

MINISTÉRIO DA EDUCAÇÃO  
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA  
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MINAS GERAIS  
CAMPUS SÃO JOÃO EVANGELISTA

Curso Superior de Sistemas de Informação

Glaysson Gonçalves de Oliveira  
Jean Christopher de Oliveira Pinheiro  
Neilson Luis Medeiros

**SISTEMA DE CONTROLE DE UMIDADE E TEMPERATURA (SCUT)**

SÃO JOÃO EVANGELISTA - MINAS GERAIS  
NOVEMBRO 2013

Glaysson Gonçalves de Oliveira  
Jean Christopher de Oliveira Pinheiro  
Neilson Luis Medeiros

**SISTEMA DE CONTROLE DE UMIDADE E TEMPERATURA (SCUT)**

Monografia apresentada como requisito para  
obtenção do título de Bacharel em Sistemas de  
Informação do Instituto Federal de Minas Gerais –  
Campus São João Evangelista.

Orientador: Dr. Márcio Takeshi Sugawara  
Coorientadora: Ma. Karina Dutra de Carvalho  
Lemos  
Coorientador: Me. Cleonir Coelho Simões

SÃO JOÃO EVANGELISTA - MINAS GERAIS  
NOVEMBRO 2013

## FICHA CATALOGRÁFICA

Elaborada pelo Serviço Técnico da Biblioteca do  
Instituto Federal Minas Gerais – Campus São João Evangelista

O48s OLIVEIRA, Glaysson Gonçalves de, 1993 -

Sistema de controle de umidade e temperatura (SCUT)./  
Glaysson Gonçalves de Oliveira; Jean Christopher de Oliveira  
Pinheiro; Neilson Luís Medeiros. São João Evangelista, MG: IFMG –  
Campus São João Evangelista, 2013.  
65 p.: il.

Trabalho de Conclusão de Curso - TCC (graduação)  
apresentado ao Instituto Federal Minas Gerais – Campus São João  
Evangelista – IFMG, Curso de Bacharelado em Sistemas de  
Informação, 2013.

Orientador: Prof. Dr. Márcio Takeshi Sugawara  
Coorientadores: Prof. Ma. Karina Dutra de Carvalho Lemos;  
Prof. Me. Cleonir Coelho Simões

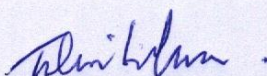
1. Sistema de informação. 2. Software livre . 3. Software. I.  
Instituto Federal Minas Gerais – Campus São João Evangelista.  
Curso de Bacharelado em Sistemas de Informação. II. Título.

CDD 006

Glaysson Gonçalves de Oliveira  
Jean Christopher de Oliveira Pinheiro  
Neilson Luis Medeiros

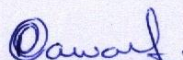
## **SISTEMA DE CONTROLE DE UMIDADE E TEMPERATURA (SCUT)**

Monografia apresentada ao Curso de Bacharelado em Sistemas de Informação do Instituto Federal de Minas Gerais – Campos São João Evangelista, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.



---

Márcio Takeshi Sugawara (Orientador) – IFMG



---

Karina Dutra de Carvalho Lemos (Coorientadora) – IFMG



---

Cleonir Coelho Simões (Coorientador) – IFMG

A Deus pela  
força, saúde e  
perseverança e aos meus  
pais, por todo incentivo  
e carinho.

São João Evangelista, 01 de Novembro de 2013.

A Deus pela  
força, saúde e  
perseverança e aos meus  
pais, por todo incentivo  
e carinho.

## **AGRADECIMENTOS**

A todos que contribuíram para a realização deste trabalho, fica aqui a nossa imensa gratidão, especialmente:

Aos Professores Márcio Takeshi Sugawara, Karina Dutra de Carvalho Lemos e Cleonir Coelho Simões, pela orientação, pelo aprendizado e apoio em todos os momentos necessários no trabalho.

A setor de manutenção em especial Técnico Sergio dos Anjos pela ajuda no desenvolvimento do trabalho quanto aos componentes.

Ao Professor Ícaro Tourino Alves pela ajuda na elaboração inicial do trabalho em conhecimentos técnicos.

A todos que, de alguma forma, contribuíram para construção deste trabalho que houve muita dedicação.

## RESUMO

Foi realizado um estudo exploratório no Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais Campus São João Evangelista, no departamento agrário, para a criação de um sistema de controle da temperatura e umidade do ar em ambientes fechados, visando proporcionar um clima ideal tanto para cultivo quanto para simulações em pesquisas acadêmicas. Pesquisas sobre o tema foram realizadas em visitas às estufas, entrevistas com profissionais da área e colhendo dados necessários para análise e estudo. Posteriormente foi desenvolvido um sistema que realiza controle de temperatura e umidade do ar. O mesmo é uma combinação de hardware e software, que gerencia de forma eficaz o controle de ambientes fechado, proporcionando um ambiente ideal para estudos e plantio. O sistema foi desenvolvido de forma que possibilite uma grande interação com o usuário, usando para isso várias formas de *feedback*. O Sistema de Controle de Temperatura e Umidade – SCUT possibilita ao usuário definir os parâmetros para controle de acordo com sua necessidade, se mostrando muito flexível quanto a sua utilização. Propiciando maior domínio ao usuário sobre sua produção, uma vez que atende as necessidades fisiológicas da planta, observadas previamente pelo usuário, no tempo em que ela se faz necessária. Desta maneira o ambiente estará o mais próximo do ideal possível, dentro do pretendido pelo usuário.

Palavras-chave: Controladores de temperatura, controladores de umidade, ambientes fechados.

## **ABSTRACT**

An exploratory study was conducted at the Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais Campus São João Evangelista, in the agricultural department for creating a system to control temperature and humidity in closed environments, aiming to provide an ideal climate both for cultivation and for simulations in academic researches. Researches on the topic were conducted in greenhouses visits, interviews with professionals of the area and collecting data needed for analysis and study. Later a system that performs temperature control and humidity was developed. The same is a combination of hardware and software, that manage effectively the control of closed environments, providing an ideal environment for study and planting. The system was developed in a way that allows a large user interaction, using for it various forms of feedback. The Control System of Temperature and Humidity - CSTH enables the user to define the parameters to control according to your needs, proving to be very flexible in its use. Provides the user greater control over his production, since it meets the physiological needs of the plant, previously observed by the user, on the time when it makes necessary. Thus the environment will be the closest possible to the ideal as within the intended by the user .

**Keywords:** temperature controllers, humidity controllers, closed environments.

## LISTA DE SIGLAS

$\mu$ A – Microfarad

°C – Grau Celsius

A/D – Analógico Digital

Amp – Ampere

CCP – Captures, Compare & PWM

CI - Circuito Integrado

CPU – Unidade Central de Processamento

EEPROM – *Electrically Erasable Programmable Read-Only Memory*

EMBRAPA – Empresa Brasileira de Pesquisa Agropecuária

EPROM – *Erasable Programmable Read-Only Memory*

fem – Força Eletromotriz

GND – Graduated Neltral Density Filter

IBGE – Instituto Brasileiro de Geografia e Estatística

IFMG - SJE -Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais Campus

São João Evangelista

Khz – Quilohertz

LCD – Tela de Cristal Líquido

LED - Diodo Emissor de Luz

mA – Micro Ampere

Mhz- Megahertz

pF – Picofarad

*PIC – Peripheral Interface Controller*

PVC – Policloreto de polivinila

RF – Requisitos Funcionais

RNF – Requisitos Não Funcionais

SCUT – Sistema de Controle de Umidade e Temperatura

TCC – Trabalho de Conclusão de Curso

UART – *Universal Asynchronous Receiver/Transmitter*

VDC – Voltagem Direct Current

## SUMÁRIO

1 INTRODUÇÃO.....	11
2 FUNDAMENTAÇÃO TEÓRICA .....	13
2.1 Adversidades climáticas .....	13
2.2 Métodos de controle de temperatura e umidade relativa do ar.....	14
2.2.1 Leitura do clima em ambientes fechados .....	15
2.2.2 Controle de ambiente fechados .....	17
2.3 Microcontroladores.....	19
2.4 Metas de usabilidade .....	21
3 METODOLOGIA.....	26
3.1 Levantamento dos requisitos .....	26
3.2 Estrutura lógica do SCUT .....	27
3.2.1 Fluxograma do SCUT.....	27
3.2.2 Desenvolvimento do código .....	30
3.3 Estrutura física do projeto SCUT .....	33
3.3.1 Construção do protótipo da estufa.....	34
3.3.2 Montagem dos módulos .....	35
3.4 Integração e testes.....	39
4 RESULTADOS E DISCUSSÕES.....	41
5 CONCLUSÃO.....	44
REFERÊNCIAS .....	45
APÊNDICE .....	49
ANEXOS.....	55

## 1 INTRODUÇÃO

O Brasil atualmente apresenta um número populacional consideravelmente elevado, segundo dados do Instituto Brasileiro de Geografia e Estatística – IBGE, há 201,03 milhões de habitantes no mesmo (IBGE, 2013). Segundo a EXAME (2011), o Brasil ocupa a quinta posição dos países mais populosos do mundo ficando atrás apenas da China, Índia, Estados Unidos e Indonésia. Com esse linear crescimento populacional em que se encontra o mundo esta questão se tornou uma dos assuntos mais debatidos e motivo de preocupação entres os líderes mundiais. A falta de alimento no mundo e a criação de métodos para melhoria da produção agrícola mundial estão intimamente relacionadas com esse crescimento.

Segundo a Empresa Brasileira de Pesquisa Agropecuária - Embrapa (2008), no Brasil o fator climático é um dos maiores desafios para a produção agrícola. Contudo técnicas foram aprimoradas o intuito de se obter melhoras na produção agrícola como a de alimentos, madeiras, entre outros. Tecnologias sugeriram para atender as necessidades fisiológicas das plantas para alcançar uma produção em alta escala e com qualidade, independente da região no qual esta sendo produzida, reduzindo os efeitos climáticos nos quais afetam o desenvolvimento da planta.

Tendo em vista o alto crescimento populacional, alta demanda de produção agrícola e as diversidades climáticas no Brasil, foi percebida a necessidade de se criar um sistema computacional que realizasse o controle de temperatura e umidade do ar de um ambiente fechado. Por meio de visitas às estufas do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais Campus São João Evangelista – IFMG-SJE e da colaboração de alguns profissionais da área, foi possível aperfeiçoar a ideia de se produzir o Sistema de Controle de Umidade e Temperatura - SCUT.

O SCUT tem por objetivo, reduzir as diversidades climáticas, proporcionando um ambiente favorável à produção agrícola, permitindo assim que pesquisadores tenham ambientes diversificados, para estudo de melhores técnicas e métodos relacionados ao desenvolvimento de plantações, com a finalidade de proporcionar melhorias na qualidade e quantidade das mesmas. Por meio do SCUT, os processos são realizados de forma automatizada, reduzindo os custos com mão-de-obra operacional.

Através do SCUT é possível automatizar o processo de controle do ambiente para cultivo, controlando a temperatura e umidade relativa do ar, fazendo o uso de sensores para identificar os estímulos do ambiente que são interpretados por um *PIC - Peripheral Interface*

*Controller*. O sistema é responsável por acionar os exaustores ou não, sistema de irrigação e/ou nebulização de acordo com os parâmetros definidos pelo usuário. Por meio dos parâmetros informados, o sistema trabalhará com variações máximas e mínimas de acordo com a demanda de determinada produção agrícola.

Este trabalho desenvolvido teve como objetivo aperfeiçoar processos importantes em ambientes fechados, mostrando que um sistema automatizado pode proporcionar um trabalho mais efetivo, minimizando falhas humanas no processo de controle de umidade e temperatura de ambientes, pois o controle passa a ser realizado por um sistema computacional, diminuindo também os custos com a mão-de-obra operacional.

O trabalho encontra-se dividido em capítulos, sendo este, a introdução, que apresentou o tema da pesquisa, a justificativa bem como a problemática tratada. Foram apresentados também os objetivos, a relevância do desenvolvimento do sistema e sua abrangência. No segundo capítulo está descrita a fundamentação teórica, descrevendo os conceitos e técnicas analisadas para o desenvolvimento do SCUT, bem como os instrumentos e procedimentos utilizados para sua elaboração e também trabalhos correlatados. O terceiro capítulo descreve a metodologia apresentando as especificações, os processos de implementação lógica e física e a funcionalidade do sistema de controle de temperatura e umidade. No quarto capítulo estão apresentados os resultados e discussões obtidos nos testes realizados com o SCUT, bem como os seus erros e suas respectivas resoluções. O quinto capítulo trata-se das conclusões obtidas no desenvolvimento e também possíveis melhorias.

## 2 FUNDAMENTAÇÃO TEÓRICA

Segundo Lakatos e Marconi (2011), para o desenvolvimento de projetos é necessário ter um assunto bem definido e embasamento teórico com referências para a projeção do trabalho. Com base em estudos realizados, foram encontradas obras que discriminam componentes, teorias e métodos que serviram de embasamento para o desenvolvimento desse trabalho. A seguir apresentamos algumas das obras anteriormente citadas.

### 2.1 Adversidades climáticas

De acordo com Vianello (2012), o Brasil é um dos países que possui considerável topologia climática, devido a sua grande extensão geográfica com interferência de fatores climáticos. Os fatores climáticos que se destacam no país são a localização geográfica e a grande extensão territorial do Brasil, a configuração geográfica triangular do território próximo à linha do equador, a maritimidade e a continentalidade, as pequenas altitudes do relevo, a dinâmica na massa de ar, a vegetação e a urbanização com grande concentração industrial.

Vianello (2012) define que no Brasil há cinco regiões com climas distintos, a região Norte é caracterizada pelas altas temperaturas e elevados índices pluviométricos, a região Nordeste é caracterizada por alta temperatura o ano todo, com uma precipitação anual de 1.600 mm anuais no litoral enquanto no interior não ultrapassa de 400 mm. Na região Centro-Oeste caracterizada por possuir duas estações com condições bastante contrastantes entre as estações do verão e do inverno, o verão chuvoso e quente, enquanto o inverno é seco e menos quente. A região Sudeste possui o inverno com poucas chuvas e geadas frequentes, no verão a elevadas temperaturas e chuvas abundantes associadas ao aquecimento superficial. Por fim a região Sul as quatro estações do ano são bem definidas, contudo as chuvas são bem distribuídas ano inteiro. Dessa forma os fatores climáticos influenciam diretamente na produção agrícola, obrigando os produtores a se adaptarem.

Para muitos produtores o fator climático é o principal processo de sua atividade no campo, visto que as variações climáticas ocorrem com muita frequência, trazendo prejuízos financeiros importantes para os produtores. Segundo Tubelis (2001), é necessário se ter um entendimento da variação climática para o controle de plantio, visando à produtividade e juntamente a isto fazer um parecer dos efeitos negativos das diversidades climáticas.

Para Tubelis (2001) os produtores devem procurar soluções para eliminar a causa desses problemas. Uma das soluções apresentadas por ele é que durante períodos onde a temperatura está muito baixa deve-se fazer o uso do plantio da cultura em ambientes fechados como túneis de plástico transparente onde a temperatura do ar é de cerca de 8 a 10°C acima da temperatura do ar a céu aberto.

Ainda segundo ele, plantas demandam de alguns períodos de dias mais longos, e outras em períodos de dias mais curtos, onde estes proporcionam o desenvolvimento fotossintetizante, para a solução do problema fotoperíodo inadequado. Para tal alteração climática é possível, através de um filme de plástico preto cobrindo as plantas, simular períodos noturnos maiores. Enquanto que para solucionar o problema de necessidade de luz, ou seja, de um período diurno com maior duração, costuma-se iluminar artificialmente o ambiente por meio de lâmpadas. (TUBELIS, 2001).

Para um melhor entendimento das adversidades climáticas torna-se necessário estudar alguns métodos utilizados na leitura e no controle de temperatura e umidade relativa do ar, conforme veremos a seguir.

## **2.2 Métodos de controle de temperatura e umidade relativa do ar**

De acordo com Tubelis (2001) para a verificação de temperatura nos mais variados ambientes há muitos equipamentos para medição da mesma: termômetro que se divide em digital, infravermelho e comum; psicrômetro que se divide em comum e fixo, comum e giratório, com aspiração e fixo; higrômetro se divide em comum e externo; termo higrômetro se divide em comum e digital; pluviométrico que se divide em leitura direta, convencional e digital.

Para o controle de umidade relativa do ar, Herech e Paciência (2008), no projeto “Controle automatizado de umidade relativa em ambientes através de psicrometria”, apresenta um sistema de ventilação responsável pela circulação do ar no ambiente. Ele é acionado pelo *PIC*, que vai emitir um pulso para o relé que aciona os exaustores pelo fechamento do circuito que lhes fornece corrente. Para controlar a umidade da estufa, foi utilizado um bico de aspersão que borrifa a água sobre o ar circulante no ambiente. Através de uma bomba pressurizadora, a água é coletada, pressurizada e conduzida até outro reservatório. Uma válvula solenoide acionada pelo *PIC* é responsável por liberar água pressurizada para os bicos de nebulização. O excesso de água aspergida pela bomba é direcionado até o primeiro

reservatório, que futuramente realimentara o sistema de aspersão de água. A água espessa pelos bicos é usada para aumentar a umidade relativa do ambiente até alcançar um índice desejável.

Macintyre define em Borré como ventilação do ambiente, o deslocamento do ar onde o mesmo pode ser realizado de forma natural ou mecânico, retirando ou fornecendo ar para um ambiente proporcionando a renovação do mesmo, bem como controle de temperatura e umidade. (MACINTYRE apud BORRÉ, 2013, p.15).

Para que se possa fazer a leitura da temperatura e da umidade interna dos ambientes são necessários o uso de sensores, que serão demonstrados na subseção seguinte.

### ***2.2.1 Leitura do clima em ambientes fechados***

De acordo com Ceron e Crespi (2013), para a medição de temperatura há vários tipos de sensores no mercado, dentre eles o LM35 que apresenta a vantagem da simplicidade do seu manuseio para medição de temperatura. O mesmo é um sensor de precisão, ou seja, não necessita calibração externa para fornecer com exatidão a temperatura. O mesmo trabalha com variações de  $\frac{1}{4}$  de Graus Celsius - °C ou até mesmo  $\frac{3}{4}$ °C em um faixa de -55°C até 150°C, com uma saída de tensão linear fornecendo o valor da temperatura do ambiente em tempo real por meio de uma alimentação com tensão de 4-20 Vdc e GND, onde terá como saída, um sinal de 10mV/°C.

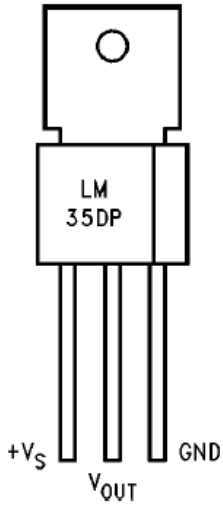
Ainda de acordo com Ceron e Crespi (2013), o LM35 pode ser alimentado de duas formas: alimentação simples ou simétrica, dependendo do que se deseja com o sinal de saída. Contudo a saída ainda assim continuará sendo de 10mV/ °C. Quanto ao encapsulamento, os autores dizem que “O sensor LM35 é apresentado com vários tipos de encapsulamentos, sendo o mais comum o TO-92, que mais se parece com um transistor, e oferece ótima relação custo benefício, por ser o mais barato dos modelos e propiciar a mesma precisão dos demais.” (CERON; CRESPI, 2013, p.4). Para compreender melhor o funcionamento do seu sistema é necessário conhecer os dados presentes em seu datasheet. A figura 1 apresenta o sensor de temperatura LM35.

**Figura 1 – Sensor de Temperatura**



**Fonte: Próprio do sistema, 2013**

Tabela 1 – Dados do LM35

Features	TO-202 Plastic Package
<ul style="list-style-type: none"> <li>• Calibrated directly in ° Celsius (Centigrade)</li> <li>• Linear + 10.0 mV/°C scale factor</li> <li>• 0.5C accuracy guaranteeable (at +25°C)</li> <li>• Rated for full -55° to +150°C range</li> <li>• Suitable for remote applications</li> <li>• Low cost due to wafer-level trimming</li> <li>• Operates from 4 to 30 volts</li> <li>• Less than 60 <math>\mu</math>A current drain</li> <li>• Low self-heating, 0.08°C in still air</li> <li>• Nonlinearity only <math>\pm 1/4^\circ\text{C}</math> typical</li> <li>• Low impedance output, 0.1 <math>\Omega</math> for 1 mA load</li> </ul>	
	TL/H/5516-24

Fonte: DatasheetCatalog, 2013

O sensor LM35 pode ser utilizado nos mais variados ambientes, no entanto, o mesmo pode ter uma interferência na medição quanto seu posicionamento. Se o sensor estiver sido colocado sobre uma superfície e a superfície tiver divergência de temperatura quanto ao ar, o sensor fará uma leitura intermediária quanto à temperatura da superfície e a do ar. De acordo com Ceron e Crespi (2013) para a solução da variância de temperatura lida pelo sensor, deverá fixar os fios e o próprio LM35 a uma superfície por meio de um leve revestimento de cola epóxi, onde o mesmo não terá mais contato com o ar, fazendo com que a leitura seja exata quanto à superfície desejada.

Ainda segundo Ceron e Crespi (2013), o LM35 assim com o a maioria dos circuitos de pequena força, possui uma limitação em chavear cargas capacitivas pesadas. O mesmo tem a capacidade de chavear a capacitâncias de até 50 pF sendo necessário o uso de resistores ou de circuitos amortecedores quando seu uso for destinado a cargas maiores.

Além de se preocupar com o controle de temperatura torna-se também necessário compreender o controle de umidade. Para Abreu e Sofia (2012), a umidade é a medida da quantidade de vapor d'água presente no ar. No entanto essa quantidade pode ser absoluta, relativa ou temperatura de ponto de orvalho. Abreu e Sofia (2012) ainda falam que podem ser usados quatro métodos para medição da umidade relativa do ar. Sendo estes por meio do psicrómetro, sensores capacitivos, sensores resistivos e sensores de condutividade térmica.

Para o desenvolvimento deste trabalho foi feito um estudo sobre os sensores capacitivos, que de acordo com Abreu e Sofia (2012, p. 15) são “[...] constituídos por uma película fina de polímero ou óxido metálico depositado entre elétrodos metálicos num substrato cerâmico ou de vidro”. Contudo existem vários tipos de sensores de umidade do ar capacitivos como o HS1101LF, HCH-1000, HIH 4000-001, HIH-5031, SHT10, entre outros. Os mesmos podem operar até uma temperatura máxima de 200 °C, tendo resistência a vapores químicos.

Dentre os sensores citados, o HIH 4000-001 foi o escolhido para o desenvolvimento do presente trabalho, cujas especificações são encontradas no ANEXO A. Para Figueiredo (2011), o sensor HIH 4000-001 de umidade, possui alta precisão na leitura, no qual pode ser integrado a um controlador programável, que recebe e analisa os dados, decidindo o que fazer, de acordo com a programação escolhida para o desenvolvimento desejado.

Feita a leitura do ambiente é necessário que seja feito o controle climático do mesmo. Para isto o uso de exaustores e de sistema de nebulização são essenciais para controle de ambientes fechados.

### ***2.2.2 Controle de ambiente fechados***

De acordo com Abreu e Sofia (2012) existem dois grandes tipos de sistemas de ventilação: ventilação natural e ventilação forçada.

Os sistemas de ventilação natural consistem em janelas laterais e/ou superiores (zenitais) que podem ser abertas a diferentes ângulos. O ar move-se devido a diferenças de pressão que podem ser causadas pelo vento no exterior da estufa, ou devido ao movimento ascendente do ar quente no interior (efeito térmico). Devem sempre existir aberturas zenitais para que a ventilação não dependa apenas do movimento do ar no exterior. A abertura/fechamento das janelas é efetuado através da utilização de um motor de corrente alternada e o ângulo de abertura é determinado por um controlador em função das variáveis climáticas na estufa (ABREU; SOFIA, 2012).

Ainda de acordo com Abreu e Sofia (2012), os sistemas mecânicos usam ventiladores elétricos para forçar o movimento do ar. Os ventiladores são, geralmente, de grande diâmetro e baixa velocidade. Existem várias estratégias para a colocação e utilização dos ventiladores. A estratégia mais simples consiste em colocar um ventilador na zona superior de uma parede da estufa ou até mesmo na cobertura para expelir o ar para o exterior, obrigando o ar exterior a entrar através de aberturas nas laterais.

Visto o controle de temperatura feito por deslocamento do ar, torna-se necessário que se faça o controle da umidade relativa do ar no ambiente.

De acordo com o Clima Certo (2011) existem diferentes métodos para o controle de umidade do ar em ambientes fechados. Um deles se baseia no aquecimento da água através de uma resistência elétrica, onde o vapor gerado é drenado por um mangote e através de tubos é conduzido ao ambiente. Outro método consiste em instalar centrais, equipadas com unidades lavadoras de ar, onde esse passa por um sistema de atomização de água, eliminadores de gotas que o resfriam e umidificam o ambiente. Há ainda os chamados umidificadores individuais, destinados a áreas menores. Nesse caso, um nebulizador faz atomização muito forte de água, que depois é dispersada no ar, processo que pode ser feito por ventilação ou bicos de ar comprimidos. Num outro método, mais recente, utiliza-se a tecnologia do ultrassom para umidificar, sem muito vapor, o ambiente. Este método é bem semelhante ao que ocorre com aparelhos de inalação, lançando uma névoa que se dissipa no ar. A dispersão das gotículas de água, neste caso, muitas vezes é feita com o auxílio de ventilação.

Para a realização do sistema foi definido que o método de pulverização seria o que mais se adequaria ao proposto no trabalho.

De acordo com Abreu e Sofia (2012) para se realizar um controle de temperatura e umidade pode-se utilizar um sistema de pulverização. Eles ainda acrescentam que um requisito essencial para os sistemas de pulverização é que as gotas de água devem ter um diâmetro suficientemente pequeno para que evaporem antes de tocar na superfície das plantas. É possível criar gotas de tamanho adequado de três formas:

- Utilizando difusores que combinam um fluxo de ar com um fluxo de água. Neste caso não são necessárias pressões muito elevadas em cada um dos fluxos, mas é necessário um compressor de ar e tubagem dupla (para o ar e para a água);
- utilizando difusores de baixa pressão, consegue-se gotas com diâmetro entre 50 a 100  $\mu\text{m}$ ;
- utilizando difusores de alta pressão consegue-se gotas com diâmetros entre 2 e 60  $\mu\text{m}$ . Neste caso, como os orifícios dos difusores têm dimensões muito reduzidas, a água necessita de estar muito limpa e bem tratada para evitar entupimentos.

Os sistemas em si são compostos por um amaciador de água, filtros para prevenir entupimentos, um reservatório de água, bombas, válvulas reguladoras de pressão e tubagem com os difusores sobre a plantação. (ABREU; SOFIA, 2012). Para que os equipamentos anteriormente citados possam executar suas respectivas funções, uma vez que esses deverão

se encontrar dispostos fisicamente de maneira a otimizar o fluxo dos processos, é necessário que haja um comando em resposta aos estímulos captados no ambiente. Esses estímulos devem ser interpretados e tratados por um microcontrolador, este por sua vez envia um comando de execução para os equipamentos de controle.

### 2.3 Microcontroladores

Para interpretar os dados recebidos pelos os sensores, torna-se necessário o uso de microcontroladores para que os métodos de controle de temperatura e umidade relativa do ar sejam executados corretamente.

Segundo Silva (2006), o Microcontrolador é um circuito integrado que pode ser programado. Ele contém todos os componentes básicos de um computador, como CPU (Unidade Central de Processamento), memória para guardar programas, memória de trabalho, portas de entrada e saída para comunicação com o meio externo, sistema de controle de tempo externo e interno, conversores A/D (análogo digital), UART (*Universal Asynchronous Receiver/Transmitter*), de comunicação, entre outros.

De acordo com Silva (2006), em 1965 a GI Microelectronics, iniciou sua caminhada rumo ao microcontrolador, fabricando memórias EPROM (*Erasable Programmable Read-Only Memory*) e EEPROM (*Electrically Erasable Programmable Read-Only Memory*). No início dos anos 70 ela desenhou um microprocessador de 16 bits CP1600. O mesmo trabalhava bem, porém ele era ineficaz no controle das portas de entrada / saída. Na tentativa de resolver o problema em 1975 foi desenhado um chip destinado a controlar portas de entrada / saída. Nascia assim o PIC (*Peripheral Interface Controller*), com uma estrutura mais simples que a do processador e este podia manipular portas de entrada e saída com facilidade, rapidez e eficiência.

“Cada tipo de microcontrolador serve para um propósito e cabe ao projetista selecionar o melhor microcontrolador para seu trabalho“. (SILVA, 2006, p. 9). Para elaboração de um projeto que faça o uso de microcontrolador, é necessário fazer antes o levantamento sobre qual o tipo de caso que o microcontrolador será implementado para poder encontrar o microcontrolador mais compatível com as necessidades e exigências requeridas pelo projeto.

Para Silva (2006, p. 10), “O consumo de corrente do microcontrolador é mínimo, consumindo menos de 2,0 mA com 5 volts, trabalhando a 4Mhz ou 15,0 mA com 3 volts, trabalhando a 32 Khz. Quando em modo *StandBy*, consome menos de 1,0 mA com 3 V.”

Devido a este fato, a alimentação do microcontrolador segue uma estrutura de corrente baixa tornando o processo de controle operacional mais simples, minimizando problemas quanto sua operação em um projeto. Para o desenvolvimento do projeto proposto foi escolhido um microcontrolador da família PIC, pelo fato das suas características serem mais favoráveis ao desenvolvimento do mesmo, conforme a Tabela 2 a seguir

**Tabela 2 – Principais fabricantes de microcontroladores** (continua)

Fabricante	Internet	Barramento	Família	Arquitetura	Núcleo	IDE
<b>Analog Device</b>	www.analog.com	8-bits	ADUC8xx	CISC	8051	-
		32-bits	ADUC7xx	RISC	ARM7	-
<b>Atmel</b>	www.atmel.com	8-bits	AT89xx	CISC	8051	Independente programs
		8-bits	AVR	RISC	-	AVR studio
		32-bits	AVR32	RISC	-	-
		32-bits	AT91xxx	RISC	ARM7/9	-
<b>Cirrus Logic</b>	www.cirrus.com	32-bits	EP73xxx	RISC	ARM7	-
		32-bits	EP93xxx	RISC	ARM9	-
<b>Cygnal</b>	www.silabs.com	8-bits	C8051F	CISC	8051	-
<b>Freescale (Motorola)</b>	www.freescale.com	8-bits	HC05	CISC	6800	-
		8-bits	HC08	CISC	6809	Code Warrior
		8-bits	HC11	CISC	6809	-
		16-bits	HC12	CISC	-	-
		16-bits	HC12	CISC	-	Code Warrior
		16-bits	HC16	CISC	-	-
		16-bits	56800	CISC	-	-
		32-bits	68K	CISC	68000	-
		32-bits	ColdFire	CISC	-	-
<b>Fujitsu</b>	www.fujitson.com	16-bits	F2MC	CISC	-	-
		32-bits	FR	RISC	-	-
<b>Infineon</b>	www.infineon.com	8-bits	C5xxx, C8xx	CISC	8051	-
		16-bits	C16xxx	CISC	-	-
		16-bits	XC16xxx	CISC	-	-
		32-bits	TCxxx	CISC	-	-
<b>Intel</b>	www.intel.com	8-bits	MC5251	CISC	8051	-
		16-bits	MC596/296	CISC	-	-
<b>Maxim (Dallas)</b>	www.maxim-ic.com	8-bits	DS80Cxxx	CISC	8051	-
		8-bits	DS83Cxxx	CISC	8051	-
		8-bits	DS89Cxx	CISC	8051	-
		16-bits	MAXQ	RISC	-	-
<b>Microchip</b>	www.microchip.com	8-bits	PIC-10, 12, 16, 18	RISC	-	MPLAB
		16-bits	dsPIC	RISC	-	MPLAB
<b>NS</b>	www.national.com	8-bits	COP8xxx	CISC	-	Webench
		16-bits	CR16Cxxx	CISC	-	-
		16-bits	CP3000	RISC	-	-
<b>Philips</b>	www.semiconductors.philips.com	8-bits	P8xxx	CISC	8051	-
		16-bits	Xaxxx	CISC	-	-
		32-bits	LPC2000	RISC	ARM7	-

(conclusão)

Fabricante	Internet	Barramento	Família	Arquitetura	Núcleo	IDE
<b>Rabbit Semiconductor</b>	www.rabbitsemi-conductor.com	8-bits	Rabbit2000	CISC	-	-
		8-bits	Rabbit3000	CISC	-	-
<b>Renesas</b>	www.renesas.com	8-bits	740	CISC	-	-
		16-bits	H8	CISC	-	HEW
		16-bits	H8S	CISC	-	HEW
		16-bits	M16C	CISC	-	-
		16-bits	7700	CISC	-	-
		32-bits	HBSX	CISC	-	-
<b>ST</b>	www.stm.com	8-bits	ST5	CISC	-	Visual FIVE
		8-bits	ST6	CISC	-	-
		8-bits	ST7	CISC	-	STVD-7
		8-bits	ST9	CISC	-	STVD-9
		16-bits	ST9	CISC	-	STVD-9
		16-bits	ST10	CISC	-	-
		32-bits	ARM7	RISC	ARM7	-
<b>Texas Instrument</b>	www.ti.com	8-bits	MSC12xx	CISC	8051	-
		16-bits	MSP430	CISC	-	Eclipse
		32-bits	TMS470	RISC	ARM7	-
<b>Toshiba</b>	Chips.toshiba.com	8-bits	870	CISC	-	-
		16-bits	900/900H	CISC	-	-
		32-bits	900/900H	CISC	-	-
<b>Ubicom(ex. Scenix)</b>	www.ubicom.com	8-bits	SXxx	RISC	-	-
<b>Zilog</b>	www.zilog.com	8-bits	Z8xxx	CISC	Z80	-
		8-bits	Z8Encore	CISC	Z80	-
		8-bits	eZ80Acloim	CISC	Z80	-

Fonte: MANTOVANI e OKI, 2013

## 2.4 Metas de usabilidade

Segundo Preece (2005), a usabilidade é constantemente considerada um fator que assegura que os produtos serão fáceis de usar, eficientes e agradáveis – da visão do usuário. A usabilidade é dividida em seis metas básicas que são:

- Ser eficaz no uso (eficácia);
- ser eficiente no uso (eficiência);
- ser seguro no uso (segurança);
- ser de boa utilidade (utilidade);

- ser fácil de aprender (*learnability*);
- ser fácil de lembrar como se usa (*memorability*).

Através de protótipo, é possível apresentar sistemas desenvolvidos mostrando a usabilidade do mesmo frente as necessidade dos clientes. De acordo com Preece (2005), protótipo é uma representação limitada de um *design* que permite aos usuários interagirem com ele e explorar a sua conveniência. A construção de um protótipo facilita a comunicação entre os membros da equipe sobre suas ideias e também é uma maneira eficaz de testá-las.

Segundo Preece (2005) a prototipação é dividida em duas: prototipação de baixa-fidelidade e prototipação de alta-fidelidade.

Os protótipos de baixa-fidelidade são os que não se assemelham muito ao produto final. Esses são úteis, pois são simples, baratos e de rápida produção. Podendo ser rapidamente modificados.

Os protótipos de alta fidelidade se assemelham muito ao produto final. Marc Rettig (apud Preece, 2005, p.265), diz que os projetos deveriam utilizar menos as prototipações de alta – fidelidade, devido aos problemas inerentes na mesma, sugerindo prioridade às prototipações de baixa fidelidade. A Tabela 3 mostra as vantagens e desvantagens das prototipações de Baixa e Alta fidelidade.

**Tabela 3 - Vantagens e desvantagens das prototipações**

(continua)

Tipo	Vantagens	Desvantagens
<b>Protótipo de Baixa-fidelidade</b>	<ul style="list-style-type: none"> <li>• Baixo custo de desenvolvimento.</li> <li>• Avalia vários conceitos de <i>design</i>.</li> <li>• Instrumento de comunicação útil.</li> <li>• Aborda questões de leiaute de tela.</li> <li>• Útil na identificação de mercado.</li> <li>• <i>Proof-of-concept</i> (Demonstração de que o conceito funciona).</li> </ul>	<ul style="list-style-type: none"> <li>• Verificação de erros limitada.</li> <li>• Especificação com poucos detalhes para codificação.</li> <li>• “Uso” conduzido pelo facilitador.</li> <li>• Utilidade limitada após estabelecimento dos requisitos</li> <li>• Utilidade limitada para realização de testes de usabilidade.</li> <li>• Limitações de fluxo e navegação.</li> </ul>

(conclusão)

Tipo	Vantagens	Desvantagens
<b>Protótipo de Alta-fidelidade</b>	<ul style="list-style-type: none"> <li>• Funcionalidade completa.</li> <li>• Completamente interativo.</li> <li>• Uso conduzido pelo usuário.</li> <li>• Define com clareza o esquema de navegação.</li> <li>• Uso para exploração e testes.</li> <li>• Mesmo <i>look and feel</i> do produtor final.</li> <li>• Mesmo <i>look and feel</i> do produtor final.</li> <li>• Serve como especificação viva.</li> </ul> Ferramenta de <i>marketing</i> e venda.	<ul style="list-style-type: none"> <li>• Custo alto no para o desenvolvimento.</li> <li>• Muito tempo para o desenvolvimento.</li> <li>• Ineficiente para <i>designers proof-of-concept</i> (demonstrações de que o conceito funciona).</li> <li>• Não funciona para coleta de requisitos.</li> </ul>

Fonte: RUDD et al., 1996.

Sistemas que são desenvolvidos a partir do uso integrado de equipamentos controlados por sistemas computacionais utilizam tensão elétrica quanto ao funcionamento e controle de equipamentos.

A tensão elétrica, segundo Afonso e Filoni (2011), é um fenômeno que ocorre quando uma carga está imersa em um campo elétrico. Ela fica sujeita a uma força e pode vir a se movimentar. Em outras palavras, essa carga adquire uma energia potencial elétrica, que talvez se transforme em energia cinética, ou seja, ela poderá realizar trabalho. Quanto maior a carga, maior a força e também maior a energia potencial. A tensão elétrica é de fundamental importância no desenvolvimento de atividades de sistemas que utilizam de equipamentos para realização de determinado tipo de trabalho, ou seja, realização de uma atividade. “Em circuitos elétricos, a diferença de potencial é imposta por geradores ou baterias.” (AFONSO; FILONI, 2011, p. 32).

Segundo Santiago (2012), o fluxo de elétrons em um circuito pode ser coordenado por corrente contínua e alternada. Na corrente contínua, considerando-se o sentido convencional, a carga sempre sai do polo positivo até o polo negativo do gerador. Esse tipo de corrente é geralmente produzido por pilhas, baterias de carro, de celular, entre outras. Já na corrente alternada a carga segue pelo polo positivo até o negativo, as cargas oscilam invertendo seu sentido de movimento, fazendo com que os polos alternem entre positivo e negativo.

Segundo Afonso e Filoni (2011, p. 37), “A potência elétrica indica quanto trabalho e/ou energia é realizado em um intervalo de tempo.”. Sistemas que trabalham com comunicação entre equipamentos, operam por meio de comunicação por potência elétrica. Um exemplo comum do uso de potência elétrica é a comunicação do usuário por meio do controle remoto.

Gomes e Parede (2011, p. 111) cita que “A oposição que um elemento de circuito apresenta à passagem de corrente elétrica é uma propriedade da física à qual chamamos de resistência”. Ainda de acordo com Gomes e Parede (2011), a unidade que representação da resistência elétrica é denominada ohm, volt/ampère, e representada pela letra ômega  $\Omega$ .

Para poder filtrar a tensão correta para o funcionamento do sistema desenvolvido quanto sua integração em um circuito elétrico se faz necessário o uso de capacitores. Os capacitores segundo Gomes e Parede (2011) são formados por duas finas chapas de metal separadas por um isolante dielétrico. Esse isolante não permite que uma carga passe de uma placa para outra. Portanto eles têm a capacidade de acumular cargas elétricas. Devido a essas características capacitores tem muitas aplicações no mundo eletrônico. Além dos capacitores, existe a necessidade de se usar outros componentes que são indispensáveis num circuito elétrico mais complexo, tais como indutores, diodos, resistores, geradores, interruptores que serão descritos a seguir.

Para Gomes e Parede (2011), a bobina ou indutor consiste, simplesmente, em um fio enrolado. A funcionalidade de uma bobina, além de gerar um campo magnético, é funcionar como filtro de frequência, uma vez que evita a passagem de sinais de alta frequência e permite a passagem de baixa frequência.

O diodo, de acordo com Gomes e Parede (2011), são elementos que deixam a corrente seguir apenas em uma direção. Os diodos emissores de luz – *LEDs* - são aqueles que apresentam a existência de passagem de corrente, assim identificando se essa esta sendo fluente quanto à necessidade de operação do sistema projetado.

Ainda de acordo com Gomes e Parede (2011) um resistor é o que o próprio nome diz, proporciona resistência à passagem de corrente no circuito. A corrente que entra em um terminal do resistor será a mesma a sair no outro terminal do resistor, porem com uma queda de tensão.

Gomes e Parede (2011) definem que a principal função do gerador é transformar um tipo de energia em outro tipo de energia podendo ser química, física ou mecânica. Através da convenção, a carga positiva sai de um polo positivo e atravessa o circuito até chegar ao polo negativo. A força que empurra essa carga é chamada de força eletromotriz (fem).

De acordo com Gomes e Parede (2011), para se ligar ou desligar um circuito elétrico é necessário um botão para que possa interromper ou permitir a passagem da corrente. Esse botão é conhecido como interruptor ou circuito elétrico de chave.

Segundo Gomes e Parede (2011), um circuito com mau funcionamento implica em perigo como o superaquecimento por excesso de carga, ou até mesmo choque por alta voltagem. Para o controle da segurança de carga em circuito faz necessário o uso de fusível ou disjuntores.

Além dos componentes importantes para o funcionamento do circuito elétrico, existem dispositivos que se fazem essenciais para que haja melhor interação do usuário com o sistema. Os principais são a utilização de *display* e teclado para visualização e inserção dos dados no sistema.

Para Melo (2011), o Display de Cristal Líquido – LCD é de grande importância em um projeto no qual agregue o desenvolvimento de uma IHM (Interface Homem Máquina). Outro ponto importante é que o mesmo oferece interação com um microcontrolador. De acordo com Barbacena e Fleury (1996), os módulos de *LCD* são projetados para de se conectarem com a maioria dos *CPU's*, basta que estes atendam a temporização de leitura e escrita de instruções e dados, fornecidos pelo fabricante do módulo. Segundo Nunes (2013), um teclado matricial é composto por linhas e colunas, onde caso um botão seja pressionado permita a conexão do mesmo. Com isso é possível definir a posição do botão. O teclado matricial é interpretado pelo microcontrolador da seguinte forma, a linha ou coluna são lidas constantemente para verificar caso tenha alguma mudança de estado. Ainda de acordo com Nunes (2013, p. 39), “a mudança de estado pode ser de um degrau positivo (0 para 1), ou um degrau negativo (1 para 0 ) dependendo da logica escolhida.”

### 3 METODOLOGIA

Segundo Lakatos e Marconi (2010, p.220), a metodologia de um trabalho, consiste em uma estruturação lógica expondo a pesquisa desenvolvida, oferecendo respostas para futuras pesquisas de âmbito mais aprofundado, ou seja, dando suporte para pesquisas contínuas da área do trabalho desenvolvido.

O trabalho proposto teve como ambiente de estudo as estufas do IFMG - SJE. Esse estudo consistiu em obter informações necessárias para elaboração, desenvolvimento e execução do projeto.

O desenvolvimento inicial implicou-se em analisar sistemas que realizassem parte dos quesitos propostos, como os sistemas que realizam controle de umidade. Com a presença de diversas tecnologias que trabalham com o âmbito de controle de ambiente fechado, foram analisadas tecnologias que pudessem ser acopladas ao projeto. Vários equipamentos de porte nacionais e internacionais foram estudados, com finalidade de fazer com que múltiplos equipamentos trabalhassem de forma concatenada, tornando o processo de controle de umidade e temperatura em ambientes fechados fatores controláveis por um sistema computacional de forma simplificada. A seguir são apresentadas as etapas de elaboração do projeto, bem como levantamento do requisitos do sistema e estruturação lógica e física.

#### 3.1 Levantamento dos requisitos

Após a elaboração da ideia e sintetização do problema, foi possível definir quais funções melhor se adequariam ao sistema. Essas funções são provenientes das análises dos projetos e das sugestões de alguns profissionais da área.

O comportamento do sistema, com relação a interação com o usuário e atuação no controle do ambiente fechados, foi baseado nos conceitos de usabilidade. Todo o processo de elaboração do SCUT exigiu uma análise criteriosa para o desenvolvimento de soluções que o tornar-se mais amigável possível. Portanto foi necessário, que na elaboração dos requisitos incluíssem diversos recursos de interação do usuário com o sistema.

Abaixo são apresentados os requisitos funcionais - RF e não funcionais do sistema - RNF.

- Leitura das informações captadas pelos sensores - RF;

- Apresentar as informações relativas aos sensores, de forma simples e clara - RF;
- Possibilitar a alteração dos parâmetros mínimos e máximos de cada sensor - RF;
- Acionar equipamentos para controle do clima de acordo com os parâmetros definidos pelo usuário - RF;
- Emitir sinal sonoro e luminoso como *feedback* para o usuário - RF;
- Ser seguro, não colocando em risco a vida do usuário - RNF;
- Possuir uma velocidade satisfatória para seu uso – RNF; e
- Realizar suas atividades de forma estável, evitando o travamento - RNF.

### 3.2 Estrutura lógica do SCUT

Após a definição dos requisitos, foi iniciado o processo criação do código do SCUT. Para o desenvolvimento desses conjuntos de instruções, foram utilizadas algumas ferramentas para sua elaboração e posterior gravação no microcontrolador.

A escolha da ferramenta para desenvolvimento foi feita através da sugestão da empresa fabricante do microcontrolador (*Microchip*). A *Microchip* possui um Ambiente de Desenvolvimento Integrado – *IDE*, criado especialmente para programação dos seus microcontroladores.

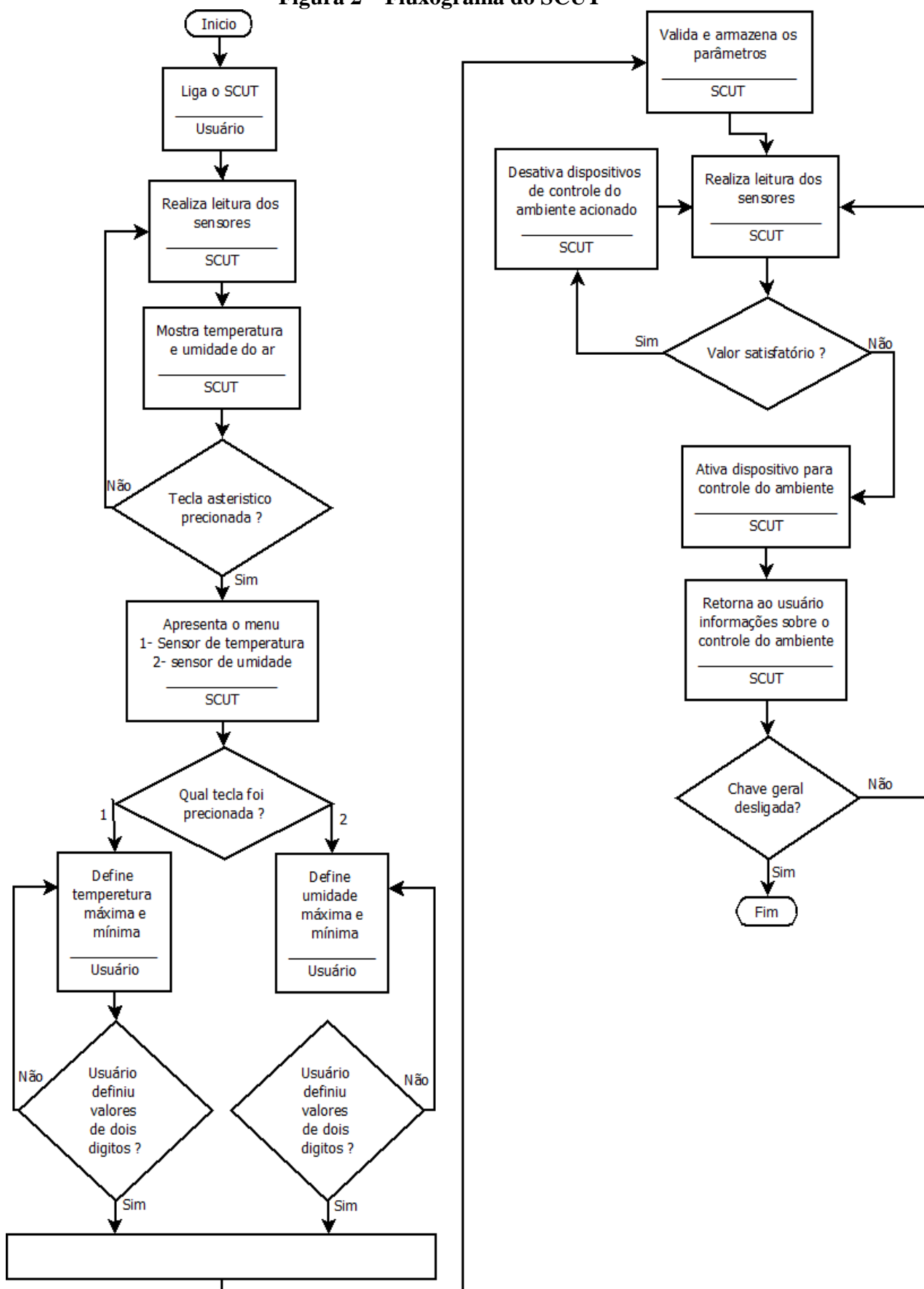
Foi utilizada a mesma estratégia para escolha do software de gravação, optando pelo *MPLAB IDE v1.40*, que é sugerido no manual do hardware gravador (*McFlash+*). Essa versão possui a ferramenta de gravação *PICSTART Plus*, que é compatível com gravadores que fazem uso de porta serial. Essa ferramenta não está presente nas versões atuais da *IDE*.

A *IDE* escolhida para o desenvolvimento foi o *MPLAB X IDE v1.85* da empresa *Microchip*. Essa versão possui uma interface amigável, suporta a criação de projetos na linguagem C e é gratuita. Esses são os principais motivos de sua escolha. O tópico a seguir descreve o funcionamento do sistema, expondo fluxograma e caso de uso.

#### 3.2.1 Fluxograma do SCUT

De acordo com Scartezini (2009), o fluxograma é uma ferramenta usada para descrever fluxos de trabalhos. São diagramas que descrevem a forma com que os processos acontece. A Figura 1 descreve o fluxograma do projeto.

Figura 2 – Fluxograma do SCUT

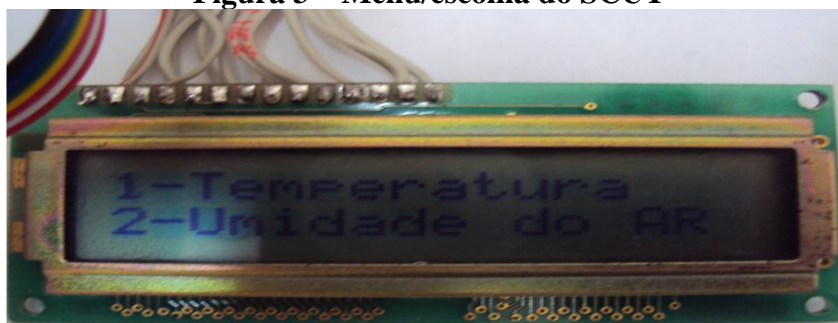


Fonte: Elaborada pelos autores, 2013

A ordem com que as atividades são realizadas é de fácil entendimento e aprendizado. O sistema inicia suas atividades a partir do momento em que se encontra devidamente energizado e após o usuário mudar o estado de uma chave para acionamento, localizada na lateral do hardware. Posteriormente são mostrados no display os dados coletados pelos sensores, porém para que o sistema funcione efetivamente é necessário que o usuário entre no modo de edição dos dados, para delimitar valores de mínima e máxima, tanto para a umidade relativa do ar, quanto para a temperatura. Ao iniciar o sistema os parâmetros de mínima e máxima tem valor padrão, respectivamente 0 e 100, tanto para a umidade quanto para temperatura.

Para acionar o modo de edição o usuário deve pressionar a tela asterisco – \*, do módulo teclado que quando acionada, interrompe a leitura de seus sensores, emite um sinal sonoro e entra no menu de edição. Nesse menu o usuário deve selecionar qual sensor deseja alterar os parâmetros de controle, como mostra a Figura 3.

**Figura 3 – Menu/escolha do SCUT**



Fonte: Próprio do sistema, 2013

Para realizar a escolha do sensor o usuário deve utilizar o teclado para pressionar a opção desejada, sendo que a tecla 1 representa o sensor de temperatura e a 2, o sensor de umidade.

Após realizar a seleção do sensor, será apresentado no *display* uma mensagem informativa dizendo ao usuário qual parâmetro digitar. Os valores de entrada são limitados à duas casa, para redução de erros por parte do usuário, permitindo que a confirmação do valor digitado seja feita apenas quando esse for representado por duas casas decimais. Assim que digitar esse valor o usuário pode confirmar os dados pressionando a tecla # do módulo de teclado ou pressionar \* para reinserir os dados. Após a confirmação, o sistema coleta os dados apresentados no *display* e inicia a comparação e controle desses parâmetros. Quando um valor é excedido, o sistema aciona um componente para realização do controle relativo ao

parâmetro. Cada componente é desativado quando esse valor apresentar uma mudança de 3°C ou 3% favorável ao parâmetro definido pelo usuário.

Caso ocorra travamento do sistema, o usuário poderá pressionar o botão de *reset* que retoma as configurações padrões do dispositivo, possibilitando ao usuário redefinir os parâmetros e retomar as atividades normalmente.

### 3.2.2 *Desenvolvimento do código*

Com o funcionamento do sistema definido, foi iniciado o processo de desenvolvimento do código, que consistiu na elaboração de conjuntos de instruções que realizassem as atividades conforme planejado.

Para desenvolvimento dos códigos foi utilizado o *IDE MPLAB X v1.85*, trabalhando em conjunto com a linguagem de programação C, voltada para microcontroladores. O procedimento consistiu da criação da estrutura lógica, por meio de um conjunto de instruções e posterior compilação.

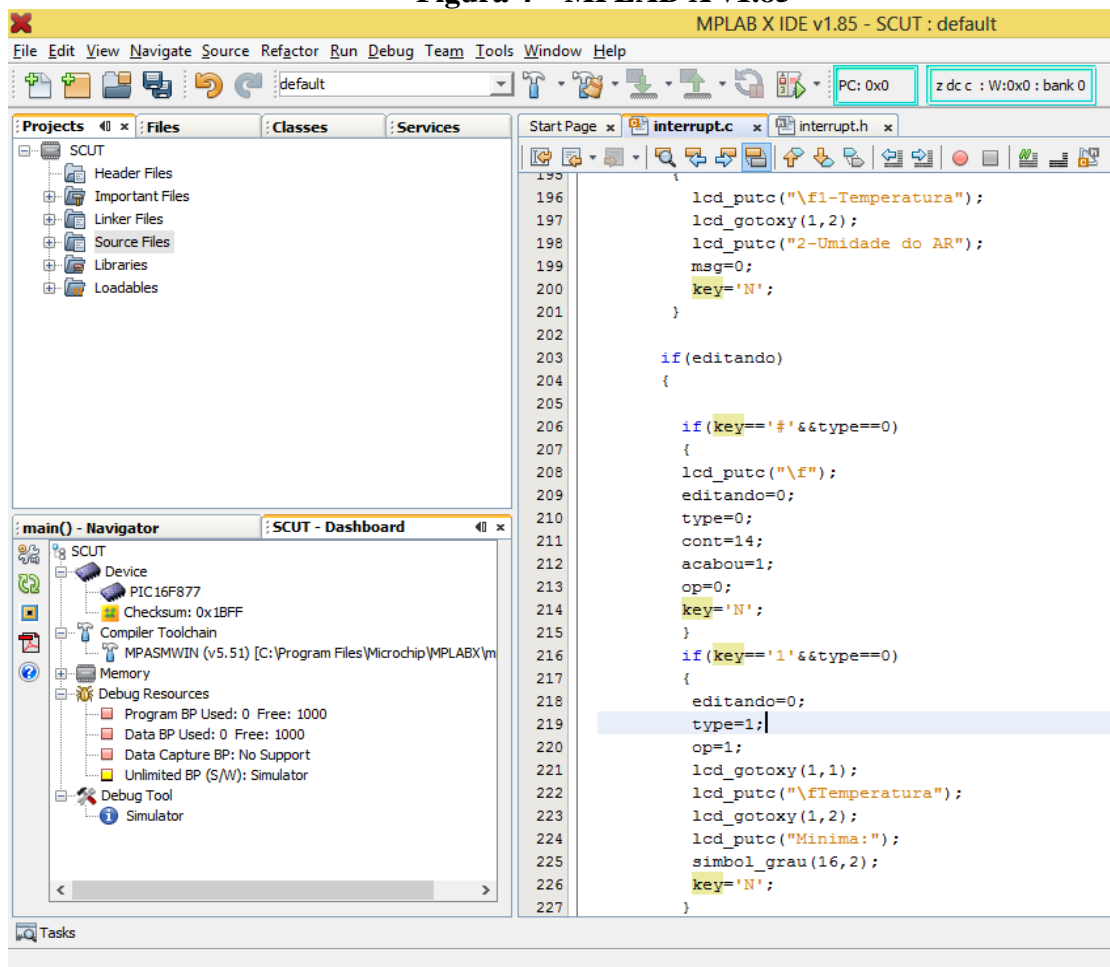
A manipulação de alguns dispositivos como *display* e teclado podem ser feitas através do uso de bibliotecas já desenvolvidas por terceiros. Essas bibliotecas contêm uma série de funções que provem o funcionamento dos dispositivos em questão, bastando apenas incluí-la no código e posteriormente utilizar as funções providas por ela.

No desenvolvimento do SCUT foram utilizadas duas bibliotecas para o provimento de funcionalidade ao *display* e teclado. Para manipulação do LCD é utilizada a biblioteca “*cuscustudio.c*” que possui uma grande variedade de funções para diversos dispositivos, porém das funções apenas as referentes a manipulação de LCD foram utilizadas. O trecho utilizado, da biblioteca, se encontra no ANEXO B. Para varredura do teclado, foi utilizada outra biblioteca, “*KEY.c*”, essa se encontra no ANEXO C. Essa biblioteca se demonstrou mais estável, por isso o motivo de sua utilização.

Além das bibliotecas para controle do *display* e teclado, foi necessário implementar todo o código relativo as funcionalidades do sistema. Esses conjuntos de instruções definem como, quando e em quais condições cada função do sistema será executada. O código do SCUT está presente no APÊNDICE A, “*SCUT.c*”. Contudo ainda foi necessário realizar a configuração do microcontrolador quanto aos seus bits de funcionalidade tais como frequência de operação, modelo do microcontrolador, entre outras. Essas estão presentes no APÊNDICE B, “*SCUT.h*”. Após desenvolver o código é necessário compilar, os conjuntos de instruções para que essas sejam inseridas do microcontrolador.

O processo de compilação resulta na criação dos códigos em linguagem de máquina do tipo hexadecimal que é utilizado na programação do microcontrolador. A Figura 4 apresenta a tela do *MPLAB X v1.85* sendo usado no desenvolvimento do código.

**Figura 4 – MPLAB X v1.85**



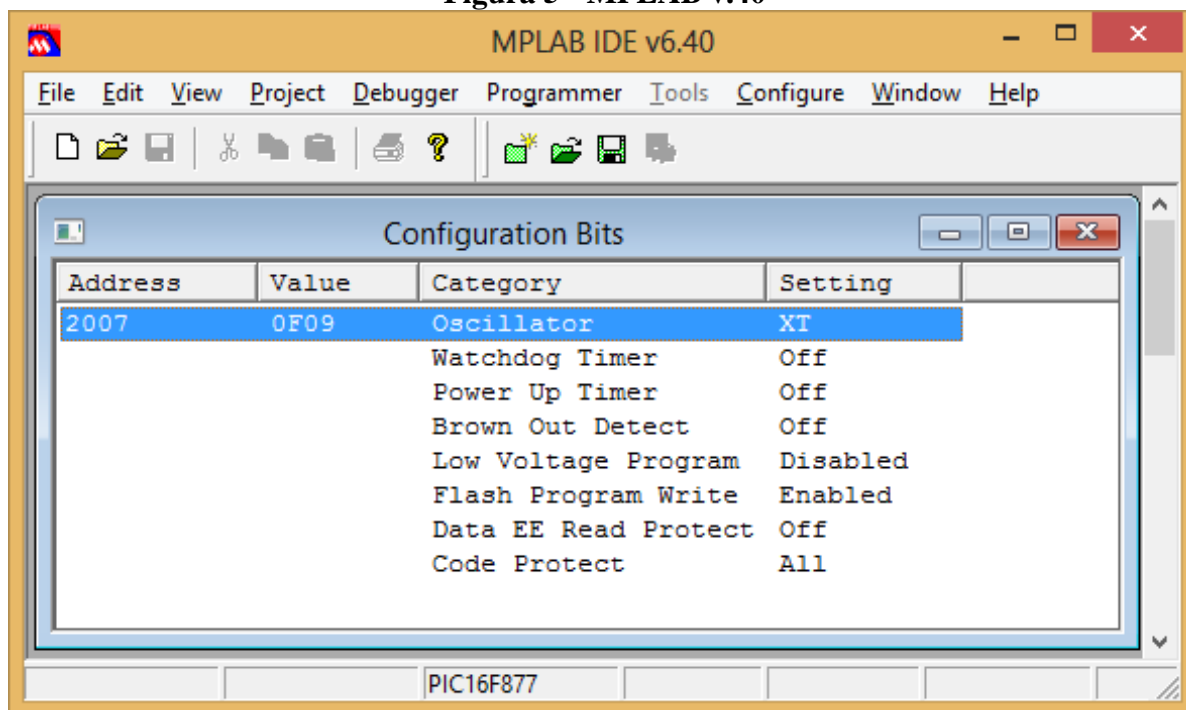
Fonte: Elaborada pelos autores, 2013

Para constatar o funcionamento dos códigos desenvolvidos tornou-se necessário realizar a gravação e teste em conjunto com o *hardware* do projeto.

Para a realização da programação do microcontrolador, foi utilizado o *hardware* gravador *McFlash+* em conjunto com a *IDE MPLAB v1.40*. A etapa de gravação consiste em realizar a importação do código hexadecimal, definir qual procedimento de gravação será utilizado, acoplar o microcontrolador ao *socket* do gravador, apagar os dados contidos dentro do microcontrolador e, por fim, solicitar que a gravação seja efetuada.

Na Figura 5, são demonstradas as definições de bits do microcontrolador, antes da gravação. Essas configurações são previamente definidas no código do sistema, no arquivo de extensão “.h”. Porém elas podem ser redefinidas no software de gravação.

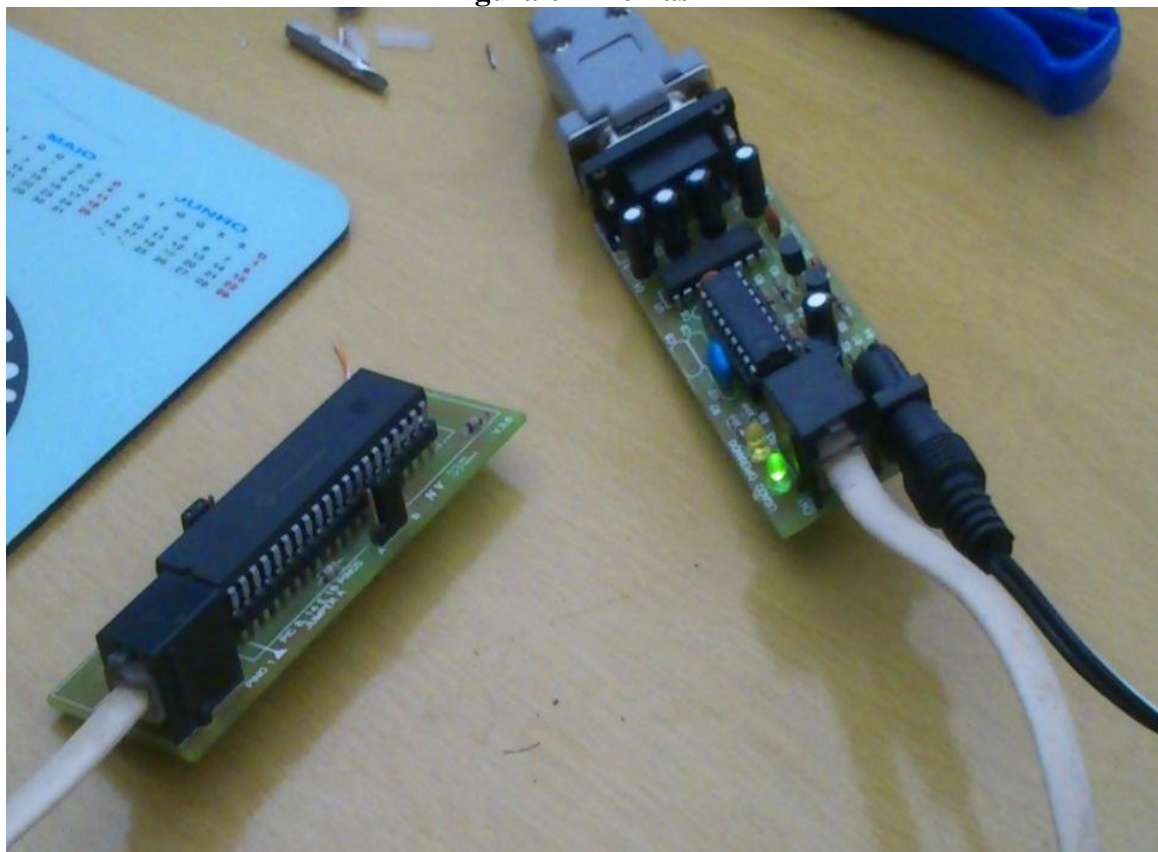
Figura 5 - MPLAB v.40



Fonte: Elaborada pelos autores, 2013

Na Figura 6 é possível visualizar o *hardware* usado para gravação, McFlash+.

Figura 6 - McFlash+



Fonte: Elaborada pelos autores, 2013

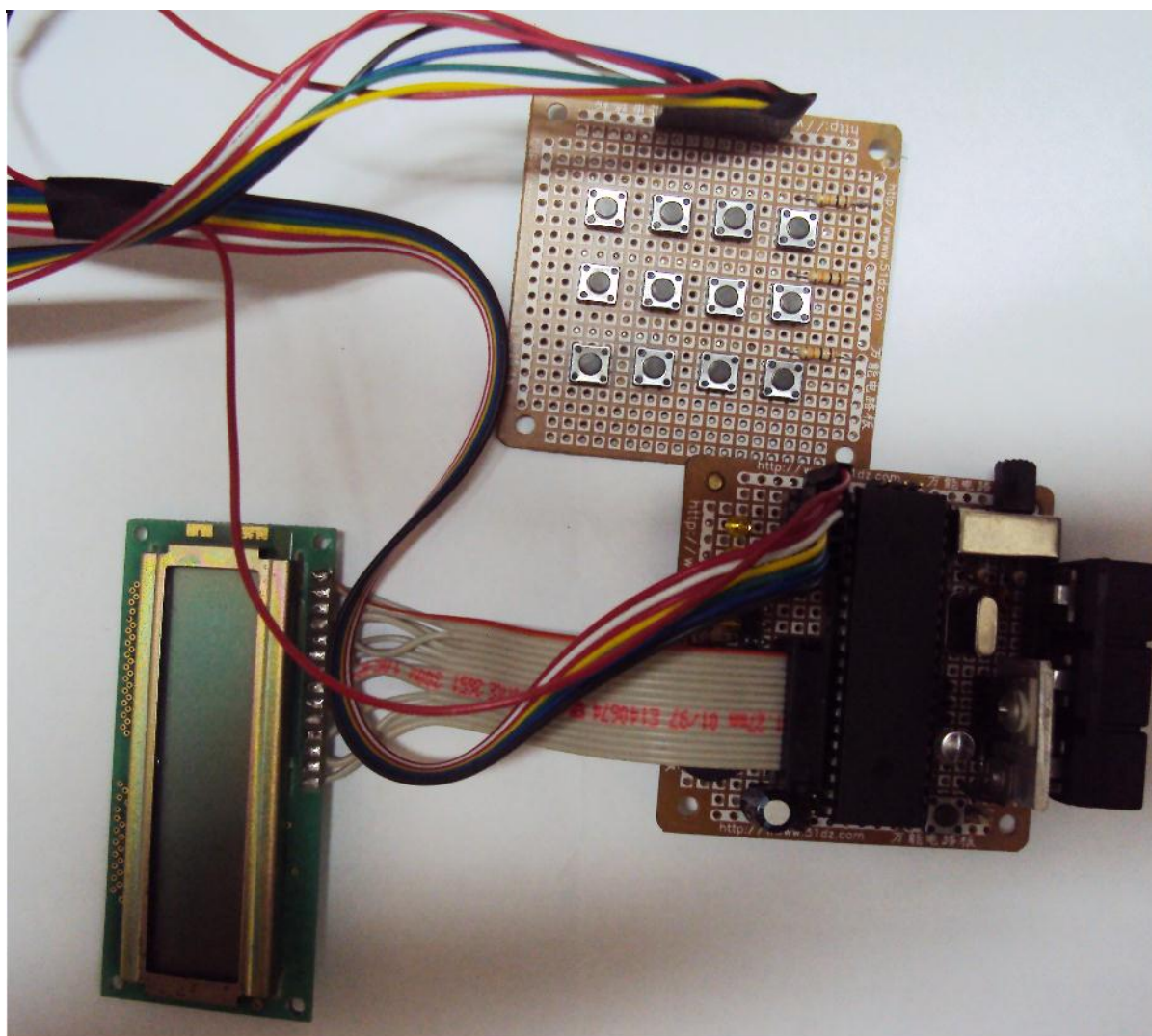
### 3.3 Estrutura física do projeto SCUT

Após a criação da parte lógica do sistema, foi necessário construir a parte física para que suas necessidades fossem atendidas. Essa consiste de um conjunto de peças e materiais que são responsáveis por agir, interligar e receber outros componentes.

Para sua concepção, foram criados protótipos com a finalidade de testar o seu funcionamento. Esses são destinados para a aplicação funcional do sistema, demonstrando o funcionamento da parte eletrônica e também como o ambiente é afetado pelas ações do SCUT. Para isso foram construídos os módulos de relé, teclado matricial e placa mãe. Esses são interligados por meio de cabos e solda, para que o funcionamento do sistema se complete.

Na Figura 7 é representada a ligação entre alguns módulos à placa mãe.

**Figura 7 - Módulos interligados**



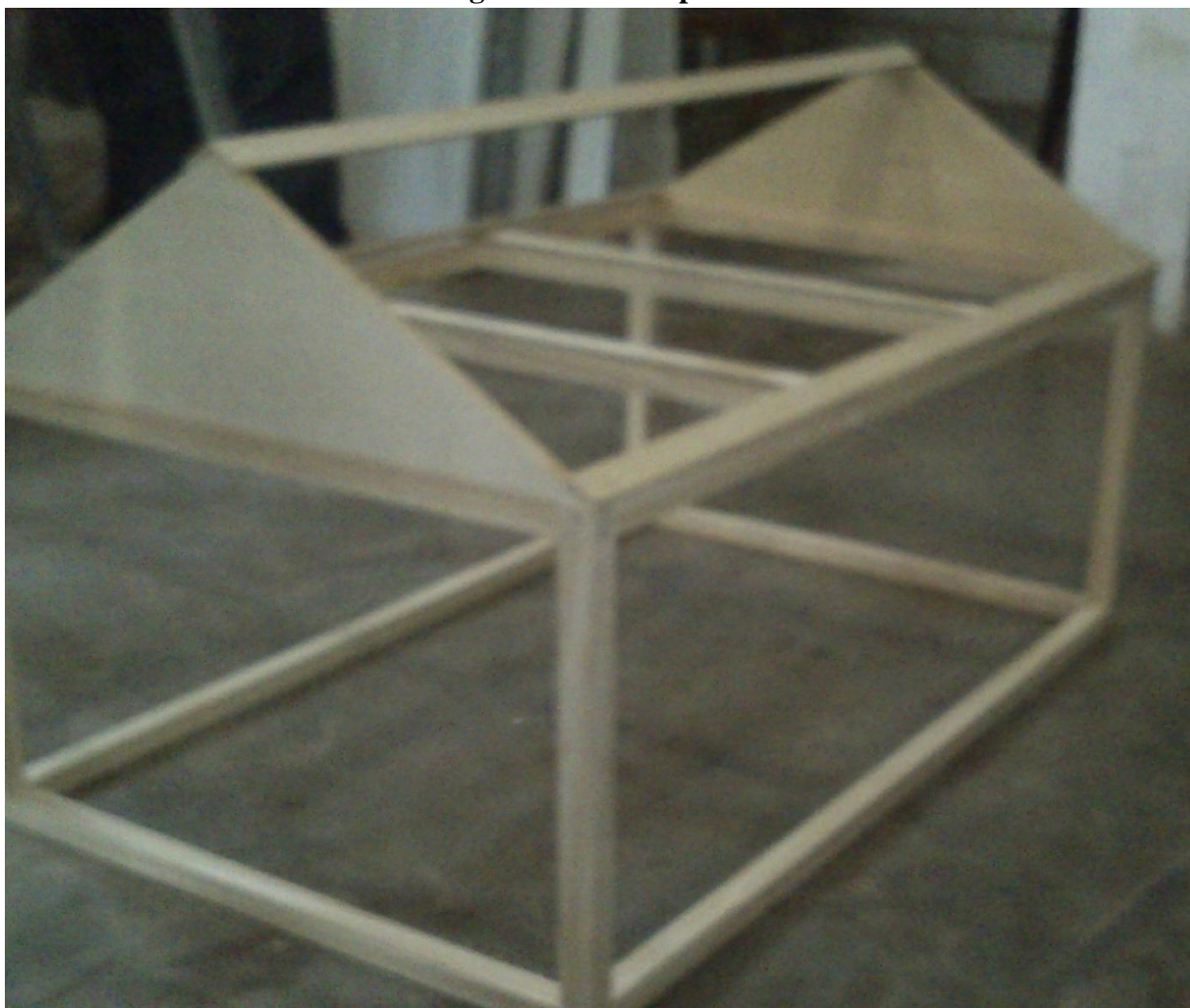
Fonte: Própria do SCUT, 2013

Para representação do ambiente controlado, foi criada uma mini estufa que é equipada com dispositivos para realizar a simulação do controle da temperatura e umidade da mesma. A seguir será detalhado o processo de desenvolvimento da mesma.

### ***3.3.1 Construção do protótipo da estufa***

O trabalho desenvolvido teve como ambiente de simulação um protótipo artesanal desenvolvido a partir de madeira, plástico transparente e tachas de metal para fixação. A estrutura principal foi feita de madeira para recepção do plástico de contenção. O plástico foi utilizado para revestir a estrutura, representando a lona de uma estufa. Para equipar o protótipo foram utilizados bomba de água com bico e reservatório, coolers de computador, fios para ligação dos mesmos e mangueiras para condução de água. Na Figura 8 é possível visualizar o ambiente desenvolvido para simulação.

**Figura 8 – Protótipo da estufa**



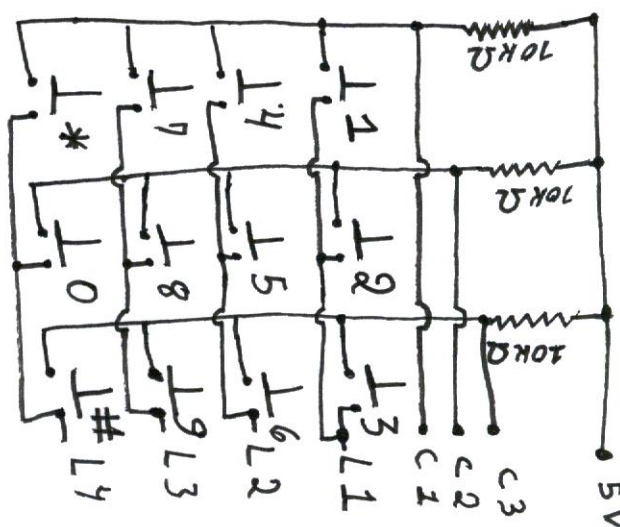
Fonte: Própria do SCUT, 2013

### 3.3.2 Montagem dos módulos

Após certificar, em testes, que o hardware da aplicação estava estável, foi iniciado o processo de criação dos módulos. Esse processo consistiu em transferir os componentes utilizados nos teste, para uma placa protoboard matriz. A seguir é detalhado cada módulo.

- O módulo de teclado é utilizado para interação do usuário com o sistema. Ele é composto de uma protoboard placa matriz 7 x 9cm, contendo 12 botões TS 1026, três resistores de 10 k e oito pinos machos protoboard. Todos os componentes foram interligado por meio de solda e fios. O teclado é uma matriz 4x3, quatro linhas e três colunas, no qual todos os botões de cada coluna são interligados e também cada linha de botões é interligada. Essa ligação é demonstrada na prototipação de baixa fidelidade usada no desenvolvimento do projeto – esquema elétrico. A Figura 9 representa o esquema elétrico do teclado.

Figura 9 – Esquema do módulo teclado



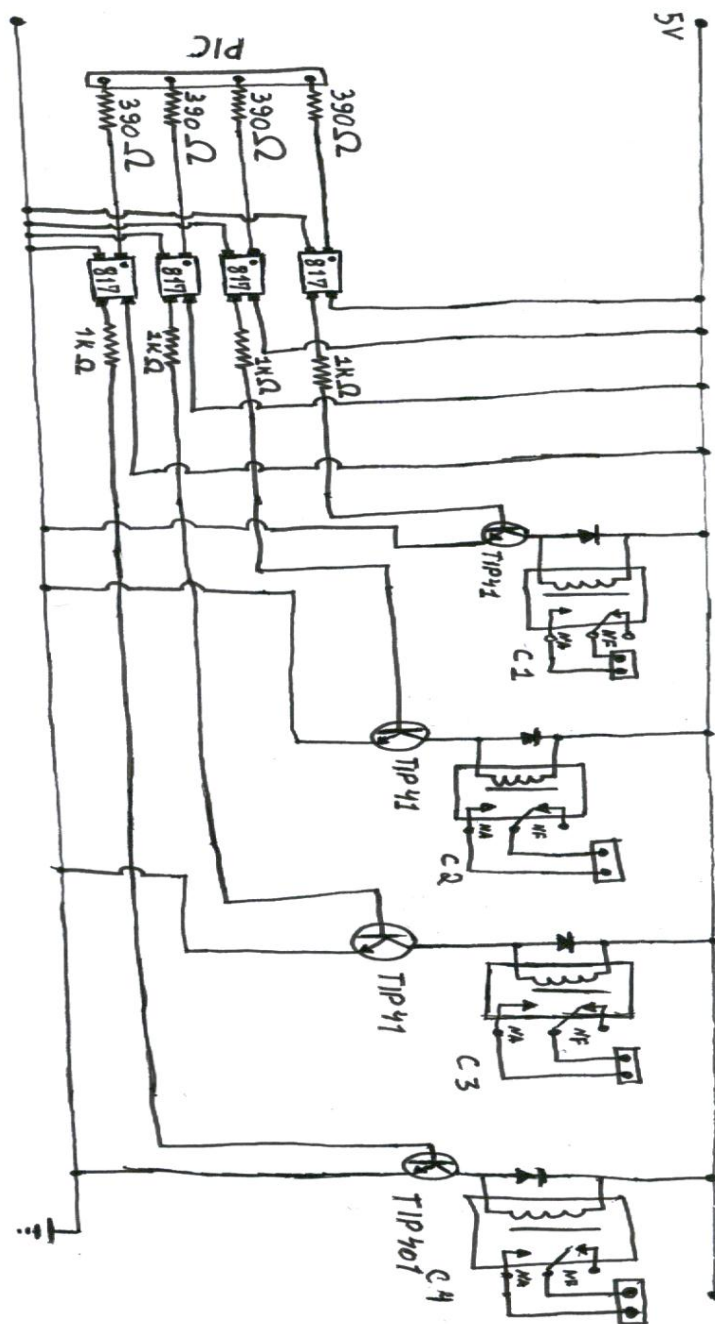
Fonte: Elaborada pelos autores, 2013

- O módulo de relé é responsável pelo acionamento dos equipamentos. Ele é composto por quatro resistores de 1k e dois resistores de 390 Ohms, um resistor de 330 Ohms, um resistor de 470 Ohms, quatro relés NRP07-C05D de 5v, quatro transistores de três pinos TIP 41, um transistor TIP 401, quatro pinos macho *protoboard* PIC e cinco borne BR2. Para compreender o funcionamento dos transistores pode-se consultar o ANEXO E. Esse módulo tem o objetivo de fazer a intermediação entre os dispositivos que fazem controle do ambiente fechado acionando e desativando os dispositivos sob

o comando do *PIC*. Todo o módulo de acionamento é baseado em cinco v. O *PIC* manda um pulso por meio de uma das portas, o mesmo chega ao optoacoplador modelo 817, permitindo a passagem de corrente pelo fototransistor, enviando um sinal a base do transistor TIP 41 e/ou TIP 401.

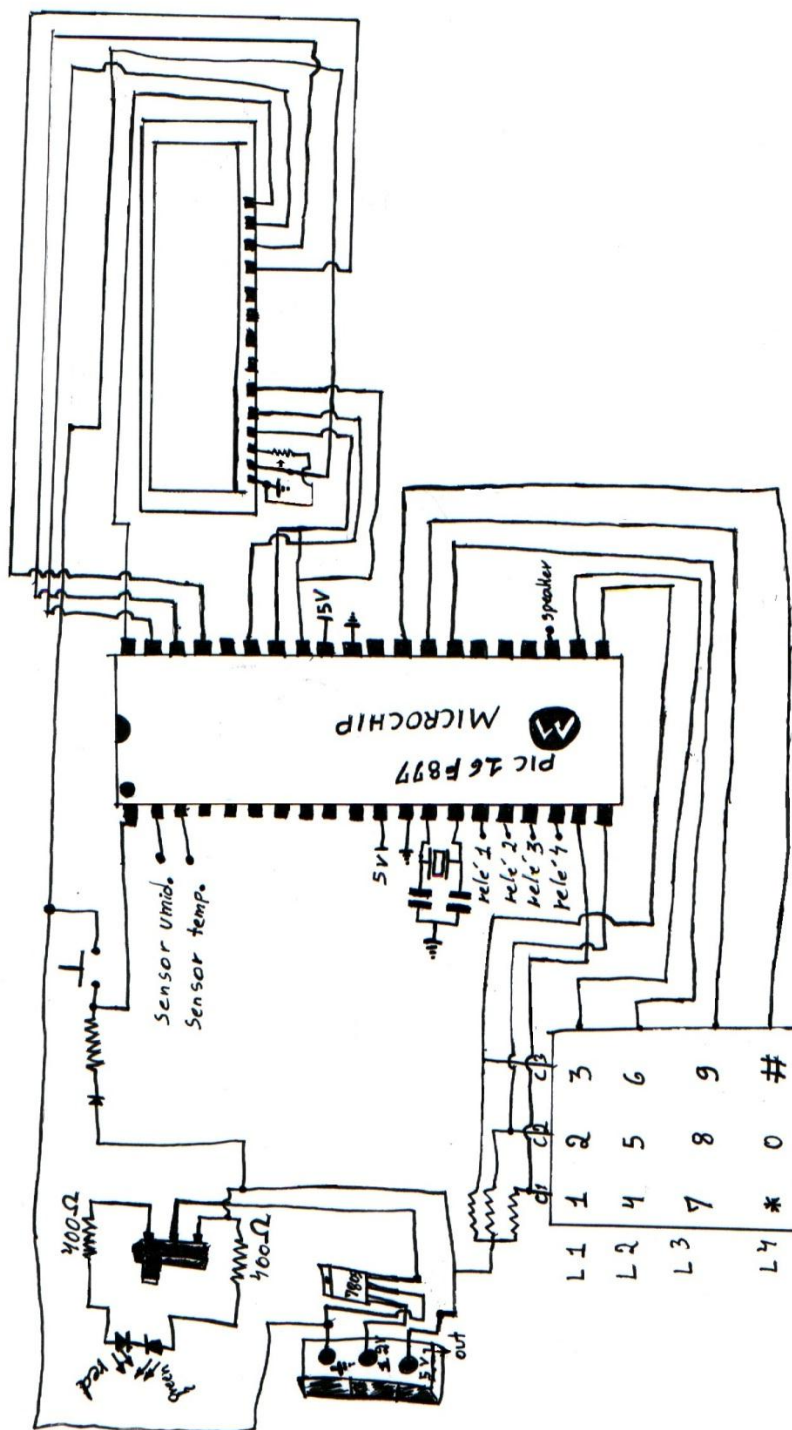
Para sua criação foi utilizado um esboço (prototipação de baixa fidelidade), que representa seu esquema elétrico. Esse esquema é apresentado na Figura 10.

Figura 10 – Esquema do módulo relé



- O módulo de placa mãe é responsável por realizar todas as operações. É a partir dela que as leituras e ações são realizadas quando necessário. Por isso todos os outros módulos devem estar conectados na mesma. Na Figura 11 é possível ver o esquema elétrico da placa mãe.

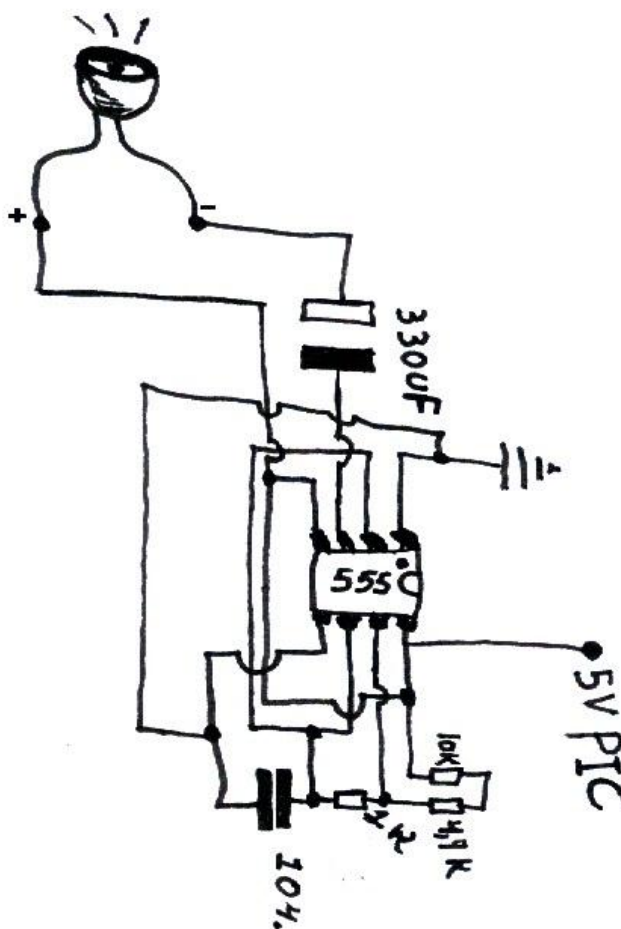
Figura 11 – Esquema do modulo placa mãe



Esse módulo é composto de diversos componentes eletrônicos, sendo eles para diversos fins. Nesse módulo foi utilizado um terminal estabilizador de tensão 7805, ele trabalha transformando 12v em 5v, mais informações presentes no ANEXO D . A partir do momento em que os cabos de energia são ligados ao dispositivo através do painel de conexão, a energia é estabilizada e satura um LED de cor vermelha se a chave geral do circuito estiver com o estado de desligado. Nessas condições, apenas o LED é energizado emitindo luz de cor vermelha. Quando a chave geral é colocada em estado de ligado, o microcontrolador, display, sensores e a porta de saída no painel, passam a ser energizados tornando o circuito operante.

Dentro do módulo da placa mãe foi colocado um pequeno circuito elétrico destinado a emitir sinais sonoros. Ele é constituído de um CI oscilador 555 que gera uma frequência para um *speaker*, que ao recebê-la emite som. Na Figura 12 é possível ver o esquema elétrico do circuito excitador de *speaker*.

**Figura 12 – Esquema do módulo teclado circuito excitador de speaker**

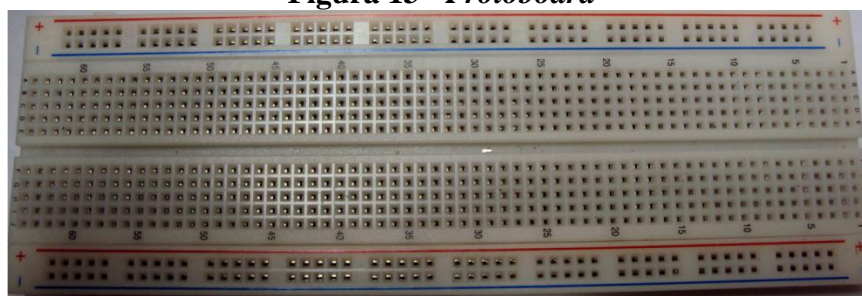


Além de saídas para display, entradas de teclado, canais de acionamento de relé, o módulo possui também quatro *LED*'s que indicam se algum relé foi acionado ou não, acendendo assim o *LED* referente ao pino de conexão do relé.

### 3.4 Integração e testes

Ao passo que a programação referente a determinadas ações específicas do sistema se desenvolvia, os circuitos elétricos iam sendo definidos em protótipos de baixa fidelidade. Depois de esboçado o funcionamento e a distribuição dos componentes e para a simulação de tal módulo, os testes e os protótipos de alta fidelidade eram criados a fim de observar o comportamento dos mesmos em situações muito próximas as reais. Para a realização dos testes de alta fidelidade foram utilizadas protoboards, essas têm por intuito prover um ambiente de fácil manipulação para realização de testes de circuitos elétricos.

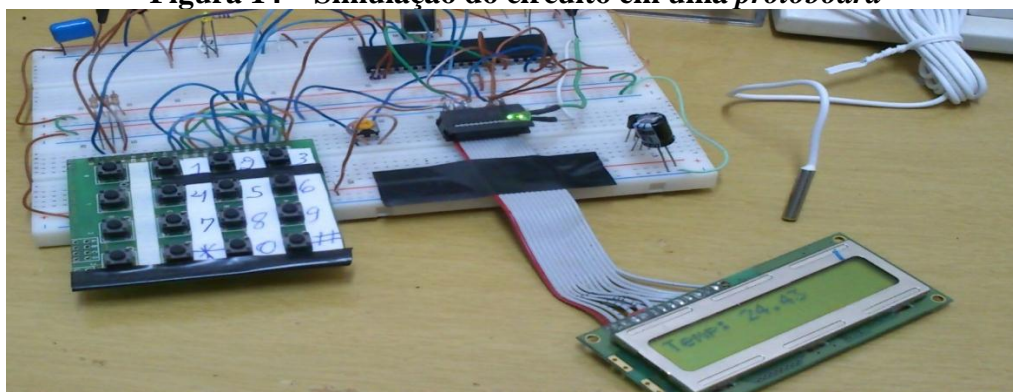
**Figura 13 - Protoboard**



Fonte: Elaborada pelos autores, 2013

Nessas *protoboards* foram testados os circuitos, o hardware do projeto, e também como se portava as instruções do sistema, software.

**Figura 14 – Simulação do circuito em uma protoboard**



Fonte: Elaborada pelos autores, 2013

Com os módulos funcionando corretamente, as instruções ia sendo aplicadas de modo que todos os módulos trabalhassem em conjunto. Feito os testes com o protótipo de alta fidelidade dos módulos e seus respectivos componentes em funcionamento, os mesmos foram transferidos para placas de circuitos e, por meio de solda, fixados a mesma. Após a constituição de cada módulo de forma independente, os mesmos foram integrados para o funcionamento em conjunto. Feita montagem dos módulos e componentes tornou-se necessário observar o comportamento do circuito e verificar os resultados obtidos.

## 4 RESULTADOS E DISCUSSÕES

O SCUT conseguiu cumprir com os requisitos propostos, realizando o controle do ambiente de forma satisfatória. Porém, durante o seu desenvolvimento, foi percebido algumas anomalias que interferiam no seu funcionamento. Essas foram posteriormente resolvidas e relatadas aqui.

No primeiro teste realizado com o SCUT, percebeu-se a existência de um ruído proveniente da fonte de energia do circuito no sistema. Pelo fato de utilizarmos sensores resistivos e de saída analógica, a tensão de funcionamento tem de ser mais precisa e constante. Porém foi constatado, através da leitura da saída do sensor, que ocorria variação de aproximadamente 10°C ou 100mV. Percebendo uma variação com tamanha amplitude, foi necessário encontrar uma forma de remover os ruídos e promover a estabilidade da tensão no circuito. O problema de variação da leitura dos sensores pode ser solucionado com a adição de um capacitor cerâmico ao circuito, que estabilizou a corrente oriunda da fonte, suprimindo as quedas de tensão e solucionando o problema. Com isso a leitura do ambiente pode ser realizada com maior precisão e os dados processados pelo sistema, reduzindo aproximadamente 95% da variação.

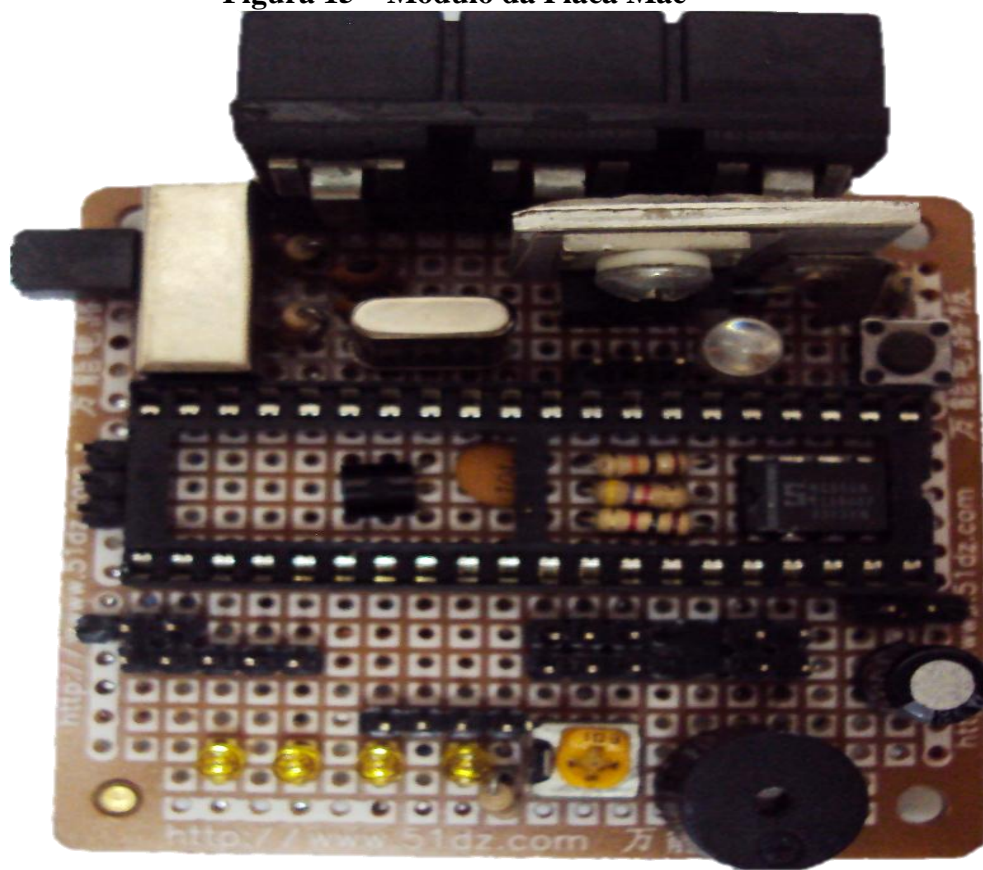
Após estabilizar os sensores, foi percebido que ao iniciar o sistema alguns dispositivos eram acionados antes da passagem dos parâmetros pelo usuário. Esse problema foi proveniente de uma atribuição nas variáveis que armazenam os valores máximos e mínimos de temperatura e umidade, onde algum(ns) se encontrava(m) contrapondo os valores do ambiente. A forma encontrada para solucionar o problema, consistiu em inicializar essas variáveis com valores improváveis para um ambiente comum. Dessa forma o sistema necessita da alteração dos parâmetros para realizar o acionamento dos dispositivos.

Contudo se o usuário necessitar de armazenamento de dados estatísticos esses não serão gerados a partir do projeto, pois o SCUT não armazena histórico, estatística do clima ou das atividades realizadas através do mesmo. Isso ocorre devido à falta de espaços de memória no *PIC* e devido a falta de tempo para a elaboração de um software que capturasse e realizasse, os cálculos necessários para levantamento dos dados estatísticos.

De forma geral, não ocorreram problemas que prejudicassem o funcionamento do sistema, resultando em módulos estáveis, de fácil locomoção e manuseio. Essas características proporcionam que o SCUT seja funcional e agradável ao uso.

Os módulos do sistema são apresentados nas figuras a seguir. A Figura 15 mostra o módulo da placa mãe pronto.

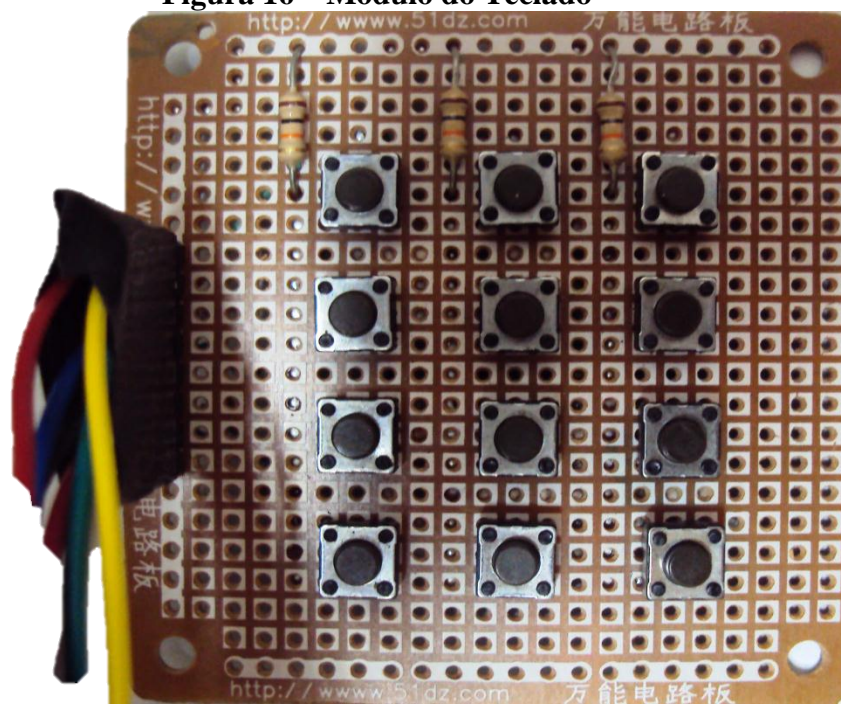
**Figura 15 – Módulo da Placa Mãe**



Fonte: Própria do SCUT, 2013.

Na Figura 16 é possível verificar como ficou o módulo do teclado.

**Figura 16 – Módulo do Teclado**



Fonte: Própria do SCUT, 2013.

Já a Figura 17 mostra o módulo de relé.

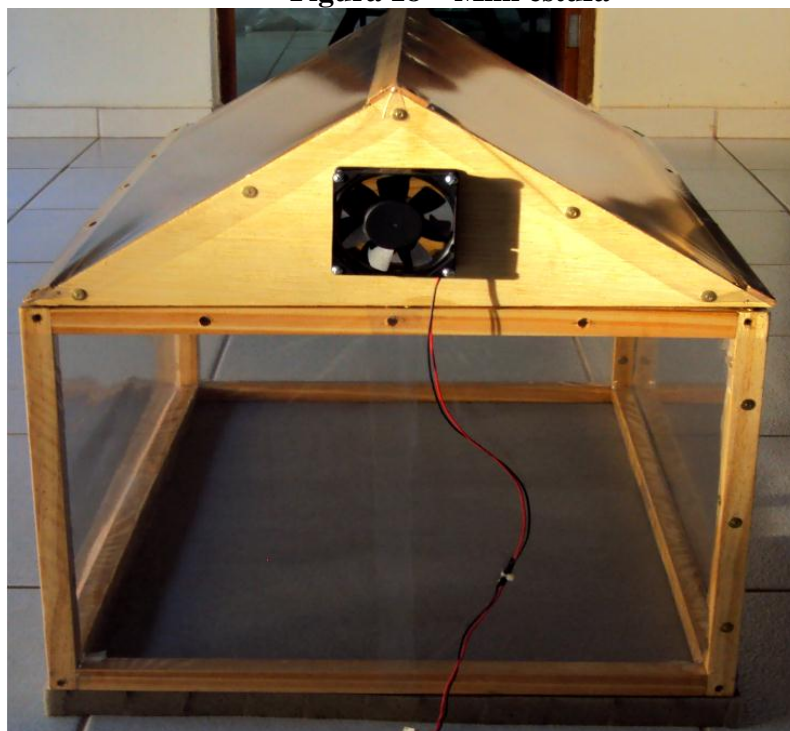
**Figura 17 - Módulo de relé**



Fonte: Própria do SCUT, 2013.

Na Figura 18 é possível verificar a mini estufa utilizada na simulação.

**Figura 18 – Mini estufa**



Fonte: Própria do SCUT, 2013.

## 5 CONCLUSÃO

Segundo Severino (2009, p.152), “para uma boa conclusão é necessário um domínio prático, sobre todos os passos realizados no trabalho bem como instrumentos, técnicas, entre outros.” Para Lakatos e Marcolni (2010, p.220), “uma conclusão consiste na argumentação das informações obtidas, bem como união das ideias, contendo assim o encerramento do trabalho.”

Todo o trabalho realizado para o desenvolvimento do SCUT obteve sucesso, pois ele atendeu as expectativas, realizando computacionalmente o controle do ambiente quanto à temperatura e umidade do ar. O protótipo utilizado para os testes do SCUT, demonstrou o quanto o sistema pode proporcionar de benefícios para situações em que é necessário um ambiente fechado controlado.

Diversas melhorias podem ser realizadas baseadas no projeto desenvolvido. Com o grande número de questões levantadas, sobre a importância de projetos não agressivos ao meio ambiente e sustentáveis, um sistema que realize o controle do ambiente fazendo uso de energias limpas e renováveis, traria não somente benefícios ecológicos, mas também financeiros.

Como um dos grandes fatores necessários para o controle de produção é a irrigação, um sistema capaz de realizar o controle de irrigação a partir da leitura da umidade do solo, seria de grande utilidade para a melhoria das produções agrárias.

O projeto não possibilita apenas uma base para criação de sistemas para controle agrário. Ele demonstra que é possível criar soluções para diversos problemas, relativos a controle de processos, através do uso de microcontroladores.

## REFERÊNCIAS

- ABREU, H. G.; SOFIA, A. **Sistema de Monitorização de Estufas Agrícolas**. 2012. Disponível em: <<https://ria.ua.pt/bitstream/10773/10269/1/Disserta%C3%A7%C3%A3o.pdf>>. Acesso em: 23 set. 2013.
- AFONSO, A. P.; FILONI, E. **Eletrônica Circuitos elétricos**. Editora Fundação Padre Anchieta. São Paulo. 2011.
- Agência Brasil, Abala Vitor. Notícia. **População brasileira cresce 0,9% entre 2012 e 2013**. 2013. Disponível em: <<http://agenciabrasil.ebc.com.br/noticia/2013-08-29/populacao-brasileira-cresce-09-entre-2012-e-2013>>. Acesso em: 12 mai.2013.
- Alldatasheet.com. **HIH-4000-001 Datasheet (PDF) - Honeywell Solid State Electronics Center - Integrated Circuitry Humidity Sensor**. Disponível em: <<http://pdf1.alldatasheet.com/datasheet-pdf/view/227310/HONEYWELL/HIH-4000-001.html>>. Acesso em: 09 out. 2013.
- BARBACENA, I. L.; FLEURY, C. A. **Display LCD**. 9.1996. Disponível em: <<ftp://ftp.dca.fee.unicamp.br/pub/docs/ea079/complementos/Lcd.pdf>>. Acesso em: 07 out. 2013.
- BORRÉ, R. L. **Sistema de Ventilação em Ambiente Industrial**. 2013. Disponível em: <<http://bibliodigital.unijui.edu.br:8080/xmlui/bitstream/handle/123456789/1656/Monografia%20Sistema%20de%20Ventila%C3%A7%C3%A3o%20em%20Ambiente%20Industrial.pdf?sequence=1>>. Acesso em: 10 out 2013
- CERON, T. A.; CRISPI, R. **Sensor de Temperatura LM35**. 2013. Disponível em: <<http://www.google.com.br/url?sa=t&rct=j&q=&esrc=s&source=web&cd=7&ved=0CE0QFjAG&url=http%3A%2F%2Fwww3.iesam-pa.edu.br%2Fojis%2Findex.php%2Fcomputacao%2Farticle%2FdownloadSuppFile%2F519%2F88&ei=NH5vUsC7M4r4kQe82oBQ&usq=AFQjCNFeWXAJTtqgxPhnnScOpmmwX5rNjw&sig2=sOBFo4MUwddpaeTXteL48g&bvm=bv.55123115,d.dmg>>. Acesso em: 10 out. 2013.
- Clima certo. Matérias. **Controle de Umidade Gera Conforto e Economia**. Disponível em: <[http://www.climacerto.com.br/Downloads/Controle-umidade\\_Artigo.pdf](http://www.climacerto.com.br/Downloads/Controle-umidade_Artigo.pdf)>. Acesso em: 23 set. 2013.
- DatasheetCatalog.com. **HD44100H**. Disponível em <[http://www.datasheetcatalog.com/datasheets\\_pdf/H/D/4/4/HD44100H.shtml](http://www.datasheetcatalog.com/datasheets_pdf/H/D/4/4/HD44100H.shtml)>. Acesso em: 10 out. 2013.
- DatasheetCatalog.com. **LM35 datasheet**. Disponível em: <[http://www.datasheetcatalog.com/datasheets\\_pdf/L/M/3/5/LM35.shtml](http://www.datasheetcatalog.com/datasheets_pdf/L/M/3/5/LM35.shtml)>. Acesso em: 10 out. 2013.

DatasheetCatalog.com. **NE555N datasheet**. Disponível em:  
<[http://www.datasheetcatalog.com/datasheets\\_pdf/N/E/5/5/NE555N.shtml](http://www.datasheetcatalog.com/datasheets_pdf/N/E/5/5/NE555N.shtml)>. Acesso em: 10 out. 2013.

DatasheetCatalog.com. **PC817**. Disponível em  
<<http://pdf.datasheetcatalog.com/datasheet/Sharp/mXruvuu.pdf>>. Acesso em: 10 out. 2013.

DatasheetCatalog.com. **TIP41 datasheet**. Disponível em  
<[http://www.datasheetcatalog.com/datasheets\\_pdf/T/I/P/4/TIP41.shtml](http://www.datasheetcatalog.com/datasheets_pdf/T/I/P/4/TIP41.shtml)>. Acesso em: 10 out. 2013.

DatasheetCatalog.com. **7895 datasheet**. Disponível em  
<[http://www.datasheetcatalog.com/datasheets\\_pdf/7/8/0/5/7805.shtml](http://www.datasheetcatalog.com/datasheets_pdf/7/8/0/5/7805.shtml)>. Acesso em: 10 out. 2013.

Embrapa, Umidade. **Mudanças Climáticas Global e Agricultura-2008**, Disponível em <<http://www.cnpma.embrapa.br/unidade>>. Acesso em: 10 jun. 2013.

EXAME. Economia. **Os 10 países mais populosos do mundo**. Disponível em: <<http://exame.abril.com.br/economia/noticias/os-10-paises-mais-populosos-do-mundo-3>>. Acesso em: 20 set. 2013.

FIGUEIREDO, R. C. **Modernização de uma Câmara Calorimétrica para Execução de Testes Climáticos de Envelhecimento Acelerado**. Disponível em:  
<[http://www.google.com.br/url?sa=t&rct=j&q=&esrc=s&source=web&cd=9&ved=0CGUQFjAI&url=http%3A%2F%2Fbdm.bce.unb.br%2Fbitstream%2F10483%2F2828%2F1%2F2011\\_PG-Rogério\\_Chaves\\_Figueiredo\\_0695114.pdf&ei=C6JnUofHM4mikQfpwoHABA&usg=AFQjC NF7PeuS0MoW\\_myHBrVZxdqfNI428g](http://www.google.com.br/url?sa=t&rct=j&q=&esrc=s&source=web&cd=9&ved=0CGUQFjAI&url=http%3A%2F%2Fbdm.bce.unb.br%2Fbitstream%2F10483%2F2828%2F1%2F2011_PG-Rogério_Chaves_Figueiredo_0695114.pdf&ei=C6JnUofHM4mikQfpwoHABA&usg=AFQjC NF7PeuS0MoW_myHBrVZxdqfNI428g)> . Acesso em: 13 out. 2013

GABI, Caio Fernandes et al. **Construção de um Protótipo de Regulação de Temperatura de Baixa Custo, como Instrumento Pedagógico de Ensino de Engenharia**. Disponível em: <<http://www.abenge.org.br/CobengeAnteriores/2012/artigos/102572.pdf>>. Acesso em: 21 mai. 2013.

GOMES, L. E. L.; PAREDE, I. M. **Eletrônica Automação industrial**. Editora Fundação Padre Anchieta. São Paulo. 2011.

HERECH, V. C.; PACIÊNCIA, J. A. A. **Controle Automatizado de Umidade Relativa em Ambientes Através de Psicrometria**. 2008. Disponível em:  
<[http://www.up.com.br/painelgpa/uploads/imagens/files/EngEletrica/Tcc/tcc\\_2008/Herech\\_Paci.pdf](http://www.up.com.br/painelgpa/uploads/imagens/files/EngEletrica/Tcc/tcc_2008/Herech_Paci.pdf)>. Acesso em: 09 set.2013.

IBGE. Sala de Imprensa. Notícias. **População brasileira deve chegar ao máximo (228,4 milhões) em 2042**. Disponível em:  
<<http://saladeimprensa.ibge.gov.br/noticias?view=noticia&id=1&idnoticia=2455&busca=1&t=populacao-brasileira-deve-chegar-maximo-228-4-milhoes-2042>> . Acesso em: 29 ago. 2013.

LAKATOS E. M.; MARCONI M. A. **Fundamentos de Metodologia Científica**, Editora Atlas S.A 2010.

Leão Consultoria e Desenvolvimento de Sistemas. **Controladores de Temperatura em Câmaras ou estufas**. 2011. Disponível em:

<<http://www.lcds.com.br/controltemperatura.asp>>. Acesso em: 10 abr.2013.

MANTOVANI, S. C. A.; OKI, N. **TEEE I-Projeto de Robôs Móveis**. 2013. Disponível em: <<http://www.feis.unesp.br/Home/departamentos/engenhariaeletrica/teee-i--projetos-de-robos-moveis-aula1-07-03-2013-final.pdf>>. Acesso em: 08 ago. 2013.

MELO, J. L. G. G. **Mini Curso Microcontrolador PIC 18f452 / Proteus**. 2011. Disponível em:

<<http://www.eletrica.ufpr.br/~james/Laboratorio%20V/arquivos/Mini%20Curso%20Microcontrolador.pdf>>. Acesso em: 23 set. 2013.

Microchip. Applications. **MPLAB IDE**. Disponível em:

<[http://www.microchip.com/stellent/idcplg?IdcService=SS\\_GET\\_PAGE&nodeId=1406&dDocName=en019469](http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en019469)>. Acesso em: 07 out. 2013.

Mosaico. Guia do Usuário Gravador – McFlash+. Disponível em:

<[http://www.mosaico.com.br/Midias/Documentacao/Manual%20McFlash\\_rev\\_02.pdf](http://www.mosaico.com.br/Midias/Documentacao/Manual%20McFlash_rev_02.pdf)>.

Acesso em: 09 out. 2013.

Mosaico. **Microchip Technology**. Disponível em:

<[http://microchip.wikidot.com/mplab:\\_start](http://microchip.wikidot.com/mplab:_start)>. Acesso em: 09 out. 2013.

NUNES, F. V. **Desenvolvimento de Sistema de Segurança Utilizando Microcontrolador PIC18F4550**. 2013. Disponível em:

<<http://www.em.ufop.br/cecau/monografias/2013/Felipe%20Vilela%20Nunes.pdf>>. Acesso em: 10 out. 2013.

PREECE, J. **Designer de Interação: Além da Interação Homem-Computador**, Editora Bookman 2005.

SANTIAGO, E. InfoEscola. Física. Eletromagnetismo. Eletricidade. **Corrente Contínua e Corrente Alternada**. 2012. Disponível em:

<<http://www.infoescola.com/eletricidade/corrente-continua-e-alternada/>>. Acesso em: 07 out. 2013.

SCARTEZINI, L. M. **Análise e Melhoria de Processos**. 2009. Disponível em:

<<http://www.aprendersempre.org.br/arqs/GE%20B%20-%20An%20E1lise-e-Melhoria-de-Processos.pdf>>. Acesso em: 12 out.2013.

SEVERINO, A. J, **Metodologia do Trabalho Científico**. Editora Cortez 2009.

SILVA, D. J. H. **Cultivo em Ambiente Protegido**. Viçosa. Editora UFV. 2004.

SILVA, R. A. **Programando Microcontroladores PIC: Linguagem “C”**. São Paulo. Ensino Profissional, 2006.

TUBELIS, A. **Conhecimentos Práticos Sobre Clima e Irrigação**. Editora Aprenda Fácil. 2001.

VIANELLO, R.L. **Meteorologia Básica e Aplicações**. Viçosa. Editora UFV. 2012.

## APÊNDICE

### APÊNDICE A

```

1: #include "C:\Users\Jean Christopher\Desktop\pic\Teste Teclado Display II\interrupt.h"
2: #include "C:\Users\Jean Christopher\Desktop\pic\Teste Teclado Display II\cuscostdio.h"
3: #include "C:\Users\Jean Christopher\Desktop\pic\Teste Teclado Display II\key.h"
4: #include <stdlib.h>
5: /*****
6:      Declaração de variáveis      *****/
7: /*****/
8: float temp;//Valor da temperatura
9: float vuma;//Voltagem do sensor de umidade
10: float uma;//Valor da umidade
11: int1 excedeu_ar=0;//Umidade máxima excedida
12: int1 excedeu_temp=0;//Temperatura máxima excedida
13: int1 insuf_ar=0;//Umidade mínima excedida
14: int1 insuf_temp=0;//Temperatura mínima excedida
15: int1 msg=0;//Mensagem do menu
16: int1 editando=0;//Editando variáveis do menu
17: int1 type=0;//Entrando com os valores
18: int1 acabou=1;//Sair do menu
19: int1 maxima=0;//Entrada dos valores máximos do sensor
20: float temp min=0,ar min=0;//Valores mínimos de temperatura e umidade
21: float temp max=100,ar max=100;//Valores máximos de temperatura e umidade
22: int8 cont=14;//Contador de quantidade de valores no display
23: int8 op=0;//Armazena opção do menu
24: const char grau[] = {0x40,0x4C,0x52,0x52,0x4C,0x40,0x40,0x40};//Símbolo Grau
25: const char porcento[] = {0x5C,0x55,0x5A,0x44,0x4B,0x55,0x47,0x40};//Símbolo Porcento
26: char valor lcd[5];//Armazena valores do display
27: float valor;//Armazena valor do display em float
28: void simbol_grau(char x, char y)//Função de impressão do símbolo grau
29: {
30:     char i;
31:     lcd_send_byte(0,0x40);
32:     for (i = 0; i<=7; i++) lcd_send_byte(1,grau[i]);
33:     lcd_gotoxy(x,y); lcd_putc(0);
34: }
35: void simbol_porcento(char x, char y)//Função de impressão do símbolo porcento
36: {
37:     char i;
38:     lcd_send_byte(0,0x40);
39:     for (i = 0; i<=7; i++) lcd_send_byte(1,porcento[i]);
40:     lcd_gotoxy(x,y); lcd_putc(0);
41: }
42: /*****/
43: /****      FIM - Declaração      *****/
44: /*****/
45:
46:
47:
48:
49: void setup (void)//Função que seta configurações do PIC
50: {
51:     set tris b(0b00000111);
52:     lcd_init();
53:     setup_adc_ports(AN0 AN1 AN2 AN3_AN4);
54:     setup_adc(ADC_CLOCK_INTERNAL);
55:     setup_psp(PSP_DISABLED);
56:     setup_spi(SPI_SS_DISABLED);
57:     setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);
58:     setup_timer_1(T1_DISABLED);
59:     setup_timer_2(T2_DISABLED,0,1);
60: }
61: void bip(int v)//Função para emitir sinal sonoro
62: {
63:     output_high(PIN_C4);//Aplica voltagem no pino C4
64:     delay ms(v); //Define tempo para desativar o o BIP
65:     output_low(PIN_C4);//Desativa voltagem no pino C4
66: }
67: void exaustor(float t)//Função para ativar e desativar sistema de exaustão
68: {
69: if(t>=temp_max&&!excedeu_temp)//verifica condicional
70: {
71:     output_high(PIN_C0);//Aplica voltagem no pino C0
72:     bip(200);//Emite bip com 200 ms de duração
73:     delay ms(200);//Aguarda 200 ms

```

```

74:     bip(200); //Emite bip com 200 ms de duração
75:     excedeu_temp=1; //Define verdadeiro para temperatura máxima excedida
76: }
77: else if(excedeu_temp&&temp<=temp_max-3) //verifica condicional
78: {
79:     output_low(PIN_C0); //Desativa voltagem pino C0
80:     lcd_gotoxy(1,2); //Posiciona cursor display
81:     lcd_putc("\f"); //Limpa o LCD
82:     excedeu_temp=0; //Define falso para temperatura excedida
83: }
84: }
85: void aquecedor(float t) //Função para ativar e desativar sistema de aquecimento
86: {
87: if(t<temp_min&&!insuf_temp) //verifica condicional
88: {
89:     output_high(PIN_C1); //Aplica voltagem no pino C1
90:     bip(200); //Emite bip com 200 ms de duração
91:     delay_ms(200); //Aguarda 200 ms
92:     bip(200); //Emite bip com 200 ms de duração
93:     insuf_temp=1; //Define verdadeiro para temperatura mínima excedida
94: }
95: else if(insuf_temp&&t>=temp_min+3) //Verifica condicional
96: {
97:     output_low(PIN_C1); //Desativa voltagem pino C1
98:     insuf_temp=0; //Define falso para temperatura mínima excedida
99: }
100: }
101:
102: void nebulizador(float h) //Função para ativar e desativar sistema de nebulização
103: {
104: if(h<ar_min&&!insuf_ar) //Verifica condicional
105: {
106:     output_high(PIN_C2); //Aplica voltagem no pino C1
107:     bip(200); //Emite bip com 200 ms de duração
108:     delay_ms(200); //Aguarda 200 ms
109:     bip(200); //Emite bip com 200 ms de duração
110:     insuf_ar=1; //Define verdadeiro para umidade mínima excedida
111: }
112: else if(insuf_ar&&h>=ar_min+3) //Verifica condicional
113: {
114:     output_low(PIN_C2); //Desativa voltagem pino C2
115:     insuf_ar=0; //Define falso para umidade mínima excedida
116: }
117: }
118: void exaustor_reverso(float h) //Função para ativar e desativar sistema de exaustão reversa
119: {
120: if(h>ar_max&&!excedeu_ar) //Verifica condicional
121: {
122:     output_high(PIN_C3); //Aplica voltagem no pino C3
123:     bip(200); //Emite bip com 200 ms de duração
124:     delay_ms(200); //Aguarda 200 ms
125:     bip(200); //Emite bip com 200 ms de duração
126:     excedeu_ar=1; //Define verdadeiro para umidade máxima excedida
127: }
128: else if(excedeu_ar&&h<=ar_max-3) //Verifica condicional
129: {
130:     output_low(PIN_C3); //Desativa voltagem pino C3
131:     excedeu_ar=0; //Define falso para umidade máxima excedida
132: }
133: }
134:
135:
136: void temperatura() //Função de leitura da temperatura
137: {
138:     set_adc_channel(1); //Define canal de leitura do ADC
139:     delay_us(92); //Espera para conversão
140:     temp=(float)read_adc()*5*100/1023; //Leitura da voltagem e conversão em temperatura
141:     lcd_gotoxy(1,1); //Posiciona cursor do display
142:     printf(lcd_putc,"Temp:"); //Escreve no display
143:     lcd_gotoxy(10,1); //Posiciona cursor do LCD
144:     printf(lcd_putc,"%2.2F",temp); //Apresenta valor da temperatura na tela
145:     simbol_grau(16,1); //Desenha simbolo de grau no display
146:     exaustor(temp); //Chama a função exaustor
147:     aquecedor(temp); //Chama a função aquecedor
148: }
149: void umidade_ar() //Função de leitura de umidade
150: {
151:     set_adc_channel(0); //Define canal de leitura do ADC
152:     delay_us(92); //Espera para conversão
153:     vuma=(float)read_adc()*0.004887585532746823069403714565; //Leitura da voltagem
154:     uma=(31.792*vuma)-26.282; //Conversão da voltagem em umidade

```

```

155:         if(uma>100)//Tratamento de variação do sensor
156:         {uma=100;}
157:         if(uma<0)//Tratamento de variação do sensor
158:         {uma=0;}
159:         lcd_gotoxy(1,2);//Posiciona cursor
160:         printf(lcd_putc,"Umidade:");//Escreve no LCD
161:         lcd_gotoxy(10,2);//Posiciona cursor
162:         printf(lcd_putc,"%2.2F",uma);//Apresenta umidade no display
163:         lcd_gotoxy(16,2);//posiciona cursor
164:         printf(lcd_putc,"%C",0x25);//Acrescenta simbolo de porcentagem ao display
165:         nebulizador(uma);//Chama a função nebulizador
166:         exaustor_reverso(uma);//Chama a função exaustor reverso
167:     }
168:
169: void main()//Função principal
170: {
171:     setup();//Chama função setup
172:     while(1)//Loop infinito
173:     {
174:         key=scan1();//Primeira verificação de teclas pressionadas
175:         key=scan2();//Segunda verificação de teclas pressionadas
176:         key=scan3();//Terceira verificação de teclas pressionadas
177:         key=scan4();//Quarta verificação de teclas pressionadas
178:
179:         if(Key=='*')//verifica se a tecla * foi pressionada
180:         {
181:             bip(200);//Emite bip com 200 ms de duração
182:             acabou=0;//Define falso para fim da edição
183:             cont=14; //Atribui 14 para a variável cont
184:             type=0;//Define falso para digitando valores
185:             editando=1;//Define verdadeiro para edição
186:             lcd_gotoxy(1,1);//posiciona cursor do display
187:             lcd_putc("\fEscolha o sensor");//Escreve na tela
188:             delay ms(1000);//Mantem o texto por um segundo no LCD
189:             msg=1;//Define verdadeiro para apresentação da mensagem
190:             Key='N';//Define que nenhuma tecla foi pressionada
191:         }
192:         if(msg)//Verifica se deve apresentar a mensagem
193:         {
194:             lcd_putc("\f1-Temperatura");//Escreve no display
195:             lcd_gotoxy(1,2);//Posiciona cursor
196:             lcd_putc("\f2-Umidade do AR");//Escreve no display
197:             msg=0;//Define falso para apresentação da mensagem
198:             key='N';//Define que nenhuma tecla foi pressionada
199:         }
200:
201:         if(editando)//Verifica se o modo de edição está ativado
202:         {
203:
204:             if(key=='#'&&type==0)//Verifica condicional
205:             {
206:                 lcd_putc("\f");//Escreve na tela
207:                 editando=0;//Desabilita modo de edição
208:                 type=0;//Desabilita modo de digitação
209:                 cont=14;//Atribui 14 para o contador
210:                 acabou=1;//Define verdade para fim da edição
211:                 op=0;//Define 0 como opção
212:                 key='N';//Define que nenhuma tecla foi pressionada
213:             }
214:             if(Key=='1'&&type==0)//Verifica condicional
215:             {
216:                 editando=0;//Define falso para modo de edição
217:                 type=1;//Define verdadeiro para o modo de digitação
218:                 op=1;//Define 1 como opção
219:                 lcd_gotoxy(1,1);//Posiciona cursor
220:                 lcd_putc("\fTemperatura");//Escreve na tela
221:                 lcd_gotoxy(1,2);//Posiciona cursor
222:                 lcd_putc("\fMinima:");//Escreve na tela
223:                 simbol_grau(16,2);//Acrescenta simbolo de grau
224:                 Key='N';//Define que nenhuma tecla foi pressionada
225:             }
226:             if(Key=='2'&&type==0)//Verifica condicional

```

```

227:     {
228:         editando=0;//Define falso para modo de edição
229:         type=1;//Define verdadeiro para o modo de digitação
230:         op=2;//Define 2 como opção
231:         lcd_gotoxy(1,1);//Posiciona cursor
232:         lcd_putc("\fUmidade do Ar");
233:         lcd_gotoxy(1,2);//Posiciona cursor
234:         lcd_putc("Minima:");//Escreve na tela
235:         simbol_porcento(16,2);//Acrescenta simbolo de porcentagem
236:         Key='N';//Define que nenhuma tecla foi pressionada
237:     }
238:
239:     if(op==4)//Verifica condicional
240:     {
241:         type=1;//Define verdadeiro para o modo de digitação
242:         editando=0;//Define falso para modo de edição
243:         lcd_gotoxy(1,1);//Posiciona cursor
244:         lcd_putc("\fTemperatura");//Escreve no display
245:         lcd_gotoxy(1,2);//Posiciona cursor
246:         lcd_putc("Maxima:");//Escreve na tela
247:         simbol_grau(16,2);//Acrescenta simbolo de grau
248:         maxima=1;//Atribui verdadeiro para entrada dos valores de máxima
249:     }
250:     if(op==5)
251:     {
252:         type=1;//Define verdadeiro para o modo de digitação
253:         editando=0;//Define falso para modo de edição
254:         lcd_gotoxy(1,1);//Posiciona cursor
255:         lcd_putc("\fUmidade do Ar");//Escreve na tela
256:         lcd_gotoxy(1,2);//Posiciona cursor
257:         lcd_putc("Maxima:");//Escreve na tela
258:         simbol_porcento(16,2);//Acrescenta simbolo de grau
259:         maxima=1;//Atribui verdadeiro para entrada dos valores de máxima
260:     }
261:
262: }
263: if(type)//Verifica se o modo de edição está ativado
264: {
265:     if(cont<16&&key!='#'&&Key!='N')//Verifica condicional
266:     {
267:         lcd_gotoxy(cont,2);//Posiciona cursor
268:         printf(lcd_putc,"%C",key);//Escreve na tela a tecla pressionada
269:         key='N';//Define que nenhuma tecla foi pressionada
270:         delay_ms(300);//Aguarda 300 ms antes de continuar
271:         cont++;//Incrementa o contador
272:     }
273: }
274:
275:     if(cont>15&&key=='#')//Verifica condicionais
276:     {
277:         bip(500);//Emite bip com 500 ms de duração
278:         valor_lcd[0]=lcd_getc(14,2);//Captura valor da posição 14 do display
279:         valor_lcd[1]=lcd_getc(15,2);//Captura valor da posição 15 do display
280:         valor_lcd[2]='.');//Acrescenta ponto ao vetor
281:         valor_lcd[3]='0';//Acrescenta zero ao vetor
282:         valor_lcd[4]='0';//Acrescenta zero ao vetor
283:         valor=atof(valor_lcd);//Converte o vetor de char em float
284:
285:         if(maxima)//Verifica se entrada dos valores de máxima está ativada
286:         {
287:             type=0;//Define falso para o modo de digitação
288:             maxima=0;//Atribui falso para entrada dos valores de máxima
289:             if(op==4)//Verifica condicional
290:             { temp_max=valor;//Atribui valor coletado do display para a variável
291:               acabou=1;//Define verdadeiro para o fim da edição
292:               op=0;//Define 0 como opção
293:               excedeu_temp=0;//Define falso para temperatura máxima excedida
294:               output_low(c1);//Desativa voltagem no pino C1
295:             }
296:             if(op==5)//Verifica condicional
297:             { ar_max=valor;//Atribui valor coletado do display para a variável
298:               acabou=1;//Define verdadeiro para o fim da edição
299:               op=0;//Define 0 como opção
300:               excedeu_ar=0;//Define falso para umidade máxima excedida
301:               output_low(c3);//Desativa voltagem no pino C3

```

```
302:         }
303:         lcd_gotoxy(1,1);//Posiciona cursor do display
304:         lcd_putc("\f");//Limpa cursor
305:         lcd_gotoxy(1,2);//Posiciona cursor do display
306:         lcd_putc("\f");//Limpa cursor
307:
308:     }
309:
310:     if(op==1)//Verifica condicional
311:     { temp_min=valor;//Atribui valor coletado do display para a variável
312:       op=4;//Define 4 como opção
313:       editando=1;//Define verdadeiro para o modo de edição
314:       acabou=0;//Define falso para fim da edição
315:       insuf temp=0;//Define falso para temperatura mínima excedida
316:       output_low(c0);//Desativa voltagem no pino C0
317:     }
318:     if(op==2)
319:     { ar_min=valor;//Atribui valor coletado do display para a variável
320:       op=5;//Define 5 como opção
321:       editando=1;//Define verdadeiro para o modo de edição
322:       acabou=0;//Define falso para fim da edição
323:       insuf ar=0;//Define falso para temperatura mínima excedida
324:       output_low(c2);//Desativa voltagem no pino C2
325:     }
326:     cont=14;//Define 14 para o contador
327:
328:     }
329:     if(acabou)//Verifica se a edição foi finalizada
330:     {
331:         temperatura();//Chama função temperatura
332:         umidade_ar();//Chama função umidade do ar
333:
334:     }
335:
336: }
337:
338: }
```

**APÊNDICE B**

```
1: #include <16F877.h>           //biblioteca do microcontrolador
2: #device adc=10                //ADC com 10 bits para conversão
3:
4: #FUSES NOWDT                  //No Watch Dog Timer
5: #FUSES XT                     //Low power osc < 200 khz
6: #FUSES PUT                    //Power Up Timer
7: #FUSES PROTECT                //Code protected from reads
8: #FUSES BROWNOUT              //brownout reset
9: #FUSES NOLVP                  //No low voltage prgming
10: #FUSES NOCPD                 //No EE protection
11: #FUSES NOWRT                 //Program memory not write protected
12: #FUSES NODEBUG               //No Debug mode for ICD
13:
14: #use delay(clock=4000000)     //clock de 4 Mhz
15:
--
```

## ANEXOS

### ANEXO A - HIH-4000-001

#### HIH-4000 Series Integrated Circuitry Humidity Sensor, 2,54 mm (0.100 in) Lead Pitch SIP



Representative photograph, actual product appearance may vary. Due to regional agency approval requirements, some products may not be available in your area. Please contact your regional Honeywell office regarding your product of choice.

#### Features

- Molded thermoset plastic housing with cover
- Linear voltage output vs %RH
- Laser trimmed interchangeability
- Low power design
- High accuracy
- Fast response time
- Stable, low drift performance
- Chemically resistant

#### Typical Applications

- Refrigeration
- Drying
- Meteorology
- Battery-powered systems
- OEM assemblies

## Description

The HIH-4000 Series Humidity Sensors are designed specifically for high volume OEM (Original Equipment Manufacturer) users. Direct input to a controller or other device is made possible by this sensor's linear voltage output. With a typical current draw of only 200 0A, the HIH-4000 Series is ideally suited for low drain, battery operated systems. Tight sensor interchangeability reduces or eliminates OEM production calibration costs. Individual sensor calibration data is available.

The HIH-4000 Series delivers instrumentation-quality RH (Relative Humidity) sensing performance in a low cost, solderable SIP (Single In-line Package). Available in two lead spacing configurations, the RH sensor is a laser trimmed, thermoset polymer capacitive sensing element with on-chip integrated signal conditioning. The sensing element's multilayer construction provides excellent resistance to most application hazards such as wetting, dust, dirt, oils and common environmental chemicals.

### HIH-4000-001

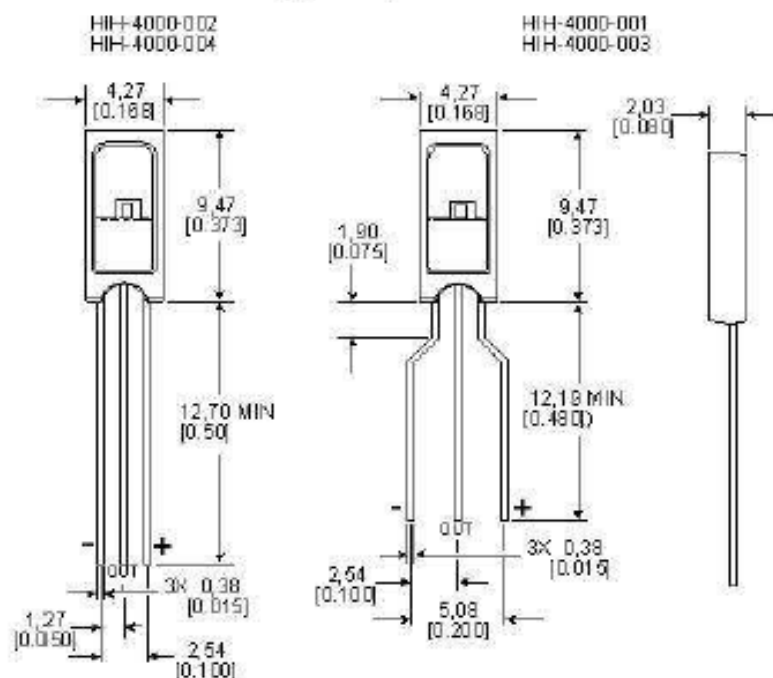
**HIH-4000 Series Integrated Circuitry Humidity Sensor, 2,54 mm (0.100 in) Lead Pitch SIP**

Product Specifications	
<b>Package Style</b>	Solderable SIP
<b>Termination Details</b>	2,54 mm [0.100 in] Lead Pitch
<b>Series Name</b>	HIH-4000 Series
<b>RH Accuracy</b>	± 3.5% RH, 0-100 % RH non-condensing, 25 °C, 5 Vdc supply
<b>RH Interchangeability</b>	± 5% RH, 0-60% RH; ± 8% @ 60-100% RH Typical
<b>RH Hysteresis</b>	± 3% of RH Span Maximum
<b>RH Repeatability</b>	± 0.5% RH
<b>RH response time, 1/e</b>	15 s in slowly moving air @ 25 °C
<b>RH Stability</b>	± 0.2% RH Typical at 50% RH in 1 Year
<b>Supply Voltage</b>	4.0 Vdc to 5.8 Vdc
<b>Supply Current</b>	500 µA Max.
<b>Operating Humidity Range</b>	0 to 100% RH, non-condensing
<b>Operating Temperature Range</b>	-40 °C to 85 °C (-40 °F to 185 °F)
<b>Temperature Compensation</b>	True RH = Sensor RH / (1.0305 + 0.000044T - 0.0000011T <sup>2</sup> ) T in °C (True RH = Sensor RH / (0.9237 - 0.0041T + 0.000040T <sup>2</sup> ) T in °C)
<b>Availability</b>	Global
<b>Comment</b>	Light sensitive, shield from bright light.
<b>UNSPSC Code</b>	411121
<b>UNSPSC Commodity</b>	411121 Transducers

## HIH-4000-001

HIH-4000 Series Integrated Circuitry Humidity Sensor, 2,54 mm (0.100 in) Lead Pitch SIP

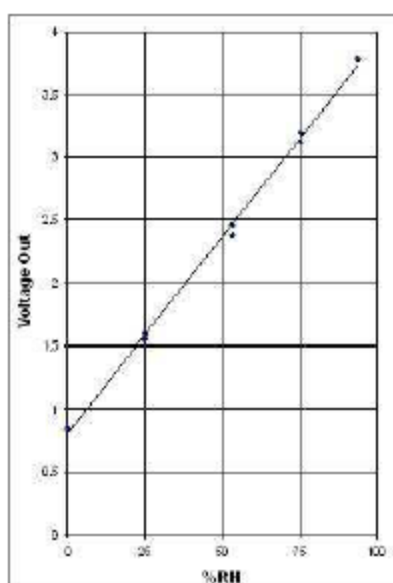
Mounting Dimensions  
For Reference Only [mm/in]



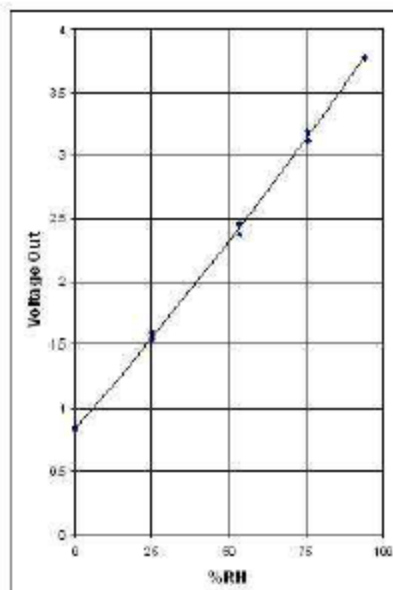
## HIH-4000-001

HIH-4000 Series Integrated Circuitry Humidity Sensor, 2,54 mm (0.100 in) Lead Pitch SIP

TYPICAL BEST FIT STRAIGHT LINE



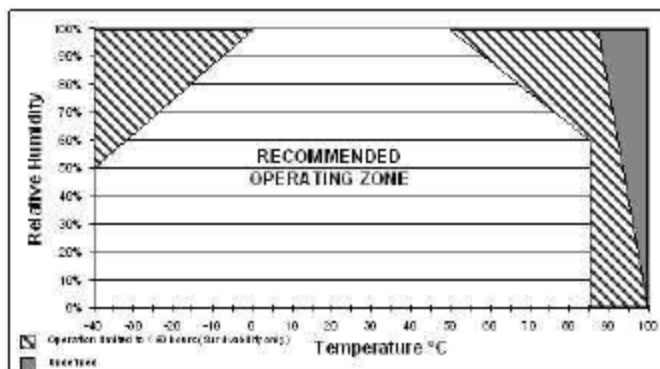
TYPICAL 2nd ORDER CURVE FIT



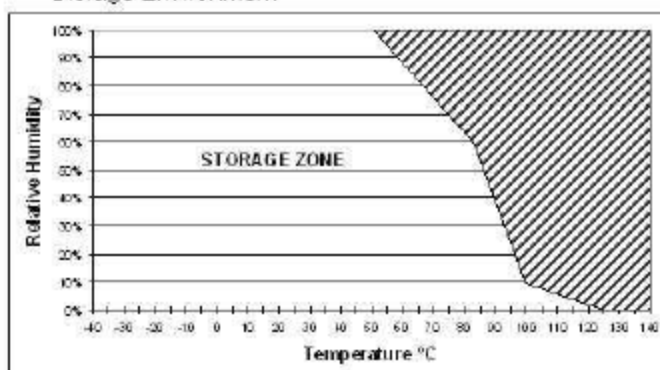
## HIH-4000-001

HIH-4000 Series Integrated Circuitry Humidity Sensor, 2,54 mm (0.100 in) Lead Pitch SIP

Recommended Operating Conditions



Storage Environment



## HIH-4000-001

HIH-4000 Series Integrated Circuitry Humidity Sensor, 2,54 mm (0.100 in) Lead Pitch SIP

### **⚠ WARNING**

#### **PERSONAL INJURY**

DO NOT USE these products as safety or emergency stop devices, or in any other application where failure of the product could result in personal injury.

**Failure to comply with these instructions could result in death or serious injury.**

### **⚠ WARNING**

#### **MISUSE OF DOCUMENTATION**

- The information presented in this product sheet (or catalog) is for reference only. DO NOT USE this document as product installation information.
- Complete installation, operation and maintenance information is provided in the instructions supplied with each product.

**Failure to comply with these instructions could result in death or serious injury.**

## ANEXO B – Cuscustudio.c

```

1:
2: /*
3: BIBLIOTECA PADRÃO PARA CUSCOVIC
4: Desenvolvida para PICs famílias 16 ou 18 de 28 ou 40 pinos.
5: Autor: Daniel Corteletti
6: Desenvolvido em agosto de 2007
7: Revisão em setembro 2007
8: Revisão em novembro 2008
9: --> Adição de timeout (mediante variável global de configuração) para função cusco_readkbd()
10: --> Variável global cusco_readkbd_timeout c/ valor padrão 0
11: --> Adição de rotinas de gravação e leitura da eeprom em 16 e 32 bits.
12:
13: */
14:
15: #define RELE1 pin_e0
16: #define RELE2 pin_e1
17: #define BUZZER pin_e2
18: #define C0 pin_c0
19: #define C1 pin_c1
20: #define C2 pin_c2
21: #define C3 pin_c3
22: #define C4 pin_c4
23: #define C5 pin_c5
24: #define C6 pin_c6
25: #define C7 pin_c7
26: #define D0 pin_d0
27: #define D1 pin_d1
28: #define D2 pin_d2
29: #define D3 pin_d3
30: #define D4 pin_d4
31: #define D5 pin_d5
32: #define D6 pin_d6
33: #define D7 pin_d7
34: #define A0 pin_a0
35: #define A1 pin_a1
36: #define A2 pin_a2
37: #define A3 pin_a3
38: #define A4 pin_a4
39: #define A5 pin_a5
40:
41:
42:
43: // Esta biblioteca foi modificada a partir da biblioteca original CCS, com o
44: // intuito de manter o padrão de programação atendendo as necessidades do LCD
45: // para a estação didática CUSCOVIC. A definição dos pinos é :
46: // PINOS PIC PINOS LCD
47: //      B0 enable
48: //      B1 rs
49: //      B2 rw
50: //      B4 D4
51: //      B5 D5
52: //      B6 D6
53: //      B7 D7
54: //
55: // Os pinos D0 a D3 do LCD não são usados.
56:
57: struct lcd_pin_map { // This structure is overlaid
58:     BOOLEAN enable; // on to an I/O port to gain
59:     BOOLEAN rs; // access to the LCD pins.
60:     BOOLEAN rw; // The bits are allocated from
61:     BOOLEAN unused; // low order up. ENABLE will
62:     int data : 4; // be pin B0.
63: } lcd;
64:
65: #byte lcd = 6
66: #define set_tris_lcd(x) set_tris_b(x)
67: #define lcd_type 2 // 0=5x7, 1=5x10, 2=2 linhas
68: #define lcd_line_two 0x40 // Endereço de RAM da segunda linha
69:
70:
71: BYTE const LCD_INIT_STRING[4] = {0x20 | (lcd_type << 2), 0xc, 1, 6};
72: // Estes bytes são enviados ao lcd para inicializa-lo
73:
74: struct lcd_pin_map const LCD_WRITE = {0,0,0,0,0}; // For write mode all pins are out
75: struct lcd_pin_map const LCD_READ = {0,0,0,0,15}; // For read mode data pins are in

```

```

76:
77: #separate
78: BYTE lcd_read_byte()
79: {
80:     BYTE low,high;
81:     set_tris_lcd(LCD_READ);
82:     lcd.rw = 1;
83:     delay_cycles(1);
84:     lcd.enable = 1;
85:     delay_cycles(1);
86:     high = lcd.data;
87:     lcd.enable = 0;
88:     delay_cycles(1);
89:     lcd.enable = 1;
90:     delay_us(1);
91:     low = lcd.data;
92:     lcd.enable = 0;
93:     set_tris_lcd(LCD_WRITE);
94:     return( (high<<4 | low);
95: }
96:
97: #separate
98: void lcd_send_nibble( BYTE n )
99: {
100:     lcd.data = n;
101:     delay_cycles(1);
102:     lcd.enable = 1;
103:     delay_us(2);
104:     lcd.enable = 0;
105: }
106:
107:
108: // funcao abaixo modificada para evitar que o LCD trave o programa no caso de nao responder.
109:
110: #separate
111: void lcd_send_byte( BYTE address, BYTE n )
112: {
113:     int cont = 200;
114:     lcd.rs = 0;
115:     while ( bit_test(lcd_read_byte(),7) )
116:     {
117:         delay_us(50);
118:         cont --;
119:         if (!cont) break;
120:     }
121:     lcd.rs = address;
122:     delay_cycles(1);
123:     lcd.rw = 0;
124:     delay_cycles(1);
125:     lcd.enable = 0;
126:     lcd_send_nibble(n >> 4);
127:     lcd_send_nibble(n & 0xf);
128: }
129:
130: short lcd_initialized = 0;
131:
132: #separate
133: void lcd_init()
134: {
135:     BYTE i;
136:     set_tris_lcd(LCD_WRITE);
137:     lcd.rs = 0;
138:     lcd.rw = 0;
139:     lcd.enable = 0;
140:     delay_ms(15);
141:     for(i=1;i<=3;++i)
142:     {
143:         lcd_send_nibble(3);
144:         delay_ms(5);
145:     }
146:     lcd_send_nibble(2);
147:     for(i=0;i<=3;++i)
148:         lcd_send_byte(0,LCD_INIT_STRING[i]);
149:     lcd_initialized = 1;
150: }
151:

```

```

152:
153: #separate
154: void lcd_gotoxy( BYTE x, BYTE y)
155: {
156:     BYTE address;
157:     if(y!=1)
158:         address=lcd_line_two;
159:     else
160:         address=0;
161:     address+=x-1;
162:     lcd_send_byte(0,0x80|address);
163: }
164:
165: #separate
166: void lcd_putc( char c)
167: {
168:     if (!lcd_initialized) lcd_init();
169:     else
170:         switch (c)
171:         {
172:             case '\f' : lcd_send_byte(0,1);
173:                         delay_ms(2);
174:                                     break;
175:             case '\n' : lcd_gotoxy(1,2);
176:                         break;
177:             case '\b' : lcd_send_byte(0,0x10);
178:                         break;
179:             default : lcd_send_byte(1,c);
180:                       break;
181:         }
182: }
183:
184: #separate
185: char lcd_getc( BYTE x, BYTE y)
186: {
187:     char value;
188:     lcd_gotoxy(x,y);
189:     while ( bit_test(lcd_read_byte(),7) ); // wait until busy flag is low
190:     lcd.rs=1;
191:     value = lcd_read_byte();
192:     lcd.rs=0;
193:     return(value);
194: }
195:
196: // funcoes adicionadas por Daniel Corteletti
197:
198: #separate
199: short lcd_connected()
200: {
201:     int cont = 200;
202:     lcd.rs = 0;
203:     while ( bit_test(lcd_read_byte(),7) )
204:     {
205:         delay us(50);
206:         cont --;
207:         if (!cont) return(0);
208:     }
209:     lcd.rs = 1;
210:     return(1);
211: }
212:
213: #separate
214: void lcd_cursor_on()
215: {
216:     lcd_send_byte(0,0x0E);
217: }
218:
219: #separate
220: void lcd_cursor_blink()
221: {
222:     lcd_send_byte(0,0x0F);
223: }
224:
225: #separate
226: void lcd_cursor_off()
227: {
228:     lcd_send_byte(0,0x0C);
229: }
230:
231: #separate
232: void lcd_shift_left()
233: {
234:     lcd_send_byte(0,0x18);
235: }
236:
237: #separate

```

## ANEXO C – KEY.c

```

1:
2:
3: /*****
4: *           Teclado Matricial com Display           *
5: *****/
6: * Rafael Varago de Aguiar           Varago Automatics *
7: *****/
8: *           www.mecatronica degaragem.blogspot.com           *
9: * Data: 24/06/2011           *
10: *           *
11: * RAM: 4%   ROM: 5%           *
12: *****/
13: * Exibe o numero em um display 7 segmentos de acordo com o *
14: * botão pressionado no teclado matricial de 0 a 9 enter e backspace *
15: *****/
16: // Definições de pinos
17: #define 10 pin_d3 // Primeira linha do teclado
18: #define 11 pin_d4 // Segunda   " do   "
19: #define 12 pin_d5 // Terceira   " do   "
20: #define 13 pin_d6 // Quarta     " do   "
21: #define c10 pin_d0 // Primeira coluna do teclado
22: #define c11 pin_d1 // Primeira   " do   "
23: #define c12 pin_d2 // Primeira   " do   "
24: // Varredura de teclas
25: char key;
26: char scan1(){ // Faz primeira varredura
27:
28:     int aux=50; // Tempo de varredura
29:     while(aux != 0){ // Laço para varredura
30:         output_low(10); // Joga 0 na coluna a ser varrida e 1 nas demais
31:         output_high(11);
32:         output_high(12);
33:         output_high(13);
34:         if(!input(c10)){ // Testa primeira coluna
35:             delay_ms(250); // Tempo para soltar tecla
36:             key= "1";
37:         }
38:         if(!input(c11)){ // Testa segunda coluna
39:             delay_ms(250); // Tempo para soltar a tecla
40:             key= "2";
41:         }
42:         if(!input(c12)){ // Testa terceira coluna
43:             delay_ms(250); // Tempo para soltar a tecla
44:             key="3";
45:         }
46:         aux--; // Decrementa 1 de aux
47:         delay_us(aux); // Tempo de atraso da varredura é igual a aux
48:     }
49:     return key;
50: } // Fim da sub-rotina
51: *****/
52: char scan2(){ // Faz segunda varredura
53:
54:     int aux=50; // Tempo de varredura
55:     while(aux != 0){ // Laço para varredura
56:         output_high(10); // Joga 0 na coluna a ser varrida e 1 nas demais
57:         output_low(11);
58:         output_high(12);
59:         output_high(13);
60:         if(!input(c10)){ // Testa primeira coluna
61:             delay_ms(250); // Tempo para soltar tecla
62:             key="4";
63:         }
64:         if(!input(c11)){ // Testa segunda coluna
65:             delay_ms(250); // Tempo para soltar a tecla
66:             key="5";
67:         }
68:         if(!input(c12)){ // Testa terceira coluna
69:             delay_ms(250); // Tempo para soltar a tecla
70:             key= "6";
71:         }
72:         aux--; // Decrementa 1 de aux
73:         delay_us(aux); // Tempo de atraso da varredura é igual a aux
74:     }
75:     return key;
76: } // Fim da sub-rotina
77: *****/

```

```

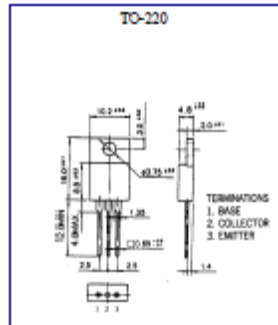
78: char scan3(){ // Faz terceira varredura
79:
80:     int aux=50; // Tempo de varredura
81:     while(aux != 0){ // Laço para varredura
82:         output_high(10); // Joga 0 na coluna a ser varrida e 1 nas demais
83:         output_high(11);
84:         output_low(12);
85:         output_high(13);
86:         if(!input(c10)){ // Testa primeira coluna
87:             delay_ms(250); // Tempo para soltar tecla
88:             key="7";
89:         }
90:         if(!input(c11)){ // Testa segunda coluna
91:             delay_ms(250); // Tempo para soltar a tecla
92:             key="8";
93:         }
94:         if(!input(c12)){ // Testa terceira coluna
95:             delay_ms(250); // Tempo para soltar a tecla
96:             key="9";
97:         }
98:         aux--; // Decrementa 1 de aux
99:         delay_us(aux); // Tempo de atraso da varredura é igual a aux
100:     }
101:     return key;
102: } // Fim da sub-rotina
103: /*****
104: char scan4(){ // Faz quarta varredura
105:
106:     int aux=50; // Tempo de varredura
107:     while(aux != 0){ // Laço para varredura
108:         output_high(10); // Joga 0 na coluna a ser varrida e 1 nas demais
109:         output_high(11);
110:         output_high(12);
111:         output_low(13);
112:         if(!input(c10)){ // Testa primeira coluna
113:             delay_ms(250); // Tempo para soltar tecla
114:             key="*";
115:         }
116:         if(!input(c11)){ // Testa segunda coluna
117:             delay_ms(250); // Tempo para soltar a tecla
118:             key="0";
119:         }
120:         if(!input(c12)){ // Testa terceira coluna
121:             delay_ms(250); // Tempo para soltar a tecla
122:             key="#";
123:         }
124:         aux--; // Decrementa 1 de aux
125:         delay_us(aux); // Tempo de atraso da varredura é igual a aux
126:     }
127:     return key; // Retorna
128: } // Fim da sub-rotina
129: /*****
130: *****/

```

## ANEXO D - TIP41 SERIES (TIP41/41A/41B/41C)

### MEDIUM POWER LINEAR SWITCHING APPLICATIONS

•Complementary to TIP42/42A/42B/42C



### ABSOLUTE MAXIMUM RATINGS (Ta=25°C)

Characteristic	Symbol	Rating	Unit
Collector-Base Voltage	V <sub>CB0</sub>	40	V
:TIP41A		60	V
:TIP41B		80	V
:TIP41C		100	V
Collector-Emitter Voltage	V <sub>CE0</sub>	40	V
:TIP41A		60	V
:TIP41B		80	V
:TIP41C		100	V
Emitter-Base voltage	V <sub>EB0</sub>	5	V
Collector Current (DC)	I <sub>C</sub>	6	A
Collector Current (Pulse)	I <sub>C</sub>	10	A
Base Current (DC)	I <sub>B</sub>	2	A
Collector Dissipation (Tc=25°C)	P <sub>C</sub>	65	W
Junction Temperature	T <sub>J</sub>	150	
Storage Temperature	T <sub>stg</sub>	-50~150	

### ELECTRICAL CHARACTERISTICS (Ta=25°C)

Characteristic	Symbol	Test Condition	Min	Typ	Max	Unit
Collector-Emitter Sustaining Voltage	BV <sub>CEO(SUS)</sub>	I <sub>C</sub> =30mA, I <sub>B</sub> =0	40			V
:TIP41A			60			V
:TIP41B			80			V
:TIP41C			100			V
Collector Cutoff Current	I <sub>CO</sub>	V <sub>CE</sub> =30V, I <sub>B</sub> =0			0.7	mA
:TIP41/41A					0.7	mA
:TIP41B/41C		V <sub>CE</sub> =60V, I <sub>B</sub> =0			0.7	mA
Collector Cutoff Current	I <sub>CES</sub>	V <sub>CE</sub> =40V, V <sub>BE</sub> =0			400	μA
:TIP41A					400	μA
:TIP41B		V <sub>CE</sub> =60V, V <sub>BE</sub> =0			400	μA
:TIP41C					400	μA
Emitter Cutoff Current	I <sub>EO</sub>	V <sub>CE</sub> =80V, V <sub>BE</sub> =0			1	mA
DC Current Gain	h <sub>FE</sub>	V <sub>CE</sub> =100V, V <sub>BE</sub> =0	30		75	
Collector-Emitter Saturation Voltage	V <sub>CE(sat)</sub>	V <sub>BE</sub> =5V, I <sub>C</sub> =0			1.5	V
Base-Emitter On Voltage	V <sub>BE(sat)</sub>	V <sub>CE</sub> =4V, I <sub>C</sub> =0.3A			2.0	V
Current Gain Bandwidth Product	f <sub>T</sub>	V <sub>CE</sub> =4V, I <sub>C</sub> =3A I <sub>C</sub> =6A, I <sub>B</sub> =600mA V <sub>CE</sub> =4V, I <sub>C</sub> =6A V <sub>CE</sub> =10V, I <sub>C</sub> =500mA f=1MHz	3.0			MHZ

## ANEXO E - KA78XX/KA78XXA

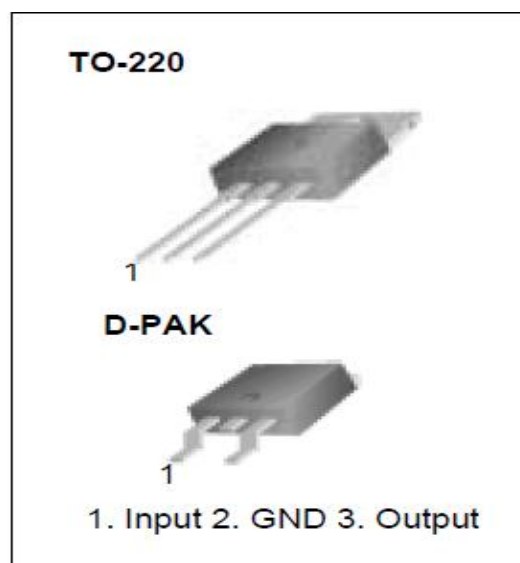
### 3-Terminal 1A Positive Voltage Regulator

#### Features

- Output Current up to 1A
- Output Voltages of 5, 6, 8, 9, 10, 12, 15, 18, 24V
- Thermal Overload Protection
- Short Circuit Protection
- Output Transistor Safe Operating Area Protection

#### Description

The KA78XX/KA78XXA series of three-terminal positive regulator are available in the TO-220/D-PAK package and with several fixed output voltages, making them useful in a wide range of applications. Each type employs internal current limiting, thermal shut down and safe operating area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltages and currents.



#### Absolute Maximum Ratings

Parameter	Symbol	Value	Unit
Input Voltage (for $V_O = 5V$ to $18V$ )	$V_I$	35	V
(for $V_O = 24V$ )	$V_I$	40	V
Thermal Resistance Junction-Cases (TO-220)	$R_{\theta JC}$	5	$^{\circ}C/W$
Thermal Resistance Junction-Air (TO-220)	$R_{\theta JA}$	65	$^{\circ}C/W$
Operating Temperature Range (KA78XX/A/R)	$T_{OPR}$	0 ~ +125	$^{\circ}C$
Storage Temperature Range	$T_{STG}$	-65 ~ +150	$^{\circ}C$

### Electrical Characteristics (KA7805/KA7805R)

(Refer to test circuit , $0^{\circ}C < T_J < 125^{\circ}C$ ,  $I_O = 500mA$ ,  $V_I = 10V$ ,  $C_I = 0.33\mu F$ ,  $C_O = 0.1\mu F$ , unless otherwise specified)

Parameter	Symbol	Conditions	KA7805			Unit	
			Min.	Typ.	Max.		
Output Voltage	$V_O$	$T_J = +25^{\circ}C$	4.8	5.0	5.2	V	
		$5.0mA \leq I_O \leq 1.0A$ , $P_O \leq 15W$ $V_I = 7V$ to $20V$	4.75	5.0	5.25		
Line Regulation (Note1)	Regline	$T_J = +25^{\circ}C$	$V_O = 7V$ to $25V$	-	4.0	100	mV
			$V_I = 8V$ to $12V$	-	1.6	50	
Load Regulation (Note1)	Regload	$T_J = +25^{\circ}C$	$I_O = 5.0mA$ to $1.5A$	-	9	100	mV
			$I_O = 250mA$ to $750mA$	-	4	50	
Quiescent Current	$I_Q$	$T_J = +25^{\circ}C$	-	5.0	8.0	mA	
Quiescent Current Change	$\Delta I_Q$	$I_O = 5mA$ to $1.0A$	-	0.03	0.5	mA	
		$V_I = 7V$ to $25V$	-	0.3	1.3		
Output Voltage Drift	$\Delta V_O / \Delta T$	$I_O = 5mA$	-	-0.8	-	$mV / ^{\circ}C$	
Output Noise Voltage	$V_N$	$f = 10Hz$ to $100KHz$ , $T_A = +25^{\circ}C$	-	42	-	$\mu V / V_O$	
Ripple Rejection	RR	$f = 120Hz$ $V_O = 8V$ to $18V$	62	73	-	dB	
Dropout Voltage	$V_{Drop}$	$I_O = 1A$ , $T_J = +25^{\circ}C$	-	2	-	V	
Output Resistance	$r_O$	$f = 1KHz$	-	15	-	$m\Omega$	
Short Circuit Current	$I_{SC}$	$V_I = 35V$ , $T_A = +25^{\circ}C$	-	230	-	mA	
Peak Current	$I_{PK}$	$T_J = +25^{\circ}C$	-	2.2	-	A	