



CENTRO UNIVERSITÁRIO DE BRASÍLIA - UniCEUB
CURSO DE ENGENHARIA DE COMPUTAÇÃO

THIAGO SILVINO ROCHA OLIVEIRA

**SISTEMA DE MONITORAMENTO E CONTROLE DE ACIONAMENTO DE
DISPOSITIVO UMIDIFICADOR**

Orientadora: Prof.^a MsC Maria Marony Sousa Farias

Brasília
Junho, 2012

THIAGO SILVINO ROCHA OLIVEIRA

**SISTEMA DE MONITORAMENTO E CONTROLE DE ACIONAMENTO DE
DISPOSITIVO UMIDIFICADOR**

Trabalho apresentado ao Centro
Universitário de Brasília como pré-
requisito para a obtenção de
Certificado de Conclusão de Curso
de Engenharia de Computação.

Orientadora: Prof.^a MsC Maria
Marony Sousa Farias

Brasília

Junho, 2012

THIAGO SILVINO ROCHA OLIVEIRA

**SISTEMA DE MONITORAMENTO E CONTROLE DE ACIONAMENTO DE
DISPOSITIVO UMIDIFICADOR**

Trabalho apresentado ao Centro
Universitário de Brasília como pré-
requisito para a obtenção de
Certificado de Conclusão de Curso
de Engenharia de Computação.

Orientadora: Professora MS Maria
Marony Sousa Farias

Este Trabalho foi julgado adequado para a obtenção do Título de Engenheiro de Computação, e aprovado em sua forma final pela Faculdade de Tecnologia e Ciências Sociais Aplicadas -FATECS.

Prof. Abiezer Amarilia Fernandes
Coordenador do Curso

Banca Examinadora:

Prof.^a Maria Marony Sousa Farias, mestre em Engenharia Elétrica.
Orientadora

Prof. Carlo Kleber da Silva Rodrigues, doutor em Engenharia de Sistemas e Computação.
UniCEUB

Prof. Sidney Cerqueira Bispo dos Santos, doutor em Telecomunicações.
UniCEUB

AGRADECIMENTOS

Primeiramente, a DEUS por toda a força que depositou em mim possibilitando-me alcançar sonhos, muitas das vezes tão distantes.

A minha mãe e ao meu pai (in memoriam) que de vocês, recebi o dom mais precioso do Universo: a vida, como também o respeito e a educação, por isto serei infinitamente grato.

Aos meus indispensáveis irmãos Joyce e Lucas que sempre estiveram presentes e me auxiliaram em todos os momentos.

Aos meus familiares e tios, em especial Custoneide e Custódio: se hoje cheguei onde estou devo a eles.

A minha orientadora Maria Marony Sousa Farias pela disponibilidade em me auxiliar.

Thiago Silvino Rocha Oliveira

SUMÁRIO

AGRADECIMENTOS	IV
SUMÁRIO	V
LISTA DE ILUSTRAÇÕES	VII
LISTA DE TABELAS	IX
RESUMO.....	X
ABSTRACT	XI
CAPÍTULO 1 - INTRODUÇÃO	12
1.1 APRESENTAÇÃO DO PROBLEMA	12
1.1.1 <i>Umidade do ar e interações</i>	13
1.1.2 <i>Umidade do ar em ambiente climatizado para o ser humano</i>	13
1.2 OBJETIVOS DO TRABALHO	14
1.3 JUSTIFICATIVA E IMPORTÂNCIA DO TRABALHO.....	14
1.4 ESCOPO DO TRABALHO	15
1.5 RESULTADOS ESPERADOS	15
1.6 ESTRUTURA DO TRABALHO.....	15
CAPÍTULO 2 - REFERENCIAL TEÓRICO E TECNOLÓGICO	17
2.1 ATMOSFERA TERRESTRE	17
2.2 UMIDADE RELATIVA DO AR.....	18
2.2.1 <i>Instrumentos de aferição de umidade</i>	19
2.3 SENSORES CAPACITIVOS DE UMIDADE DO AR.	21
2.3.1 <i>Sensor de umidade SHT15</i>	22
CAPÍTULO 3 - SISTEMA DE MONITORAMENTO E CONTROLE DE ACIONAMENTO DE DISPOSITIVO UMIDIFICADOR.....	30
3.1 DO HARDWARE.....	30
3.1.1 <i>Sensoriamento</i>	30
3.1.2 <i>Sistema de Controle</i>	31
3.1.3 <i>Alimentação do dispositivo</i>	34
3.2 DO FIRMWARE	39

3.2.1	<i>Sensoriamento</i>	39
3.2.2	<i>Controle e respostas</i>	43
3.3	DA IMPLEMENTAÇÃO	46
	CAPÍTULO 4 - TESTES E APLICAÇÃO PRÁTICA DO SISTEMA	50
4.1	AMBIENTE DE TESTES.....	50
4.2	DESCRIÇÃO DO SISTEMA EM AMBIENTE DE TESTES.....	53
4.3	RESULTADOS DOS TESTES	55
4.4	CUSTOS	56
4.5	ANÁLISE DO SISTEMA	56
	CAPÍTULO 5 - CONCLUSÃO	58
5.1	CONCLUSÕES	58
5.2	SUGESTÕES PARA TRABALHOS FUTUROS	58
	REFERÊNCIAS	59
	APÊNDICE	61
	<i>FIRMWARE DO MICROCONTROLADOR</i>	61

LISTA DE ILUSTRAÇÕES

Figura 2.1 – Gráfico de pressão de saturação de vapor de água por temperatura do ar.	Página 18.
Figura 2.2 – Psicrômetro.	Página 20.
Figura 2.3 – Higrômetro de fios de cabelo.....	Página 20.
Figura 2.4 – Termômetro e Higrômetro Digitais.....	Página 21.
Figura 2.5 – Sensor SMD SHTx	Página 22.
Figura 2.6 – Precisão do valor a umidade do ar do sensores SHT1x...Página 23.	
Figura 2.7 – Gráfico de precisão do valor temperatura aferida pelo SHT1x.	Página 23.
Figura 2.8 – Dimensões em milímetros do SHT11.	Página 24.
Figura 2.9– Recomendação de ligação para sensores SHT1x.....	Página 25.
Figura 2.10– Sequência de início do sensor.	Página 25.
Figura 2.11– Exemplo de sequência de um comando.	Página 27.
Figura 2.12– Gráfico de demonstração do ajuste para Umidade Relativa.	Página 28.
Figura 3.1 – Conectores RJ-11	Página 31.
Figura 3.2 – Pinos centrais de um conector RJ-11.....	Página 31.
Figura 3.3 - Microcontrolador atmega48.	Página 32.
Figura 3.4 - Ligação entre Atmega48 e <i>display</i> LCD 20x4 (4 bits)....	Página 32.
Figura 3.5 - Ligação entre Atmega48 e sensor SHT15.	Página 33.
Figura 3.6 - Interface de botões e controle do umidificador.....	Página 33.
Figura 3.7 – Chave seletora 220 ou 110.	Página 34.
Figura 3.8 – Circuito de alimentação.	Página 35.
Figura 3.9 – Simulação completa do circuito.	Página 36.
Figura 3.10 – Circuito de alimentação (Trafo).	Página 36.

Figura 3.11 – Resultado final do dispositivo construído.	Página 37.
Figura 3.12 – Resultado final do dispositivo construído 2.	Página 38.
Figura 3.13 – Sequência de início do sensor.	Página 39.
Figura 3.14 – Interface de botões para configuração do umidificador..	Página 45.
Figura 3.15 – LCD com informações do controle.	Página 46.
Figura 3.16 – ISIS Professional com simulação do sistema de controle.	Página 47.
Figura 3.17 – Ares Professional com designer do layout do circuito...	Página 48.
Figura 3.18 – Programador USB de microcontrolador AVR.....	Página 48.
Figura 3.19– Extreme Burn-AVR	Página 49.
Figura 4.1 – Desumidificação utilizando NaOH	Página 50.
Figura 4.2 – Alteração do circuito de acionamento do chuveiro	Página 51.
Figura 4.3 – Entupimento das saídas de água e acoplamento de cano para saída de vapor.	Página 52.
Figura 4.4 – Ligação entre o umidificador e ambiente de testes.	Página 52.
Figura 4.5 – Configuração inicial do dispositivo.	Página 53.
Figura 4.6 – Monitorando umidade e temperatura em ambiente aberto.....	Página 53.
Figura 4.7 – Monitorando umidade e temperatura no ambiente durante processo de desumidificação.	Página 54.
Figura 4.8 – Acionando umidificador quando umidade é menor que 40%.	Página 54.
Figura 4.9 – Umidificador desativado, umidade de 84% e temperatura de 37° Celsius.....	Página 55.

LISTA DE TABELAS

Tabela 2.1: Constituição atmosférica	Página 17.
Tabela 2.2 – Comandos do SHT15.....	Página 26.
Tabela 2.3: Coeficientes de conversão de umidade da 4ª versão do Sht1x.	Página 27.
Tabela 2.4: Coeficiente para correção de resposta de umidade relativa.....	Página 28
Tabela 2.5: Relação de tensão de alimentação (VDD) e coeficiente de temperatura d1.....	Página 29.
Tabela 2.6 – Relação à acurácia do sensor e coeficiente de temperatura d2.....	Página 29.
Tabela 3.1: Comandos do SHT15.	Página 39
Tabela 4.1: Custos do Projeto.	Página 56

RESUMO

O presente trabalho propõe um dispositivo capaz de monitorar a umidade e a temperatura do ar e acionar umidificadores, dada uma condição preestabelecida. A umidade relativa do ar muito baixa é prejudicial à saúde e é comum a utilização de umidificadores de ar em épocas muito secas. Contudo, manter um umidificador ligado o dia inteiro poder ser prejudicial à saúde, além de representar um gasto, com energia elétrica, desnecessário. O dispositivo, resultado desse trabalho, foi concebido com uma interface para que o usuário possa programar parâmetros para controlar o acionamento de umidificadores. A possibilidade de determinar parâmetros de acionamento associada à capacidade de carga de 3300 Watts flexibiliza o usos desse dispositivo em diversos ambientes.

Palavras-chave: Sensor, temperatura, umidade relativa, SHT15, ATMEGA

ABSTRACT

This project proposes a device capable of monitoring the humidity and air temperature and trigger humidifiers, given a predetermined condition. The relative humidity is very low hazard to health and is common to use air humidifiers in very dry seasons. But, keep a humidifier on all day can be harmful to health, besides being an expense, with electricity, unnecessary. The device, result of this work, was designed with an interface that the user can set parameters for controlling the drive of humidifiers. The possibility of determining parameters associated with the drive capacity of 3300 Watts eases the use of this device in various environments.

Keywords: sensor, temperature, relative humidity, SHT15, ATMEGA 48.

CAPÍTULO 1 - INTRODUÇÃO

Em diferentes ambientes a quantidade de água presente no ar pode influir positiva ou negativamente na manutenção adequada da vida, substâncias ou objetos. É possível, em ambientes fechados, elevar a umidade relativa do ar por meio de dispositivos umidificadores, contudo, muito desses dispositivos disponíveis no mercado não possuem um sistema de controle de acionamento ou desligamento para quando a umidade do ar estiver em determinados patamares.

Manter um umidificador ligado confere um custo com energia elétrica. Caso ligado sem interrupção, o uso de umidificadores pode representar um gasto desnecessário e um excesso de umidade. Controlar os níveis de umidade relativa é determinante para qualidade do ar de um ambiente adequado, seja ele, para o homem ou para substâncias.

Esse trabalho apresenta um dispositivo monitor de temperatura e umidade no qual o usuário pode predeterminar critérios de acionamento e desativação de umidificadores a ele conectados. A possibilidade de determinar quando o umidificador estará ligado possibilitará o uso do dispositivo em diversos ambientes.

1.1 Apresentação do Problema

O planeta terra é muito vasto e em cada ponto da atmosfera pode-se detectar diversos níveis de vapor de água. A umidade relativa do ar está relacionada a diversos fatores tais como temperatura, altitude, ventos, época do ano, proximidade de rios lagos ou mares, arborização, substâncias químicas, etc. Por isso, a umidade relativa do ar varia muito em relação ao local e à época em que se a afere.

A quantidade de vapor de água no ar é determinante para a melhor manutenção da vida de organismos, processos de corrosão ou oxidação, resistência elétrica do ar, processos de produção ou conservação de alimentos, entre outros. Faz-se, então, necessário um local em que se obtenha condições ideais de umidade para cada finalidade, um ambiente interno de umidade controlada.

Cada ambiente interno, dependendo do seu desígnio, pode requerer diferentes níveis de temperatura, umidade ou ventilação; ou seja, é necessário climatizá-lo conforme a aplicação. Ambientes climatizados são recintos fisicamente delimitados e caracterizados por

dimensões e instalações próprias, submetidos ao processo de climatização, através de equipamentos [Re. Nº 9 ANVISA, 2003].

1.1.1 Umidade do ar e interações

A umidade relativa do ar, capacidade higrométrica do ar, influi em diversos processos; conservação dos suportes informacionais, amadurecimento de vinhos, conservação de alimentos ou sementes e salubridade de ambientes para o homem.

Arquivos, biblioteca e museus são exemplos típicos de ambientes que requerem controle de umidade para a preservação de seus acervos ou obras. Existem diversos estudos que associam a temperatura e a umidade a maior durabilidade de papéis e sendo o papel a grande maioria dos suportes materiais de arquivos e bibliotecas é necessário que se mantenha um controle nas variáveis temperatura e umidade do ar. Conforme recomendado no manual de gestão de documentos do estado do Paraná devem ser mantidas com índices de 20°C (Celsius) de temperatura e 50% de Umidade Relativa do Ar evitando que aceleração de processos químicos de deterioração, a proliferação de insetos e o ataque de microrganismos (fungos e bactérias). Ressalta-se que documentos iconográficos como fotografias, negativos e microfilmes necessitam de um local específico com temperatura mais baixa. [DERP - PA, 2007]

1.1.2 Umidade do ar em ambiente climatizado para o ser humano

As variáveis temperatura e umidade estão intrinsecamente ligadas à sensação térmica e ao conforto e qualidade de vida das pessoas em um determinado ambiente. Essas variáveis também estão vinculadas à proliferação de fungos e agravamento de situações asmáticas.

Segundo a resolução nº 9 de 2003 da Agencia Nacional de Vigilância Sanitária, em um ambiente climatizado artificialmente, de uso público e coletivo, a faixa recomendável de operação da umidade relativa do ar e temperatura, nas condições internas para verão, deverá variar de 40% a 65% e 23°C a 26°C, respectivamente. Para condições internas para inverno, a faixa recomendável de operação deverá variar de 35% a 65% e 26,5°C a 27°C. Vale ressaltar que a seleção da faixa depende da finalidade e do local da instalação, em ambientes de arte, por exemplo, recomenda-se de 40% a 55% de umidade relativa e temperaturas entre 21°C e 23°C durante todo o ano. [Re. Nº9 ANVISA, 2003]

Para o ser humano a qualidade do ar em um ambiente interno está vinculada a diversos fatores, tais como temperatura, pressão, compostos químicos ou umidade relativa do ar. Doenças podem surgir ou serem agravadas em ambientes muito secos ou muito úmidos, como, por exemplo, “a asma não alérgica que pode ser causada por circunstâncias como baixa umidade do ar” [GALVÃO, 2009].

Ressecamento das vias respiratórias, falta de ar, nariz entupido e garganta seca. Os incômodos provocados pela baixa umidade relativa do ar, típica deste período, costumam atingir grande parte da população. Mas para os idosos os dias muito secos e o longo período de estiagem podem causar problemas muito mais graves. Segundo um estudo realizado pela Universidade de São Paulo (USP), o risco de infarto em pessoas com mais de 65 anos aumenta em 30% em dias muito secos. [ROMAN, 2010]

Entretanto, não somente o tempo seco é prejudicial, a umidade em excesso também pode tornar-se maléfica por propiciar o surgimento de mofo e bolores no ambiente e esses o agravamento, bem como o aparecimento, de doenças respiratórias [JAAKKOLA et al., 2002].

1.2 Objetivos do Trabalho

O controle de dispositivos umidificadores, por meio de parâmetros predeterminados, é a solução proposta neste trabalho. O sistema sugerido tem o intuito de manter a umidade relativa do ar de um ambiente acima de uma taxa escolhida previamente. Como objetivos específicos neste trabalho têm-se:

- Exposição do funcionamento de sensores capacitivos de umidade, especificamente o SHT 15 produzido pela empresa Sensirios;
- Apresentar uma solução utilizando microcontroladores AVR (Atmega48);
- Desenvolver um ambiente de simulação em que se possa elevar a umidade relativa quando estiver abaixo de uma taxa da qual o usuário determine.

1.3 Justificativa e Importância do Trabalho

A relevância do trabalho dar-se à necessidade de, em ambientes fechados específicos, manter um percentual mínimo e máximo de vapor de água no ar por meio de um

umidificador, visto que muitas vezes os umidificadores disponíveis no mercado não dispõem de controle de acionamento e desligamento do dispositivo e esse controle é executado pelo ser humano. A interferência direta e repetitiva do homem no controle de um sistema de umidificação do ar pode ser evitada e às vezes necessária, seja por requisitos de segurança, de salubridade, confiabilidade ou disponibilidade. O dispositivo proposto além efetuar o controle da necessidade que o justifica apresenta uma solução microcontrolada com uma interface LCD e uma comunicação serial do tipo *half-duplex* com um sensor tornando-o, também, um recurso de consulta no desenvolvimento de trabalhos que utiliza soluções semelhantes.

1.4 Escopo do Trabalho

O trabalho se limita a monitorar a temperatura e a umidade relativa de algum ambiente e caso a umidade detectada esteja abaixo de um nível programado pelo usuário, o controlador do sistema ativa uma das portas responsáveis por ligar o umidificador.

Caso o umidificador já esteja ligado, o mesmo só será desativado se a umidade aferida estiver acima de uma taxa máxima de umidade programada pelo usuário.

1.5 Resultados Esperados

Almeja-se com o presente trabalho desenvolver uma solução de controle de umidade bastante flexível, possibilitando o uso do dispositivo em diferentes ambientes. O aparelho construído deve permitir a definição de parâmetros de umidade por meio de uma interface com o usuário.

1.6 Estrutura do Trabalho

Este trabalho é composto por cinco capítulos, incluindo essa introdução que delimita a abrangência do trabalho desenvolvido expondo a relevância, objetivos e motivação do projeto.

O Capítulo 2 é um arcabouço teórico necessário para o entendimento e desenvolvimento do projeto. Trata-se de esclarecimentos teóricos de umidade relativa do ar, dispositivos componentes do projeto, métodos utilizados e a justificativa da escolha dos métodos e dos componentes do projeto.

O Capítulo 3 contempla o modelo adotado para a resolução do problema de monitorar e controlar a variável umidade relativa. Esse capítulo está dividido em hardware projetado e *firmware* desenvolvido.

O ambiente e o umidificador construídos para realização de testes estão descritos no Capítulo 4, o qual apresenta também os procedimentos e os resultados dos testes.

O Capítulo 5 versa sobre as conclusões e possibilidade de projetos futuros relacionados a esse trabalho.

CAPÍTULO 2 - REFERENCIAL TEÓRICO E TECNOLÓGICO

Neste capítulo são expostos as tecnologias e os fundamentos teóricos para o correto entendimento da metodologia adotada no desenvolvimento do projeto.

2.1 Atmosfera Terrestre

A atmosfera da terra é um aglomerado de gases, partículas e vapor de água que envolve o planeta, considerada com cerca de 80 a 100 quilômetros de espessura. Caracteriza-se por ser um sistema aberto, plurifásico e multicomponente, a atmosfera terrestre é composta em por nitrogênio, oxigênio, argônio, dióxido de carbono, e outros gases juntamente com vapor de água conforme disposto na tabela a seguir. [SILVA, 2006]

Tabela 2.1 – Constituição atmosférica.

CONSTITUINTES	PORCENTAGEM
NITROGÊNIO (N)	78,00
OXIGÊNIO (O ²)	20,90
ARGÔNIO (Ar)	0,90
DIÓXIDO DE CARBONO (CO ²)	0,03
NEÔNIO (Ne)	18,18 x 10 ⁻⁴
HÉLIO (He)	5,24 x 10 ⁻⁴
KRIPTÔNIO (Kr)	1,14 x 10 ⁻⁴
XENÔNIO (Xe)	0,89 x 10 ⁻⁴
HIDROGÊNIO (H ²)	0,50 x 10 ⁻⁴
METANO (CH ⁴)	1,50 x 10 ⁻⁴
ÓXIDO NITROSO (N ² O)	0,27 x 10 ⁻⁴
MONÓXIDO DE CARBONO (CO)	0,19 x 10 ⁻⁴
VAPOR D'ÁGUA (H ² O)*	0,0 a 0,04
OZÔNIO (O ³)*	0,0 a 12,00 x 10 ⁻⁴
DIÓXIDO DE ENXOFRE (SO ²)*	0,10 x 10 ⁻⁶
DIÓXIDO DE NITROGÊNIO (NO ²)*	0,10 x 10 ⁻⁶
AMÔNIA (NH ³)*	0,40 x 10 ⁻⁶
ÓXIDO NÍTRICO (NO)*	0,50 x 10 ⁻⁷
SULFITO DE HIDROGÊNIO (H ² S)*	0,50 x 10 ⁻⁸

* Constituintes de concentração variável

(Fonte: [LATORRE et al., 2002])

O vapor de água na atmosfera é um componente cuja concentração varia no tempo e espaço. Oriundo de processos de evaporação dos solos e das superfícies líquidas —

oceanos, mares, lagos rios, etc.— e também da transpiração dos animais e, especialmente, das plantas, o vapor de água se mistura ao ar por processo de difusão e é conduzido por correntes de ar aos diversos pontos da atmosfera. Posteriormente a atmosfera fornece água à superfície terrestre seja líquida, por meio das chuvas, ou sólida (gelo e neve). Esse conjunto de fenômenos naturais de evaporação e sublimação e precipitação da água forma o círculo hidrológico[LATORRE et al., 2002].

2.2 Umidade Relativa do Ar

A quantidade de vapor de água difundido em determinado gás ou conjunto de gases, como o ar atmosférico, é a umidade. A quantidade máxima de vapor de água que o ar comporta é limitada e é denominada concentração de saturação –ou pressão de saturação–. A pressão de saturação do vapor de água está vinculada a temperatura e obedece aproximadamente a equação a seguir [COLLISCHONN, 2011]:

$$P_s = 0,611 \cdot \exp\left(\frac{17,27 \cdot T}{237,3+T}\right) \quad (1)$$

Na qual P_s é a concentração de saturação em Pascal e T a temperatura em graus Celsius.

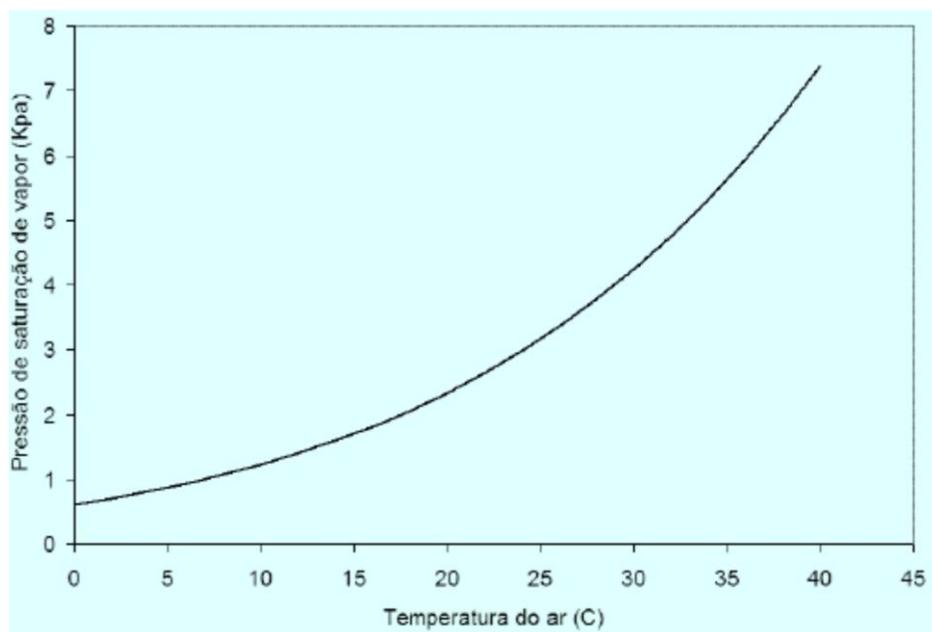


Figura 2.1 – Gráfico de pressão de saturação de vapor de água por temperatura do ar.

(Fonte: [COLLISCHONN, 2011])

A figura 2.1 apresenta graficamente a relação exponencial entre a pressão de saturação e a temperatura, de 0°C a 45°C , denotando que quanto maior a temperatura do ar maior será a capacidade desse de deter a água em seu estado gasoso.

A relação entre pressão parcial do vapor no ar ‘P’ e a pressão de saturação ‘ P_S ’ é a umidade relativa do ar ‘UR’:

$$UR = 100 \cdot \frac{P}{P_S}, \text{ em \%}. \quad (2)$$

Analogamente, a umidade relativa do ar pode se calculada relacionando a massa de vapor no ar ‘w’ pela massa de vapor no ponto de saturação w_S :

$$UR = 100 \cdot \frac{w}{w_S}, \text{ em \%}. \quad (3)$$

2.2.1 Instrumentos de aferição de umidade

Dois tipos de aparelhos destacam-se como medidores de umidade do ar são eles os higrômetros e os psicrômetros. O primeiro é um instrumento baseado em substâncias capazes de absorver a umidade do ar e o segundo na queda de temperatura devido à taxa de evaporação de água em um ambiente.

Um psicrômetro consiste de dois termômetros idênticos que são expostos ao ar. Um dos termômetros é chamado de bulbo seco e o outro, termômetro de bulbo úmido. O ar que passa pelo termômetro de bulbo úmido faz com que o vapor de água de um tecido umedecido que compõem esse termômetro evapore. O processo de evaporação causa uma queda de temperatura que é proporcional à secura do ar, quanto mais seco o ar, maior o resfriamento. Comparando a diferença de temperatura entre os dois termômetros pode-se calcular a umidade relativa do ar. [SILVA, 2006]

A figura 2.2 apresenta um psicrômetro com seus dois termômetros e uma tabela entre eles. Essa é a tabela de calibração, a diferença entre as leituras dos dois termômetros pode ser transformada em umidade quando associada ao respectivo valor da tabela.

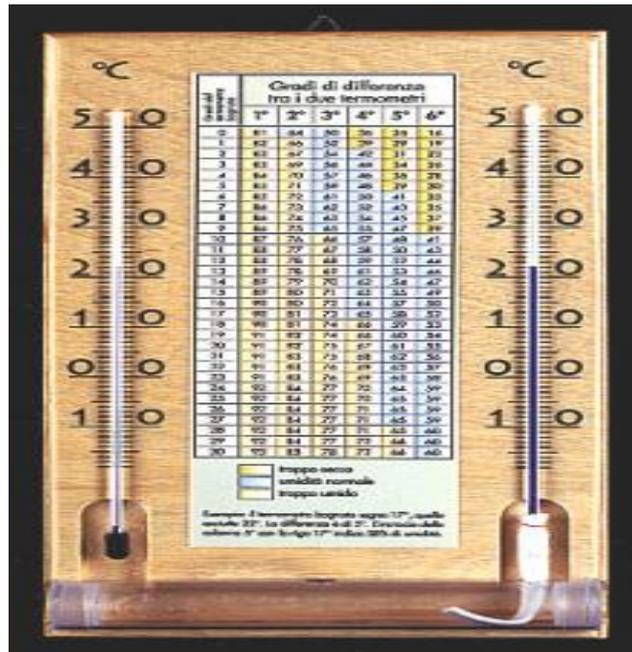


Figura 2.2 – Psicrômetro.

(Fonte: pt.wikipedia.org/wiki/Psicrômetro)

Frequentemente os higrômetros existentes utilizam sais de lítio, fios de cabelo ou células capacitivas polímeras. Quando a umidade do ar absorvida por essas substâncias varia ocorre também a variação de alguma característica física da substância. O fio de cabelo humano se alonga quando a umidade do ar se eleva e encolhe com o ar mais seco, devido a essa característica é utilizado em higrômetros, como o da figura 2.3:

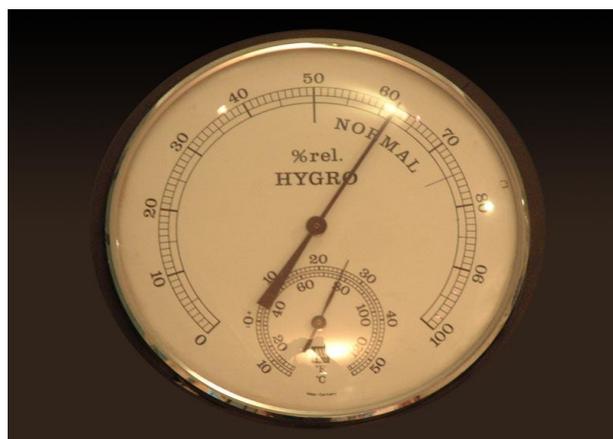


Figura 2.3 – Higrômetro de fios de cabelo.

(Fonte: <http://www.receita.fazenda.gov.br/Memoria/acervo/objetos/default.asp>)

A característica dos higrômetros baseados resistores de sais de lítio que se altera com a umidade é a condutividade elétrica, para isso um amperímetro com uma escala calibrada mede a corrente que passa pelo resistor, dada uma tensão constante e fornece os valores da umidade do ar. [INDRIUNAS, 2008]



Figura 2.4 – Termômetro e Higrômetro Digitais.

(Fonte: http://www.indupropil.com.br/produtos/0,17835_Termometro-e-Higrometro-Digital.html)

Os higrômetros digitais normalmente são construídos utilizando sensores de sais de lítio ou capacitivos. Nesses tipos de sensores há a necessidade de se calibrar a resposta conforme a temperatura e isso justifica o fato de que os higrômetros digitais também serem compostos por um termômetro, tal como o apresentado na figura 2.4.

O projeto proposto por este trabalho apresenta uma solução utilizando um sensor capacitivo, por isso foi elaborado o tópico a seguir tratará que tratará desse tipo de sensor.

2.3 Sensores capacitivos de umidade do ar.

Dentre os diversos sensores de umidade do ar existentes no mercado os sensores que utilizam células capacitivas são os mais usados na construção de higrômetros digitais.

A célula capacitiva desses sensores de umidade é constituída de um polímero dielétrico capaz de absorver moléculas de água. A quantidade de umidade absorvida pela célula determina a constante dielétrica do condensador, alterando a capacitância do sensor.

Para detectar a capacitância do sensor, este é atrelado a um circuito oscilador. A constante dielétrica da célula capacitiva determina a frequência de oscilação da saída do circuito, logo o teor de umidade do ar é então entendido como a resposta de oscilação [Gripp, 2010]. Entretanto, a temperatura do ambiente deve ser considerada para o cálculo correto da umidade relativa e cada fabricante de sensor disponibiliza uma equação relacionada à temperatura para essa correção.

Existem no mercado diversos sensores capacitivos como o HS1101, o 2322 691 90001, os DHTxx, e o SHTxx. A família de sensores SHTxx dispõem sensores de umidade e de temperatura integrados em um único circuito com uma saída de comunicação digital com um protocolo de comunicação muito semelhante ao protocolo de comunicação mestre-escravo IC2 desenvolvido pela Philips.

2.3.1 Sensor de umidade SHT15

O SHT15 é um componente SMD¹ para aferição de temperatura e umidade relativa do ar da família de sensores SHTxx da empresa suíça Sensirios. Esse componente é um chip de silício composto de uma célula capacitiva, com a finalidade de alterar sua capacitância conforme quantidade de vapor de água no ar, em consonância com o descrito no item 2.3, um sensor de temperatura do tipo *bandgap*², uma memória OTP³, um amplificador, um conversor analógico para digital e uma interface digital.

Esse sensor, vide figura 2.5, é previamente calibrado e os coeficientes de calibração são programados na memória OTP, entretanto, a precisão dessa calibração esta vinculada à tensão de alimentação. As características elétricas, tais como consumo de energia, entrada de nível alto e baixo e tensões de saída também dependem da tensão de alimentação.



Figura 2.5 – Sensor SMD SHTx. (Fonte: Folha de dados do fabricante)

¹ SMD: de “Superficial Monting Device” são comoponetes de montagem em superfície caracterizado pelo seu tamanho reduzido.

² Bandgap: Tipo comum de sensor de temperatura cuja maior vantagem é a de ser possível integra-lo a um circuito integrado de silício.

³ OTP: Memórias programáveis uma única vez, One Time Programmable

O sensor SHT15 é um dispositivo composto por dois sensores integrados para umidade relativa e temperatura capaz de medir a temperatura entre o -40°C e o 140°C e a umidade entre 0 e 100%. Dentro da família de sensores SHT1x o SHT15 apresenta a melhor precisão, contudo a precisão dos sensores é variável conforme a temperatura e umidade aferida. Para umidade relativa a precisão entre 10% e 90% é de mais ou menos 2%, podendo chegar a mais ou menos 4% nos outros valores como mostra os gráfico a seguir (Figura 2.6):

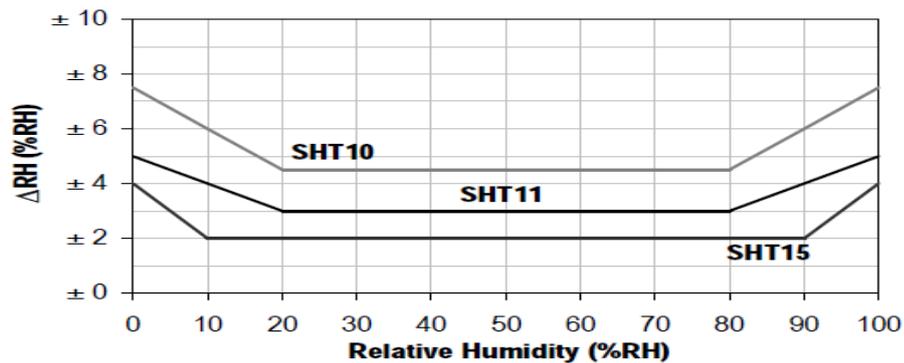


Figura 2.6 – Precisão do valor a umidade do ar dos sensores SHT1x.

(Fonte: Folha de dados do fabricante.)

Para temperatura a precisão é menos linear, mas apresenta uma ótima resposta comparada as dos outros sensores disponíveis no mercado. As temperaturas medidas entre -10°C e 60°C podem ter cerca de menos 1 grau de variação para mais ou para menos e dentro da faixa de operação do sensor a variação máxima é de $1,5^{\circ}\text{C}$, observe o gráfico a seguir (Figura 2.7):

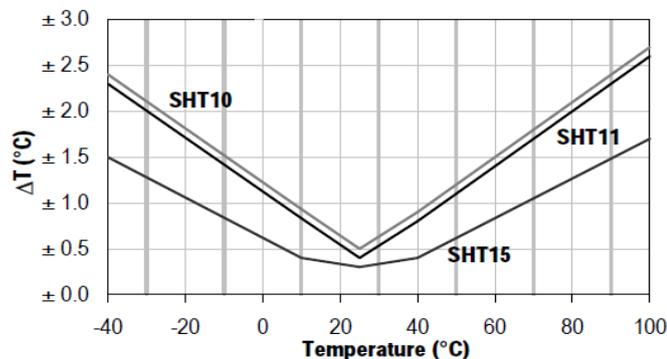


Figura 2.7 – Gráfico de precisão do valor temperatura aferida pelo SHT1x.

(Fonte: Folha de dados do fabricante.)

Fisicamente os sensores da família SHT1x , por serem componentes SMD, tem dimensões reduzidas, de aproximadamente 8 milímetro por 5 milímetros (Figura 2.8). O tamanho dos SHT1x facilita a alocação do componente em pequenos espaços das placas de circuitos, contudo é necessário atenção quanto ao processo de soldagem do componente à placa. Conforme informação do fabricante esses componentes suportam solda de acordo com IPC/JEDEC J-STD-020d⁴ para um pico de temperatura de até 260° durante 40 segundos.

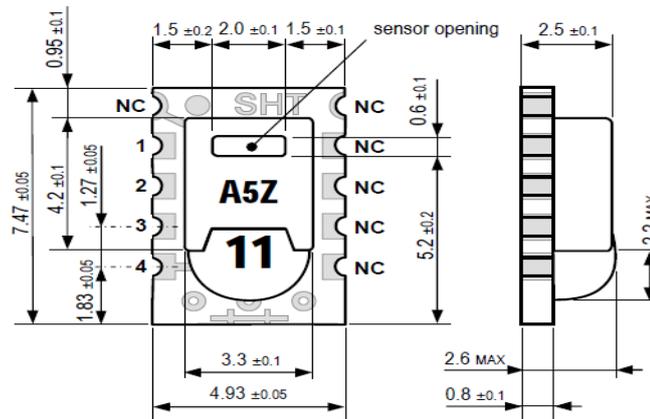


Figura 2.8 – Dimensões em milímetros do SHT11.

(Fonte: Folha de dados do fabricante.)

É importante ressaltar que durante processos de solda dos sensores às placas o polímero da célula capacitiva sofre desidratação e para que o componente funcione adequadamente deve ser armazenado durante 12 horas, após o processo de soldagem, em ambiente com umidade relativa superior à 75%.

Para o funcionamento do sensor são dispostos 4 pinos, sendo 2 de alimentação (GND e VCC), um de dados (DATA) e um de sincronização (SCK) as demais conexões do sensor não tem nenhuma funcionalidades (observe a Figura 2.9).

⁴ Norma sobre processos de solda por refluxo (fornos convencionais), elaborada para padronizar e garantir maior confiabilidade.

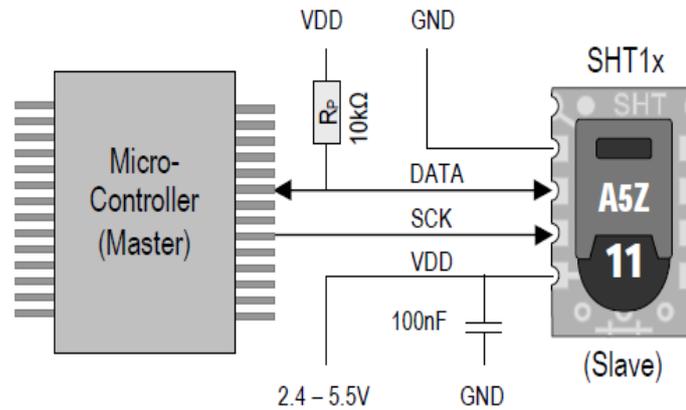


Figura 2.9 – Recomendação de ligação para sensores SHT1x

(Fonte: Folha de dados do fabricante.)

Na imagem anterior nota-se que entre os pinos de alimentação e entre o VDD e o DATA estão conectados um capacitor e um resistor, respectivamente de 10nF e 10KΩ, essa é uma recomendação do fabricante, logo que o capacitor, que deverá ser colocado o mais próximo possível do sensor, atenuará interferência oriundas da fonte de alimentação. O resistor de 10kΩ funcionará como *pull-up*⁵ no barramento de dados, também se pode utilizar um resistor no barramento de sincronização com a mesma finalidade.

2.3.1.1 Comunicação com o SHT15

Na transferência de dados de e para o sensor utiliza-se o pino DATA, trata-se de uma comunicação *half-duplex*, que se inicia após uma transmissão de sequência de início. Essa transmissão deverá aguardar 11 milésimos de segundos após o sensor ser ligado. A Sequência de início do sensor é constituída por uma diminuição da linha de dados enquanto SCK é alta durante dois ciclos de *clock* (observe a Figura 2.10).

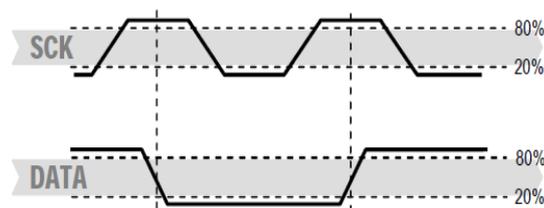


Figura 2.10 – Sequência de início do sensor.

(Fonte: Folha de dados do fabricante)

⁵ Pull-up: Estado de nível alto (1).

O comando posterior é constituído por três bits de endereço (Apenas '000' é suportado) e cinco bits de comando. O SHT15 indica, por meio de um bit —ACK— após a decida do clock, que recebeu adequadamente os dados transmitidos e a linha de dados é liberado (Pull-up). A tabela 2.2 mostra os comandos que o sensor reconhece:

Tabela 2.2 – Comandos do SHT15:

COMANDO	CÓDIGO
Reservado	0000x
Aferir Temperatura	00011
Aferir Umidade do Ar	00101
Ler o registrador de status	00111
Escrever no registrador de status	00110
Reinicialização ⁶	11110

(Fonte: Folha de dados do fabricante)

Após emitir um comando de medição ('00000101' para umidade relativa, '00000011' para a temperatura), é necessário um tempo de espera para que o senso possa efetuar o processo completo de aferição. Esse tempo é de 20, 80 ou 320 milissegundos para 8, 12 ou 14 bits de precisão, respectivamente.

Nenhuma leitura deve ser efetuada antes que haja o sinal de conclusão. O sensor indica que o sinal já pode ser lido quando após, o comando de medição, o barramento DATA o sensor é colocado em baixa então o sensor entra em espera. Depois do sinal de conclusão o microcontrolador pode reiniciar a SCK e prover a leitura dos dados, que são armazenados até leitura, observe a exemplificação do processo de comandos e leitura de aferição de umidade na figura 2.11.

⁶ Caso o senso seja reinicializado deve-se aguardar o intervalo de tempo de 11ms até o próximo comando.

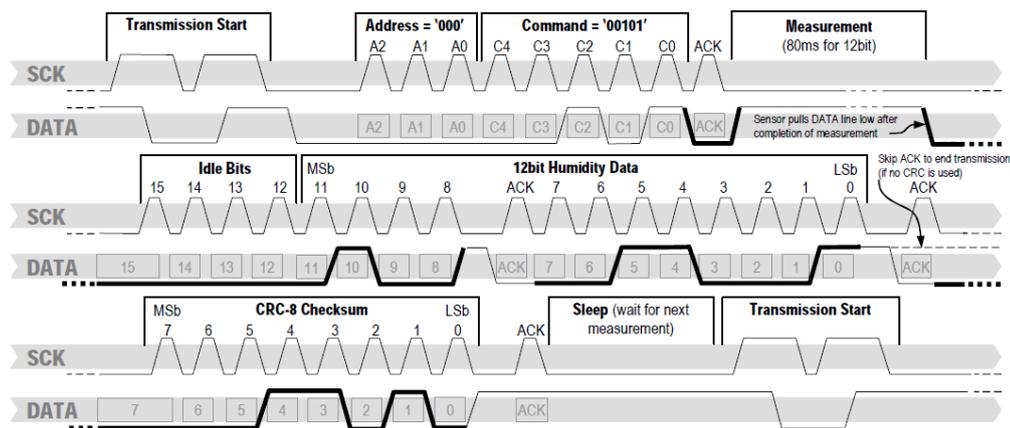


Figure 17: Example RH measurement sequence for value "0000'0100'0011'0001" = 1073 = 35.50%RH (without temperature compensation). DATA valid times are given and referenced in boxes on DATA line. Bold DATA lines are controlled by sensor while plain lines are controlled by the micro-controller.

Figura 2.11 – Exemplo de sequência de um comando.

(Fonte: Folha de dados do fabricante)

O microcontrolador deve reconhecer cada byte no barramento de dados sendo que dois bytes de dados de medição e um byte de CRC checksum (opcional) serão transmitidos. Todos os valores são com o bit mais significativo primeiro. O dispositivo retorna automaticamente ao modo de latência após a medição e comunicação estão concluídas.

2.3.1.2 Conversão de sinal de saída

Tanto para umidade relativa quanto para temperatura é necessário que se ajustem os valores de saída do sensor para que se obtenha a precisão máxima descrita pelo fabricante. No caso do sensor de umidade para compensar a não linearidade – figura 2.6 – é recomendado converter a leitura da umidade (SO_{RH}) com a seguinte equação:

$$HR_{LINEAR} = c_1 + c_2 * SO_{HR} + c_3 * (SO_{HR})^2 \quad (4)$$

Em que SO_{HR} é a umidade obtida diretamente da saída do sensor, HR_{LINEAR} é a umidade relativa com ajuste e c_1 , c_2 e c_3 são coeficientes, e estão expressos na tabela a seguir:

Tabela 2.3 – Coeficientes de conversão de umidade da 4ª versão do Sht1x.

SO_{HR}	c_1	c_2	c_3
12 bit	-2,0468	0,0367	$-1.5955 * 10^{-6}$
8 bit	-2,0468	0,5872	$-4.0845 * 10^{-4}$

(Fonte: Folha de dados do fabricante)

Os valores apresentados na Tabela 3.2 são coeficientes são otimizados para 4ª versão do Sht1x, contudo parâmetro estabelecido para 3ª versão, que foram propostos em fichas anteriores, ainda se é aplicável. Valores superiores a 99% de umidade relativa indicam que o ar esta em concentração de saturação e deve ser entendido e apresentado como 100%.

O gráfico a seguir (Figura 2.12) retrata a relação entre os valores de saída do sensor e o valor equivalente de umidade relativa:

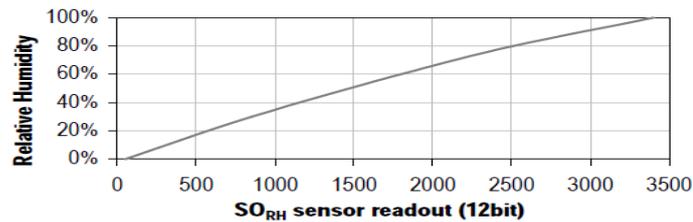


Figura 2.12: Gráfico de demonstração do ajuste para Umidade Relativa.

(Fonte: Folha de dados do fabricante)

Como relatado no tópico 2.2 deste capítulo a umidade relativa está vinculada à temperatura do ambiente. O sensor SHT 15 considera como padrão de temperatura 25° C. Uma compensação de umidade relativa, quando o sensor é utilizado em temperaturas bastante diferentes de 25 ° C, é imprescindível. Recomenda-se a correção da conforme a equação a seguir:

$$RH_{\text{true}} = (T_{\text{C}} - 25) \cdot (t1 + t2 \cdot SO_{\text{RH}}) + HR_{\text{LINEAR}} \quad (5)$$

Em que RH_{true} é o valor mais adequado para taxa de umidade relativa em temperaturas diferentes de 25° C e $t1$ e $t2$ são coeficientes conforme a tabela a seguir:

Tabela 2.4 – Coeficiente para correção de resposta de umidade relativa.

SO _{RH}	t1	t2
12 bit	0,01	0,00008
8 bit	0,01	0,00128

(Fonte: Folha de dados do fabricante)

Para a temperatura também é necessário um tratamento em relação ao dado oriundo do sensor, recomenda-se a seguinte equação de conversão:

$$T = d_1 + d_2 \cdot SO_T \quad (6)$$

Em que d_1 é um valor vinculado a tensão de alimentação do sensor e d_2 é um coeficiente relacionado à acurácia do sensor:

Tabela 2.5 – Relação de tensão de alimentação (VDD) e coeficiente de temperatura d_1 .

VDD	d_1 (°C)	d_1 (°F)
5	-40,1	-40,2
4	-39,8	-39,6
3,5	-39,7	-39,5
3	-39,6	-39,3
3,5	-39,4	-38,9

(Fonte: Folha de dados do fabricante)

Tabela 2.6 – Relação à acurácia do sensor e coeficiente de temperatura d_2 .

SO_{RH}	d_2	d_2
14 bit	0,01	0,018
12 bit	0,04	0,072

(Fonte: Folha de dados do fabricante)

Seguindo essas recomendações de conversão do sinal de saída do sensor e os padrões de comunicação é possível, segundo o fabricante do sensor, obter valores muito próximos aos reais.

CAPÍTULO 3 - SISTEMA DE MONITORAMENTO E CONTROLE DE ACIONAMENTO DE DISPOSITIVO UMIDIFICADOR

Neste capítulo será explicitado como foi desenvolvido o dispositivo bolado para monitorar a temperatura e a umidade do ar e controlar a ativação de umidificadores. Dividido em três subtítulos principais o capítulo delinea quais as ferramentas utilizadas, detalhes da construção física — componentes e técnicas escolhidos — e da lógica de funcionamento do aparelho.

3.1 Do Hardware

Para construir o projeto proposto, inicialmente, foi feito um levantamento dos principais microcontroladores, dos sensores de umidade e *display* de LCD existentes no mercado e que atendessem ao planejamento inicial do projeto.

Foi escolhido o microcontrolador Atmega48 devido sua capacidade de memória e quantidade de portas lógicas, que atendem com folga o que o projeto propõe. A opção do sensor de umidade foi o SHT15 produzido pela empresa Sensirios. Conforme descrito no capítulo anterior esse sensor apresenta uma saída digital em que a comunicação é implementada com base em um protocolo mestre-escravo. Dentre os sensores que apresentam a mesmas características o SHT15 foi o mais fácil de ser encontrado no mercado brasileiro. Como já dispunha de um *display* de LCD 20X4 e esse se adequava ao projeto e o mesmo foi utilizado. Os demais componentes do projeto foram escolhidos conforme a disponibilidade e compatibilidade com o projeto.

3.1.1 Sensoriamento

O Sensor SHT15 foi soldado a um cabo de quatro vias fora da placa de controle para que aquele pudesse ser posto em um ambiente de testes e efetuar o monitoramento sem a necessidade de se colocar todo o sistema de controle. Na outra ponta desse cabo foi colocado um conector macho RJ-11 que se acopla à placa de controle por meio de um conector RJ-11 fêmea (Figura 3.1).

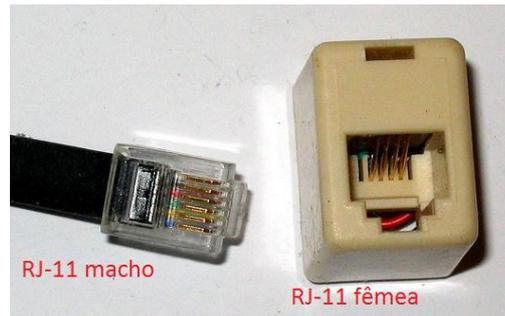


Figura 3.1 – Conectores RJ-11.

(Fonte: <http://pt.wikipedia.org/wiki/Ficheiro:Photo-RJ11-MF.jpg> (adaptado).)

O conector RJ-11 é comumente utilizado em terminações de linhas telefônicas. Foi empregado nesse projeto com o intuito de facilitar a troca do sensor em caso de dano. Os quatro pinos centrais –1, 2, 3 e 4 (Figura 3.2) – foram conectados ao VCC, DATA, SCK e GND do sensor, respectivamente.

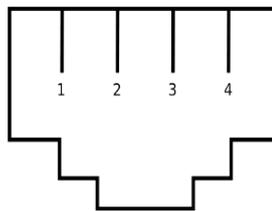


Figura 3.2 – Pinos centrais de um conector RJ-11

(Fonte: <http://pt.wikipedia.org/wiki/Ficheiro:RJ-11.svg>)

A alimentação do sensor é de aproximadamente 5V (Volts), pinos 1(positivo) e 4(terra) (Figura 3.2).A comunicação com o controlador é feita pelo pino 2 e 3, os quais a linha de dados e o *clock* de comunicação, respectivamente.

3.1.2 Sistema de Controle

O microcontrolador atmega48 (Figura 3.3) é o componente responsável pela comunicação com o sensor de umidade, gerenciamento do acionamento do relé para alimentação do umidificador e exibição dos dados no *display* alfanumérico. Todo o conjunto funciona centralizado em um único microcontrolador, barateando e facilitando o projeto.

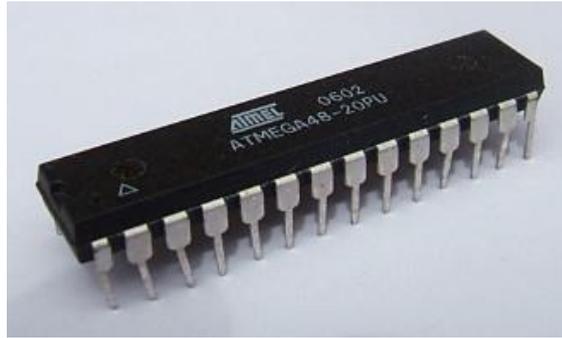


Figura 3.3 – Microcontrolador atmega48.

(Fonte: <http://hobby-electrons.sourceforge.net/components/ATmega48/index.html>)

Neste projeto utilizou-se cada porta do atmega48 para uma finalidade. A porta D foi utilizada para o controle do modulo de LCD por meio de uma interface de dados de 4 bits. Para escrever ou ler um dado no LCD utilizou-se o pino PD2 (0 para escrita e 1 para leitura). Os bits PD7, PD6, PD5 e PD4 foram empregados na transmissão dos bytes de comando do LCD, como são necessário 8 bits o controlador deve primeiro enviar os 4 bits mais significativos gerar um pulso no pino PD3 (*enable*) e depois enviar os bits menos significativos. A figura 3.4 ilustra a conexão entre o ATMEGA48 e o *display* LCD.

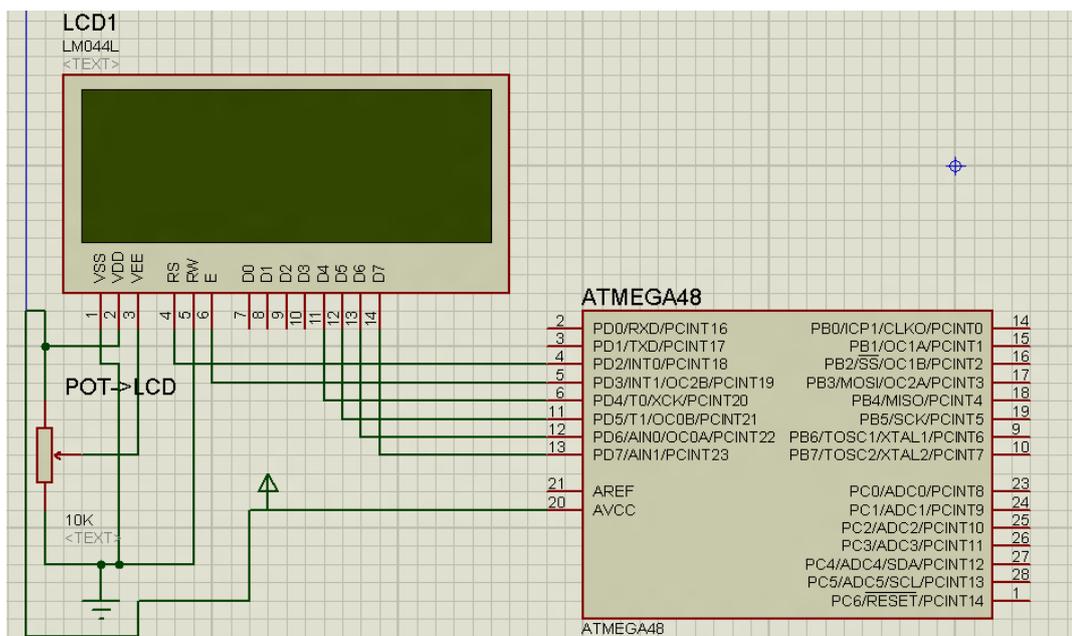


Figura 3.4 – Ligação entre Atmega48 e *display* LCD 20x4 (4 bits).

(Fonte: Autor)

Para a comunicação com o sensor SHT15 foi utilizada a porta C, sendo PC0 para dados e PC1 para o sinal de clock, conforme mostra a figura 3.5.

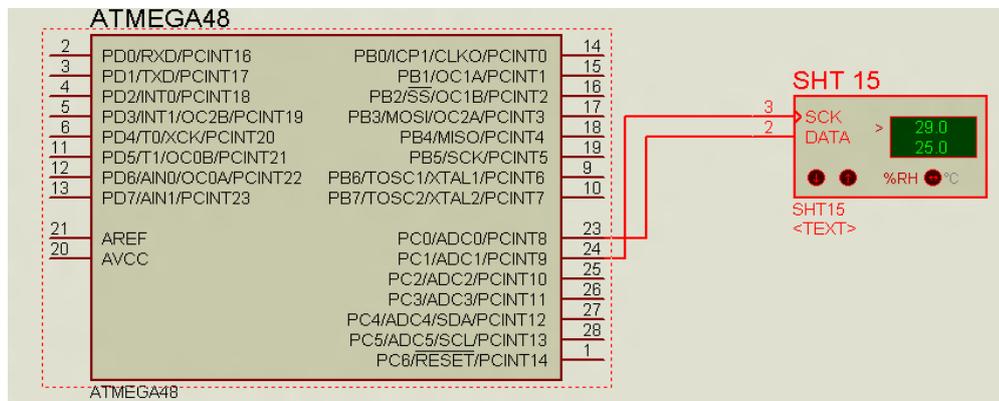


Figura 3.5 – Ligação entre Atmega48 e sensor SHT15

(Fonte: Autor)

A porta B foi designada para uma interface com três botões para que o usuário programe os parâmetros de ‘umidade mínima’, abaixo da qual o umidificador é ativado, e ‘umidade máxima’, acima da qual o umidificador, se ligado, é desativado (figura 3.6).

A saída para o controle do umidificador é feito pelo bit PB4. Um led conectado a PB3 sinaliza quando o umidificador está ligado(observe a figura 3.6).

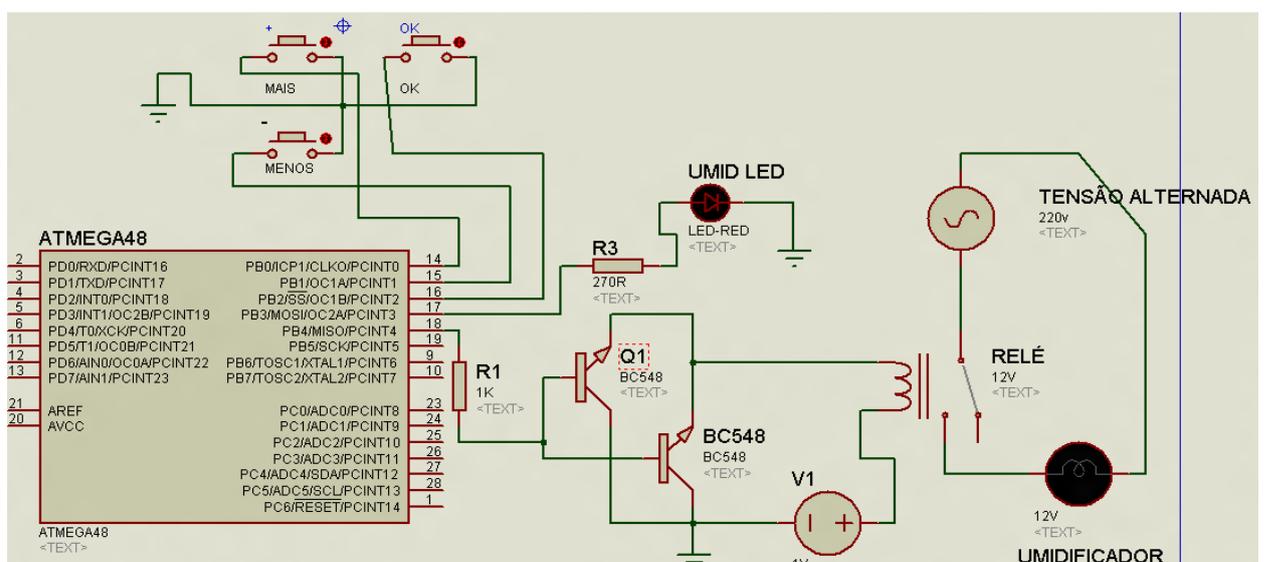


Figura 3.6 – Interface de botões e controle do umidificador

(Fonte: Autor)

A corrente fornecida por cada pino do atmega48 é de até 40 mA , insuficiente para ativar um relé. Para que o umidificador seja ativado através de um relé 12V, foi posto dois transistores, BC548, para funcionar como chave de ativação do relé (Figura 3.6).

Quando o microcontrolador identifica que o SHT15 acusa umidade relativa abaixo do valor parametrizado pelo usuário o bit PB4 passa a ser nível logico 1 (tensão positiva) e o transistor , NPN BC548, conectado a porta PB4 fecha o circuito e permite a passagem de corrente para que ative o relé. Como o relé ativado a tensão alternada (VCA) é liberada para alimentação do umidificador.

No dispositivo desenvolvido foi utilizado dois relés com capacidade de corrente de 10A (Ampères) a 220V, cada. A corrente máxima suportada pelo circuito de acionamento seria então de 20A, contudo para que o circuito trabalhe com uma boa margem de segurança considera-se a corrente máxima de 15A. O dispositivo, então, é capaz de acionar uma carga de 3300W (Watts) a 220V.

3.1.3 Alimentação do dispositivo

O dispositivo foi concebido com a ideia de ser flexível quanto á alimentação. Por isso o aparelho apresenta uma única entrada para alimentação do circuito. Essa entrada é bivolt, a tensão a qual o dispositivo se conecta deve ser definida previamente por meio de uma chave (figura 3.7).



Figura 3.7 – Chave seletora 220 ou 110.

(Fonte: http://www.proesi.com.br/site/components/com_virtuemart/shop_image/product/CHAVE_HH_SS22K15_4c05118e76860.gif)

Para alimentar o microcontrolador, o LCD e o sensor é necessário uma tensão constante de 5V. Para que se obtivesse essa tensão foi utilizado o componente regulador de tensão LM7805 (Figura 3.8).

Para os relés serem acionados uma tensão 12 V é disponibilizada após uma ponte de retificação na saída de um transformador (220/110 V para 12V). A figura 3.8 mostra a ideia básica do circuito de alimentação (Figura 3.8).

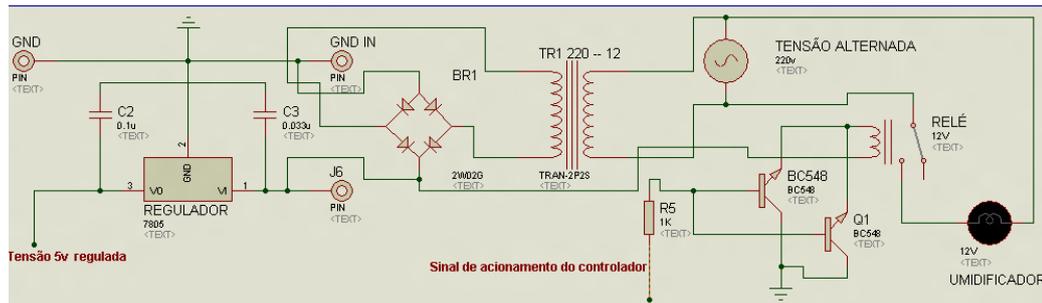


Figura 3.8 – Circuito de alimentação

(Fonte: Autor)

A figura 3.9 apresenta a simulação completa do circuito. O microcontrolador e suas conexões com o LCD, o Sht15 e a interface de botões são alimentados pela tensão de 5V oriunda do LM7805. O transformador utilizado na simulação apresenta apenas um entrada 220 V e uma saída 12V, contudo para que o dispositivo fosse bivolt, operando em 220V ou 110V, foi utilizado um transformador tal como o da figura 3.10.

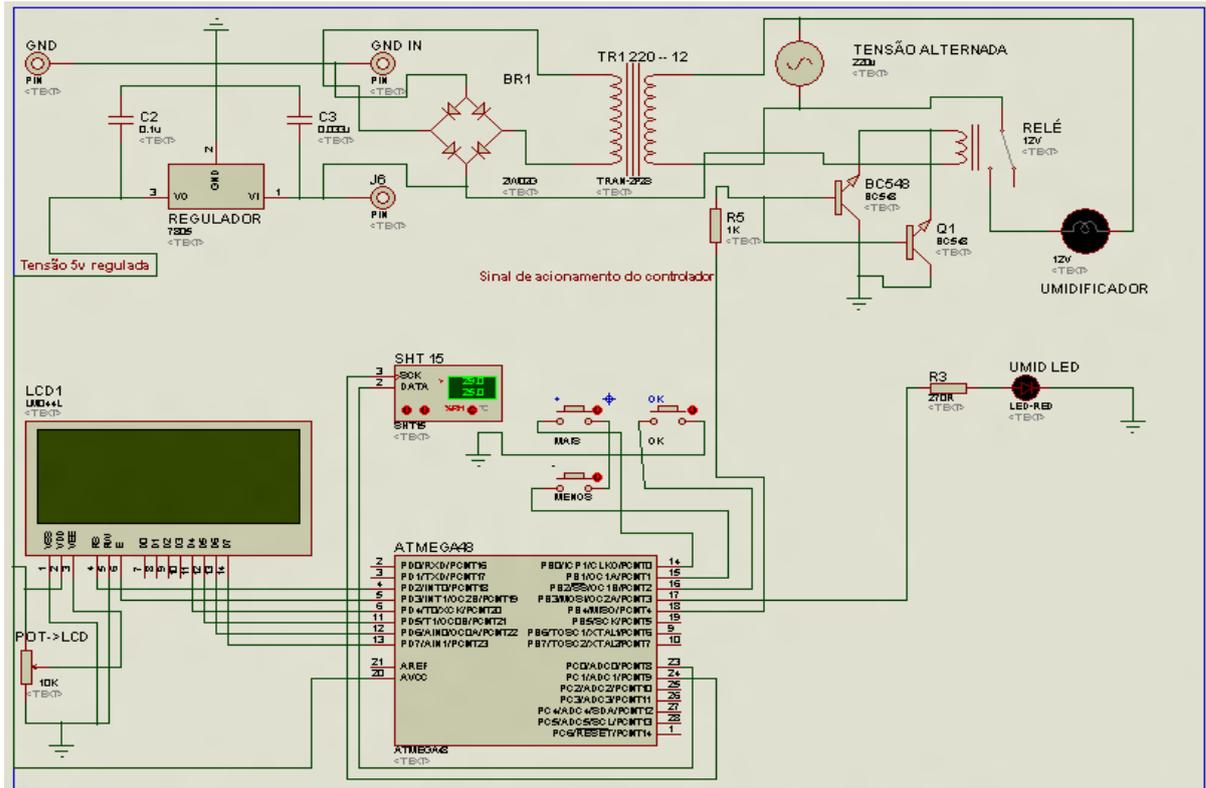


Figura 3.9 – Simulação completa do circuito.

(Fonte: Autor)

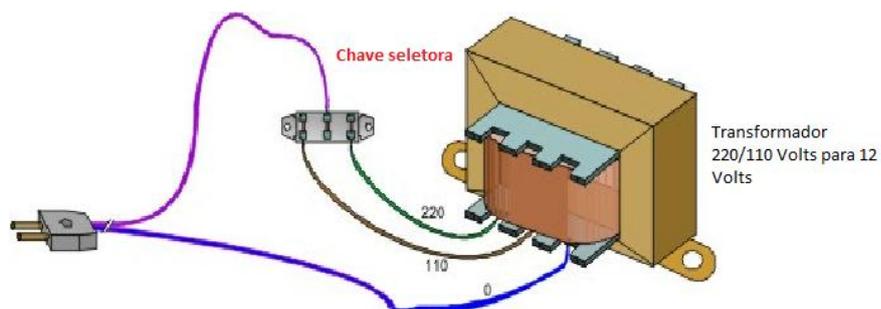


Figura 3.10 – Circuito de alimentação (Trafo).

(Fonte: <http://eletronicanoel.blogspot.com.br/2012/06/curso-de-eletronica-transformadores.html> (adaptado))

Em seguida à simulação foi produzida a placa de circuito e a ela soldado os componentes conforme simulação. Para facilitar a demonstração do circuito algumas ligações entre os componentes foram feitas por fios. A figura 3.11 apresenta o resultado final dos dispositivos conectados entre si:

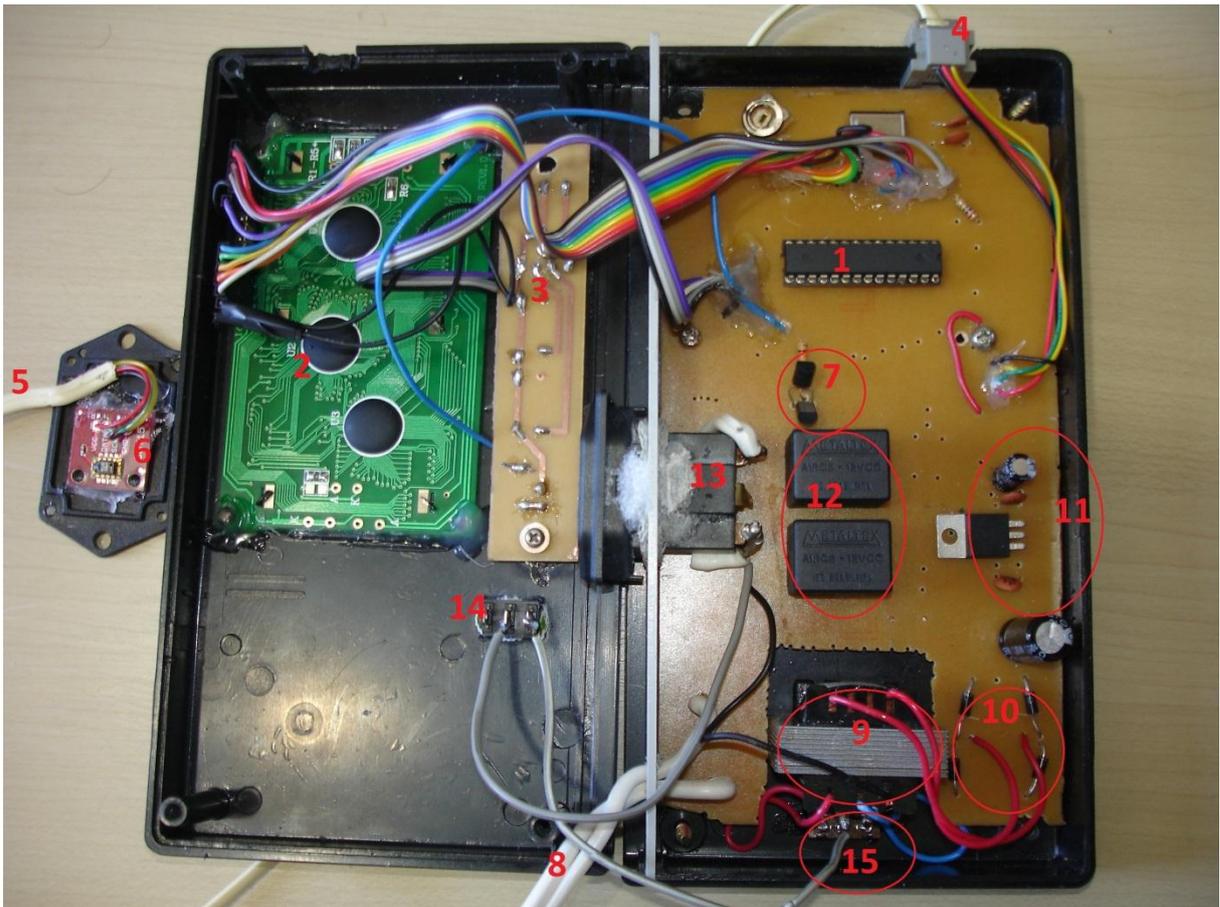


Figura 3.11 – Resultado final do dispositivo construído.

(Fonte: Autor)

Nota-se na figura 3.11 a disposição dos componentes, os números, em vermelho são:

- 1 – Microcontrolador ATMEGA48;
- 2 – *Display* LCD (4 linhas e 20 colunas);
- 3 – Placa com botões para interface de controle;
- 4 – Conector RJ-11(macho e fêmea conectados);
- 5 – Cabo de 4 vias (conexão entre sensor SHT15 e conector RJ-11);
- 6 – Placa com sensor SHT15 ;
- 7 – Transistores BC548, responsáveis pelo acionamento dos relés;
- 8 – Cabo para alimentação do dispositivo (220V ou 110 V);
- 9 – Transformador 220V/110V para 12 V;
- 10 – Ponte de diodos (retificação);

- **11** – Circuito regulador de tensão (entrada retificada 12 V e saída 5V);
- **12** – Relés para acionamento de carga 220V para umidificadores;
- **13** – Conector fêmea de tomadas, saída para umidificador (20A);
- **14** – Chave para ativar dispositivo;
- **15** – Chave seletora de tensão;

Todo o circuito foi alocado em uma caixa patola. Para tornar o uso do dispositivo mais intuitivo foi desenhado e colado uma etiqueta à caixa. O resultado é o dispositivo apresentado na figura 3.12.



Figura 3.12 – Resultado final do dispositivo construído 2.

(Fonte: Autor)

3.2 Do Firmware

O microcontrolador funciona após ter sido carregado um código hexadecimal. Esse código é resultado de uma compilação de uma linguagem de um nível mais elevado, seja em Assembler, Basic, C, etc. A programação do microcontrolador do sistema proposto neste trabalho foi desenvolvida em linguagem C. Esse código em C aplica-se a outros microcontroladores AVR, tal como o Atmega88 e o Atmega168.

3.2.1 Sensoriamento

A comunicação entre o microcontrolador e o sensor SHT15 é baseada em um protocolo mestre-escravo através de 2 barramentos, o de dados e o de clock. O microcontrolador e o sensor podem enviar dados através do respectivo barramento, contudo o clock da comunicação é ditado pelo microcontrolador, o mestre.

A rotina básica de comunicação com o sensor é composta por uma sequência de início (Figura 4.6), três bits de endereço (Apenas '000' é suportado) e cinco bits de comando (Tabela 4.1). O SHT15 indica, por meio de um bit —ACK— após a decida do clock, que recebeu adequadamente os dados transmitidos e a linha de dados é liberado (*Pull-up*).

Tabela 3.1: Comandos do SHT15:

COMANDO	CÓDIGO
Reservado	0000x
Aferir Temperatura	00011
Aferir Umidade do Ar	00101
Ler o registrador de status	00111
Escrever no registrador de status	00110
Reinicialização ⁷	11110

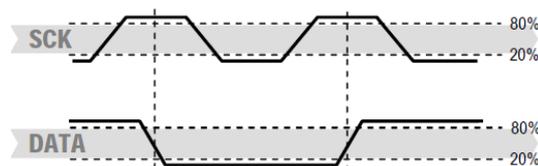


Figura 3.13 – Sequência de início do sensor.

(Fonte: Folha de dados do fabricante)

⁷ Caso o senso seja iniciado ou reinicializado deve-se aguardar o intervalo de tempo de 11ms até o próximo comando.

As rotinas responsáveis por esses comandos são “s_transstart” e “s_measure”, expostas a seguir:

```

//-----
// Gerando transmissão de início
void s_transstart(void)
{
    DDRC = 0b00000011; // SCK E DATA COMO SAÍDA
    DATA_1; SCK_0; //Initial state
    _delay_us(2);
    SCK_1;
    _delay_us(2);
    DATA_0;
    _delay_us(2);
    SCK_0;
    _delay_us(5);
    SCK_1;
    _delay_us(2);
    DATA_1;
    _delay_us(2);
    SCK_0;
    DDRC = 0b00000010;
}
//rotina de escrita no barramento do sensor
char s_escreve_byte(unsigned char value)
{
    unsigned char i,error=0;
    DDRC = 0b00000011; // PC0 e PC1 como saídas
    for (i=0x80;i>0;i/=2) //mudar bit para mascarar
    {
        if (i & value) DATA_1; //mascarando valor com i, escreva no
barramento do sensor
        else DATA_0;
        _delay_us(1);
        SCK_1; //clk para o sensor
        _delay_us(5); //tempo de 5 us
        SCK_0;
        _delay_us(1);
    }
    DATA_1; //libera barramento de dados
    DDRC = 0b00000010; // PINO de dados como entrada PC0
    _delay_us(1); //tempo de configuração
    SCK_1; //nono clock para o ack
    _delay_us(2);
    error=DATA_IN; //verificar ack (DATA será puxado para baixo por SHT11)
    SCK_0;
    return error; //erro = 1 em caso de não confirmado
}
//-----
// faz uma medição (umidade / temperatura) com checksum
char s_measure(unsigned char *p_value, unsigned char *p_checksum, unsigned char mode)
{
    unsigned char error=0;
    unsigned int temp=0;
    s_transstart(); //transmissão de início
    switch(mode)//envia comando ao sensor
    {
        case TEMP : error+=s_escreve_byte(MEDIR_TEMP); break;
        case HUMI : error+=s_escreve_byte(MEDIR_UMID); break;
        default : break;
    }
}

```

```

DDRC = 0b00000010; //pino de dados como entrada
while(1)
{
    if(!DATA_IN) break; //espera o sensor efetuar as medidas
}
if(DATA_IN) error+=1;

switch(mode) // envio de comando ao sensor
{
    case TEMP    :
        temp=0;
        temp=s_le_byte(ACK);
        temp<<=8;
        tempvalue[0]=temp;
        temp=0;
        temp=s_le_byte(ACK);
        tempvalue[0]|=temp;
        break;

    case HUMI    :
        temp=0;
        temp=s_le_byte(ACK);
        temp<<=8;
        tempvalue[1]=temp;
        temp=0;
        temp=s_le_byte(ACK);
        tempvalue[1]|=temp;
        break;

    default     : break;
}

```

A Tabela 4.1, com comandos do SHT15, é definida no início como:

```

#define STATUS_REG_W 0x06 //000 0011 0
#define STATUS_REG_R 0x07 //000 0011 1
#define MEDIR_TEMP 0x03 //000 0001 1
#define MEDIR_UMID 0x05 //000 0010 1
#define RESET 0x1e //000 1111 0

```

Após enviar o comando serial de medição ('00000101' para umidade relativa, '00000011' para a temperatura) aguarda se o intervalo de tempo de cerca de 320 milissegundos, esse é o tempo necessário para que o sensor efetue as medidas. O sensor também indica que os resultados já podem ser lidos quando após, o comando de medição, o barramento DATA do sensor é colocado em baixa (0). Nenhuma leitura deve ser efetuada antes que haja o sinal de conclusão.

O comando de leitura é ditado pelo microcontrolador e cada byte no barramento de dados deve ser reconhecido. São dois bytes de dados de medição e um byte de CRC, *checksum*. Todos os valores são com o bit mais significativo primeiro. O dispositivo retorna automaticamente ao modo de latência após a medição e comunicação está concluída.

```

//Lê dados do barramento do sensor e dá uma confirmação em caso de "ack = 1"
char s_le_byte(unsigned char ack)
{
    unsigned char i,val=0;
    DATA_0;
    DDRC = 0b00000010; // PINO de dados como entrada PC0
    for (i=0x80;i>0;i/=2)
    {
        SCK_1; //clk do barramento do sensor
        if (DATA_IN) val=(val | i); //ler bit
        _delay_us(2);
        SCK_0;
    }
    DDRC = 0b00000011;
    if (ack>=1)
    {
        DATA_0;
    }
    else
    {
        DATA_1;
    }
    SCK_1; //clk #9 para ack
    _delay_us(5); //
    SCK_0;
    _delay_us(1); //observe hold time***
    DATA_1; //libera barramento de dados do sht15
    return val;
}

```

Conforme já descrito no referencial teórico deste trabalho as medidas efetuadas pelo sensor devem ser ajustadas segundo coeficientes fornecidos pelo fabricante do sensor, essa correção é efetuada pela rotina “calc_sht15”:

```

void calc_sht15(float *p_humidity ,float *p_temperature)
{
    const float C1=-2.0468; // 12 Bit UH
    const float C2=+0.0367; // 12 Bit UH
    const float C3=-0.0000015955; // 12 Bit UH
    const float T1=+0.01; // 12 Bit UH
    const float T2=+0.00008; // 12 Bit UH
    float ur=*p_humidity; // Uh: umidade 12 Bit
    float t=*p_temperature; // t: Temperature [Ticks] 14 Bit
    float ur_lin; // ur_lin: umidade linear
    float ur_true; // ur_true: Temperatura, umidade compensado
    float t_C; // t_C : Temperature [°C]
    t_C=t*0.01 - 40.1; //calc. temperatura [° C] a partir de 14 bit temp.ticks @ 5V
    ur_lin=C3*ur*ur + C2*ur + C1; //calc. umidade do ticks para [% UR]
    ur_true=(t_C-25)*(T1+T2*ur)+ur_lin; //calc. temperatura, umidade
    compensado[%UH]
    if(ur_true>100)ur_true=100; //cortado se o valor está fora do
    if(ur_true<0.1)ur_true=0.1; //o alcance físico possível
    *p_temperature=t_C; //retorna temperatura [° C]
    *p_humidity=ur_true; //retorna umidade [% UH]
}

```

Com os dados corretos das medidas o controlador pode então exibir os resultados no LCD e compara-los aos parâmetros predeterminados pelo usuário.

3.2.2 Controle e respostas

Para poder efetuar o controle uma interface é exibida logo após o sistema ser ligado. O usuário deve definir dois valores: “a umidade mínima” e a “umidade máxima”. O primeiro valor representa qual a menor umidade relativa para qual o umidificador pode ficar desligado e o segundo valor é a maior umidade relativa aceitável pelo sistema, quando um o sensor acusar umidade acima de “umidade máxima” o umidificador deve ser desligado. A rotina de controle é ‘config_control_umidade’:

```
//definições iniciais:
#define b_mais PB0 //B_mais (Botão de incremento) é o substituto de PB2
#define b_menos PB1 //B_menos (Botão de decremento) é o substituto de PB3
#define b_ok PB2 //B_ok (Botão de decremento) é o substituto de PB3
char umidade_minima=50;
char umidade_maxima=90;
//rotina de configuração:
void config_control_umidade(void)
{
    PORTB=0xFF;
    int cc1=0;
    int cc2=0;
    cmd_LCD (0x01,0); //limpa o lcd
    escreve_LCD(" CONFIGURAR ");
    cmd_LCD(0xC0,0);
    escreve_LCD("UMIDADE RELATIVA");
    _delay_ms(150);
    cmd_LCD (0x01,0); //limpa o lcd
    while (cc1==0)
    {
        cmd_LCD(0xC0,0); // Cursor na 2ª linha e 1ª coluna
        escreve_LCD(" UMIDADE MINIMA");
        cmd_LCD(0x9C,0); // Cursor na 3ª linha 8ª coluna
        escreve_variavel_LCD(umidade_minima);
        escreve_LCD("%");
        if(!tst_bit(PINB,b_mais))//se o botão for pressionado incrementa valor
da umidade mínima
        {
            umidade_minima++;
            _delay_ms(20);
        }
        if (umidade_minima>=91)
        {
            cmd_LCD (0x01,0); //limpa o lcd
            escreve_LCD("Valores entre 0%-90%");
            _delay_ms(200);
            cmd_LCD (0x01,0); //limpa o lcd
            umidade_minima=90;
        }
        if(!tst_bit(PINB,b_menos))//se o botão for pressionado decrementa valor
da umidade mínima
        {
            umidade_minima--;
            _delay_ms(20);
            if (umidade_minima<0)umidade_minima++;
        }
    }
    if(!tst_bit(PINB,b_ok))//se o botão for pressionado finda o loop while
```

```

        {
            cc1=1;
            _delay_ms(20);
        }
    }
    cmd_LCD (0x01,0); //limpa o lcd
    while (cc2==0)
    {
        cmd_LCD(0xC0,0); // Cursor na 2ª linha e 1ª coluna
        escreve_LCD("  UMIDADE MAXIMA");
        cmd_LCD(0x9C,0); // Cursor na 3ª linha
        escreve_variavel_LCD(umidade_maxima);
        escreve_LCD("%");
        if (umidade_maxima-umidade_minima<10)
        {
            cmd_LCD (0x01,0); //limpa o lcd
            escreve_LCD("Valor deve ser Maior");
            cmd_LCD(0xC0,0); // Cursor na 3ª linha
            escreve_LCD("em 10% da Minima");
            _delay_ms(150);
            cmd_LCD (0x01,0); //limpa o lcd
            umidade_maxima=umidade_minima+10;
        }
        if(!tst_bit(PINB,b_mais))//se o botão for pressionado incrementa valor
da umidade máxima
        {
            umidade_maxima++;
            _delay_ms(20);
        }
        if(!tst_bit(PINB,b_menos))//se o botão for pressionado decrementa valor
da umidade máxima
        {
            umidade_maxima--;
            _delay_ms(20);
        }
        if(!tst_bit(PINB,b_ok))//se o botão for pressionado quebra o loop while
        {
            cc2=1;
            _delay_ms(20);
        }
    }
    cmd_LCD (0x01,0);// limpa lcd
    cmd_LCD(0x94,0); // Cursor na 3ª linha e 1ª coluna
    escreve_LCD("### CONFIGURADO ###");
    _delay_ms(100);
    cmd_LCD (0x01,0); //limpa lcd
}

```

Para alterar os valores de “umidade máxima” e “umidade mínima” o usuário utiliza os botões **^**, **v** e **ok**, de incremento, de decremento e de confirmação, respectivamente (figura 3.14).



Figura 3.14 – Interface de botões para configuração do umidificador

(Fonte: Autor)

O microcontrolador efetua duas simples comparações para determinar se liga ou deliga o umidificador conforme a “umidade máxima” e “umidade mínima” e, essa comparação é efetuada após cada medida do sensor e é executada dentro da instrução principal do *firmware*.

```

if (umidade_minima<=umidade)
    {
        if(umidade_maxima<umidade)
            {
                cmd_LCD(0xD4,0); // Cursor na 4ª linha e 1ª coluna
                A=0;
                escreve_LCD("UMIFICADOR DESLIGADO");

            }
            if (A==0)
            {
                clr_bit(PORTB,PB4);
                clr_bit(PORTB,LED_UMID);
                cmd_LCD(0xD4,0); // Cursor na 4ª linha e 1ª coluna
                escreve_LCD("UMIFICADOR DESLIGADO");
            }
            _delay_ms(100);
    }
if ((umidade_minima>umidade)|| (A==1))
{
    set_bit(PORTB,PB4);
    set_bit(PORTB,LED_UMID);
    A=1;
    cmd_LCD(0xD4,0); // Cursor na 4ª linha e 1ª coluna
    escreve_LCD(" UMIFICADOR LIGADO");
    cmd_LCD(0x98,0); // Cursor na 3ª linha e 3ª coluna
    escreve_LCD("~~~");
    _delay_ms(50);
    escreve_LCD("~~~");
    _delay_ms(50);
    escreve_LCD("~~~");
    _delay_ms(50);
    escreve_LCD("~~~");
    _delay_ms(50);
}

```

O LCD exibe as informações de umidade aferida, temperatura e status do umidificador. Além do aviso de status do umidificador no LCD um led ao lado dos botões acende quando há tensão na saída (220V) para o umidificador (figura 3.15).



Figura 3.15 – LCD com informações do controle.

(Fonte: Autor)

3.3 Da Implementação

Um conjunto de fermentas foi utilizado para o desenvolvimento deste trabalho. As primeiras ferramentas a serem utilizadas foram o ambiente de simulação Isis do Proteus 7.7 e o ambiente de desenvolvimento Atmel Studio 6.0, empregadas de maneira integrada.

Na simulação dos possíveis circuitos do projeto realizaram-se diversas configurações e teste com componentes, com o intuito de dispor um circuito que mais se adequasse às pretensões do sistema a ser desenvolvido (Figura 3.16).

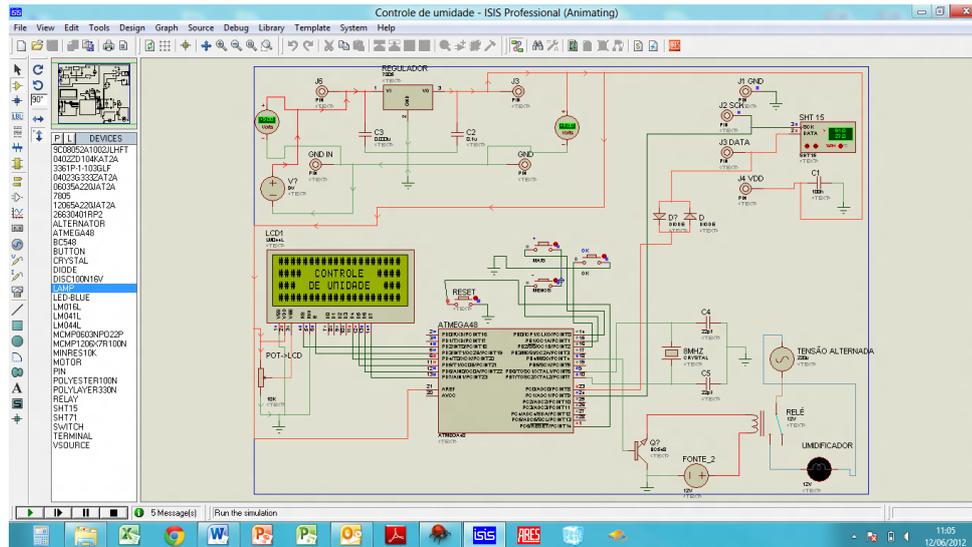


Figura 3.16 – ISIS Professional com simulação do sistema de controle

(Fonte: Autor)

O *firmware* do Atemega48 foi elaborado na plataforma de desenvolvimento Atmel Studio 6.0, uma ferramenta gratuita disponibilizada pelo fabricante do microcontrolador. O código foi desenvolvido em linguagem C e a cada rotina formada o código era compilado e os resultados dessas compilações, em instruções hexadecimais, foram testados na simulação construída no Isis do Proteus.

Essas primeiras ferramentas foram imprescindíveis na construção do projeto, dado que foi permitido corrigir erros de planejamento do circuito e alterar o código para que o sistema funcionasse adequadamente sem ter que construir e modificar fisicamente todo o circuito. Trata-se de ferramentas de grande valia em diversos projetos de eletrônicos, servindo de base de planejamento para vários projetos de engenharia.

Compõe o Proteus 7.7 a ferramenta Ares, que auxilia o desenho das placas de circuito. Graças a integração do Ares com o Isis os componentes simulados no segundo em suas dimensões reais, bem como as ligações entre eles, são automaticamente exportados para o Ares e o processo de desenho da placa de circuito impresso é facilitada (Figura 3.17).

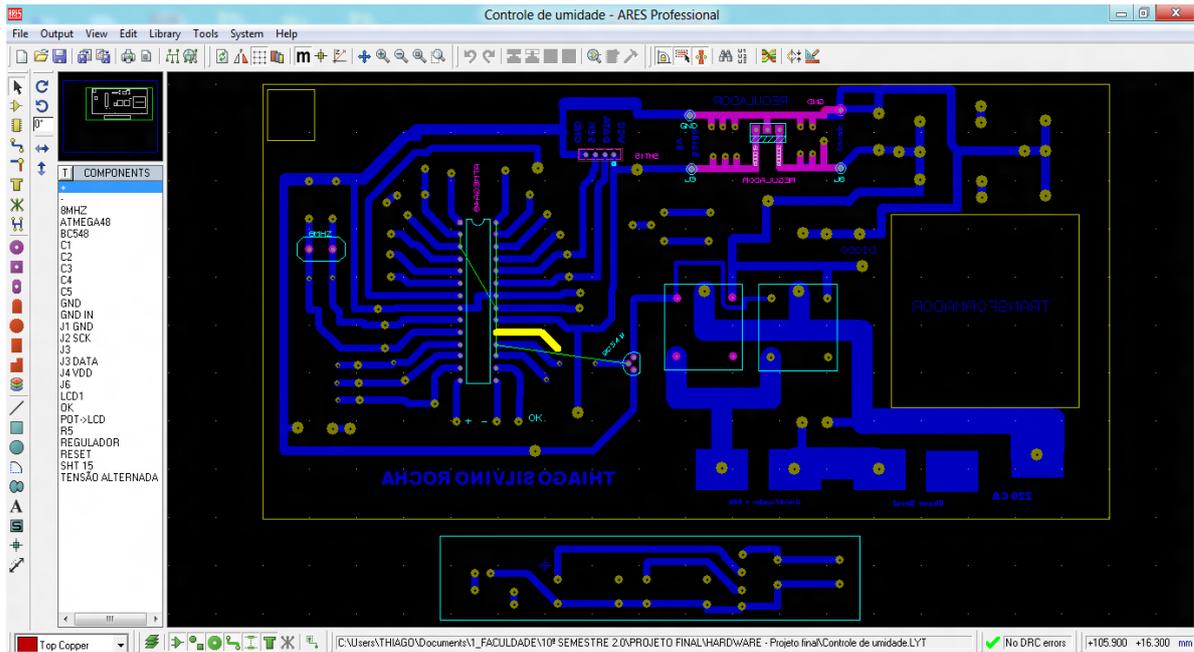


Figura 3.17 – Ares Professional com designer do layout do circuito

(Fonte: Autor)

Todo o circuito resultante do *layout* elaborado no Ares foi transferido para uma placa de fenolite e essa perfurada para receber os componentes e ligações, conforme simulado no Isis.

Com circuito de controle pronto, o código compilado no Atmel Studio 6.0 precisava ser carregado no microcontrolador. Para isso foi empregado um programador USB de microcontrolador AVR (figura 3.18) e o Extreme Burn-AVR (figura 3.19), ferramenta responsável por transmitir para o microcontrolador Atmega48 ligado ao programador, as instruções compiladas.

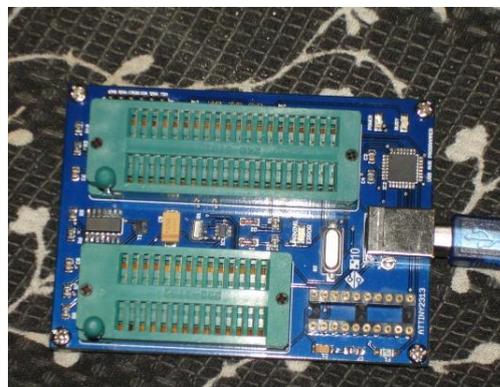


Figura 3.18 – Programador USB de microcontrolador AVR

(Fonte: Autor)

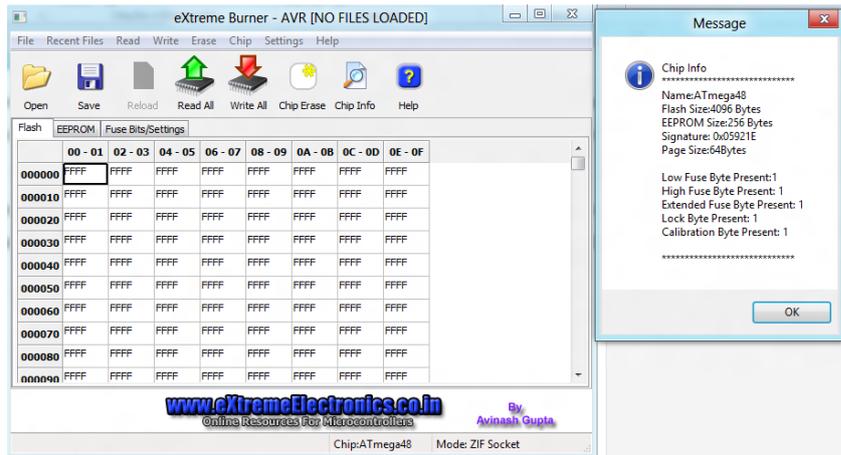


Figura 3.19 – Extreme Burn-AVR

(Fonte: Autor)

CAPITULO 4 - TESTES E APLICAÇÃO PRÁTICA DO SISTEMA

Visando efetuar o monitoramento assim como o controle da umidade de uma atmosfera, foi utilizado como ambiente de testes um vasilhame de cerca de 20 litros. Dentro desse ambiente foi colocado um sistema para prover o ressecamento do ar.

O dispositivo, fruto deste trabalho, foi idealizado para suportar uma carga elevada, 3300 w na saída para umidificadores. Entretanto, não é comum no mercado umidificador dessa potência. Com o intuito de avaliar o dispositivo trabalhando próximo ao regime máximo de potência foi desenvolvido um umidificador, baseado em um chuveiro elétrico, que utiliza uma grande corrente elétrica, condizente com a carga suporta pelo equipamento construído.

4.1 Ambiente de testes

A umidade relativa é muito variável em relação ao ambiente em que é aferida. Para que um ambiente fosse passível de testes foi necessário algo que causasse o ressecamento do ar. Uma substância, soda cáustica —NaOH— (soda caustica), tem grande capacidade de absorver água, e foi utilizada como desumidificador do ambiente de testes.

Para agilizar o processo de deumidificação do ambiente porções de NaOH foi colocada dentro de um tecido poroso formando pequenass bolsas de soda cáustica. Um cooler foi utilizado aumentar o fluxo de ar nesse tecido acelerando o processo de desumidificação (Figura 4.1).

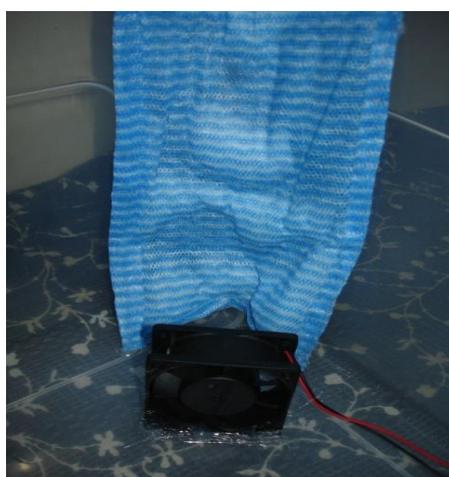


Figura 4.1 – Desumidificação utilizando NaOH

(Fonte: Autor)

Com o ambiente pronto foi construído o umidificador a partir de um chuveiro elétrico para funcionar como uma caldeira que aquece a água até evaporar. Foram necessárias alterações no chuveiro para que esse pudesse trabalhar da forma pretendida. Primeiramente o circuito de acionamento do chuveiro foi fechado a fim de que esse funcionasse independente de pressão da água (Figura 4.2).



Figura 4.2 – Alteração do circuito de acionamento do chuveiro

(Fonte: Autor)

Foi escolhido o modo com maior resistência elétrica, o modo verão. Dessa maneira a caldeira tem a potência de cerca de 3200 W.

Todas as entradas e saídas de água do chuveiro foram seladas, de tal forma que somente por um cano acoplado ao chuveiro (conforme figura 4.3) possa sair o vapor de água. Esse cano é conectado, por uma mangueira, ao ambiente de teste a fim de umidificar o ar dentro do ambiente, quando a caldeira estiver ligada (figura 4.4).



Figura 4.3 – Entupimento das saídas de água e acoplamento de cano para saída de vapor.

(Fonte: Autor)



Figura 4.4 – Ligação entre o umidificador e ambiente de testes.

(Fonte: Autor)

O resultado é uma atmosfera que pode ser ressecada para que o dispositivo de controle possa detectar um ambiente de baixa umidade e acionar o umidificador (caldeira) efetuando o controle de umidade, conforme predeterminado pelo usuário.

4.2 Descrição do sistema em ambiente de testes

O teste do funcionamento do dispositivo construído neste trabalho foi feito utilizando o ambiente descrito no item 4.1. Ao ligar o aparelho é necessário determinar a umidade mínima e máxima para que seja efetuado o controle do acionamento do umidificador. Foram definidos as taxas de 40% e 75% de umidade mínima e umidade máxima, respectivamente (Figura 4.5).



Figura 4.5 – Configuração inicial do dispositivo.

(Fonte: Autor)

O sensor de umidade foi inserido no ambiente de testes. Inicialmente o ambiente estava aberto e foi detectado a temperatura de 21° Celsius e umidade relativa de 61% (Figura 4.6).



Figura 4.6 – Monitorando umidade e temperatura em ambiente aberto.

(Fonte: Autor)

Foi ativado o cooler responsável por acelerar o processo de desumidificação do ambiente. A umidade relativa começou a cair a uma taxa de aproximadamente 2% por segundo (Figura 4.7).

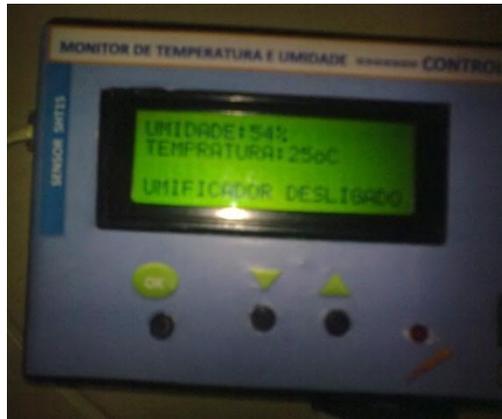


Figura 4.7 – Monitorando umidade e temperatura no ambiente durante processo de desumidificação.

(Fonte: Autor)

O umidificador é acionado quando o sensor acusa a umidade de 39%. Entretanto o ambiente não é umidificado logo de início, isso porque a caldeira usada como umidificador leva, aproximadamente, 30 segundos para aquecer a água e iniciar o processo de umidificação do ambiente (Figura 4.8).



Figura 4.8 – Acionando umidificador quando umidade é menor que 40%.

(Fonte: Autor)

Devido à potência do umidificador, em menos de 8 segundos, após o início do processo de umidificação, o ambiente de teste ultrapassou a umidade máxima (75%). O umidificador é desativado e o sensor detecta a umidade relativa de 84% e uma temperatura de 37° Celsius (Figura 4.9).



Figura 4.9 – Umidificador desativado, umidade de 84% e temperatura de 37° Celsius.

(Fonte: Autor)

Mas por que o umidificador foi desativado somente a 84% de umidade relativa e não logo depois de 75% como configurado? Isso se deve ao fato de o dispositivo efetuar medidas somente a cada 1,2 segundos. Após o ambiente ter realmente atingido mais de 75% de umidade, o sensor ainda não aferiu a umidade e o ambiente continuou sendo umidificado. Quando o sensor detecta que já ultrapassou a umidade máxima o umidificador é desativado, mas o ambiente já está mais úmido do que se esperava.

4.3 Resultados dos testes

Dado o exposto no item 4.2 conclui-se que o sistema desenvolvido atende ao que foi proposto, executando o monitoramento da umidade relativa e também da temperatura e controlando o acionamento de um dispositivo de umidificação.

A temperatura detectada pelo sensor ao final do teste (Figura 4.9) deve-se ao vapor sob alta temperatura exaurido do umidificador.

A diferença entre a umidade em que o umidificador foi desligado e a que ele foi configurado para desligar já era prevista. Primeiramente essa diferença é atribuída ao ambiente de volume pequeno e a grande potência do umidificador. O fator secundário dessa diferença é o intervalo de tempo entre as medidas.

O teste do dispositivo foi considerado concluído e satisfatório, conforme os resultados esperados foram alcançados. Para um ambiente maior, como uma sala, a diferença descrita no último parágrafo do item 4.2 seria bem menor.

4.4 Custos

O custo total do projeto foi de cerca de 700,00 reais. Entretanto o custo efetivo do dispositivo construído e do ambiente de simulação foi de, aproximadamente, R\$ **498,00** (quatrocentos e cinquenta e oito reais) distribuídos conforme tabela a seguir.

Tabela 4.1 – Custos do Projeto

Item	Preço (R\$)
Programador AVR	98,00
Microcontrolador ATMEGA 48	32,00
Sensor SHT15	180,00
Chuveiro	33,00
Vasilhame (Ambiente)	29,00
<i>Display</i> LSD	56,00
Outros Componentes (Resistores, capacitores, fios, placas, relés)	70,00

(Fonte: Autor)

A diferença entre o valor total do projeto e o valor efetivo é devido a um sensor SHT15 queimado (componente mais caro do projeto) e a alguns ajustes realizados ao longo do projeto.

4.5 Análise do sistema

De maneira geral o Sistema de Monitoramento e Controle de Acionamento de Dispositivo Umidificador pode ser aplicado em qualquer tipo de ambiente, apresentando vantagem de ser bem flexível no que tange as configurações de controle do umidificador.

Os testes realizados no item 4.2 indicam que aparelho construído é capaz de ativar uma carga elevada de potência, 3200 W. Considerando um umidificador de ultrassom,

comumente encontrado no mercado, que consome apenas 55 W, seria possível conectar paralelamente 58 umidificadores. O dispositivo, então, poderia controlar a umidade de ambientes razoavelmente grandes.

CAPÍTULO 5 - CONCLUSÃO

5.1 Conclusões

A solução desenvolvida atende as perspectivas quanto à automação do processo de umidificação do ar. O uso de uma solução de controle de umidade, conforme proposto neste trabalho, poderia evitar um gasto a mais com um ou mais umidificadores ligados todo o tempo.

A possibilidade de se configurar em qual taxa de umidade o dispositivo aciona umidificadores a ele conectados e a capacidade de potência do dispositivo viabiliza a utilização do aparelho em diversos ambientes. Outro fato que o torna maleável, quanto ao uso, é o fato de ser bivolt.

O resultado do projeto foi satisfatório, atendeu-se a todos os objetivos inicialmente enumerados, foi produzido um aparelho que automatiza o controle de qualidade do ar no que tange a umidade relativa.

5.2 Sugestões para Trabalhos Futuros

Para projetos futuros baseados neste trabalho sugiro soluções que acresçam aos processos de automação dos dispositivos controladores de qualidade do ar:

- Integrando o sistema de controle desse projeto ao sistema de sensoriamento por via sem fio;
- Realizando o controle, não somente da umidade do ar, mas também da temperatura e das substâncias presentes no ar, como CO₂ e O₂.

Como soluções de melhorias a esse projeto, indico trabalhos futuros:

- Com componentes de menor custo, mas que atendam aos objetivos de automação do controle de umidade. Existem diversos componentes análogos aos utilizados neste projeto, contudo a falta de bibliografia relacionada a esses componentes dificulta o desenvolvimento de aparelhos como o deste trabalho, mas com um custo reduzido;
- Com uma interface de comunicação com computadores para que se possa guardar e analisar os dados oriundos dos sensores.

REFERÊNCIAS

[AYOADE, 1998] J.o. Ayoade INTRODUÇÃO À CLIMATOLOGIA DOS TRÓPICOS Editora: BERTRAND Páginas:332

[Cocota, 2005] **Sistema de Monitoramento De Temperatura e Umidade**, José Alberto Naves Cocota Júnior, Disponível em: http://users.cjb.net/cocotajr/eneca_2005_smtu.pdf (Acesso em 19 de maio de 2012).

[COLLISCHONN, 2011] Walter Collischonn e Rutinéia Tassi. **Introduzindo hidrologia**. Apostila disponível em: http://galileu.iph.ufrgs.br/collischonn/apostila_hidrologia/apostila.html (Acesso em 03 de abril de 2011)

[DERP - PA, 2007] **Manual de gestão de documentos do estado do Paraná**, 3ª edição, 2007; Departamento Estadual de Arquivo Público. Disponível em: <http://www.arquivopublico.pr.gov.br/arquivos/File/pdf/gestao.pdf> . (Acesso em 05 de março de 2011)

[GALVÃO, 2009] Stella Galvão. **Respire melhor**: Entenda como se manifestam e como são tratadas três doenças respiratórias cujas crises podem aumentar durante os meses mais frios: asma, bronquite e rinite alérgica. Reportagem tipo artigo publicada em 27 de abril de 2009. Disponível em: <http://revistavivasaude.uol.com.br/saude-nutricao/73/artigo132440-1.asp> (Acesso em 03 de abril de 2011)

[Gripp, 2010] **Desenvolvimento de Instrumentação para Aquisição de Dados Com sensores Capacitivos Interdigitados Nanométricos**, Alexandre Bittencourt Gripp e Michael Willer Souza Rodrigues. Disponível em: http://www.uems.br/portal/biblioteca/repositorio/2011-08-31_07-31-00.pdf (Acesso em 19 de maio de 2012).

[INDRIUNAS, 2008] "**HowStuffWorks - Como funciona a umidade do ar**", Alexandre Indriunas, Publicado em 31 de julho de 2008. <http://ciencia.hsw.uol.com.br/umidade-do-ar.htm> (Acesso em 26 de março de 2012).

[JAAKKOLA et al., 2002] Maritta S. Jaakkola, Henrik Nordman, Ritva Piipari, Jukka Uitti, Jukka Laitinen, Antti Karjalainen, Paula Hahtola and Jouni J.K. Jaakkola. **Indoor Dampness and Molds and Development of Adult-Onset Asthma: A Population-Based**

Incident Case–Control Study. Disponível em:
<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1240846/> (Acesso em 12 de abril de 2011)

[LATORRE et al., 2002] Latorre, Marcelo; Carvalho Júnior, Osmar Abílio de; Carvalho, Ana Paula Ferreira de; Shimabukuro, Yosio Edemir. **Correção atmosférica: conceitos e fundamentos. Espaço & Geografia, Vol.5, No 1 (2002), 153:178 ISSN: 1516-9375**

[Re. N°9 ANVISA, 2003] Brasil. Agência Nacional de Vigilância Sanitária **Resolução nº 9, de 16 de janeiro de 2003.** Determina a publicação de Orientação Técnica elaborada por Grupo Técnico Assessor, sobre Padrões Referenciais de Qualidade do Ar Interior, em ambientes climatizados artificialmente de uso público e coletivo. Publicação: D.O.U. – Diário Oficial da União –; Poder Executivo, de 20 de janeiro de 2003.

[ROMAN, 2010] Marcos Roman. **Risco de infarto em idosos é 30% maior em dias secos.** 20 de Setembro de 2010. Disponível em:
<http://www.jornalfolhanorte.com.br/component/content/article/55-saude/328-risco-de-infarto-em-idosos-e-30-maior-em-dias-secos.html>. (Acesso em 05 de março de 2011)

[SANTOS, 1998] Santos, Marília de Oliveira. **Conservação dos Suportes Informacionais:** do papel ao meio magnético. Porto Alegre, ABEED, 1998.

[SILVA, 2006] **Meteorologia e Climatologia**, versão digital 2, 2006; Mário Adelmo Varejão Silva. Disponível em:
http://www.agritempo.gov.br/publish/publicacoes/livros/METEOROLOGIA_E_CLIMATOLOGIA_VD2_Mar_2006.pdf (Acesso em 25 de março de 2012).

APÊNDICE

Firmware do microcontrolador

```

/*
***CONTROLADOR_DE_UMIDADE.cpp
*Chip          : ATMEGA 48
*Frequência   : 8.000.000 Hz
*
*SHT15 Humidity and Temperature Sensor
* Created: 22/05/2011 03:45:15
* Author: THIAGO SILVINO ROCHA OLIVEIRA
*/
#define F_CPU 8000000UL //define a frequência do microcontrolador (base de cálculo das
funções Delay)
#include <avr/io.h>
#include <stdio.h>
#include <stdlib.h>
#include <util/delay.h>
#include <compat/deprecated.h>
#include <string.h>

//Definições para o LCD
#define DADOS_LCD PORTD //4 bits de dados do LCD na porta D
#define RS PD2 //pino de instrução ou dado para o LCD
#define E PD3 //pino de enable do LCD
unsigned char J=1; //variável para inicializar corretamente o LCD

//Macros empregados para o trabalho com os bits de uma das portas(Adaptação do livro
Técnicas de Projeto eletrônicos com Microcontroladores AVR)
#define set_bit(Y,bit_x) (Y|=(1<<bit_x)) //ativa o bit_x da variável Y (coloca
em 1)
#define clr_bit(Y,bit_x) (Y&=~(1<<bit_x)) //limpa o bit_x da variável Y (coloca
em 0)
#define tst_bit(Y,bit_x) (Y&(1<<bit_x)) //testa o bit_x da variável Y (retorna
0 ou 1)
#define cpl_bit(Y,bit_x) (Y^=(1<<bit_x)) //troca o estado do bit x da variável Y
(complementa)

#define b_mais PB0 //B_mais (Botão de incremento) é o substituto de PB2 na
programação
#define b_menos PB1 //B_menos (Botão de decremento) é o substituto de PB3 na
programação
#define b_ok PB2 //B_ok (Botão de decremento) é o substituto de PB3 na
programação

#define LED_UMID PB3

enum {TEMP,UMI};

#define DATA_1 set_bit(PORTC,PC0)
#define DATA_0 clr_bit(PORTC,PC0)
#define SCK_1 set_bit(PORTC,PC1)
#define SCK_0 clr_bit(PORTC,PC1)
#define DATA_IN tst_bit(PINC,PC0)

#define noACK 0
#define ACK 1

```

```

//Padrão de comandos para o SHT15
#define STATUS_REG_W 0x06 //000 0011 0
#define STATUS_REG_R 0x07 //000 0011 1
#define MEDIR_TEMP 0x03 //000 0001 1
#define MEDIR_UMID 0x05 //000 0010 1
#define RESET 0x1e //000 1111 0

//Variáveis globais
unsigned int tempvalue[2]={1,0};
char umidade_minima=50;
char umidade_maxima=90;

typedef union
{
    unsigned int i;
    float f;
}value;
//-----
//Sub-rotina para enviar comandos ao LCD com dados de 4 bits
void cmd_LCD(unsigned char c, char cd)
{
    DADOS_LCD = c;//primeiro os 4 MSB (PD4 - PD5 - PD6 - PD7) -> (D4 - D5 - D6 - D7
LCD)
    for(;J!=0;J--)
    {
        if(cd==0)
            cbi(PORTD,RS);
        else
            sbi(PORTD,RS);
            sbi(PORTD,E);
            cbi(PORTD,E);
            _delay_us(45);
            if((cd==0)&&(c<127))
                _delay_ms(2); //se for instrução espera tempo de resposta do display
            DADOS_LCD = c<<4; //4 bits menos significativos (nibble)
    }
    J = 2;
}

//Sub-rotina de inicializacao do LCD
void inic_LCD(void) //envio de instrucoes para o LCD
{
    // Como o LCD inicia no modo de 8 bits e somente 4 são conectados
    // o primeiro comando deve ser mandado duas vezes
    // a primeira para chavear para o modo de 4 bits (os bits menos
    // significativos não são vistos
    // a segunda para enviá-lo como 2 nibbles, então os menos
    // significativos são reconhecidos
    cmd_LCD(0x28,0); //interface de 4 bits
    cmd_LCD(0x28,0); //interface de 4 bits 2 linhas (aqui serao habilitadas as 2
linhas)
    cmd_LCD(0x0C,0); //mensagem aparente cursor ativo piscando
    cmd_LCD(0x01,0); //limpa todo o display
    cmd_LCD(0x80,0); //escreve na primeira posição a esquerda - 1ª linha
}

//Sub-rotina de escrita no LCD
void escreve_LCD(char *c)
{
    for (; *c!=0;c++) cmd_LCD(*c,1);
}

```

```

//Sub-rotina de escrita de variáveis do tipo int no LCD
void escreve_variavel_LCD(int i)
{
    char buff[17];

    itoa(i,buff,10); //extern char *itoa(int __val, char *__s, int __radix);
    escreve_LCD(buff);
    /**
    \brief Convert an integer to a string.

    The function itoa() converts the integer value from \c val into an
    ASCII representation that will be stored under \c s. The caller
    is responsible for providing sufficient storage in \c s.

    \note The minimal size of the buffer \c s depends on the choice of
    radix. For example, if the radix is 2 (binary), you need to supply a buffer
    with a minimal length of 8 * sizeof (int) + 1 characters, i.e. one
    character for each bit plus one for the string terminator. Using a larger
    radix will require a smaller minimal buffer size.

    \warning If the buffer is too small, you risk a buffer overflow.

    Conversion is done using the \c radix as base, which may be a
    number between 2 (binary conversion) and up to 36. If \c radix
    is greater than 10, the next digit after \c '9' will be the letter
    \c 'a'.

    If radix is 10 and val is negative, a minus sign will be prepended.

    The itoa() function returns the pointer passed as \c s.
    */
}
//-----
//-----
// escreve um byte no no barramento do Sensor e checa confirmação ack=1
char sht_escreve_byte(unsigned char dado)
{
    unsigned char i,error=0;
    DDRC = 0b00000011; // PC0 e PC1 como saídas
    for (i=0x80;i>0;i/=2) //mudar bit para mascarar
    {
        if (i & dado)
            DATA_1; //mascarando valor com i, escreva no barramento do sensor
        else
            DATA_0;
        _delay_us(6);
        SCK_1; //clk para o sensor
        _delay_us(10); //tempo de 5 us
        SCK_0;
        _delay_us(6);
    }
    DATA_1; //libera barramento de dados
    DDRC = 0b00000010; // PINO de dados como entrada PC0
    _delay_us(6); //tempo de configuração
    SCK_1; //nono clock para o ack
    _delay_us(7);
    error=DATA_IN; //verificar ack (DATA será puxado para baixo por SHT11)
    _delay_us(6);
    SCK_0;
    return error; //erro = 1 em caso de não confirmado
}

```

```

}
//-----
//-----
/*/Lê dados do barramento do sensor e dá uma confirmação em caso de "ack = 1"
char sht_le_byte(unsigned char ack)
{
    unsigned char i,val=0;
    DDRC = 0b00000010;    // PINO de dados como entrada PC0
    _delay_us(5); //
    for (i=0x80;i>0;i/=2)
    {
        SCK_1; //clk do barramento do sensor
        if (DATA_IN) val=(val | i); //ler bit
        _delay_us(7);
        SCK_0;
    }
    DDRC = 0b00000011;
    _delay_us(5); //
    if (ack>=1)
    {
        DATA_0;
        _delay_us(5); //
    }
    else
    {
        DATA_1;
        _delay_us(5); //
    }
    SCK_1; //clk #9 para ack
    _delay_us(10); //
    SCK_0;
    _delay_us(6);
    DATA_1; //libera barramento de dados do sht15
    _delay_us(5); //
    return val;
}
*/
//Lê dados do barramento do sensor e dá uma confirmação em caso de "ack = 1"
char sht_le_byte(unsigned char ack)
{
    unsigned char i,val=0;
    DDRC = 0b00000010;    // PINO de dados como entrada PC0
    _delay_us(5); //
    for (i=0x80;i>0;i/=2)
    {
        SCK_1; //clk do barramento do sensor
        if (DATA_IN) val=(val | i); //ler bit
        _delay_us(7);
        SCK_0;
    }
    DDRC = 0b00000011;
    _delay_us(5); //
    if (ack>=1)
    {
        DATA_0;
        _delay_us(5); //
    }
    else
    {
        DATA_1;
        _delay_us(5); //
    }
}

```



```

//-----
//-----
// Reinicia o sensor por softreset
char sht_softreset(void)
{
    unsigned char error=0;
    sht_connectionreset(); //reiniciando comunicação
    error+=sht_escreve_byte(RESET); //envia comando de reinício ao sensor
    return error; //error=1 caso não haja resposta do sensor
}
//-----
//-----
// lê o registrador de status com checksum (8-bit)
char sht_read_statusreg(unsigned char *p_value, unsigned char *p_checksum)
{
    unsigned char error=0;
    sht_transstart(); //transmissão de início
    error=sht_escreve_byte(STATUS_REG_R); //envia comando para o sensor
    *p_value=sht_le_byte(ACK); //Lê o registra dor de Status (8-bit)
    *p_checksum=sht_le_byte(noACK); //Lê checksum (8-bit)
    return error; //error=1 in case of no response form the sensor
}
//-----
//-----
// escreve no registrador de status com checksum( 8-bit)
char sht_write_statusreg(unsigned char *p_value)
{
    unsigned char error=0;
    sht_transstart(); //transmission start
    error+=sht_escreve_byte(STATUS_REG_W); //send command to sensor
    error+=sht_escreve_byte(*p_value); //send value of status register
    return error; //error>=1 in case of no response form the sensor
}
//-----
//-----
// faz uma medição (umidade / temperatura) com checksum
char sht_measure(unsigned char *p_value, unsigned char *p_checksum, unsigned char
mode)
{
    unsigned char error=0;
    unsigned int temp=0;
    sht_transstart(); //transmissão de início
    switch(mode)//envia comando ao sensor
    {
        case TEMP : error+=sht_escreve_byte(MEDIR_TEMP); break;
        case UMI : error+=sht_escreve_byte(MEDIR_UMID); break;
        default : break;
    }
    DDRC = 0b00000010; //pino de dados como entrada
    _delay_us(5); //
    while(1)
    {
        if(!DATA_IN) break; //espera o sensor efetuar as medidas
    }
    if(DATA_IN) error+=1;

    switch(mode) // envio de comando ao sensor
    {

```

```

        case TEMP :
            temp=0;
            temp=sht_le_byte(ACK);
            temp<<=8;
            tempvalue[0]=temp;
            temp=0;
            temp=sht_le_byte(ACK);
            tempvalue[0]|=temp;
            break;

        case UMI :
            temp=0;
            temp=sht_le_byte(ACK);
            temp<<=8;
            tempvalue[1]=temp;
            temp=0;
            temp=sht_le_byte(ACK);
            tempvalue[1]|=temp;
            break;

        default : break;
    }
    *p_checksum =temp=sht_le_byte(noACK); //read checksum
    return error;
}

//-----
//-----
// calculates temperature [°C] and humidity [%UH]
// input : umidade [Ticks] (12 bit) temp [Ticks] (14 bit)
// output: umidade [%UH] temp [°C]
float calc_sht15(float *p_humidity ,float *p_temperature)
{
    const float C1=-2.0468; // 12 Bit UH
    const float C2=0.0367; // 12 Bit UH
    const float C3=-0.0000015955; // 12 Bit UH
    const float T1=0.01; // 12 Bit UH
    const float T2=0.00008; // 12 Bit UH
    float ur=*p_humidity; // Uh: umidade 12 Bit
    float t=*p_temperature; // t: Temperature [Ticks] 14 Bit
    float ur_lin; // ur_lin: umidade linear
    float ur_true; // ur_true: Temperatura, umidade compensado
    float t_C; // t_C : Temperature [°C]
    t_C=t*0.01 - 40.1; //calc. temperatura [° C] a partir de 14 bit temp.ticks @ 5V
    ur_lin=C3*ur*ur + C2*ur + C1; //calc. umidade do ticks para [% UR]
    ur_true=(t_C-25)*(T1+T2*ur)+ur_lin; //calc. temperatura, umidade compensado[%UH]

    if(ur_true>100)ur_true=100; //cortado se o valor está fora do
    if(ur_true<0.1)ur_true=0.1; //o alcance físico possível
    *p_temperature=t_C; //retorna temperatura [° C]
    *p_humidity=ur_true; //retorna umidade [% UH]
}
//-----

void config_control_umidade(void)
{
    PORTB=0b00000111;
    int cc1=0;
    int cc2=0;
    cmd_LCD (0x01,0); //limpa o lcd
    escreve_LCD(" CONFIGURAR ");
}

```

```

cmd_LCD(0xC0,0);
escreve_LCD("  UMIDADE RELATIVA");
while(!tst_bit(PINB,b_ok));
_delay_ms(250);
cmd_LCD (0x01,0); //limpa o lcd
while (cc1==0)
{
    cmd_LCD(0xC0,0); // Cursor na 2ª linha e 1ª coluna
    escreve_LCD("  UMIDADE MINIMA");
    cmd_LCD(0x9C,0); // Cursor na 3ª linha 8ª coluna
    escreve_variavel_LCD(umidade_minima);
    escreve_LCD("%");
    if(!tst_bit(PINB,b_mais))//se o botão for pressionado incrementa valor da
umidade mínima
    {
        umidade_minima++;
        _delay_ms(40);
    }
    if (umidade_minima>=91)
    {
        cmd_LCD (0x01,0); //limpa o lcd
        escreve_LCD("Valores entre 1%-90%");
        _delay_ms(200);
        cmd_LCD (0x01,0); //limpa o lcd
        umidade_minima--;
    }
    if(!tst_bit(PINB,b_menos))//se o botão for pressionado decrementa valor
da umidade mínima
    {
        umidade_minima--;
        _delay_ms(40);
        cmd_LCD (0x01,0); //limpa o lcd
    if (umidade_minima<1)umidade_minima++;
    }

    if(!tst_bit(PINB,b_ok))//se o botão for pressionado finda o loop while
    {
        while(!tst_bit(PINB,b_ok));
        cc1=1;
        _delay_ms(20);
    }
}
cmd_LCD (0x01,0); //limpa o lcd
while (cc2==0)
{
    cmd_LCD(0xC0,0); // Cursor na 2ª linha e 1ª coluna
    escreve_LCD("  UMIDADE MAXIMA");
    cmd_LCD(0x9C,0); // Cursor na 3ª linha
    escreve_variavel_LCD(umidade_maxima);
    escreve_LCD("%");
    if (umidade_maxima-umidade_minima<8)
    {
        cmd_LCD (0x01,0); //limpa o lcd
        escreve_LCD("Valor deve ser Maior");
        cmd_LCD(0xC0,0); // Cursor na 3ª linha
        escreve_LCD("em 8% da Minima");
        _delay_ms(150);
        cmd_LCD (0x01,0); //limpa o lcd
        umidade_maxima=umidade_minima+8;
    }
    if(!tst_bit(PINB,b_mais))//se o botão for pressionado incrementa valor da

```

```

umidade máxima
{
    umidade_maxima++;
    _delay_ms(40);
    if (umidade_maxima>100)umidade_maxima--;
}
if(!tst_bit(PINB,b_menos))//se o botão for pressionado decrementa valor
da umidade máxima
{
    umidade_maxima--;
    _delay_ms(40);
    cmd_LCD (0x01,0); //limpa o lcd
}
if(!tst_bit(PINB,b_ok))//se o botão for pressionado quebra o loop while
{
    while(!tst_bit(PINB,b_ok));
    cc2=1;
    _delay_ms(40);
}
}
cmd_LCD (0x01,0);// limpa lcd
cmd_LCD(0x94,0); // Cursor na 3ª linha e 1ª coluna
escreve_LCD("### CONFIGURADO ###");
_delay_ms(180);
cmd_LCD (0x01,0); //limpa lcd
}

void config_control(void)//!!! corrigir exibição de variável
{
    config_control_umidade();
}

void msg_inicial(void)
{
    DDRC = 0xFF; //Porta D como entrada
    escreve_LCD("#####");
    cmd_LCD(0xC0,0); // Cursor na 2ª linha e 1ª coluna
    escreve_LCD("#### CONTROLE ####");
    cmd_LCD(0x94,0); // Cursor na 3ª linha e 1ª coluna
    escreve_LCD("### DE UMIDADE ###");
    cmd_LCD(0xD4,0); // Cursor na 4ª linha e 1ª coluna
    escreve_LCD("#####");
    _delay_ms(1000);//*****
*
    cmd_LCD (0x01,0); //limpa lcd
}

int main()
{
    value umi_val,temp_val;
    unsigned char error, checksum;
    unsigned int i;
    unsigned int A=0;
    unsigned int B;
    char inp;
    char _umidade [5]={0,0,0,0,0};
    char _temperatura [5]={0,0,0,0,0};
    int temperatura;
    int umidade;
    PORTD = 0x00;
    DDRD = 0xFF; //Porta D como saída (Display LCD)
    PORTB = 0b00000111;
}

```

```

DDRB = 0b00011000;
PORTC = 0x00; // todos os bits da porta C = 0
DDRC = 0x00; //Porta D como entrada

sht_connectionreset();
inic_LCD( ); //Inicializa o LCD
msg_inicial();//DESABILITADA P TESTES*****
config_control();//DESABILITADA P TESTES*****
escreve_LCD("EFETUANDO MEDIDAS");
while(1)
{
    _delay_ms(230);
    error=0;
    error+=sht_measure((unsigned char*) &umi_val.i,&checksum,UMI); //medir
umidade
    error+=sht_measure((unsigned char*) &temp_val.i,&checksum,TEMP); //medir
temperatura

    if(error!=0)
    {
        sht_connectionreset();
        cmd_LCD (0x01,0); //limpa lcd
        escreve_LCD("ERRO DE MEDIÇÃO!!");
        _delay_ms(200);
    } //em caso de erro reinicia conexão
    else
    {
        umi_val.f=(float)tempvalue[1]; //integer to float
        temp_val.f=(float)tempvalue[0]; //integer to float
        calc_sth15(&umi_val.f,&temp_val.f); //calculate humidity,
temperature

        umidade = (int) umi_val.f;
        temperatura = (int) temp_val.f;

        cmd_LCD (0x01,0); //limpa lcd
        escreve_LCD("UMIDADE:");
        escreve_variavel_LCD(umidade);
        escreve_LCD("%");
        cmd_LCD(0xC0,0); // Cursor na 2ª linha e 1ª coluna
        escreve_LCD("TEMPRATURA:");
        escreve_variavel_LCD(temperatura);
        escreve_LCD("oC");

        if (umidade_minima<=umidade)
        {
            if(umidade_maxima<umidade)
            {
                cmd_LCD(0xD4,0); // Cursor na 4ª linha e 1ª coluna
                A=0;
                escreve_LCD("UMIFICADOR DESLIGADO");
            }
            if (A==0)
            {
                clr_bit(PORTB,PB4);
                clr_bit(PORTB,LED_UMID);
                cmd_LCD(0xD4,0); // Cursor na 4ª linha e 1ª coluna
                escreve_LCD("UMIFICADOR DESLIGADO");
            }
            _delay_ms(100);
        }
    }
}

```