMMER 1 |
| 15 | P. DIGITAL 09 | BT SOBE |
| 16 | P. DIGITAL 10 | BT DIREITA |
| 17 | P. DIGITAL 11 | BT ESQUERDA |
| 18 | P. DIGITAL 12 | BT DESCE |
| 19 | P. DIGITAL 13 | SENSOR IR |
| 20 | VCC | VCC |
| 21 | VCC | VCC |
| 22 | GND | GND |
| 23 | A0 | SENSOR TEMPERATURA |
| 24 | A1 | ATIVA LUZ |
| 25 | A2 | BUZZER |
| 26 | A3 | SENSOR DE CORRENTE |
| 27 | A4 | LCD SCL |
| 28 | A5 | LCD SDA |

APÊNDICE E EMAIL - VENTILADORES DELTA

19/11/2017

Gmail - Re: Contato - gilmarcorrearmoraes@gmail.com



Gilmar C. Moraes <gilmarcorrearmoraes@gmail.com>

Re: Contato - gilmarcorrearmoraes@gmail.com

1 mensagem

Dejair <dejair@ventidelta.com.br>
Para: gilmarcorrearmoraes@gmail.com

30 de outubro de 2017 16:56

Boa Tarde ; Gilmar !!

Em 127 Volts a corrente é:- 1,409 A com FP = 0,95 . Capacitor de 6 uF X 400 Vac.

Em 220 Volts a corrente é:- 0,77 A com FP = 0,98 . Capacitor de 6 uF X 400 Vac.

Att;

Dejair.

----- Original Message -----

From: [Vendas - VentiDelta](#)
To: dejair@ventidelta.com.br
Sent: Monday, October 30, 2017 4:13 PM
Subject: Fw: Contato - gilmarcorrearmoraes@gmail.com**From:** [Formulário](#)
Sent: Monday, October 30, 2017 1:17 AM
To: vendas@ventidelta.com.br
Subject: Contato - gilmarcorrearmoraes@gmail.comNome: Gilmar Correa Moraes
Email: gilmarcorrearmoraes@gmail.com
Telefone: 3333221702

Mensagem: Oi tudo bom ? Meu nome é Gilmar, sou estudante de engenharia elétrica, em meu TCC estou fazendo um estudo de eficiência utilizando sistemas automatizados, estou usando uma sala de aula para fundamentalizar minha argumentação, as salas possuem ventiladores de sua empresa do modelo de parede PREMIUM 60CM - GRADE DE AÇO, no site só tem a potencia de 170W, más seria interessante se eu soubesse a corrente e o fator de potencia para a minha revisão bibliográfica, gostaria de saber se existe a possibilidade de me enviarem esta informação, desde já agradeço.

APÊNDICE F CÓDIGO FONTE EM C, UTILIZADO NO MICROCONTROLADOR

```

// Programa feito por Gilmar C. Moraes.
// TCC2 – Eng. Eletrica.
// Agradecimentos ao RUI do site Labdegaragem pela importante ajuda.

#include <OneWire.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h> // Biblioteca LCD I2C
#include <TimerOne.h> // Biblioteca do timer1
#include <Limits.h>
#include "EmonLib.h"

EnergyMonitor emon1;
int rede = 127;
int pino_sct = A3;
int DS18S20_Pin = 14; // DS18S20 Signal pin on digital 14
int PinoAtivaLuz = A1; // Pino para o acionamento da lampada (
const int PinoBuzzer = A2; // Pino de saida para o buzzer, ao muda
ok testado
int PinOnOff = 4; // Pino de saida para comutacao (feita
int PinoSensPres = 3; // Pino ligado ao sensor de movimento (
int acionamento; // Variavel para guardar valor do senso
int ConRegre = 240; // Variavel com o valor definido de 120
int ConRegreReg = ConRegre; // Recebe o valor de referencia definid
int MenuCarga = 1; // Estado do menu AUTOMATIZACAO onde 1
int ConfigTemp = 26; // variavel que estabelece a temperatura
int ConfigTempMuda = 26;
int ContaDesliga = 0;
int AtivaSist = 1;
int JaContei = 0; // declarando variavel e atribuindo o v
int seg = 0, min = 0, hor = 0;
int SomaTempo = 0;
int TempAtual = 0;
unsigned long Cinco_Min = 30000; // 5 minutos
unsigned long Tempo = 0; // Comparador do tempo
int TempMem = 0;
int ConRegLCD = 10;
float temperature;
double Irms;

//Envio de dados

String sinais;
unsigned long tempoDecorrido = 0;

OneWire ds(DS18S20_Pin); // on digital pin 2

int val1 = 0; // variavel que recebe valores entre 0
int val2 = 0; // variavel que recebe valores entre 0
int val3 = 0; // variavel que recebe valores entre 0
int val4 = 0; // variavel que recebe valores entre 0

```

```

int TempDisp1 = 0;
int TempDisp2 = 0;
int TempDisp3 = 0;
int TempDisp4 = 0;

// Substituindo o delay do termometro por millis

unsigned long tBase = 0;           // recebe valor inicial
int tInt = 1000;                 // tempo a ser subtraido

#define bDir 10                   // Definindo entrada para os botoes
#define bEsq 11
#define bSobe 9
#define bDesce 12

#define bDir0 90                  // Valor de referencia que a
#define bEsq0 91                  // funcao CheckButton() passa
#define bSobe0 92                  // indicando que um botao foi
#define bDesce0 93                // solto

boolean aLeft, aRight, aUp, aDown; // Grava o ultimo valor lidos nos botoes

int valorSensorTemp = 0;
int menorValorTemp = INT_MAX;

int state = 1;                   // variavel que guarda posicao do menu

LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE); // Endereco para LCD

byte seta_direita[8] = {0x00, 0x04, 0x02, 0x1F, 0x02, 0x04, 0x00, 0x00}; // cria seta direita
byte seta_esquerda[8] = {0x00, 0x04, 0x08, 0x1F, 0x08, 0x04, 0x00, 0x00}; // cria seta esquerda
byte seta_abaixo[8] = {0x00, 0x04, 0x04, 0x15, 0x0E, 0x04, 0x00, 0x00}; // cria seta abaixo
byte seta_acima[8] = {0x00, 0x04, 0x0E, 0x15, 0x04, 0x04, 0x00, 0x00}; // cria seta acima
byte seta_indica[8] = {0x00, 0x06, 0x0C, 0x18, 0x18, 0x0C, 0x06, 0x00}; // cria seta indica

byte Dir = 1;
byte Esq = 2;
byte Desce = 3;
byte Sobe = 4;
byte Indica = 5;

unsigned char Canal1 = 8;        // Port para Triac 1 (pin 4)
unsigned char Canal2 = 5;        // Port para Triac 2 (pin 5)
unsigned char Canal3 = 6;        // Port para Triac 3 (pin 6)
unsigned char Canal4 = 7;        // Port para Triac 4 (pin 7)
unsigned char Pulso_Timer1;      // Pulsos gerados pelo Timer1

unsigned char Disparo[4] = {100, 100, 100, 100}; // Valor iniciais para cargas
(Apagadas)
int Contador = 0;

//-----

void setup()
{
  Serial.begin(9600);
  attachInterrupt(0, zero_cross, RISING); // Define rotina chamada pela passagem d

```

```

Timer1.initialize(83); // Define interrupt de Timer1 a cada 83
Timer1.attachInterrupt(Timer1Zr); // OverFlow de Timer1
lcd.backlight();
emon1.current(pino_sct, 60);

pinMode(bDir, INPUT_PULLUP); // Botoes
pinMode(bEsq, INPUT_PULLUP);
pinMode(bSobe, INPUT_PULLUP);
pinMode(bDesce, INPUT_PULLUP);

pinMode(PinoBuzzer, OUTPUT); // Define port como saída para o buzzer
pinMode(PinoAtivaLuz, OUTPUT); // Define port como saída para comando
pinMode(PinoSensPres, INPUT); // Define port sensor como entrada

pinMode(PinOnOff, OUTPUT); // Define port como saída para comutaca
digitalWrite(PinOnOff, HIGH); // Comuta para o controle digital digit

digitalWrite(PinoAtivaLuz, HIGH); // Comuta para o controle digital digit
pinMode(Canal1, OUTPUT); // Define port para Canal1 como saída
pinMode(Canal2, OUTPUT); // Define port para Canal2 como saída
pinMode(Canal3, OUTPUT); // Define port para Canal3 como saída
pinMode(Canal4, OUTPUT); // Define port para Canal4 como saída

lcd.begin(16, 2); // Iniciando a biblioteca do LCD
lcd.print("INICIANDO... "); // splash
delay(1000); // Parada de 1segundo

lcd.createChar(Dir, seta_direita); // vai criar novos caracteres
lcd.createChar(Esq, seta_esquerda);
lcd.createChar(Desce, seta_abaixo);
lcd.createChar(Sobe, seta_acima);
lcd.createChar(Indica, seta_indica);
}
//-----
void loop()
{
  if (millis() - tempoDecorrido > 1000) {
    tempoDecorrido = millis(); //Se passou mais de 1s zera o contador de tempo
    sinais = String(ConRegreReg) + "|" + String(ConRegre) + "|" + String(TempMem) + "|" + Strin
    + String(MenuCarga)+ "|" + String(temperature)+ "|" + String(Disparo[0])+ "|" + String(Disp
    + "|" + String(Disparo[3]) + "|" + String(ConfigTemp) + "|" + String(ConfigTempMuda)+ "|" +
    Serial.println(sinais);
  }
  temperature = getTemp();
  if (ConRegLCD <= 0) {
    lcd.noBacklight();
  }
  else {
    lcd.backlight();
  }
  MemoriazaTemperatura();

  if (JaContei == 1) {
    CalcTempDes();
  }
  if (AtivaSist != 0)
  {

```

```

    if (MenuCarga == 1)
    {
        Controla_Vent_Temp(); // pelo sensor de temperatura e pelas
de carga predeterminadas pela funcao.
    }
}
acionamento = digitalRead(PinoSensPres); // Le o valor do sensor
if (acionamento != 0)
{
    digitalWrite(PinoAtivaLuz, HIGH); // aciona a luz se estiver desligada(p
ConRegre = ConRegreReg; // reseta o contador para o valor defi
JaContei = 0;
}
if (ConRegre <= 0)
{
    digitalWrite(PinoAtivaLuz, LOW); // estado baixo da saida
    if (JaContei == 0 )
    {
        ContaDesliga = ContaDesliga + 1; // pega o valor armazenado e soma + 1
        JaContei = 1;
    }
}

switch (state) // Estrutura do menu – botoes e acoes
{ // O state captura a posicao em que o switch vai estar
  case 1: // menu (CONTROLE) acoes (LIGADO/DESLIGADO)
    if (digitalRead(4) == HIGH) // Se o sistema estiver ligado (CONTRO
    {
        lcd.setCursor(0, 1);
        lcd.print("LIGADO "); // Print LIGADO
    }
    else // Se o sistema estiver desligado
    {
        lcd.setCursor(0, 1);
        lcd.print("DESLIGADO "); // Print DESLIGADO
    }
    switch (CheckButton()) // verifica qual botao foi pressionado
    {
        case bDesce: // botao bDesce pressionado
            lcd.clear(); // limpa a tela do LCD
            Set_state(2); // seta para o menu (ILUMINACAO)
            tone(PinoBuzzer, 1500, 50); // o buzzer emite um beep
            break;
        case bDir: // botao bDir pressionado (neste caso
            if (digitalRead(4) == HIGH) // Se o sistema estiver ligado entao f
            {
                digitalWrite(4, LOW); // Desliga saida
                AtivaSist = 0;
                lcd.setCursor(0, 1); // Posiciona cursor
                lcd.print("DESLIGADO "); // imprime DESLIGADO no LCD
                tone(PinoBuzzer, 1500, 50); // o buzzer emite um beep
            }
            break;
        case bEsq: // botao bEsq pressionado (neste caso
            if (digitalRead(4) == LOW ) // Se o sistema estiver desligado enta
            {

```

```

        digitalWrite(4, HIGH);                // Liga Saida
        AtivaSist = 1;
        lcd.setCursor(0, 1);                 // Posiciona cursor
        lcd.print("LIGADO");                 // imprime LIGADO no LCD
        tone(PinoBuzzer, 1500, 50);         // o buzzer emite um beep
    }
    break;
default:
    Set_state(1);                            // sempre vai exibir as informacoes do
    break;
}
break;
// informacoes repetidas no restante do codigo considerar referencias acima ja citadas
case 2:                                     // menu (ILUMINACAO) exibe o tempo pre
    switch (CheckButton())
    {
        case bDir:                          // botao bDir pressionado
            lcd.clear();                     // limpa a tela do LCD
            tone(PinoBuzzer, 1500, 50);     // o buzzer emite um beep
            Set_state(3);                    // seta para o menu (CONF TEMPO)
            break;
        case bDesce:                         // botao bDesce pressionado
            lcd.clear();
            tone(PinoBuzzer, 1500, 50);
            Set_state(5);                    // exibe para o menu (CARGAS)
            break;
        case bSobe:                          // botao bSobe pressionado
            lcd.clear();
            tone(PinoBuzzer, 1500, 50);
            Set_state(1);                    // Retorna ao menu (CONTROLE)
            break;
        default:
            Set_state(2);                    // sempre vai exibir as informacao do
    }
    break;
case 3:                                     // exibe o menu (CONFIGURA TEMPO) que
    switch (CheckButton())
    {
        case bDir:                          // botao bDir pressionado
            lcd.clear();
            tone(PinoBuzzer, 1500, 50);
            Set_state(4);                    // exibe o menu (MONIT. TEMPO) que e u
            break;
        case bEsq:                           // botao bEsq pressionado
            lcd.clear();
            tone(PinoBuzzer, 1500, 50);
            Set_state(2);                    // Retorna ao menu principal (ILUMINAC
            break;
        case bDesce:                         // botao bDesce pressionado
            if (ConRegreReg >= 0)            // verifica o valor de ConRegreReg, se
            {
                ConRegreReg = ConRegreReg - 30; // ConRegreReg recebe o valor dele mes
                ConRegre = ConRegreReg;        // ConRegre recebe o valor de ConRegre
                tone(PinoBuzzer, 1500, 50);
            }
            break;
        case bSobe:                          // botao bSobe pressionado

```

```

        ConRegreReg = ConRegreReg + 30; // ConRegreReg recebe o valor dele mes
        ConRegre = ConRegreReg; // ConRegre recebe o valor de ConRegre
        tone(PinoBuzzer , 1500, 50);
        break;
    default: // sempre vai exibir as informacao do
        Set_state(3);
    }
    break; // exibe o menu (MONITORACAO)que e um
case 4: // botao bDir pressionado
    switch (CheckButton())
    {
        case bDir: // volta ao nemu (ILUMINACAO)
            lcd.clear();
            tone(PinoBuzzer , 1500, 50);
            Set_state(2);
            break;
        case bEsq: // botao bEsq pressionado
            lcd.clear();
            tone(PinoBuzzer , 1500, 50);
            Set_state(3); // volta ao nemu (CONFIGURA TEMPO)
            break;
        default: // sempre vai exibir as informacao do
            Set_state(4); // aqui termina o controle sobre a ilu
    }
    break;
case 5: //exibe o menu (CARGAS)
    switch (CheckButton())
    {
        case bDir: // exibe o menu (AUTOMACAO)
            lcd.clear();
            tone(PinoBuzzer , 1500, 50);
            Set_state(11);
            break;
        case bDesce: // retorna ao menu (MONITORACAO)
            lcd.clear();
            tone(PinoBuzzer , 1500, 50);
            Set_state(13);
            break;
        case bSobe: // retorna ao menu (ILUMINACAO)
            lcd.clear();
            tone(PinoBuzzer , 1500, 50);
            Set_state(2);
            break;
        default: // sempre vai exibir as informacao do n
            Set_state(5); // este menu pode-se voltar no menu ILU
    }
    break;
case 6: // menu (CARGA - 1) caso a opcao (AUTOM
    switch (CheckButton())
    {
        case bDir: //exibe menu (CARGA - 2)
            lcd.clear();
            tone(PinoBuzzer , 1500, 50);
            Set_state(7);
            break;
        case bEsq:
            lcd.clear();

```

```

    tone(PinoBuzzer , 1500, 50);
    Set_state(11);
    break;
case bSobe:
    if (val1 < 100 )
    {
        val1 = val1 + 5;
        Disparo[0] = 100 - val1;
        tone(PinoBuzzer , 1500, 50);
    }
    break;
case bDesce:
    if (val1 > 0)
    {
        val1 = val1 - 5;
        Disparo[0] = 100 - val1;
        tone(PinoBuzzer , 1500, 50);
    }
    break;
default:
    Set_state(6);
}
break;
case 7:
    switch (CheckButton())
    {
        case bDir:
            lcd.clear();
            tone(PinoBuzzer , 1500, 50);
            Set_state(8);
            break;
        case bEsq: //
            lcd.clear();
            tone(PinoBuzzer , 1500, 50);
            Set_state(6);
            break;
        case bSobe:
            if (val2 < 100 )
            {
                val2 = val2 + 5;
                Disparo[1] = 100 - val2;
                tone(PinoBuzzer , 1500, 50);
            }
            break;
        case bDesce:
            if (val2 > 0)
            {
                val2 = val2 - 5;
                Disparo[1] = 100 - val2;
                tone(PinoBuzzer , 1500, 50);
            }
            break;
        default:
            Set_state(7);
    }
    break;
case 8:

```

// retorna ao menu (AUTOMACAO)

// botao bSobe pressionado

//Recebe o valor de val1 acrecido de 5

// botao bDesce pressionado

//Recebe o valor de val1 decrescido de

// sempre vai exibir as informacao do ne

// menu (CARGA - 2) caso a opcao (AUTOMA

//exibe menu (CARGA - 3)

// retorna ao menu (CARGA - 2)

// botao bSobe e bDesce, muda os valore

// sempre vai exibir as informacao do

// menu (CARGA - 3) caso a opcao (AUTO

```
switch (CheckButton())
{
  case bDir:
    lcd.clear();
    tone(PinoBuzzer, 1500, 50);
    Set_state(9); //exibe o menu (CARGA - 4)
    break;
  case bEsq:
    lcd.clear();
    tone(PinoBuzzer, 1500, 50);
    Set_state(7); // retorna ao menu (CARGA - 2)
    break;
  case bSobe: // botao bSobe e bDesce, muda os valores
    if (val3 < 100 )
    {
      val3 = val3 + 5;
      Disparo[2] = 100 - val3;
      tone(PinoBuzzer, 1500, 50);
    }
    break;
  case bDesce:
    if (val3 > 0)
    {
      val3 = val3 - 5;
      Disparo[2] = 100 - val3;
      tone(PinoBuzzer, 1500, 50);
    }
    break;
  default:
    Set_state(8); // sempre vai exibir as informacao do nem
}
break;
case 9: // menu (CARGA - 1) caso a opcao (AUTOMAC
switch (CheckButton())
{
  case bDir:
    lcd.clear();
    tone(PinoBuzzer, 1500, 50);
    Set_state(10); // exibe o menu (TEMPERATURA)
    break;
  case bEsq:
    lcd.clear();
    tone(PinoBuzzer, 1500, 50);
    Set_state(8); // retorna ao menu (CARGA - 3)
    break;
  case bSobe: // botao bSobe e bDesce, muda os valores
    if (val4 < 100 )
    {
      val4 = val4 + 5;
      Disparo[3] = 100 - val4;
      tone(PinoBuzzer, 1500, 50);
    }
    break;
  case bDesce:
    if (val4 > 0)
    {
      val4 = val4 - 5;
```

```

        Disparo[3] = 100 - val4;
        tone(PinoBuzzer, 1500, 50);
    }
    break;
default:
    Set_state(9); // sempre vai exibir as informacao do nemu
}
break;
case 10: // menu (TEMPERATURA) esta opcao mostra a
    switch (CheckButton())
    {
        case bDir:
            lcd.clear();
            tone(PinoBuzzer, 1500, 50);
            Set_state(5); // retorna ao menu (CARGAS)
            break;
        case bEsq:
            if (MenuCarga == 1) // verifica se a variavel esta em 1 do men
            {
                lcd.clear();
                tone(PinoBuzzer, 1500, 50);
                Set_state(12); // exibe menu (AJUSTE TEMP)
            }
            else if (MenuCarga == 0)
            {
                lcd.clear();
                tone(PinoBuzzer, 1500, 50);
                Set_state(9); // retorna ao menu (CARGA - 4)
            }
            break;
        default:
            Set_state(10); // sempre vai exibir as informacao do nemu
    }
    break;
case 11: // menu (AUTOMACAO) neste menu pode mudar
    switch (CheckButton())
    {
        case bDir:
            if (MenuCarga == 1) // verifica se o estado do MenuCarga e igu
            {
                lcd.clear();
                tone(PinoBuzzer, 1500, 50);
                Set_state(12); // exibe o menu (AJUSTE TEM)
            }
            else if (MenuCarga == 0) // verifica se o estado do MenuCarga e igu
            {
                lcd.clear();
                tone(PinoBuzzer, 1500, 50);
                Set_state(6); // exibe menu (CARGA - 1)
            }
            break;
        case bEsq:
            lcd.clear();
            tone(PinoBuzzer, 1500, 50);
            Set_state(5); // retorna ao meu (CARGAS)
            break;
        case bSobe: // importante! o botao bSobe deste menu (A

```

```

if (MenuCarga == 0) // verifica MenuCarga = 0 se tiver faca
{
    MenuCarga = 1; // atribuiu o MenuCarga = 1 (LIGADO)

    val1 = 100-Disparo[0];
    val2 = 100-Disparo[1];
    val3 = 100-Disparo[2];
    val4 = 100-Disparo[3];
    lcd.setCursor(0, 1);
    lcd.print("LIGADO "); // imprime na tela LCD "LIGADO"
    tone(PinoBuzzer, 1500, 50);
}
break;
case bDesce: // importante! o botao bDesce deste menu (
if (MenuCarga == 1) // verifica MenuCarga = 1 se tive faca
{
    MenuCarga = 0; // atribuiu o MenuCarga = 0 (DESLIGADO)
    val1 = 100-Disparo[0];
    val2 = 100-Disparo[1];
    val3 = 100-Disparo[2];
    val4 = 100-Disparo[3];
    lcd.setCursor(0, 1);
    lcd.print("DESLIGADO ");
    tone(PinoBuzzer, 1500, 50);
}
break;
default:
    Set_state(11); // sempre vai exibir as informacao do nemu
    // esta funcao imprime no LCD em estado esta o menu (AUTOMACAO) que pode ser (LIGADO/
if (MenuCarga == 1) // verifica MenuCarga = 1 se tiver faca
{
    lcd.setCursor(0, 1);
    lcd.print("LIGADO "); // imprime na tela LCD "LIGADO"
}
if (MenuCarga == 0) // verifica MenuCarga = 0 se tiver faca
{
    lcd.setCursor(0, 1);
    lcd.print("DESLIGADO "); // imprime na tela LCD "DESLIGADO"
} // aqui termina o controle sobre as cargas
// [ CARGAS -> AUTOMACAO <> CARGA - 1 <> C
}
break;
case 12: // menu (AJUSTE TEMP) aqui faz o ajuste da
switch (CheckButton())
{
case bDir:
    lcd.clear();
    tone(PinoBuzzer, 1500, 50);
    Set_state(10); // exhibe o menu (TEMPERATURA)
    break;
case bEsq:
    lcd.clear();
    tone(PinoBuzzer, 1500, 50);
    Set_state(11); // retorna ao menu (AUTOMACAO)

    break;
case bSobe:
    if (ConfigTemp < 100) // verifica se a o valor da variavel Config

```

```

    {
        ConfigTempMuda++;
        lcd.setCursor(0, 1);
        tone(PinoBuzzer, 1500, 50);
    }
    break;
case bDesce:
    if (ConfigTempMuda > 0) // verifica se a o valor da variavel C
    {
        ConfigTempMuda--; // subtrai 1 em 1 a variavel ConfigTemp
        lcd.setCursor(0, 1);
        tone(PinoBuzzer, 1500, 50);
    }
    break;
default:
    Set_state(12); // sempre vai exibir as informacao do nemu
} // aqui termina o controle sobre as cargas
break;
case 13: // menu (MONITORAMENTO TEMPERATURA)
    switch (CheckButton())
    {
        case bDir:
            lcd.clear();
            tone(PinoBuzzer, 1500, 50);
            Set_state(14);
            break;
        case bSobe:
            lcd.clear();
            tone(PinoBuzzer, 1500, 50);
            Set_state(5); //exibe o menu (CARGAS CONFIGURACAO)
            break;
        default:
            Set_state(13);
    }
    break;
case 14: // menu (MONITORAMENTO CONTADOR)
    switch (CheckButton())
    {
        case bDir:
            lcd.clear();
            tone(PinoBuzzer, 1500, 50);
            Set_state(15);
            break;
        case bEsq:
            lcd.clear();
            tone(PinoBuzzer, 1500, 50);
            Set_state(13); //exibe o menu (CARGAS CONFIGURACAO)
            break;
        default:
            Set_state(14);
    }
    break;
case 15: // menu (MONITORAMENTO CONTADOR)
    switch (CheckButton())
    {
        case bDir:
            lcd.clear();

```

```

        tone(PinoBuzzer , 1500, 50);
        Set_state(16);
        break;
    case bEsq:
        lcd.clear();
        tone(PinoBuzzer , 1500, 50);
        Set_state(14);
        break;
        //exibe o menu (CARGAS CONFIGURACAO)
    default:
        Set_state(15);
    }
    break;

case 16:
    // menu (MONITORAMENTO CONTADOR)
    switch (CheckButton())
    {
        case bDir:
            lcd.clear();
            tone(PinoBuzzer , 1500, 50);
            Set_state(13);
            break;
        case bEsq:
            lcd.clear();
            tone(PinoBuzzer , 1500, 50);
            Set_state(15);
            //exibe o menu (CARGAS CONFIGURACAO)
            break;
        default:
            Set_state(16);
    }
    break;
default: ;
}
}
//-----
char CheckButton()
// Checa botoes acionados aLeft Inical
{
    if (aLeft != digitalRead(bDir))
    // Se aLeft diferente de bDir
    { // Faca
        aLeft = !aLeft;
        // Inverste aLeft
        ConRegLCD = 10;
        if (aLeft) return bDir0;
        // Se aLeft diferente de 0 retorne bDir
        else return bDir;
        // Se aLeft igual a 0 retorne bDir
    }
    else
    // Se aLeft igual de bDir
    // Se aRight diferente de bEsq
    if (aRight != digitalRead(bEsq))
    // Se aRight diferente de bEsq
    { // Faca
        aRight = !aRight;
        // Inverste aRight
        ConRegLCD = 10;
        if (aRight) return bEsq0;
        // Se aRight diferente de 0 retorne bEsq
        else return bEsq;
        // Se aRight igual a 0 retorne bEsq
    }
    else
    // Se aRight igual de bEsq
    // Se aUp diferente de bSobe
    if (aUp != digitalRead(bSobe))
    // Se aUp diferente de bSobe
    { // Faca
        aUp = !aUp;
        // Inverste aUp
        ConRegLCD = 10;
        if (aUp) return bSobe0;
        // Se aUp diferente de 0 retorne bSobe0
    }
}

```

```

        else return bSobe; // Se aUp igual a 0 retorne bSobe
    }
    else // Se aUp igual de bSobe
        if (aDown != digitalRead(bDesce)) // Se aDown diferente de bDesce
        { // Faca
            aDown = !aDown; // Inverste aDown
            ConRegLCD = 10;
            if (aDown) return bDesce0; // Se aDown diferente de 0 retorne bDesce0
            else return bDesce; // Se aDown igual a 0 retorne bDesce
        }
    else // Se aDown igual de bDesce
        return 0; // Retorne 0
}
//-----
void Set_state(char index) // este switch serve para exibir os menus
{ // note que o variavel state recebe o mesmo valor do switch para que cada CASE receba sua per
    state = index; // Atualiza a variavel state para a nova tela

    switch (state) // verifica qual a tela atual e exibe o menu
    {
        case 1: //===== state 1 menu (CONTROLE)
            lcd.setCursor(0, 0);
            lcd.print("CONTROLE "); // imprime no LCD
            lcd.write(Desce); // imprime seta DESCE no LCD
            lcd.write(Esq); // imprime seta ESQUERDA no LCD
            lcd.write(Dir);
            lcd.setCursor(0, 1);
            break;
        case 2: //===== state 2 menu (ILUMINACAO)
            lcd.setCursor(0, 0);
            lcd.print("ILUMINACAO ");
            lcd.write(Sobe);
            lcd.write(Desce);
            // lcd.write(Esq);
            lcd.write(Dir);
            lcd.setCursor(0, 1);
            lcd.print("T. CONF:");
            lcd.print(ConRegreReg); // imprime o valor da variavel ConRegreReg
            lcd.print("s");
            break;
        case 3: //===== state 3 menu (CONF. TEMPO)
            lcd.setCursor(0, 0);
            lcd.print("CONF. TEMPO ");
            lcd.write(Sobe);
            lcd.write(Desce);
            lcd.write(Esq);
            lcd.write(Dir);
            lcd.setCursor(0, 1);
            lcd.print("TEMPO: ");
            lcd.print(ConRegreReg); // imprime o valor da variavel ConRegreReg
            lcd.print("s ");
            break;
        case 4: //===== state 4 menu (MONIT. TEMPO)
            lcd.setCursor(0, 0);
            lcd.print("MONIT. TEMPO ");
            lcd.write(Esq);
            lcd.write(Dir);
    }
}

```

```
    lcd.setCursor(0, 1);
    lcd.print("TEMPO: ");
    lcd.print(ConRegre); // imprime o valor da variavel ConRegre
    lcd.print("s");
    break;
case 5: //===== state 5 menu (CARGAS)
    lcd.setCursor(0, 0);
    lcd.print("VENTILADOR ");
    lcd.write(Sobe);
    lcd.write(Desce);
    // lcd.write(Esq);
    lcd.write(Dir);
    lcd.setCursor(0, 1);
    lcd.print("CONFIGURACAO");
    break;
case 6: //===== state 6 - menu (CARGA - 1)
    lcd.setCursor(0, 0);
    lcd.print("VENT. 1");
    lcd.setCursor(12, 0);
    lcd.write(4);
    lcd.write(3);
    lcd.write(2);
    lcd.write(1);
    lcd.setCursor(0, 1);
    lcd.print(val1, DEC); // mostra o valor de "variavel"
    lcd.print("% ");
    lcd.setCursor(3, 1);
    lcd.write(5); // imprime seta indicadora no LCD
    lcd.setCursor(4, 1);
    lcd.print(val2, DEC); // mostra o valor de "variavel"
    lcd.print("% ");
    lcd.setCursor(8, 1);
    lcd.print(val3, DEC); // mostra o valor de "variavel"
    lcd.print("% ");
    lcd.setCursor(12, 1);
    lcd.print(val4, DEC); // mostra o valor de "variavel"
    lcd.print("% ");
    break;
case 7: //===== state 7 menu (CARGA - 2)
    lcd.setCursor(0, 0);
    lcd.print("VENT. 2");
    lcd.setCursor(12, 0);
    lcd.write(4);
    lcd.write(3);
    lcd.write(2);
    lcd.write(1);
    lcd.setCursor(0, 1);
    lcd.print(val1, DEC); // mostra o valor de "variavel"
    lcd.print("% ");
    lcd.setCursor(4, 1);
    lcd.print(val2, DEC); // mostra o valor de "variavel"
    lcd.print("% ");
    lcd.setCursor(7, 1);
    lcd.write(5); // imprime seta indicadora no LCD
    lcd.setCursor(8, 1);
    lcd.print(val3, DEC); // mostra o valor de "variavel"
    lcd.print("% ");
```

```
    lcd.setCursor(12, 1);
    lcd.print(val4, DEC); // mostra o valor de "variavel"
    lcd.print("% ");
    break;
case 8: //===== state 8 menu (CARGA - 3)
    lcd.setCursor(0, 0);
    lcd.print("VENT. 3");
    lcd.setCursor(12, 0);
    lcd.write(4);
    lcd.write(3);
    lcd.write(2);
    lcd.write(1);
    lcd.setCursor(0, 1);
    lcd.print(val1, DEC); // mostra o valor de "variavel"
    lcd.print("% ");
    lcd.setCursor(4, 1);
    lcd.print(val2, DEC); // mostra o valor de "variavel"
    lcd.print("% ");
    lcd.setCursor(8, 1);
    lcd.print(val3, DEC); // mostra o valor de "variavel"
    lcd.print("% ");
    lcd.setCursor(11, 1);
    lcd.write(5); // imprime seta indicadora no LCD
    lcd.setCursor(12, 1);
    lcd.print(val4, DEC); // mostra o valor de "variavel"
    lcd.print("% ");
    break;
case 9: //===== state 9 menu (CARGA - 4)
    lcd.setCursor(0, 0);
    lcd.print("VENT. 4");
    lcd.setCursor(12, 0);
    lcd.write(4);
    lcd.write(3);
    lcd.write(2);
    lcd.write(1);
    lcd.setCursor(0, 1);
    lcd.print(val1, DEC); // mostra o valor de "variavel"
    lcd.print("% ");
    lcd.setCursor(4, 1);
    lcd.print(val2, DEC); // mostra o valor de "variavel"
    lcd.print("% ");
    lcd.setCursor(8, 1);
    lcd.print(val3, DEC); // mostra o valor de "variavel"
    lcd.print("% ");
    lcd.setCursor(12, 1);
    lcd.print(val4, DEC); // mostra o valor de "variavel"
    lcd.print("% ");
    lcd.setCursor(15, 1);
    lcd.write(5); // imprime seta indicadora no LCD
    break;
case 10: //===== state 10 menu (TEMPERATURA)
    lcd.setCursor(0, 0);
    lcd.print("TEMPERATURA ");
    lcd.write(Esq);
    lcd.write(Dir);
    lcd.setCursor(0, 1);
    if ((millis() - tBase) >= tInt)
```

```

    {
        tBase = millis();
        long xpto = (long)(temperature);
        lcd.print(xpto); // Imprime menor valor
        lcd.write(B11011111); // Simbolo de graus celsius
        lcd.print("C ");
    }
    break;
case 11: //===== state 11 menu (AUTOMACAO)
    lcd.setCursor(0, 0);
    lcd.print("AUTOMACAO ");
    lcd.write(Sobe);
    lcd.write(Desce);
    lcd.write(Esq);
    lcd.write(Dir);
    lcd.setCursor(0, 1);
    break;
case 12: //===== state 12 menu (AJUSTE TEMP)
    lcd.setCursor(0, 0);
    lcd.print("AJUSTE TEMP ");
    lcd.write(Sobe);
    lcd.write(Desce);
    lcd.write(Esq);
    lcd.write(Dir);
    lcd.setCursor(0, 1);
    lcd.print("BASE:");
    lcd.print(ConfigTemp); // imprime no LCD o valor de ConfigTemp
    lcd.write(B11011111); // Simbolo de graus celsius
    lcd.print("NOVO:");
    lcd.print(ConfigTempMuda); // imprime no LCD o valor de ConfigTemp
    lcd.write(B11011111); // Simbolo de graus celsius
    if (ConfigTempMuda == 0) // apenas uma brincadeira, se o ajuste chega
    { lcd.setCursor(5, 1);
      lcd.print("GELO ");
    }
    if (ConfigTempMuda == 100) // apenas uma brincadeira, se o ajuste cheg
    { lcd.setCursor(5, 1);
      lcd.print("EBULICAO ");
    }
    break;
case 13: //===== state 13 menu (MONITORACAO TEMPERATURA)
    lcd.setCursor(0, 0);
    lcd.print("MONITORAMENTO ");
    lcd.write(Sobe);
    lcd.write(Dir);
    lcd.setCursor(0, 1);
    lcd.print("TEMP: ");
    lcd.setCursor(6, 1);
    if ((millis() - tBase) >= tInt)
    {
        tBase = millis();
        long xpto = (long)(temperature);
        lcd.print(xpto); // Imprime menor valor
        lcd.write(B11011111); // Simbolo de graus celsius
        lcd.print("C ");
    }
    break;

```

```

case 14: //===== state 14 menu (MONITORACAO CONTADOR)
    lcd.setCursor(0, 0);
    lcd.print("MONITORAMENTO ");
    lcd.write(Esq);
    lcd.write(Dir);
    lcd.setCursor(0, 1);
    lcd.print("CONTADOR: ");
    lcd.setCursor(10, 1);
    lcd.print(ContaDesliga);
    break;
case 15: //===== state 15 menu (tempo decorrido)
    CalcTempDes();
    lcd.setCursor(0, 0);
    lcd.print("MONITORAMENTO ");
    lcd.write(Esq);
    lcd.write(Dir);
    lcd.setCursor(0, 1);
    lcd.print("T.DESL.: ");
    lcd.print(hor);
    lcd.print(":");
    lcd.print(min);
    lcd.print(":");
    lcd.print(seg);
    break;
case 16: //===== state 15 menu (tempo decorrido)
    Irms = emon1.calcIrms(1480);
    lcd.setCursor(0, 0);
    lcd.print("CORRENTE CONS.");
    lcd.write(Esq);
    lcd.write(Dir);
    lcd.setCursor(0, 1);
    lcd.print(Irms);
    lcd.print(" A");
    break;
default;;
}
}

void MemoriazaTemperatura()
{
    if ((millis() - Tempo) >= Cinco_Min) // Se passou tempo previsto
    {
        long xpto = (long)(temperature);
        TempMem = xpto; // Incrementa variavel
        Tempo = millis(); // Restaura tempo
    }
}

//-----
//Esta funcao controla automaticamente a velocidade dos ventiladores

void Controla_Vent_Temp() // define qual valor a variavel str1 va
{
    //O ResultTemp recebe a subtracao de ConfigTemp e ConfigTempMuda.
    //Por definicao o valor para inicio de ativacao dos ventiladores e 26 graus definido na varia
    //mas ConfigTempMuda e o valor que o usuario pode mudar, com isso o valor minimo para ativaca

```

```
int ResultTemp = ConfigTemp - ConfigTempMuda;           // o ConfigTemp e o valor inicial de 26
int NovaTemp = ConfigTemp + ResultTemp;
int Temp26 = 26 + ResultTemp;
int Temp27 = 27 + ResultTemp;
int Temp28 = 28 + ResultTemp;
int Temp29 = 29 + ResultTemp;

int verificaLuz = digitalRead(A1);                       // le o valor na saida PinoAtivaLuz par

if (verificaLuz == HIGH)                                 // verifica se a luz esta ou nao ligado
{
    if (Temp26 >= Temp26 )
    {
        if (TempMem == Temp26)
        {
            Disparo[0] = 50;
            Disparo[1] = 50;
            Disparo[2] = 50;
            Disparo[3] = 50;
        }
        else if (TempMem == Temp27)
        {
            Disparo[0] = 40;
            Disparo[1] = 40;
            Disparo[2] = 40;
            Disparo[3] = 40;
        }
        else if (TempMem == Temp28)
        {
            Disparo[0] = 30;
            Disparo[1] = 30;
            Disparo[2] = 30;
            Disparo[3] = 30;
        }
        else if (TempMem == Temp29)
        {
            Disparo[0] = 20;
            Disparo[1] = 20;
            Disparo[2] = 20;
            Disparo[3] = 20;
        }
        else if (TempMem >= 30)
        {
            Disparo[0] = 10;
            Disparo[1] = 10;
            Disparo[2] = 10;
            Disparo[3] = 10;
        }
        else
        {
            Disparo[0] = 0;
            Disparo[1] = 0;
            Disparo[2] = 0;
            Disparo[3] = 0;
        }
    }
}
```

```

else
if (verificaLuz == LOW)
{
    Disparo[0] = 0;
    Disparo[1] = 0;
    Disparo[2] = 0;
    Disparo[3] = 0;
}
}

void CalcTempDes() // define qual valor a variavel str1 va
{
    if (JaContei == 1)
    {
        static unsigned long ult_tempo = 0;
        int tempo = millis();
        if (tempo - ult_tempo >= 1000) {
            ult_tempo = tempo;
            seg++;
        }
        if (seg >= 60) {
            seg = 0; min++;
        }
        if (min >= 60) {
            min = 0; hor++;
        }
        if (hor >= 24) {
            hor = 0; min = 0;
        }
    }
}
//-----
void Timer1Zr() // Rotina chamada pelo Interrupt de Zer
{
    Contador++; // Incrementa contador para contar
    if (Contador == 13333) // Valor de contagem para 1 segundo
    {
        ConRegre--; // Decrementa tempo programado
        ConRegLCD--; // Se chegar em zero, fica em zero
        if (ConRegre <= 0) ConRegre = 0;
        Contador = 0; // Se chegar em zero, fica em zero
        if (ConRegLCD <= 0) ConRegLCD = 0; // Zera contador// Zera contador
        // Contador = 0;
    }
    Pulso_Timer1++; // Incrementa o tempo de controle de di

    if (Disparo[0] == Pulso_Timer1) // Se o atraso do Triac 0 for igual a t
    {
        digitalWrite(Canal1, HIGH); // Dispara Triac
        delayMicroseconds(8.33); // Tempo de disparo ligado
        digitalWrite(Canal1, LOW); // Corta disparo Triac
    }
    if (Disparo[3] == Pulso_Timer1) // Se o atraso do Triac 1 for igual a t
    {
        digitalWrite(Canal2, HIGH); // Dispara Triac
        delayMicroseconds(8.33); // Tempo de disparo ligado
        digitalWrite(Canal2, LOW); // Corta disparo Triac
    }
}

```

```

}
if (Disparo[2] == Pulso_Timer1) // Se o atraso do Triac 2 for igual a t
{
    digitalWrite(Canal3, HIGH); // Dispara Triac
    delayMicroseconds(8.33); // Tempo de disparo ligado
    digitalWrite(Canal3, LOW); // Corta disparo Triac
}
if (Disparo[1] == Pulso_Timer1) // Se o atraso do Triac 3 for igual a t
{
    digitalWrite(Canal4, HIGH); // Dispara Triac
    delayMicroseconds(8.33); // Tempo de disparo ligado
    digitalWrite(Canal4, LOW); // Corta disparo Triac
}
}
//-----
void zero_cross() // Senoide Passou pelo zero
{
    Pulso_Timer1 = 0; // Zera contador de tempo
}

float getTemp() {
    //returns the temperature from one DS18S20 in DEG Celsius

    byte data[12];
    byte addr[8];

    if ( !ds.search(addr)) {
        //no more sensors on chain, reset search
        ds.reset_search();
        return -1000;
    }

    if ( OneWire::crc8( addr, 7) != addr[7]) {
        // Serial.println("CRC is not valid!");
        return -1000;
    }

    if ( addr[0] != 0x10 && addr[0] != 0x28) {
        // Serial.print("Device is not recognized");
        return -1000;
    }

    ds.reset();
    ds.select(addr);
    ds.write(0x44, 1); // start conversion, with parasite power on at the end

    byte present = ds.reset();
    ds.select(addr);
    ds.write(0xBE); // Read Scratchpad

    for (int i = 0; i < 9; i++) { // we need 9 bytes
        data[i] = ds.read();
    }

    ds.reset_search();

```

```
byte MSB = data[1];
byte LSB = data[0];

float tempRead = ((MSB << 8) | LSB); //using two's compliment
float TemperatureSum = tempRead / 16;

return TemperatureSum;
}
```