

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E
TECNOLOGIA DE SANTA CATARINA
CAMPUS JOINVILLE
CURSO SUPERIOR DE TECNOLOGIA EM
MECATRÔNICA INDUSTRIAL**

**JEAN CARLOS BORGES
THIAGO FRÖHLICH**

**DESENVOLVIMENTO DE DISPOSITIVO DE CONTROLE
AUTOMÁTICO PARA OS PROCESSOS DE MOSTURAÇÃO E
FERVURA DA FABRICAÇÃO DE CERVEJA ARTESANAL**

TRABALHO DE CONCLUSÃO DE CURSO

**JEAN CARLOS BORGES
THIAGO FRÖHLICH**

**DESENVOLVIMENTO DE DISPOSITIVO DE CONTROLE
AUTOMÁTICO PARA OS PROCESSOS DE MOSTURAÇÃO E
FERVURA DA FABRICAÇÃO DE CERVEJA ARTESANAL**

JOINVILLE, 2015

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E
TECNOLOGIA DE SANTA CATARINA.
CAMPUS JOINVILLE
CURSO MECATRÔNICA INDUSTRIAL**

**JEAN CARLOS BORGES
THIAGO FRÖHLICH**

**DESENVOLVIMENTO DE DISPOSITIVO DE CONTROLE
AUTOMÁTICO PARA OS PROCESSOS DE MOSTURAÇÃO E
FERVURA DA FABRICAÇÃO DE CERVEJA ARTESANAL**

**Submetido ao Instituto
Federal de Educação, Ciência e
Tecnologia de Santa Catarina
como parte dos requisitos de
obtenção do título de Tecnólogo
em Mecatrônica Industrial.**

Orientador: Roberli L. Cantidio, MSC.

JOINVILLE, 2015

Borges, Jean Carlos. Fröhlich, Thiago.
Desenvolvimento de Dispositivo de Controle Automático para os Processos de Mosturação e Fervura da Fabricação de Cerveja Artesanal / Borges, Jean Carlos.; Fröhlich, Thiago – Joinville: Instituto Federal de Santa Catarina, 2015. 104p.

Trabalho de Conclusão de Curso - Instituto Federal de Santa Catarina, 2015. Graduação. Curso Superior de Tecnologia em Mecatrônica Industrial. Modalidade: Presencial. Orientador: Roberli Leopoldo Cantidio, MSc.

1. Cerveja 2. Automação 3. Controle 4. Interface 5. Arduino

I. DESENVOLVIMENTO DE DISPOSITIVO DE CONTROLE AUTOMÁTICO PARA OS PROCESSOS DE MOSTURAÇÃO E FERVURA DA FABRICAÇÃO DE CERVEJA ARTESANAL

**DESENVOLVIMENTO DE DISPOSITIVO DE CONTROLE
AUTOMÁTICO PARA OS PROCESSOS DE MOSTURAÇÃO E
FERVURA DA FABRICAÇÃO DE CERVEJA ARTESANAL**

**JEAN CARLOS BORGES
THIAGO FRÖHLICH**

Este trabalho foi julgado adequado para obtenção do título de Tecnólogo em Mecatrônica Industrial e aprovado na sua forma final pela banca examinadora do Curso Mecatrônica Industrial do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.

Joinville, 22 de junho de 2015.

Banca Examinadora:

**Prof. Roberli L. Cantidio, Mestre
Orientador**

**Prof. Leo Schirmer
Avaliador**

**Prof. Luis Mariano Nodari
Avaliador**

AGRADECIMENTOS

Agradecemos a esta instituição, seu corpo docente, direção, administração e demais funcionários por nos possibilitar a conclusão deste curso.

Ao orientador Roberli L. Cantidio e aos demais professores que nos auxiliaram, em especial ao professor Michael Klug pelo acompanhamento, apoio e ajuda neste projeto.

Ao senhor Douglas Tiago Hardt pelo equipamento fornecido e pela oportunidade de acompanhar os processos de fabricação da cerveja, possibilitando os testes deste projeto.

Aos colegas e amigos pelos conselhos, opiniões e partilha de conhecimentos.

Aos familiares pelo apoio, incentivo e compreensão.

E, sobretudo, a Deus que na sua infinita bondade nos concede a vida e a capacidade de desbravarmos as mais diversificadas áreas do conhecimento.

RESUMO

Este projeto descreve a automação de um equipamento de mosturação e fervura, que será utilizado na fabricação artesanal de cerveja. Com esta automação será possível realizar o controle de determinadas temperaturas e períodos de tempo, que serão definidas pelo usuário do equipamento. Esta automação consiste em um controlador de temperatura temporizado, podendo ser programado facilmente pelo operador através de uma Interface Homem Máquina (IHM), composta por uma tela touchscreen. O controle e manutenção da temperatura entre determinados períodos de tempo será feita dentro de uma plataforma programável Arduino, com realimentação e correção através de ação Proporcional-Integral-Derivativa (PID).

Palavras-chave: Cerveja. Automação. Controle. Interface. Arduino.

ABSTRACT

This project describes the automation of a mashing and boiling equipment that will be used in craft brewing. With this automation will be possible to control certain temperatures and time periods, which will be defined by the user of the equipment. This automation is at a temperature controller timer which can be programmed easily by the operator via a Man Machine Interface (MMI) consisting of a touchscreen *display*. The control and maintenance of temperature between certain time periods will be made within a programmable Arduino platform, with feedback and correction by Proportional-Integral-Derivative (PID) action.

Keywords: Beer. Automation. Control. Interface. Arduino.

LISTA DE FIGURAS

FIGURA 1 – Processos da cerveja	25
FIGURA 2 – Micro Cervejaria Profissional 100 litros	29
FIGURA 3 – Sistema <i>single vessel</i>	30
FIGURA 4 – Equipamento BeerBot 60	31
FIGURA 5 – Sensor de temperatura DS18B20 imersível	34
FIGURA 6 – Controle potência de ciclo integral	36
FIGURA 7 – Componentes do Arduino Mega 2560	38
FIGURA 8 – Tela sensível ao toque	42
FIGURA 9 – Fonte chaveada 12 Volts 10 Amperes	43
FIGURA 10 – Motor CC Bosch modelo CHP 9390082064	44
FIGURA 11 – Módulo relé 5V 10A de 2 canais	45
FIGURA 12 – Esboço do equipamento	47
FIGURA 13 – Resistência elétrica	50
FIGURA 14 – Dimensões da panela	51
FIGURA 15 – Misturador	52
FIGURA 16 – Diagrama de blocos do controlador	53
FIGURA 17 – Gráfico Temperatura X Tempo	54
FIGURA 18 – Esquema do controlador de potência CA	58
FIGURA 19 – Aspecto físico TRIAC TIC263	58
FIGURA 20 – Aspecto físico do optoacoplador MOC3063	59
FIGURA 21 – Exemplo de controle de potência	59
FIGURA 22 – Sequência de funcionamento do equipamento	61
FIGURA 23 – Fluxograma do processo	62
FIGURA 24 – Sistema de coordenadas para desenho de telas	63
FIGURA 25 – Tela inicial	65
FIGURA 26 – Tela inicial	67
FIGURA 27 – Tela de seleção de receitas	68
FIGURA 28 – Tela de seleção de receitas	68
FIGURA 29 – Editando a mosturação	69
FIGURA 30 – Editando a fervura	70
FIGURA 31 – Configurando o pré-aquecimento	70
FIGURA 32 – Aquecendo	71
FIGURA 33 – Pré-aquecimento concluído	72
FIGURA 34 – Executando a mosturação	72
FIGURA 35 – Gráfico da mosturação	73
FIGURA 36 – Mosturação concluída	74
FIGURA 37 – Clarificação	75

FIGURA 38 – Iniciando a fervura.....	75
FIGURA 39 – Tempo para adição de lúpulos.....	76
FIGURA 40 – Adicionar lúpulos.....	76
FIGURA 41 – Lupulagem concluída.....	77
FIGURA 42 – Receita concluída.....	77
FIGURA 43 – Projeto do <i>shield</i> principal.....	80
FIGURA 44 – <i>Shield</i> principal.....	80
FIGURA 45 – Caixa Light utilizada para montagem.....	81
FIGURA 46 – Esquema elétrico do painel de controle.....	82
FIGURA 47 – Testes antes da montagem.....	83
FIGURA 48 – Interior do painel elétrico.....	84
FIGURA 49 – Chapa lateral.....	84
FIGURA 50 – Interior do painel elétrico com vista lateral.....	85
FIGURA 51 – Vista frontal do painel elétrico fechado.....	86
FIGURA 52 – Início da fabricação de cerveja.....	87
FIGURA 53 – Malte queimado aderido à resistência.....	88
FIGURA 54 – Fabricação de cerveja.....	89
FIGURA 55 – Comparação: aspecto físico Arduino x CLP.....	95

LISTA DE QUADROS

QUADRO 1 – Características do Arduino Mega 2560.....	39
QUADRO 2 – Bibliotecas utilizadas no Arduino	40
QUADRO 3 – Requisitos do projeto.....	48
QUADRO 4 – Propriedades da resistência elétrica.....	50
QUADRO 5 – Ajuste de controlador PID	57
QUADRO 6 – Esquema elétrico.....	79
QUADRO 7 – Comparativo de custo: Arduino x CLP.....	96

LISTA DE SIGLAS, ABREVIATURAS E SÍMBOLOS

<u>Sigla / Abreviatura</u>	<u>Significado Original</u>	<u>Tradução</u>
A	Ampère	---
CA	Corrente alternada	---
CC	Corrente contínua	---
EEPROM	<i>Electrically- Erasable Programmable Read- Only Memory</i>	Memória de Leitura Programável Apagável Eletricamente
I/O	<i>Input / Output</i>	Entrada / Saída
IDE	<i>Integrated Development Environment</i>	Ambiente de Desenvolvimento Integrado
IHM	Interface homem máquina	---
KB	<i>Kilobyte</i>	---
Kg	<i>Kilogram</i>	Quilograma
KW	<i>Kilowatt</i>	Quilowatt
IFSC	Instituto Federal de Santa Catarina	
LCD	<i>Liquid Crystal Display</i>	Tela de Cristal Líquido
LED	<i>Light Emitting Diode</i>	Diodo Emissor de luz
mA	miliAmpère	---
MgO	Óxido de Magnésio	---
MHz	Mega-hertz	---
Nm	Newton por metro	---
PD	Proporcional derivativo	---
PI	Proporcional integral	---
PID	Proporcional integral derivativo	---

PWM	<i>Pulse Width Modulation</i>	Modulação por Largura de Pulso
RGB	<i>Red Green Blue</i>	Vermelho Verde Azul
RISC	<i>Reduced Instruction Set Computer</i>	Computador com Conjunto Reduzido de Instruções
RPM	Rotações por minuto	---
SCR	<i>Silicon controlled rectifier</i>	Retificadores Controlados de Silício
SRAM	<i>Static Random Access Memory</i>	Memória Estática de Acesso Aleatório
TRIAC	<i>Triode for Alternating Current</i>	Tríodo para Corrente Alternada
USB	<i>Universal Serial Bus</i>	Barramento Serial Universal
UTFT	<i>Universal Thin Film Transistor</i>	Transistor de Película Fina Universal
V	Volts	---
Vac	Volts em corrente alternada	---
Vcc	Volts em corrente contínua	---
W	Watt	---
W/cm ²	Watts por centímetro quadrado	---
Ω	Ohm	---

SUMÁRIO

1 INTRODUÇÃO	19
1.1 Justificativa	19
1.1 Objetivos	20
2 REVISÃO DE LITERATURA	22
2.1 Cerveja	22
2.1.1 História.....	22
2.1.2 Produção artesanal	24
2.1.3 Fabricação	24
2.1.4 Importância da temperatura na produção da cerveja	26
2.2 Automação.....	27
2.2.1 Automação na produção artesanal de cerveja	28
2.3 Controle automático	32
2.4 Sensor de temperatura.....	34
2.5 Aquecimento resistivo indireto	35
2.6 Controle de potência em corrente alternada	35
2.7 Arduino.....	37
2.8 Interface homem-máquina	40
2.8.1 Tela sensível ao toque	41
2.9 Fonte de alimentação	42
2.10 Motor de corrente contínua	43
2.11 Relé eletromecânico	44
3 DESENVOLVIMENTO	46
3.1 Concepção do projeto	46
3.1.1 Requisitos de projeto.....	47
3.2 Aquecedor do líquido	49
3.3 Controlador automático	52

3.3.1 Controlador Proporcional Integral Derivativo	53
3.3.2 Controle de potência CA	57
3.4 Programação do software.....	60
3.5 Interface Homem Máquina.....	67
3.6 Circuito de comando	78
3.7 Painel elétrico.....	81
4 RESULTADOS OBTIDOS	87
5 CONCLUSÕES	90
5.1 Desenvolvimento	90
5.1.1 Pesquisas na área da cerveja.....	91
5.1.2 Pesquisas na área de automação.....	91
5.1.3 Desenvolvimento do <i>software</i>	92
5.1.4 Desenvolvimento do <i>hardware</i>	94
5.2 Baixo custo de implementação.....	94
5.3 Discussão dos resultados	96
5.4 Oportunidades de melhoria.....	97
REFERÊNCIAS.....	99
APÊNDICE A – Dificuldades encontradas.....	103

1 INTRODUÇÃO

Atualmente, observa-se uma crescente busca na sofisticação de processos a fim de se obter um determinado produto da forma mais eficiente possível.

Com base nesta demanda introduz-se a automação, um processo em fluxo contínuo que associa vários métodos para obtenção de um produto com a mínima intervenção do trabalho humano. Muitos processos automatizados utilizam máquinas autorreguláveis, ou seja, máquinas que são capazes de interpretar a atual situação do sistema por meio de sensores e tomar a devida ação a fim de corrigir um eventual erro observado, representando uma alta performance, estabilidade e precisão para o processo produtivo.

Muitos processos industriais necessitam do constante monitoramento e controle de temperatura para se atingir os resultados esperados, sendo este de fundamental importância numa vasta gama de aplicações, abrangendo desde processos físicos e químicos até a proteção de equipamentos (FRANKLIN, 2009; BEGA, 2006). Um dos processos que necessita deste controle rigoroso é a fabricação de cerveja, mais precisamente em duas etapas iniciais, chamadas de mosturação e fervura, onde se trabalham com as temperaturas mais elevadas do processo, que ficam em faixas estreitas e precisas e por isso a importância de um controle perfeito no qual se devem utilizar equipamentos de medição condizentes com a precisão pretendida.

1.1 Justificativa

A ideia para o projeto em questão surgiu da necessidade de automatizar um processo de produção de cerveja artesanal, que até então era feito manualmente. Um fabricante artesanal, cuja produção é inteiramente manual, necessita estar presente em tempo integral durante os processos de mosturação e fervura, que fazem parte da brassagem do mosto cervejeiro, controlando a temperatura, observando um termômetro e operando um fogareiro. Este acompanhamento é necessário,

pois nesta parte controla-se a transformação do amido em açúcares, pelas enzimas do malte, que precisam de temperaturas específicas para realizar cada tipo de transformação. Além disso, o fabricante artesanal precisa acompanhar o tempo para adição de ingredientes, observando um relógio. Apenas os processos mencionados duram cerca de duas horas, tornando esta rotina extremamente cansativa.

1.1 Objetivos

Este projeto visa o desenvolvimento de um equipamento automatizado de brassagem do mosto cervejeiro. O equipamento consiste em um aquecedor de líquido, com controlador de temperatura temporizado, que pode ser programado facilmente pelo operador através de uma Interface Homem Máquina (IHM).

Os componentes do equipamento são: panela de contenção do mosto, resistência elétrica para aquecimento, misturador para homogeneização da temperatura e unidade de controle, esta última contemplando: IHM para configuração e monitoramento do processo, sensoriamento da temperatura e controle de acionamento do aquecedor e do misturador.

Este projeto visa desenvolver a unidade de controle e integrá-la aos demais componentes que serão fornecidos por um fabricante artesanal interessado na automação do processo mencionado. Nesta unidade será utilizada a plataforma programável Arduino para efetuar o processamento das informações do sistema e controle das variáveis desejadas.

O método utilizado para atingir e manter a temperatura conforme as pré-definições será o controle com realimentação. Este tipo de controle refere-se a uma operação que, na presença de distúrbios, tende a diminuir a diferença entre a saída de um sistema e alguma entrada de referência (OGATA, 2003). Uma das maneiras de viabilizar este controle é através do conhecimento da planta, a fim de se obter as equações que a representam, para depois elaborar uma equação que consiga corrigir o erro com ação Proporcional-Integral-Derivativa (PID) (MONTEIRO, 2006; NISE, 2009)

Desta forma um sensor digital fará a medição da temperatura do líquido. A leitura deste sensor será

constantemente comparada, na unidade de processamento, com a temperatura pré-definida, computando uma ação de controle. A resistência elétrica aquecerá o líquido, com potência definida por esta ação de controle, para se atingir a temperatura pré-definida. O misturador será acionado para homogeneizar a temperatura nos momentos de aquecimento do líquido.

A IHM será composta por uma tela *touchscreen*, onde o operador irá configurar a receita que será executada no sistema, determinando os períodos de tempo e temperatura em cada um deles. Esta interface irá mostrar os valores de temperatura atuais e tempo decorrido, além de apresentar alarmes personalizados para o processo em que o sistema será usado.

Toda a eletrônica de controle será desenvolvida no projeto (sem a utilização de controladores comerciais), sendo uma das grandes contribuições deste trabalho. O projeto deverá possuir um bom acabamento, resistente às condições do ambiente onde provavelmente estará exposto a umidade e resíduos do processo.

2 REVISÃO DE LITERATURA

Este capítulo apresenta os aspectos já conhecidos sobre a fabricação de cerveja e técnicas de automação que, juntamente com os componentes eletrônicos disponíveis, contribuíram para a implementação do projeto proposto. Além disto, também são apresentados os equipamentos já encontrados no mercado cujos objetivos são correlatos a este trabalho.

2.1 Cerveja

Cerveja é uma bebida alcoólica carbonatada produzida a partir da fermentação de cereais, principalmente a cevada maltada. É a terceira bebida mais popular do mundo, ficando atrás apenas da água e do chá. Na categoria das bebidas alcoólicas, esta é a mais consumida atualmente.

2.1.1 História

Segundo Morado (2009), registros encontrados em desenhos rupestres indicam que uma bebida semelhante à cerveja já era produzida antes mesmo da invenção da escrita. Especialistas afirmam que a cerveja teve papel importante na origem e estabelecimento da sociedade civilizada, por volta de 6000 a.C. Peças arqueológicas de 4000 a.C. apresentam uma bebida semelhante, chamada *sikaru*, utilizada pelos sumérios como remédio, moeda e oferendas. O mais antigo registro sobre cervejarias tem origem no Egito em 3400 a.C., sendo o primeiro grande centro produtor de cerveja a cidade de Pelesium, atual Port-Said. Chineses também dominavam a produção de uma bebida chamada *tsiou* por volta de 2000 a.C. Regras para produção de cerveja foram encontradas no Código de Hamurábi, escrita pelo Império Mesopotâmico em 1730 a.C.

Através de movimentos migratórios, a técnica para produção de cerveja chegou à Europa, primeiramente aos povos grego e romano. Logo se tornou uma popular alternativa ao

vinho, que possuía uma forma produtiva muito delicada e de alto custo. Plínio, o Velho, em sua obra *Naturalis Historia* publicada por volta de 78 d.C., mencionou a produção de cerveja nos territórios hoje pertencentes a França e Espanha. Foi nesta região que a bebida obteve sua etimologia, nomeada de *cerevisia* ou *cervisia* em homenagem a Ceres, deusa da colheita e fertilidade (MORADO, 2009).

Conforme Morado (2009), a partir da Idade Média, por volta do século VI, a produção de cerveja em larga escala desenvolveu-se principalmente em mosteiros europeus. Nesta época, estes eram os locais detentores do conhecimento, com capacidade para desenvolver e registrar novas receitas. Assim, a história da cerveja esteve intrinsecamente ligada à Igreja, pois, por pertencerem à elite alfabetizada, os religiosos foram os pioneiros nas pesquisas sobre o assunto. Como exemplo da inovação implementada nesta época, foram estes que introduziram a técnica de conservação a frio da bebida.

Com a urbanização observada após o século XII, algumas cervejarias comerciais começaram a surgir pela Europa, sendo pioneiras as cidades alemãs de Frankfurt e Munique. (MORADO, 2009).

Os séculos seguintes foram de notáveis desenvolvimentos nos processos produtivos, como nas etapas de fermentação e secagem de malte. Nas palavras de Morado (2009, p.46) “A cerveja, como é conhecida hoje, [...] é resultado do desenvolvimento científico e industrial ocorrido no século XIX”.

Segundo Morado (2009), no Brasil, a cerveja teve sua primeira aparição no século XVII, trazida pela Companhia Holandesa das Índias Orientais. Porém, em 1654, os holandeses deixaram o país, levando consigo o hábito pela bebida. O produto retornou apenas em 1808 com a chegada da Família Real portuguesa à colônia. Até os anos de 1870 a cerveja inglesa monopolizou o mercado brasileiro devido à influência comercial da Inglaterra sobre Portugal. Porém, com o aumento dos impostos sobre produtos importados esta prática tornou-se inviável. Até então, a produção no Brasil era feita de forma artesanal e obstada pela falta de cevada e lúpulo, ambos importados, e pela dificuldade de refrigeração do produto.

Em meados do século XIX começam a surgir pequenas cervejarias nas regiões sul e sudeste. Algumas alcançam escala industrial de produção até o final deste mesmo século, surgindo a

partir de então marcas consagradas nos dias atuais. (MORADO, 2009).

2.1.2 Produção artesanal

Após a era da industrialização, a cerveja voltou a ser fabricada artesanalmente a partir do ano de 1979, primeiramente nos Estados Unidos, quando o presidente americano Jimmy Carter anulou as leis que restringiam a produção artesanal da bebida.

Nas décadas seguintes, o *hobby* pela produção da cerveja se espalhou pelos Estados Unidos e Canadá, chegando posteriormente ao Brasil.

Apesar da produção limitada, a fabricação artesanal proporciona bebidas especiais com aroma e sabores diferenciados, sendo esta sua principal característica e contribuição para o mundo da cerveja.

2.1.3 Fabricação

Segundo Morado (2009), produzir cerveja requer três processos básicos: mostura, fervura e fermentação. Os dois primeiros geralmente ocorrem no mesmo dia. Já a fermentação pode durar até duas semanas. A FIGURA 1 ilustra o fluxograma típico da produção de cerveja.

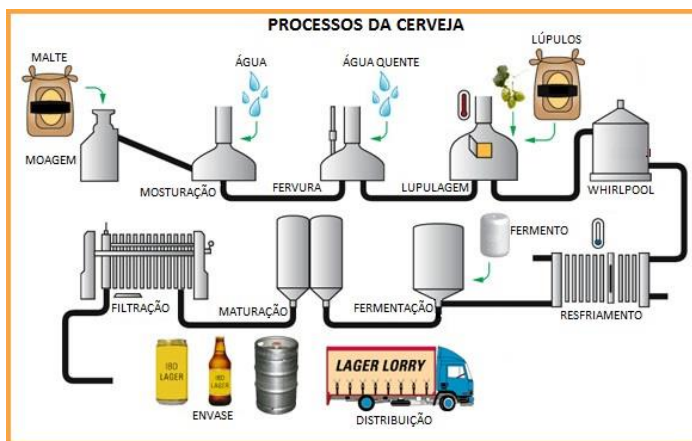


FIGURA 1 – Processos da cerveja

Adaptado de <http://people.cs.pitt.edu/~mehmud/cs134-2084/projects/Team4/brewing.html>. Acesso em: 09 jun. 2015.

Os processos iniciais para produção da cerveja chamam-se brassagem, ou fabricação de mosto, que consiste no cozimento do malte para extrair os açúcares necessários para fermentação.

A moagem do malte tem por função quebrar o grão a fim de expor o amido contido no seu interior. Ao malte moído é adicionado água e esta solução é então submetida a diferentes temperaturas por períodos de tempo determinados, constituindo o processo chamado de mosturação. Em seguida, no processo de clarificação, o mosto é filtrado para remoção do bagaço de malte e então o líquido é fervido.

A fervura é uma das etapas determinantes da qualidade da cerveja produzida. É nesta etapa que ocorre a esterilização do malte e definição do aroma e sabor da bebida, também como resultado da adição do lúpulo à solução.

A etapa seguinte é o resfriamento, necessário para que o mosto atinja a temperatura correta para fermentação. Este processo, também conhecido como *whirlpool*, deve ocorrer de forma rápida, para que sejam evitados a formação de aromas indesejáveis e contaminação da solução. Em seguida o mosto é aerado a fim de fornecer o oxigênio necessário para multiplicação celular.

Na fermentação, a levedura é adicionada ao mosto para que ocorra a transformação dos açúcares em dióxido de carbono e etanol. A duração deste processo pode variar entre algumas horas a alguns dias, dependendo do fermento e receita aplicada.

Após a retirada da levedura que ficou decantada ao final do ciclo de fermentação, ocorre o processo de maturação. Esta etapa é uma fermentação lenta e complementar onde acontecem reações físico-químicas que irão modificar e definir o aroma e sabor da bebida. Considerada como o “afinamento” da cerveja, este processo dura em média de 10 dias.

Em seguida é feita a filtração, com o objetivo de remover resíduos de levedura do líquido. Concluído estes procedimentos, pode ocorrer o processo de carbonatação, que consiste em injetar gás carbônico na cerveja caso os níveis do gás tenham ficado inferiores aos esperados. Ao final destes processos, a cerveja está pronta para ser engarrafada.

2.1.4 Importância da temperatura na produção da cerveja

Durante a produção da cerveja, muitos fatores influenciam no rendimento da receita e na qualidade do produto final. Dentre eles, o controle do diagrama tempo/temperatura no processo de mosturação é de suma importância (TSCHOPE, 2001).

A mostura destina-se prioritariamente à ativação de enzimas que farão a conversão do amido do malte em açúcares, em razões variáveis de acordo com o perfil da cerveja desejada. Para isso, cada enzima do malte atua melhor em determinada faixa de temperatura. Existem ainda rampas de temperatura que podem ser utilizadas para ativar enzimas específicas, conferindo características particulares à cerveja. Variações na temperatura ou no tempo de exposição podem comprometer alguns ciclos enzimáticos, gerando sabores desagradáveis na cerveja (GOMES, 2013; COSTA, 2014). Temperaturas acima de 74°C ocasionam a desativação das enzimas, comprometendo o rendimento e o processo.

Durante o processo de fervura, os tempos das receitas também devem ser cuidadosamente controlados, vistos que são estes os fatores responsáveis pelo aroma e sabor da bebida.

2.2 Automação

Segundo Moraes (2013) a palavra *automation* foi inventada pelo *marketing* da indústria de equipamentos na década de 1960, sendo posteriormente traduzido para automação. O termo buscava enfatizar a participação do computador no controle automático industrial.

A automação teve início com a necessidade da modernização dos processos produtivos das indústrias, visando a realização de sistemas otimizados capazes de produzir bens com menor custo, com maior quantidade, em menor tempo e com maior qualidade (ROSÁRIO, 2009).

Portanto, é comum pensar que a automação resulta tão-somente do objetivo de reduzir custos de produção para que as empresas se tornem mais competitivas. Contudo isto não é verdade, pois ela decorre mais de necessidades tais como maior nível de qualidade, maior flexibilidade de modelos para o mercado, maior segurança pública e dos operários, menores perdas de materiais e de energia, mais disponibilidade e qualidade da informação sobre o processo e melhor planejamento e controle da produção. Para que se tenha esta qualidade na informação, a automação compreende também a implantação de sistemas interligados e assistidos, sendo estes os sistemas supervisórios e as interfaces homem-máquina (IHM). (MORAES, 2013)

Nas palavras de Silveira (1998, p.23)

A automação é um conceito e um conjunto de técnicas por meio das quais se constroem sistemas ativos capazes de atuar com uma eficiência ótima pelo uso de informações recebidas do meio sobre o qual atuam.

Desta forma a automação de um sistema busca o controle de variáveis do processo para atender a determinações conferidas ao sistema pelo ser humano.

Para isto a automação envolve contribuições de duas especialidades de controle: o controle dinâmico e o controle lógico. A especialidade de controle dinâmico (controle acionado pelo tempo) busca projetar as malhas de realimentação capazes

de manter as condições operacionais nos valores eficientes, elaborando um algoritmo de controle que aperfeiçoe a eficiência e ainda simule o conjunto em computador. Já a especialidade de controle lógico (controle acionado por eventos) busca implementar regras desejadas para os eventos discretos, além de considerar os níveis de segurança para os componentes e para as pessoas assim como os requisitos de monitoração, alarme e intervenção humana e os relatórios gerenciais do processo. Portanto é preciso selecionar os equipamentos de computação e depois programá-los, testá-los e acompanhar o desempenho no *start-up* (MORAES, 2013).

2.2.1 Automação na produção artesanal de cerveja

Com a popularização da fabricação artesanal de cerveja, equipamentos automáticos, que executam o processo de brassagem do mosto cervejeiro, começaram a ser desenvolvidos e comercializados entre os fabricantes artesanais. Estes equipamentos unem as tecnologias da automação, desenvolvidas para indústria, ao trabalho artesanal, buscando além da facilidade no trabalho, maior precisão na repetição das receitas.

Existem no mercado equipamentos automáticos destinados à microcervejarias artesanais, onde a capacidade de produção é a partir de 100 litros, denominado linha profissional, e aos cervejeiros caseiros, onde a capacidade de produção varia de 20 a 60 litros, denominado de linha hobby.

Existem dois tipos de equipamentos automáticos para brassagem: os convencionais e os sistemas *single vessel*¹.

Os equipamentos convencionais seguem a sequência de fabricação de cerveja conforme apresentado no tópico 2.1.3. Estes possuem um tanque próprio para a mosturação, outro para aquecer a água utilizada na clarificação, e mais um para fervura. A FIGURA 2 apresenta um exemplo de equipamento de

¹ *Single vessel*: tanque único

produção convencional da marca Jhonas Bier modelo Micro Cervejaria Profissional 100 litros.



FIGURA 2 – Micro Cervejaria Profissional 100 litros

Fonte: <http://jhonasbier.webnode.com.br/products/cervejaria-de-100-litros-jb/>. Acesso em: 08 jun. 2015.

Nos sistemas *single vessel*, a brassagem é feita em um único tanque, característica que atribui o nome do sistema, com a diferença de que este tanque possui um tubo interno removível. Neste tipo de equipamento o mosto recircula do tanque interno (tubo), onde o malte fica contido por telas, para o tanque externo através da ação de uma bomba. A agitação do mosto para equilibrar a temperatura é feita pela ação da recirculação.

A FIGURA 3 apresenta um esboço em corte de um equipamento com sistema *single vessel*, feito através de análises do mesmo, para explicar o funcionamento durante o processo de mosturação:

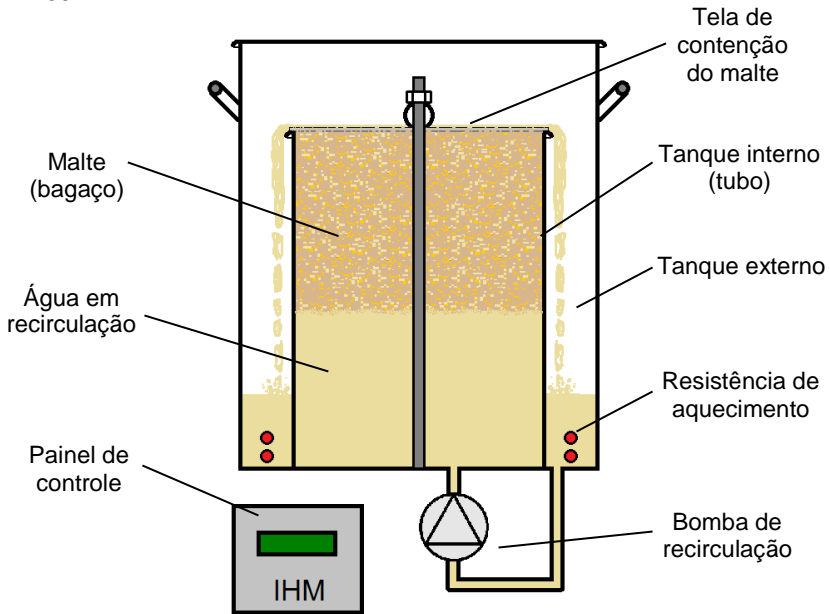


FIGURA 3 – Sistema *single vessel*

Nestes equipamentos o tanque interno, juntamente com bagaço, é retirado após a conclusão da mosturação, deixando apenas o líquido no tanque para ser feita a fervura.

A maioria dos equipamentos de brassagem automática da linha hobby são sistemas *single vessel*, conforme modelo apresentado acima. Porém, cabe ressaltar que a proposta deste projeto é o desenvolvimento de um equipamento linha hobby do tipo convencional.

Um dos equipamentos automáticos com sistema *single vessel* que se destaca no mercado nacional é o BeerBot 60. A FIGURA 4 apresenta o aspecto físico do equipamento.



FIGURA 4 – Equipamento BeerBot 60

Fonte: <http://www.cervejantes.com.br/panela-eletrica-beerbot-50>
(acesso em 01/06/2015)

Na imagem é possível observar que o tanque e toda a mecânica foram fabricados especificamente para este equipamento para possibilitar a brassagem com uma só panela. Verifica-se nesta imagem, e também nos outros equipamentos similares, que a IHM para programar as receitas, fica abaixo do tanque, na base do equipamento. Esta posição prejudica a ergonomia do equipamento, pois é preciso se abaixar para configurá-lo. Como é necessário retirar o tubo interno, contendo o malte, durante o processo, o equipamento não pode ficar em altura muito elevada.

A grande vantagem deste e de outros equipamentos com sistema *single vessel*, além da automação do processo, é a economia de espaço, uma vez que não é preciso uma panela para mosturação, outra para esquentar a água para a clarificação e eventualmente uma outra para a fervura. Porém esta vantagem pode se tornar uma desvantagem se o cervejeiro pretende fazer, de uma só vez, a fervura de mais de um lote em produção. Outro ponto que pode se tornar uma desvantagem, é o de um cervejeiro já possuir todas as panelas e estar procurando apenas

pela automação do processo, se assim adquirir este equipamento, terá que inutilizar as painéis.

2.3 Controle automático

O controle automático de um processo tem por objetivo manter a saída de um sistema dentro de uma faixa de valores ou em um valor constante pré-determinado.

Esta técnica, segundo Ogata (2003), originou-se durante a Revolução Industrial, quando o matemático e engenheiro James Watt implementou um controle de velocidade para uma máquina a vapor. Em 1922, Nicolas Minorsky, engenheiro russo radicado nos Estados Unidos, determinou a estabilidade para o seu controle de direcionamento marítimo a partir de equações diferenciais. Em 1932, Harry Nyquist desenvolveu importante trabalho para a estabilidade de sistemas de malha fechada na empresa americana de telecomunicações Bell Labs. A partir de então este ramo da engenharia obteve grande desenvolvimento, sendo amplamente aplicada nos dias atuais em áreas como biologia, biomedicina e economia, além da própria engenharia.

O controle automático tem papel fundamental nos processos industriais e de produção, sendo essencial em operações como controle de pressão, umidade, viscosidade, vazão e temperatura. A implementação de um sistema de controle automático pode otimizar o desempenho de sistemas dinâmicos, melhorar a produtividade e diminuir o trabalho árduo de várias rotinas manuais repetitivas (OGATA, 2003).

Em um sistema típico, a grandeza física que necessita ser controlada recebe o nome de variável de processo, cujo valor estabelecido para o processo é denominado *setpoint*. O sistema de controle utiliza um sensor para medir a variável de processo e receber uma resposta, assim, ao comparar com o *setpoint*, irá determinar o sinal de controle para o atuador. Este processo denomina-se sistema de controle em malha fechada e a maneira pela qual o controlador automático produz o sinal de controle é chamada de ação de controle.

Segundo Ogata (2003), os controladores automáticos podem ser classificados de acordo com suas ações de controle em: controladores on-off, proporcionais, integrais, proporcional-

integrais (PI), proporcional-derivativos (PD) e proporcional-integral-derivativos (PID).

Normalmente, na necessidade de um controle preciso, utiliza-se uma ação PID. Estes controladores utilizam as três ações de controle (proporcional, integral, derivativo) simultaneamente para atingir resultados mais satisfatórios.

A ação proporcional determina a taxa de saída do sistema tendo por base apenas a diferença entre a variável de processo e o *setpoint*, diferença esta chamada de erro ou *offset*. Para Dias (2012, p.3) “[...] a saída de um controlador proporcional pode assumir qualquer valor desde que compreendido entre os limites de saída máxima e mínima, em função do *offset* verificado.” O aumento do ganho proporcional resulta em uma maior velocidade de resposta do sistema de controle. Porém, um aumento desordenado provocará oscilações e tornará o sistema instável.

A ação integral soma o valor do erro ao longo do tempo, ou seja, a velocidade da correção no sinal de saída é proporcional à amplitude do desvio. Mesmo um pequeno erro fará com que a componente integral aumente lentamente. Assim, o termo de natureza integral pode reduzir ou eliminar os erros estacionários, porém este processo torna o controle lento, haja vista o elevado tempo de transição.

A ação derivativa tem por função melhorar o desempenho do sistema, atuando apenas nos momentos em que há erro e tornando-se nula quando o processo está estabilizado. Aumentar este termo torna o sistema mais sensível às mudanças no parâmetro de erro, conferindo maior velocidade de resposta (DIAS, 2012; NATIONAL INSTRUMENTS, 2014).

Em síntese, um sistema PID reúne o controle básico da ação proporcional, a redução do erro devido à ação integrativa e aceleração da saída com redução de oscilações pela ação derivativa. Este controlador oferece boa relação entre precisão e velocidade, sendo muito aplicado em plantas térmicas.

A companhia National Instruments (2014) expressa em resumo a operação do controlador citado

A ideia básica por trás de um controlador PID é ler um sensor, calcular a resposta de saída do atuador através do cálculo proporcional, integral e derivativo e então somar os três componentes para calcular a saída.

Os controladores PID são largamente utilizados em sistemas que não se conseguem aplicar métodos de projeto analítico, por não se conhecer o modelo matemático da planta. Para estes casos, quando o sistema precisa ser ajustado em campo, diferentes tipos de regras de sintonia podem ser encontrados na literatura, permitindo ajustes finos no controlador.

2.4 Sensor de temperatura

Sistemas que necessitam de um acompanhamento de temperatura rigoroso requerem sensores apropriados que façam leituras precisas. Nesta área, o sensor DS18B20 se apresenta como uma boa opção custo/benefício. Este termômetro digital possui resolução ajustável de 9 a 12 bits, escala de -55°C a 125°C com precisão de $\pm 0,5^{\circ}\text{C}$ na faixa que compreende de 10°C a 85°C , faixa de alimentação de 3V a 5,5V, conta com opção de alarmes personalizáveis e converte diretamente a temperatura para a forma digital (ARAÚJO, 2012). Na FIGURA 5 observa-se o sensor DS18B20 e seu encapsulamento imersível.



FIGURA 5 – Sensor de temperatura DS18B20 imersível

Fonte: www.elecrow.com/wiki/index.php?title=One_Wire_Waterproof_Temperature_Sensor. Acesso em: 06 out. 2014.

Este dispositivo se comunica com o microprocessador através do protocolo 1-Wire, ou seja, necessita de apenas uma linha para transmissão de dados. Este sistema, desenvolvido pela empresa Dallas Semiconductor, é amplamente utilizado em situações que possam existir ruídos elétricos, casos onde sensores analógicos não se mostram eficientes. Uma característica notável desta tecnologia é a baixa corrente de

alimentação necessária, podendo esta ser provida a partir da própria linha de dados.

2.5 Aquecimento resistivo indireto

O aquecimento resistivo utiliza o calor gerado por efeito Joule numa resistência que é atravessada por uma corrente elétrica. No aquecimento resistivo indireto, o material a ser aquecido está contido em uma câmara e o calor é transferido da resistência elétrica através dos fenômenos de condução, convecção e irradiação (MANEDE FILHO, 2011).

2.6 Controle de potência em corrente alternada

Para se aplicar um percentual de potência determinado à uma carga em corrente alternada (CA) ou mesmo acioná-la com um comando, é necessário um método de controle de potência. Segundo Ahmed (2000), há dois métodos básicos para o controle de potência da carga: o de ciclo integral ou liga-desliga, e o de fase.

O controle de fase é indicado para cargas com constantes de tempo pequenas, encontradas nas aplicações como controle de iluminação e controle de velocidade para motores. Já o controle de ciclo integral serve para sistemas com uma constante de tempo grande, como os de controle de temperatura.

Portanto, este último foi selecionado para ser implementado no projeto. Neste método, a potência na carga pode ser controlada com a ligação e o desligamento, da fonte de tensão à carga, por alguns ciclos completos de condução de corrente e depois com a repetição do chaveamento. A duração relativa dos períodos nos estados ligado e desligado, isto é, o ciclo de trabalho, é ajustado de tal modo que a potência média entregue à carga atenda a algum objetivo particular. A FIGURA 6 ilustra o funcionamento deste método.

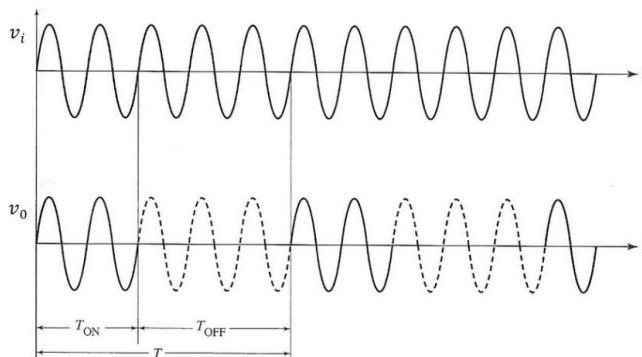


FIGURA 6 – Controle potência de ciclo integral
Fonte: AHMED, 2000, p.405

Na imagem apresentada, T_{ON} é o tempo de condução: a chave está ligada e a potência na carga é máxima; T_{OFF} é o tempo de extinção: a chave está desligada e a potência na carga é nula; T é o número de ciclos no período completo de operação.

Segundo Ahmed (2000), a aplicação deste método de controle, em que há longos tempos de desligamentos, é aceitável em sistemas de aquecimento como fornos elétricos e tanques de aquecimento de líquido, pois a resposta da carga é relativamente longa (segundos, e não frações de segundo). Portanto, a carga responderá bem a potência média aplicada.

Uma das formas de executar este controle de potência CA é através do uso de um retificador controlado de silício (SCR) bidirecional, como um TRIAC. Este dispositivo é um tipo de tiristor (semicondutor de potência) que possui três terminais: principal 1, principal 2 e porta. O TRIAC é capaz de conduzir corrente em ambas as direções, direta e inversa, e pode ser controlado por um sinal na porta, positivo ou negativo. Isso o torna útil para o controle de potência CA (AHMED, 2000).

Algumas limitações do dispositivo são a baixa capacidade de regulação de potência, se comparado aos SCRs e a baixa velocidade de chaveamento, que restringe a frequência operacional a algumas centenas de hertz. Os TRIACs são, portanto usados apenas para regular a tensão CA em torno dos 60Hz em aplicações como iluminação, controles de velocidade de motores e de aquecimento e em relés CA de estado sólido (AHMED, 2000).

O circuito que gera o pulso de porta, o qual aciona o dispositivo tiristor, deve ser protegido contra transitórios de tensão induzidos. É necessário separar este circuito da fiação de força, o tanto quanto possível. O isolamento elétrico entre o dispositivo tiristor e o circuito de porta costuma ser feito por um transformador de pulso ou por um acoplador óptico.

Neste projeto utilizou-se o método de isolamento com acoplador ótico. Segundo Ahmed (2000 p. 125)

Os acopladores ópticos de uso mais comum consistem na combinação de um diodo emissor de luz (light-emitting diode – LED) e de um fototransistor, montados em um invólucro único. Este arranjo permite o acoplamento do sinal de um circuito para outro, ao mesmo tempo que fornece um isolamento elétrico quase completo entre os dois circuitos.

Portanto, a proteção do circuito de porta de um TRIAC, feito com acoplador ótico, garante a confiabilidade necessária para o controle de uma carga CA feito com o uso dessa chave.

2.7 Arduino

O Arduino é uma plataforma que pode interagir com o ambiente através do processamento de entradas e saídas entre o dispositivo e os componentes externos conectados a ele (MCROBERTS, 2011). Trata-se de um microcontrolador acessível, tanto pelo baixo custo bem como pela facilidade de utilização devido ao código livre, possibilitando a aplicação em diversas operações.

Microcontroladores são componentes eletrônicos utilizados no controle de periféricos, tais como: LEDs, botões, *displays*, resistências, relés, sensores diversos (pressão, temperatura, etc), dentre outros. Com base no estado destes periféricos, o microcontrolador executa ações lógicas conforme a programação gravada pelo usuário. São componentes capazes de controlar todo um processo, pois, em um único elemento, possui memória de programa, memória de dados, portas de entrada e/ou saída

paralelas, *timers*, contadores, comunicação serial, PWMs, conversores analógico-digitais, etc. (SOUZA, 2008)

O primeiro Arduino surgiu em 2005 a partir da ideia de professores de Computação Física do curso de Arte e Design em uma cidade italiana. O objetivo principal era desenvolver uma ferramenta única de *hardware*, facilmente programável e de baixo custo. Com a colaboração de especialistas da área, desenvolveu-se a interface de programação em código aberto (SILVEIRA, 2011).

A placa Arduino Mega 2560 possui um processador ATmega2560, baseado na família de microprocessadores Atmel AVR RISC (computador com conjunto reduzido de instruções, em inglês) com memória *flash* de 256 KB. Este modelo possui 54 pinos digitais de entrada-saída, 16 pinos de entrada analógica, oscilador de cristal de 16 MHz, *chipset* ATmega16U2 com comunicação serial USB e fonte de alimentação (EVANS, 2013).

Na FIGURA 7 pode-se observar a placa com a localização dos seus elementos constituintes, destacando-se os pinos de entrada/saída utilizados para leitura de sensores e acionamento dos componentes externos.

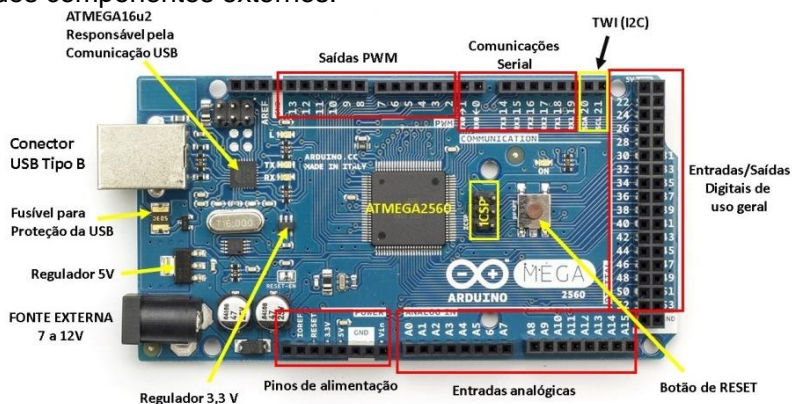


FIGURA 7 – Componentes do Arduino Mega 2560

Fonte: <http://www.embarcados.com.br/arduino-mega-2560/>. Acesso em: 30 set. 2014.

O modelo Mega se sobressai frente aos outros devido sua maior capacidade de memória, ideal para aplicações que necessitarão grande quantidade de informações, como Interfaces

Homem-Máquina (IHMs). No QUADRO 1 são apresentadas as características do Arduino Mega 2560.

QUADRO 1 – Características do Arduino Mega 2560

Microcontrolador	ATmega2560
Tensão de operação	5V
Tensão de entrada (recomendada)	7 - 12V
Tensão de entrada (limites)	6 - 20V
Pinos de entrada/saída digitais	54 (15 podem ser saídas PWM)
Pinos de entradas analógicas	16
Corrente CC por pino (I/O)	40mA
Corrente CC para pino de 3,3V	50mA
Memória <i>Flash</i>	256 KB (8 Kb para <i>bootloader</i>)
SRAM	8 KB
EEPROM	4 KB
Velocidade de <i>clock</i>	16 MHz

Adaptado de www.arduino.cc. Acesso em: 30 set. 2014.

A programação do Arduino é feita no IDE (Ambiente de Desenvolvimento Integrado, em inglês), *software* livre onde são escritos os códigos modelados a partir da linguagem *Processing*. Silveira (2011, p.42) define IDE como “[...] conjunto de aplicativos acessíveis por uma única interface gráfica, uma só tela principal, para a edição, depuração, compilação e gravação de programas [...]”. Ao fazer o *upload* para a placa, o código escrito é traduzido para a linguagem C e transmitido para o compilador (BANZI, 2011). Em suma, o programa escrito no IDE do Arduino é baseado na linguagem C, que ao ser transmitido para a placa fica salvo em sua memória *flash*.

Assim como em outras plataformas, o Arduino também admite o uso de bibliotecas nos códigos a serem desenvolvidos. Por definição, uma biblioteca em linguagem de programação é um conjunto de funções já desenvolvido e disponível para o uso em qualquer aplicação, sem a necessidade de inserir no programa grandes trechos de código. No QUADRO 2 abaixo

apresenta-se as bibliotecas utilizadas neste projeto com suas respectivas funções.

QUADRO 2 – Bibliotecas utilizadas no Arduino

Biblioteca	Função
UTFT	Desenvolvimento de <i>layouts</i> e interfaces para o <i>display</i> LCD
Utouch	Utilização da tela sensível ao toque
TimerOne	Códigos de temporizadores e contadores
OneWire	Leitura do termômetro digital
PID	Cálculos para o controlador
EEPROM	Inserção e leitura de dados na memória não-volátil do Arduino
avr/pgmspace	Utilização de imagens no formato <i>bitmap</i>

O Arduino pode expandir seu *hardware* com o uso de *shields*, placas de circuito que são conectadas a ele, adicionando novas funções e possibilitando uma infinidade de aplicações. Existem *shields* para acoplamento de sensores, controle de motores, módulos de comunicação, *displays*, dentre outros.

2.8 Interface homem-máquina

As interfaces homem-máquina, também conhecidas pela sigla IHM, são equipamentos que possibilitam a interação entre o usuário e a máquina, seja para intervir em processos ou apenas para visualização de informações, de dados, de controle e de comandos. Trata-se de um *hardware* industrial composto normalmente por uma tela de cristal líquido e um conjunto de teclas para navegação e/ou inserção de dados, baseado na programação escrita pelo desenvolvedor.

Cybis (2003) lista as funções que uma IHM pode desempenhar.

[...] solicita e recebe as entradas de dados, de controles e de comandos. Finalmente, ela

controla o diálogo entre as apresentações e as entradas. Uma interface tanto define as estratégias para a realização da tarefa, como conduz, orienta, recepciona, alerta, ajuda e responde ao usuário durante as interações.

2.8.1 Tela sensível ao toque

As telas sensíveis ao toque, do inglês *touch-screen*, são *displays* sensíveis à pressão exercida sobre sua superfície, realizando operações de acordo com os botões programados, dispensando assim a necessidade de teclados ou botões físicos (COSTA, 2011).

O primeiro registro de dispositivos sensíveis ao toque data do ano de 1965, um equipamento britânico desenvolvido para uso em radares de controle de tráfego aéreo. Telas com maior precisão surgiram a partir da década de 70, com o intuito de organizar dados na área da física nuclear (CIRIACO, 2014).

Estes dispositivos estão muito difundidos em indústrias de processos devido às vantagens apresentadas. Como características citam-se: agilidade, facilidade de operação, alto grau de personalização e configuração dos comandos (MUNHOZ, 2011; COSTA, 2011).

Uma tela sensível ao toque pode ser controlada através de uma plataforma Arduino. Para tanto, é necessário um *shield* específico para esta aplicação, possibilitando a conexão da tela com a placa.

Neste projeto utilizou-se uma tela do fabricante Itead Studio, modelo ITDB02 de 3,2 polegadas. Este dispositivo possui resolução de 320 x 240 *pixels* com 65000 cores e suporte para cartão de memória. Na FIGURA 8 abaixo observa-se a tela citada já conectada à placa Arduino Mega com o *shield* correspondente.

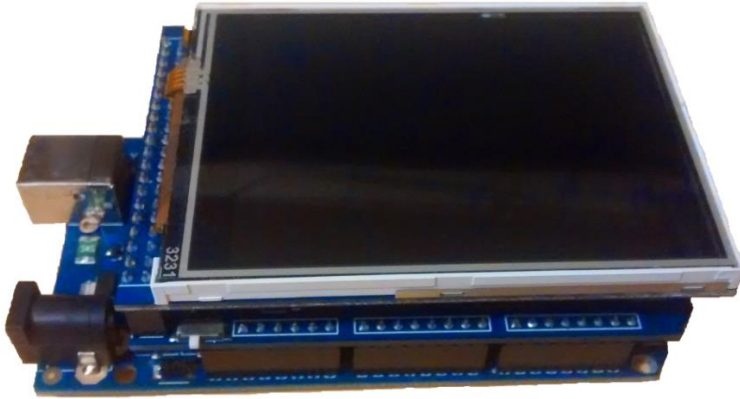


FIGURA 8 – Tela sensível ao toque

2.9 Fonte de alimentação

Das fontes de alimentação provém a energia elétrica adequada que os circuitos eletrônicos necessitam. Braga (2005b, p.7), define fonte de alimentação como “[...] um circuito que a partir da tensão elétrica disponível (alternada ou contínua) fornece a tensão contínua (ou mesmo alternada) na forma requerida pelo circuito alimentado.”

Os componentes deste projeto requerem tensões contínuas na faixa dos 12 Volts, assim, como esta forma de energia elétrica não está disponível em uma tomada comum da rede, torna-se necessário a utilização de uma fonte que converta a tensão alternada de 220 Volts para tensão contínua de 12 Volts.

Fontes chaveadas apresentam-se como uma boa opção para este caso. Tal topologia possibilita circuitos de alto rendimento, além de tamanho e peso reduzido devido à operação em alta frequência (BRAGA, 2005b).

Este tipo de fonte controla a tensão em uma carga através de um circuito comutador, alternando o tempo de abertura e fechamento deste circuito a fim de manter a tensão de saída no valor desejado.

Optou-se pela aquisição de um modelo comercial de fonte de alimentação, pelo fato de não ser este o foco do projeto. A FIGURA 9 apresenta o modelo utilizado, cuja conversão da tensão alternada de 220 Volts para 12 Volts contínua oferece corrente máxima de 10 Amperes.



FIGURA 9 – Fonte chaveada 12 Volts 10 Amperes

Fonte: http://produto.mercadolivre.com.br/MLB-635066568-fonte-chaveada-12v-10a-120w-p-cftv-fita-led-som-automotivo-_JM. Acesso em: 05 mar. 2015.

2.10 Motor de corrente contínua

Motores CC são utilizados para conversão de energia elétrica em movimentos mecânicos. Basicamente, são compostos por duas partes principais, estator e rotor, sendo que seu princípio de funcionamento se deve a forças magnéticas de atração e repulsão geradas nos seus componentes (BRAGA, 2005a).

Este motor pode ser utilizado em diversas aplicações, sendo encontrados no mercado com características variadas. Porém, quando o projeto pretendido requer um torque mais

elevado com menor rotação deve-se aplicar um motor CC com caixa de redução.

Neste projeto, um motor de corrente contínua com caixa de redução foi empregado como um misturador, cujo objetivo é homogeneizar a temperatura do líquido nos momentos de aquecimento. O modelo escolhido foi o CHP 9390082064 do fabricante Bosch, largamente utilizado em limpadores de para-brisa automotivos, pois suas características como torque e rotação atendem perfeitamente aos requisitos deste trabalho.

A FIGURA 10 a seguir apresenta o motor utilizado e suas características de funcionamento.

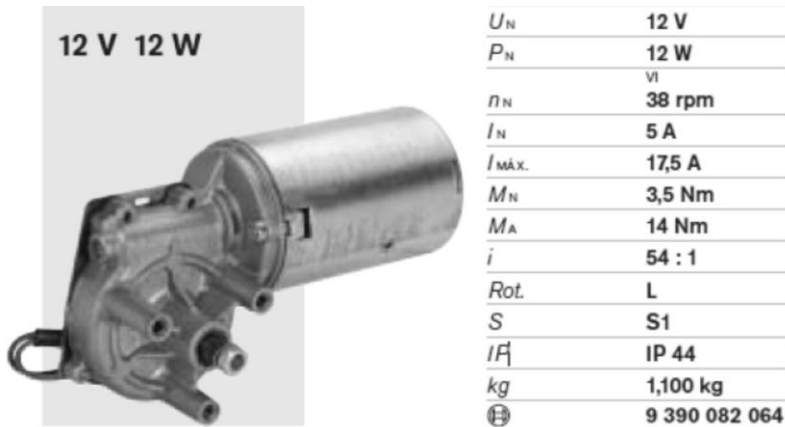


FIGURA 10 – Motor CC Bosch modelo CHP 9390082064
 Fonte: http://www.kalatec.com.br/Motor_Bosch/9390_082_064.pdf.
 Acesso em: 05 mar. 2015.

2.11 Relé eletromecânico

Relés são componentes que atuam como interruptores para circuitos elétricos, muito utilizados no acionamento de motores. Quando o relé recebe um sinal do seu circuito de comando, uma corrente circula por sua bobina criando um campo magnético, atraindo um mais de seus contatos, permitindo ou bloqueando a passagem de corrente no circuito da carga. Ao

cessar a corrente na bobina, uma mola faz os contatos voltarem à posição original.

Sua principal vantagem é o isolamento entre o circuito de comando e o circuito da carga, propiciando a operação com tensões diferentes em cada circuito (BRAGA, 2005a). A baixa tensão necessária para o seu acionamento permite controlá-lo através de uma saída digital do Arduino.

Circuitos com relés podem ser confeccionados ou adquiridos comercialmente, como optado neste projeto. A FIGURA 11 apresenta o módulo relé utilizado.



FIGURA 11 – Módulo relé 5V 10A de 2 canais

Fonte: http://produto.mercadolivre.com.br/MLB-625153371-modulo-rele-2-canais-5v-arduino-pic_JM. Acesso em: 05 mar. 2015.

3 DESENVOLVIMENTO

Este capítulo apresenta as etapas de desenvolvimento do equipamento de controle automático dos processos de mosturação e fervura da produção artesanal de cerveja, sendo estas: concepção do projeto, pesquisa do processo e definição dos requisitos, idealização do princípio de funcionamento, desenvolvimento do controlador automático, programação do *software*, desenvolvimento da interface homem máquina, desenvolvimento do circuito de comando, montagem do painel elétrico e os testes que foram realizados em cada uma das etapas.

3.1 Concepção do projeto

A concepção deste projeto foi inspirada no método de brassagem convencional da produção de cerveja artesanal, feita pela maioria dos cervejeiros caseiros. Neste método se utilizam geralmente três painéis: uma panela para fazer a mosturação e depois a clarificação, outra para a fervura e mais uma para fazer o aquecimento da água utilizada na clarificação. Para realizar o aquecimento do líquido utilizam-se fogareiros. Para alcançar o equilíbrio da temperatura, nos momentos de aquecimento, utiliza-se uma grande espátula para a constante agitação do mosto. Durante a mosturação esta ação também evita a queima do malte no fundo da panela.

Um método alternativo, de custo inferior, utiliza apenas duas painéis: uma para a mosturação, clarificação e fervura; e outra para aquecer a água utilizada na clarificação. Neste método, depois de feita a mosturação e clarificação, coloca-se o mosto em um recipiente temporário (balde), retira-se o bagaço (malte), do interior da panela utilizada, e inicia-se a fervura recolocando o mosto na mesma.

A ideia do equipamento fixou-se na automação deste processo alternativo, fazendo o aquecimento da panela onde se executa a mosturação e posteriormente a fervura do mosto cervejeiro, utilizando uma resistência elétrica para o aquecimento, um misturador motorizado para fazer a agitação do

mosto e uma unidade de controle para fazer a automação destes dispositivos. A FIGURA 12 ilustra a concepção deste equipamento.

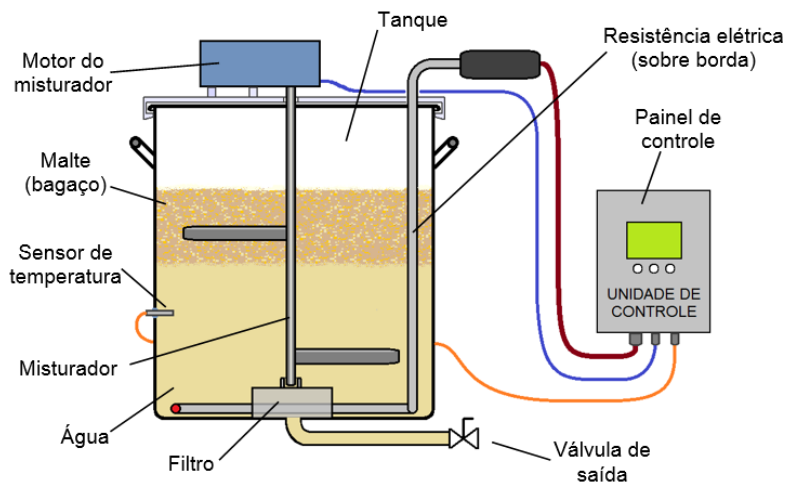


FIGURA 12 – Esboço do equipamento

Através da unidade de controle faz-se a programação da receita, e através desta o equipamento assume os *setpoints* para controlar as temperaturas e os tempos do processo. Com este equipamento atende-se a necessidade da precisão no controle da temperatura no processo, proporcionando repetitividade nas execuções das receitas, vantagem que é muito importante para conseguir a reprodução de cervejas feitas anteriormente. Além disso, esta automação confere elevada comodidade para fazer o processo de brassagem.

3.1.1 Requisitos de projeto

Inicialmente foi realizada uma pesquisa a fim de compreender os processos necessários para fabricação de cerveja e quais etapas um cervejeiro artesanal deve seguir para conseguir produzi-la.

Após o cervejeiro definir os parâmetros que irá utilizar na sua receita, obtidos através da literatura e experiência própria, seu primeiro procedimento é aquecer a água à uma temperatura ligeiramente acima da necessária para a primeira etapa da mosturação, prevendo o equilíbrio térmico após inserido o malte. Em seguida, a temperatura de cada etapa deve ficar estabilizada durante os tempos solicitados na receita. Terminada a mosturação, é feita a clarificação, processo manual onde é feita a circulação do líquido para extrair açúcares do malte ainda retidos, enquanto se coloca o mosto no recipiente temporário. Em seguida, depois de retirado o bagaço e recolocado o mosto na panela, o processo de fervura aquece o líquido até a sua ebulição e então cada tipo de lúpulo deve ser inserido na mistura no momento exato para permanecer o tempo determinado pela receita.

Com base nestes procedimentos necessários, determinou-se os requisitos do projeto bem como a escolha pela utilização de determinados componentes e técnicas, feita através de pesquisas, para assim alcançar aos mesmos, conforme apresenta o QUADRO 3.

QUADRO 3 – Requisitos do projeto

Requisito	Componentes	Técnicas
Configuração de receitas de mosturação e fervura (edição; gravação; escolha c/ visualização)	Arduino + LCD touch	Layouts de tela personalizados + comandos em tela sensível ao toque
Aquecimento e controle preciso de temperatura conforme receitas	Resistência elétrica + controlador automático (Arduino) + circuito de acionamento	Aquecimento resistivo + controle PID desenvolvido no Arduino (programação)
Controle de tempos de permanência conforme receitas	Arduino	Programação do controle lógico
Controle e acionamento do misturador	Arduino + motor CC + relé eletromecânico	Programação do controle lógico + acionamento por relé
Monitoramento do processo: status, temperatura, tempo, alarmes.	Arduino + LCD touch + alto falante	Exibição das variáveis do processo nos <i>layouts</i> da tela + gráficos + reprodução de som com Arduino

Possibilitar intervenções: pausa na mosturação, cancelamento; ajuste durante a fervura p/ uma ebulição mais intensa ou mais branda.	Arduino + LCD touch	Layouts de tela personalizados + comandos em tela sensível ao toque (programação)
Baixo custo do equipamento	Arduino + Shields	Desenvolvimento de IHM e controlador embarcados no Arduino
Bom acabamento	Caixa + conectores + molduras	Projeto + montagem de boa qualidade
Conter toda a eletrônica em um único painel c/ conexão p/ os periféricos	Painel de controle	Projeto do painel
Alimentação em rede elétrica residencial 220V		

Com base nestas definições iniciou-se o projeto e o desenvolvimento do equipamento.

3.2 Aquecedor do líquido

O aquecedor do líquido é um conjunto composto por uma resistência elétrica e um misturador, ambos controlados pelo painel de controle. Tem por objetivo aquecer o líquido com temperatura uniforme, de acordo com comando recebido do controlador do equipamento. O conjunto foi projetado para poder ser removido com facilidade para se efetuar a limpeza da panela.

A resistência elétrica foi construída especificamente para o tanque deste equipamento. As dimensões são mostradas na FIGURA 13.

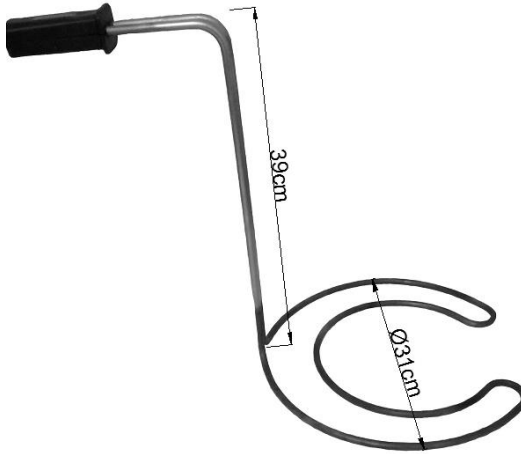


FIGURA 13 – Resistência elétrica

A resistência elétrica foi projetada para ser utilizada em uma rede elétrica residencial de 220V e, em sua capacidade máxima, drena uma corrente de 10A, o que possibilita a ligação do equipamento em tomadas comuns. O QUADRO 4 abaixo mostra as suas propriedades:

QUADRO 4 – Propriedades da resistência elétrica

Material de blindagem	Aço inox 304
Material de isolamento interna	98% Óxido de Magnésio (MgO)
Tensão	220V
Resistência (aprox.)	22Ω
Potência máxima	2200W
Dissipação máxima	15W/cm ²

A panela para o aquecimento é de alumínio e foi modificada pelo cervejeiro artesanal especificamente para a função proposta, onde foi inserido as válvulas para saída do líquido ao final do processo. Na FIGURA 14 a seguir pode-se visualizar a panela com as suas dimensões.



FIGURA 14 – Dimensões da panela

O misturador é de suma importância para a homogeneização da temperatura do líquido. O mesmo é automático, substituindo o trabalho manual do cervejeiro, que executava esta operação com o auxílio de uma grande colher.

Nos testes realizados sem um misturador, constatou-se a cristalização do malte, isto é, o cozimento e a impregnação em torno do tubo da resistência elétrica, região de maior transferência de calor. Além disso, constataram-se diferenças elevadas de temperatura entre as diferentes regiões do líquido. Contudo, o misturador só deve ser ligado quando o equipamento estiver aquecendo, enquanto a temperatura estiver estabilizada o líquido deve ficar em repouso.

O misturador foi projetado especificamente para o tanque deste equipamento, sendo um conjunto composto por um eixo com duas hastes, mancalizado em uma bucha soldada em um filtro que se localiza no fundo do tanque, um motor elétrico CC,

que rotaciona o eixo, fixado em um suporte preso na borda do tanque, conforme visto na FIGURA 15. O acionamento do motor é comandado no painel de controle.



FIGURA 15 – Misturador

3.3 Controlador automático

O controlador automático é responsável por controlar a temperatura do líquido. Para tanto, um controlador PID, implementado na plataforma Arduino compara o valor real de saída do sistema (temperatura) com a entrada de referência (*setpoint* de temperatura assumido conforme a receita programada) e assim calcula uma ação de controle para ser aplicada pelo atuador para se atingir esta referência. Este

controlador pode ser representado através de um diagrama de blocos conforme a FIGURA 16.

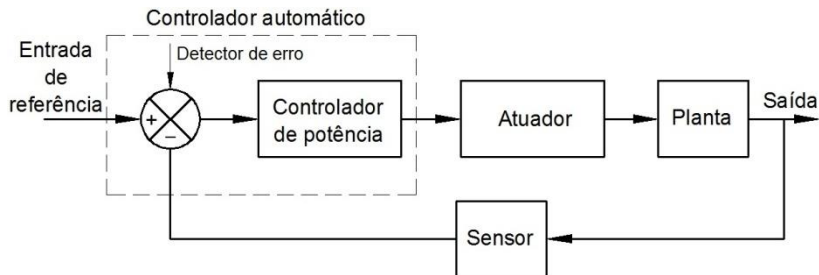


FIGURA 16 – Diagrama de blocos do controlador

Neste diagrama de blocos o detector de erro representa a comparação feita através da plataforma Arduino. O controlador de potência é o sistema responsável por aplicar a ação de controle calculada, fornecendo a tensão média necessária em CA para a resistência elétrica de aquecimento (atuador). A planta representa o sistema a ser controlado, ou seja, o tanque contendo o mosto cervejeiro a ser aquecido. O sensor é um termômetro digital de temperatura que faz a leitura da saída da planta.

3.3.1 Controlador Proporcional Integral Derivativo

O projeto do controlador do sistema iniciou-se na análise de um gráfico Temperatura X Tempo. Com a resistência de aquecimento operando em 30% da sua potência ao longo de aproximadamente três horas, coletou-se os dados necessários através da função *Serial Monitor* presente na plataforma Arduino.

Na FIGURA 17 abaixo observa-se o gráfico gerado com os dados obtidos, onde a temperatura inicial da análise foi de 21,56°C e a temperatura final foi de 73,94°C.

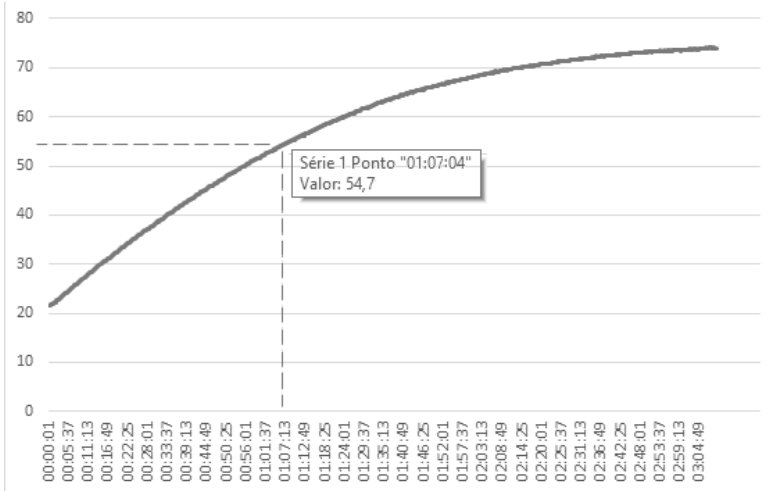


FIGURA 17 – Gráfico Temperatura X Tempo

De acordo com Ogata (2003, p.559), um sistema de primeira ordem, como neste caso, é caracterizado pela seguinte função de transferência:

$$\frac{C(s)}{U(s)} = \frac{K \cdot e^{-Ls}}{Ts + 1}$$

Onde:

$C(s)$ = referência (temperatura do sistema)

$U(s)$ = resposta do sistema

K = ganho em malha aberta

L = tempo de retardo

T = constante de tempo

A constante de tempo T é definida como o tempo necessário para o sistema atingir 63% da variação total final. Porém, para este cálculo, é necessário levar em consideração a temperatura inicial de 21,56°C:

$$T_t = (0,632 \cdot \Delta T) + T_i$$

$$T_t = (0,632 \cdot (73,94 - 21,56)) + 21,56$$

$$T_t = 54,7^\circ\text{C}$$

Através do gráfico observa-se que o sistema alcançou a temperatura de $54,7^\circ\text{C}$ no instante aproximado de:

$$t = 4020 \text{ s}$$

O ganho estático K é definido por:

$$K = \frac{T_t}{\Delta T} = \frac{54,7}{52,44} = 1,04$$

O tempo de retardo L é o intervalo gasto para variar a temperatura do sistema em 1°C . De acordo com os dados do gráfico mostrado verifica-se que, em média, $L = 216$ segundos.

Com a determinação destes valores obtém-se a planta do sistema, conforme mostrado abaixo.

$$\frac{C_{(s)}}{U_{(s)}} = \frac{1,04 \cdot e^{-216s}}{4020s + 1}$$

Para se obter as constantes proporcional, integral e derivativa que irão atuar no sistema, aplicou-se o método de sintonia de Ziegler-Nichols. Este método resulta em uma estabilidade marginal do sistema. As equações apresentadas na TABELA 1 abaixo fornecem estimativas dos parâmetros que deverão ser aplicados, proporcionando um ponto de partida na sintonia fina que deverá ser realizada posteriormente, através dos testes em campo (OGATA, 2003).

TABELA 1 – Regra de sintonia de Ziegler-Nichols

Tipo de controlador	K_p	T_i	T_d
P	$\frac{T}{L}$	∞	0
PI	$0,9 \frac{T}{L}$	$\frac{L}{0,3}$	0
PID	$1,2 \frac{T}{L}$	$2L$	$0,5L$

Fonte: OGATA (2003, p.559)

Assim, para a componente proporcional tem-se:

$$K_p = 1,2 \frac{T}{L} = 1,2 \frac{4020}{216} = 22,3$$

Para a componente integral:

$$T_i = 2 \cdot L = 2 \cdot 216 = 432$$

E para componente derivativa tem-se:

$$T_d = 0,5 \cdot L = 0,5 \cdot 216 = 108$$

Estes parâmetros são então inseridos na linguagem de programação do Arduino através de uma biblioteca disponível no site da plataforma. Em uma de suas funções, esta biblioteca faz a comparação da temperatura lida com a temperatura de *setpoint* e com base nesta diferença calcula a ação de controle. Para este cálculo, é necessário informar previamente os parâmetros do PID (K_p , T_i e T_d), bem como o valor da temperatura atual lida pelo sensor, denominada de *input*, e a temperatura a ser atingida, denominada de *setpoint*. Este cálculo resulta em uma ação de controle denominada *output*, sendo esta um valor variável de 0 a 255. Na programação do Arduino, desenvolveu-se uma função que faz a conversão deste resultado para um percentual, que então será utilizado para variar a potência entregue a resistência de aquecimento.

Após implementar a ação PID no microcontrolador, foram feitos alguns testes a fim de verificar a resposta do sistema. Com os dados obtidos nos cálculos, observou-se um sobressinal considerável, fazendo com que a temperatura estabelecida fosse ultrapassada em cerca de 1,5°C. Pesquisando por métodos de ajuste PID, encontrou-se o QUADRO 5 que apresenta algumas soluções para possíveis problemas neste tipo de controlador.

QUADRO 5 – Ajuste de controlador PID

Problema	Medida de ajuste
Resposta muito lenta	Aumentar ganho proporcional
Resposta excessivamente oscilatória	Aumentar tempo derivativo
Sobrepassagem excessiva	Reduzir taxa integral
Resposta inicialmente rápida e em seguida muito lenta	Aumentar taxa integral

Fonte: www.ece.ufrgs.br/~jmgomes/pid/Apostila/apostila/node46.html.
Acesso em: 18 mai. 2015.

Assim, nos cálculos apresentados anteriormente, foi reduzido o tempo de retardo L para 180 segundos, obtendo os valores de $K_p = 27$, $T_i = 360$ e $T_d = 90$. Estes novos parâmetros reduziram consideravelmente o sobressinal do sistema, estabilizando cerca de $0,5^\circ\text{C}$ acima da temperatura esperada, considerado uma boa margem para o processo pretendido.

3.3.2 Controle de potência CA

Neste projeto, o atuador é uma resistência elétrica projetada para alimentação em 220Vac com 2200W de potência, gerando um consumo de 10A. O método de controle CA de ciclos inteiros foi escolhido para acionar a resistência elétrica de acordo com a ação de controle recebida do controlador PID.

Neste método, o chaveamento da alimentação de corrente elétrica da resistência gera uma potência média para o aquecedor. Controlando este chaveamento é possível aplicar um percentual de potência determinado pelo controlador PID, ou seja, a ação de controle.

Um circuito elétrico foi então projetado para possibilitar este método de controle de potência e posteriormente implementado no circuito de comando do equipamento. A FIGURA 18 mostra o esquema do circuito controlador de potência projetado.

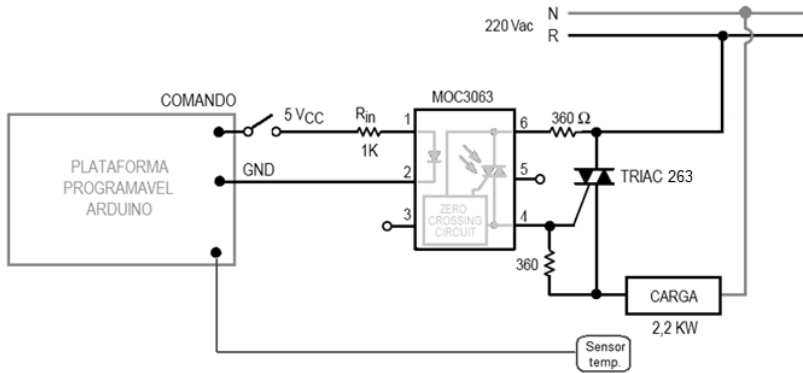


FIGURA 18 – Esquema do controlador de potência CA

Neste circuito a chave utilizada foi um TRIAC TIC263D, cuja capacidade de corrente é de 25A, atribuindo uma boa margem de segurança para o componente. A FIGURA 19 abaixo mostra o aspecto físico deste componente.

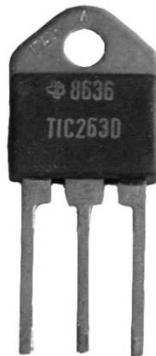


FIGURA 19 – Aspecto físico TRIAC TIC263

Fonte: <http://www.ebay.ph/itm/Triac-TIC-263-D-400V-25A-/121378531022>. Acesso em: 15 mar. 2015.

O comando desta chave foi feito com um opto-acoplador MOC3063, que possui um circuito interno detector de passagem de tensão zero. A FIGURA 20 mostra o aspecto físico do componente.



FIGURA 20 – Aspecto físico do optoacoplador MOC3063

Fonte: <http://www.oncomponents.com/moc3063-optocoupler-triac-o-p-6-dip.html>. Acesso em: 15 mar. 2015.

Conforme o circuito da FIGURA 18, o comando em 5Vcc vindo do Arduino faz parte do controlador de potência. O programa executado no Arduino converte a ação de controle (percentual a ser aplicado) em períodos de tempo ligado e desligado dentro de um ciclo de trabalho, conforme o método de controle de potência de ciclo integral. Neste projeto o ciclo de trabalho é de 2 segundos.

Por exemplo, para aplicar 40% de potência do aquecedor o programa calcula o tempo de estado ligado da seguinte forma:

$$0,4 \cdot 2 = 0,8s$$

Neste caso a resistência elétrica repete um chaveamento em que 0,8 segundo permanece ligada e 1,2 segundos desligada. Como a frequência de alimentação elétrica é de 60Hz, em 0,8 segundo são permitidos 48 ciclos de alternância da corrente elétrica de um total de 120 ciclos disponíveis. A FIGURA 21 abaixo ilustra este exemplo de aplicação de potência.

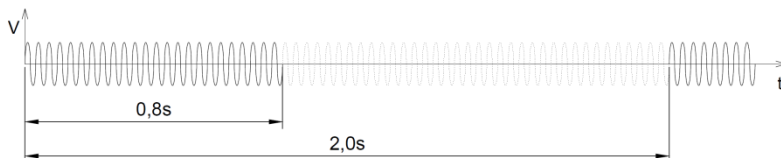


FIGURA 21 – Exemplo de controle de potência

No tempo de estado ligado, o pino digital 10 do Arduino, configurado como saída, permanece em nível lógico alto.

Quando isto acontece, o opto-acoplador MOC3063 está pronto para ativar o TRIAC, enviando o sinal de porta, e só o faz quando o seu circuito interno detector de passagem por zero, identifica o início de um semi-ciclo de alternância da corrente elétrica.

Enquanto esta saída permanece ligada, o opto-acoplador fornece um disparo para o TRIAC a cada detecção de passagem por 0V, garantindo assim a condução de corrente elétrica.

Ao desligar esta saída, o opto-acoplador retira o comando de porta do TRIAC, e este por sua vez só interrompe a condução quando a corrente elétrica alternada chega à tensão zero. Este método garante o chaveamento da alimentação elétrica somente nos momentos em que a tensão passa por zero, o que diminui significativamente o ruído gerado se comparado ao método de controle de potência de fase.

3.4 Programação do software

A programação deste equipamento foi desenvolvida na IDE Arduino, na linguagem *processing*. O microcontrolador, Arduino Mega 2560, processa as informações recebidas nas portas de entrada e, de acordo com o as condições escritas no programa, envia sinais para as portas de saída que irão atuar no sistema.

Durante a execução de uma receita, o programa desenvolvido tomará as ações necessárias para sua execução de acordo com a temperatura do sistema ou então com base nos tempos decorridos durante os processos. A FIGURA 22 apresenta em síntese a sequência de funcionamento do equipamento.

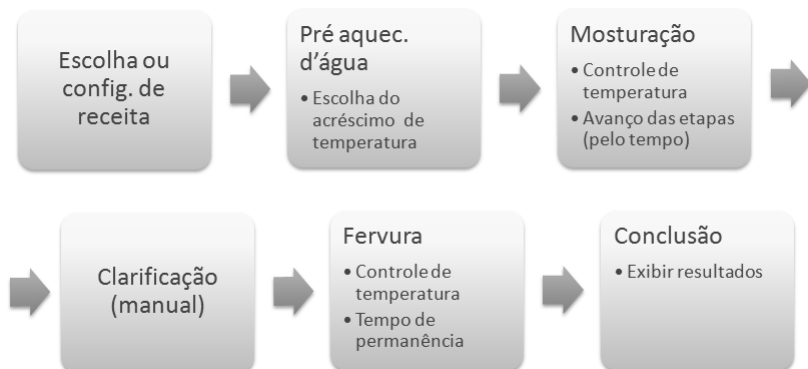


FIGURA 22 – Sequência de funcionamento do equipamento

Para o controle de temperatura, o sistema faz a leitura do sensor correspondente, calcula a ação necessária para atingir a temperatura estabelecida através do método PID e com o resultado obtido atua no sistema variando a potência aplicada sobre a resistência de aquecimento, através do método de ciclos inteiros.

O acionamento do misturador é feito de acordo com o resultado do cálculo PID. Caso a potência aplicada seja superior a 5%, o sistema envia um sinal para o chaveamento do relé que faz o acionamento do motor do misturador.

Para determinar a conclusão de etapas, processos e receitas, o sistema monitora o tempo decorrido após alcançada cada temperatura estabelecida. Com base neste processamento um novo *setpoint* é assumido, de acordo com a próxima etapa, e em determinadas condições, é enviado o sinal lógico para o acionamento dos alarmes sonoros.

Há, também, procedimentos da receita que devem ser realizados manualmente, como a clarificação, confirmação de ebulição e adição de lúpulos. Nestes casos, o sistema aguarda o usuário pressionar o botão correspondente na tela sensível ao toque para então continuar a execução da receita.

Através do único botão físico do equipamento, o sistema recebe o sinal para ligar e desligar a tela, caso este coincida com o momento em que esta função possa ser executada.

Na FIGURA 23 apresenta-se o fluxograma do processo realizado durante a execução de uma receita.

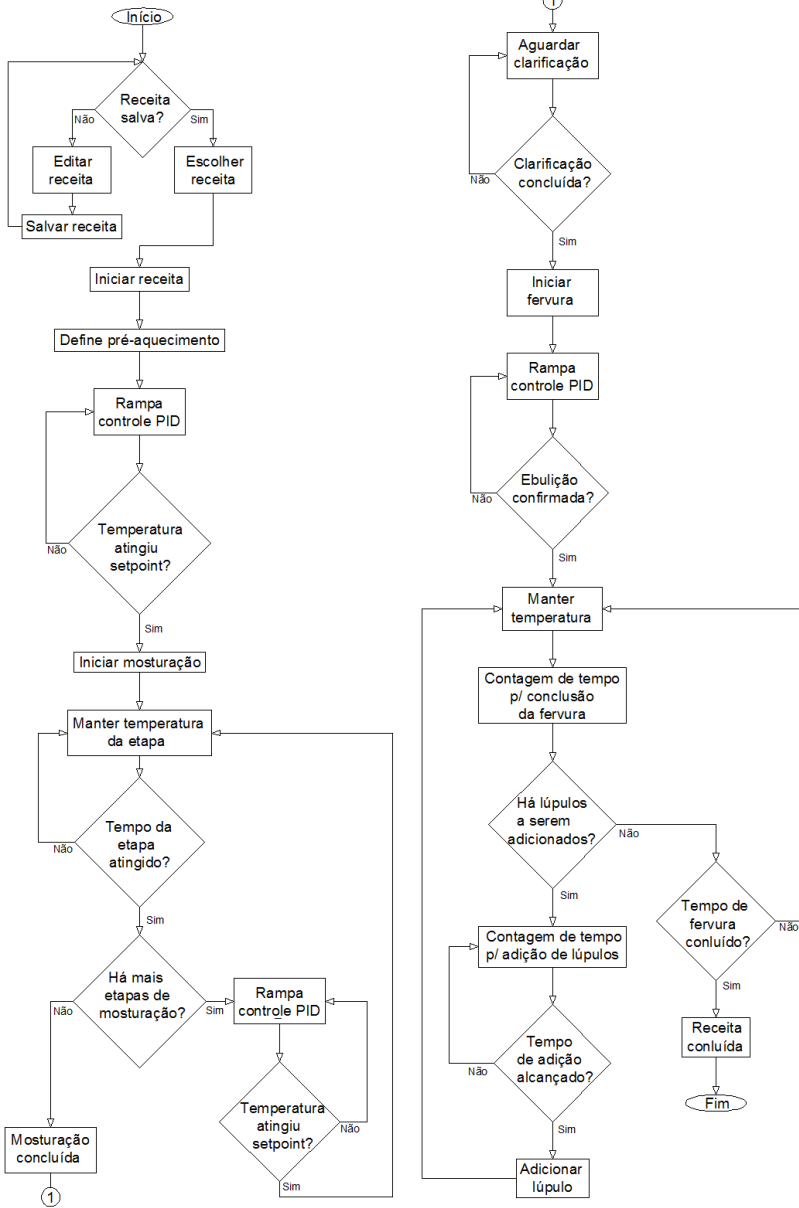


FIGURA 23 – Fluxograma do processo

Durante a execução das receitas, é necessário que as informações do sistema, como por exemplo temperatura e tempo decorrido, sejam apresentadas ao usuário. Assim, desenvolveu-se os *layouts* de tela para possibilitar a interação com o equipamento, permitindo que o sistema tome algumas ações de acordo com os toques exercidos no *display*.

Para o desenvolvimento de *layouts* de telas necessários para o processo, utilizou-se uma biblioteca disponibilizada pelo fabricante do *display*, chamada *UTFT*. Esta contém funções que permitem desenhar retângulos e círculos, configurar tamanho de fonte, cor da fonte e preenchimento, além de imprimir na tela os textos necessários.

As cores são definidas através do sistema RGB (red, green, blue). O modelo de *display* aqui utilizado pode exibir uma variedade de 65000 cores através da combinação das cores vermelho, verde e azul. Para tanto, deve-se atribuir valores que variam de 0 a 255 para cada uma das componentes. Por exemplo, atribuindo-se o valor de 255 para a componente vermelho, 255 para componente verde e 0 para componente azul, define-se a cor amarela, através da função *setColor* ou *setBackColor*.

Para distribuir os elementos no espaço da tela, utiliza-se o sistema de coordenadas cartesianas, ou seja, no plano x-y. O *display* utilizado neste projeto tem o tamanho de 3,2 polegadas, 320x240 *pixels*. Assim, o plano de coordenadas poderá assumir valores de $x = 0$ e $y = 0$ até $x = 319$ e $y = 239$, conforme visto na FIGURA 24.

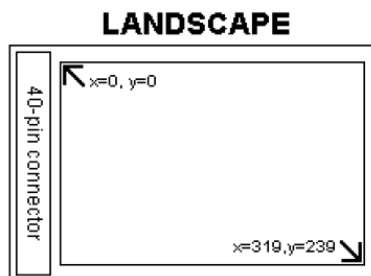


FIGURA 24 – Sistema de coordenadas para desenho de telas
 Fonte: <http://www.rinkydinkelectronics.com/download.php?f=UTFT.zip>.
 Acesso em: 30 set. 2014.

Para desenhar retângulos na tela, indica-se os pontos x e y onde se deseja iniciar o desenho e os seus respectivos pontos finais, ou seja, x_1, y_1, x_2, y_2 . Utiliza-se a função *drawRect* para desenhar o contorno de um retângulo ou a função *fillRect* para uma área preenchida.

Para círculos deve-se indicar as coordenadas do centro e o raio do mesmo. Assim como no exemplo anterior, há a função *drawCircle* para contornos e *fillCircle* para áreas preenchidas.

Para a exibição de textos, deve-se indicar as coordenadas iniciais da primeira letra, utilizando a função *print*. Há dois tamanhos de fontes possíveis: definindo-se como *SmallFont* cada letra terá o tamanho de 8×12 pixels e para *BigFont* cada letra terá 16×16 pixels.

Há também a possibilidade de inserir imagens ou figuras prontas através da função *drawBitmap*, indicando as coordenadas iniciais e finais e o nome do arquivo salvo.

No exemplo abaixo apresenta-se o código desenvolvido para exibição da tela inicial do sistema.

```
void telainicial() { // Desenha a Tela Inicial - Início
  myGLCD.fillScr(255, 255, 255); // Preenche a tela com cor branca
  myGLCD.drawBitmap(240, 110, 64, 81, chopprbranco); // Insere figura
  myGLCD.setBackgroundColor(255, 255, 255); // Cor de preenchimento (branco)
  myGLCD.setFont(BigFont); // Define tamanho da fonte
  myGLCD.setColor(255, 192, 0); // Define cor da fonte (dourado)
  myGLCD.print("Sist. Automatico de", CENTER, 10); // Texto
  myGLCD.print("Mosturacao e Fervura", CENTER, 26); // Texto
  myGLCD.print("do mosto cervejeiro", CENTER, 42); // Texto
  drawLinha(1, 8, 319, 8); // Desenha uma linha
  drawLinha(1, 60, 319, 60); // Desenha uma linha
  drawBotao2(8, 66, 310, 196); // Função desenvolvida durante este projeto
  myGLCD.setColor(92, 92, 92); // Define cor da fonte (cinza)
  myGLCD.setFont(SmallFont); // Define tamanho da fonte
  myGLCD.print("Ao iniciar, o sistema", 20, 70); // Texto
  myGLCD.print("seguira as seguintes etapas", 20, 84); // Texto
  myGLCD.print("1. Escolha/Edicao da receita", 20, 106); // Texto
  myGLCD.print("2. Pre-aquecimento", 20, 120); // Texto
  myGLCD.print("3. Mosturacao", 20, 134); // Texto
  myGLCD.print("4. Clarificacao", 20, 148); // Texto
  myGLCD.print("5. Fervura", 20, 162); // Texto
  myGLCD.print("6. Conclusao", 20, 176); // Texto
  myGLCD.setColor(255, 192, 0); // Define cor da fonte (dourado)
  myGLCD.fillRect(90, 200, 230, 232); // Desenha um retângulo preenchido
  drawBotao2(90, 200, 230, 232); // Função desenvolvida durante o projeto
```

```

myGLCD.setFont(BigFont); // Define tamanho da fonte
myGLCD.setColor(255, 255, 255); // Define cor da fonte (branco)
myGLCD.setBackColor(255, 192, 0); // Cor de preenchimento (dourado)
myGLCD.print("INICIAR", CENTER, 208); // Texto
} // Fim

```

A função `drawBotao2` não pertence à biblioteca citada, foi elaborada especificamente para este projeto para facilitar o desenvolvimento das telas. Este comando utiliza uma combinação de retângulos com cores de linhas diferentes para criar a impressão de um botão físico funcional. Os parâmetros para sua aplicação são iguais às funções para o desenho de retângulos, ou seja, deve-se indicar as coordenadas iniciais e finais do elemento.

Na FIGURA 25 observa-se a tela gerada a partir deste código.

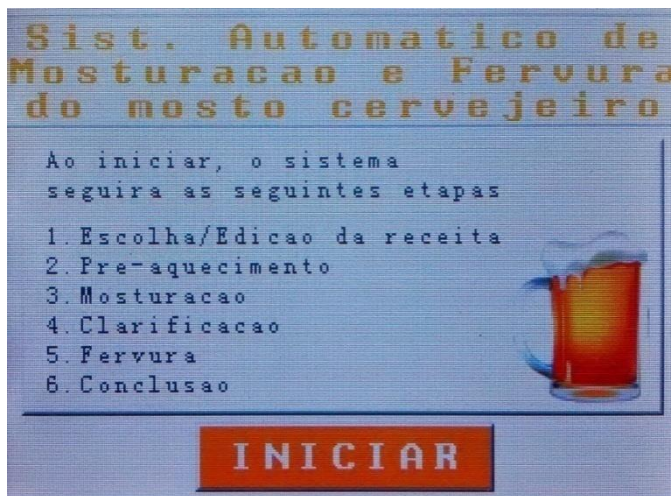


FIGURA 25 – Tela inicial

Para que o sistema interprete os toques em sua superfície é necessário a utilização da biblioteca *UTouch*. Durante a execução do programa, a função *read* faz a leitura de toda a área da tela a fim de receber os sinais de um toque e, ao receber estes sinais, as funções *getX* e *getY* coletam as coordenadas do ponto pressionado.

Para que o sistema tome ações a partir de um toque na tela é necessário indicar a área de um eventual botão. Ao receber sinais de um toque nesta região o programa entrará em uma condição para realizar a ação programada.

No exemplo abaixo apresenta-se a programação de ações para a tela inicial do sistema.

```
void loop()
{
  while (true)
  {
    .
    .
    .
    if (myTouch.dataAvailable())
    {
      myTouch.read();
      x = myTouch.getX();
      y = myTouch.getY();

      switch (flag)
      {

        case 0: // Tela Inicial -----

          if ((x>=90) && (x<230) && (y>=200) && (y<=232)) // Botão iniciar
          {
            botaoPress(90,200,230,232); // Função desenvolvida para o projeto
            flag = 2;
            myGLCD.fillScr(192, 255, 192); // Preenche a tela com a cor verde
            telaconfig(); // Carrega a tela configurar
          }

          break;

          .
          .
          .
        }
      }
    }
  }
}
```

A função *botaoPress* não pertence à biblioteca, foi desenvolvida durante este projeto para criar a imagem de um

botão pressionado. Os três pontos em linhas sequenciais indicam supressão de código.

3.5 Interface Homem Máquina

A interface para configuração e acompanhamento dos processos foi implementada através de uma tela sensível ao toque de 3,2 polegadas, controlada pelo Arduino Mega 2560.

Com os conhecimentos obtidos acerca da produção artesanal de cerveja, desenvolveu-se através da programação no Arduino as interfaces necessárias para os processos.

A tela inicial exibe as informações básicas do sistema e o botão principal para iniciar a configuração das receitas. Os procedimentos necessários para a inicialização do sistema são apresentados no momento em que o equipamento entra em operação, a fim de guiar os usuários que ainda não estão habituados com o mesmo. Na FIGURA 26 apresenta-se a interface inicial do sistema.

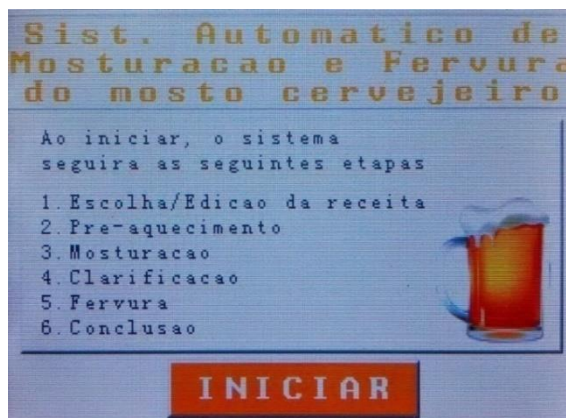


FIGURA 26 – Tela inicial

A interface para seleção e configuração das receitas, apresentada na FIGURA 27, exibe os parâmetros de temperatura e tempo conforme configurado pelo usuário. Botões alternam a exibição dos parâmetros entre os processos de mosturação e

fervura. Cinco receitas podem ser editadas, cada uma com sete etapas. O sistema salva as edições feitas em cada receita para que possam ser exibidas na próxima inicialização. Para o caso da mosturação, ao lado das etapas é apresentada o tempo em minutos da rampa para se atingir a próxima etapa.



FIGURA 27 – Tela de seleção de receitas

Para o processo de fervura, é apresentado o tempo de permanência e adição dos lúpulos, conforme FIGURA 28.



FIGURA 28 – Tela de seleção de receitas

Estando selecionado o botão Mosturação, ao pressionar o botão Editar o usuário tem acesso à tela mostrada na FIGURA 29, onde poderá alterar os parâmetros da receita de mosturação.



FIGURA 29 – Editando a mosturação

Para cada etapa deve ser atribuído um valor de temperatura que se deseja atingir e o tempo em que o sistema manterá a temperatura escolhida. O usuário pode alterar qualquer etapa aleatoriamente tocando no campo correspondente ou avançar por cada parâmetro através do botão "Prox". Nesta tela, implementou-se restrições para que a temperatura da etapa selecionada não seja inferior a temperatura anterior e nem superior a temperatura seguinte, haja vista que a diminuição da temperatura no meio do processo compromete o resultado final da cerveja produzida. Os campos de rampa são calculados e preenchidos automaticamente conforme a edição da temperatura de cada etapa. O incremento pode ser feito com uma ou cinco unidades, conforme escolhido na caixa seletora. O botão Zerar torna a etapa selecionada e as seguintes sem efeito para o sistema.

Ao término da configuração, o botão Ok salva a receita editada. Já o botão *Cancel* descarta as edições feitas e o sistema volta a exibir os parâmetros salvos anteriormente.

Selecionando o botão Fervura/Lúpulos, o usuário pode editar os parâmetros deste processo conforme visto na FIGURA 30.

Para este caso, deve-se atribuir o tempo total da fervura e o tempo de permanência dos lúpulos.



FIGURA 30 – Editando a fervura

Após a edição, ao pressionar o botão *Start* é exibida a tela para a configuração da temperatura de pré-aquecimento, visto na FIGURA 31.

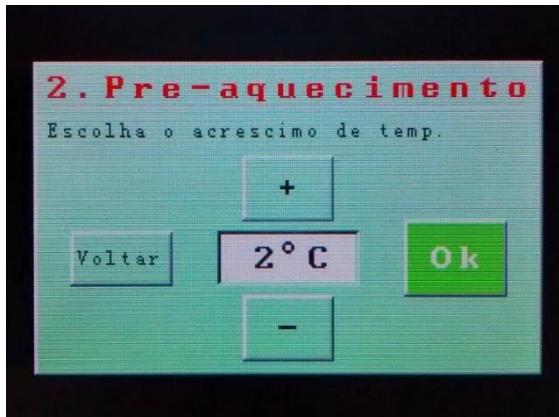


FIGURA 31 – Configurando o pré-aquecimento

Este valor de acréscimo no pré-aquecimento será incrementado na temperatura da etapa 1. No momento em que o

malte é depositado, a transferência de calor da água para o malte tende a diminuir a temperatura do sistema. Assim se faz necessário um acréscimo na primeira temperatura da receita, prevendo o equilíbrio térmico entre os ingredientes iniciais do processo. Normalmente este pré-aquecimento é fixado em 2°C, mas nesta tela optou-se por permitir a alteração deste parâmetro conforme a experiência do usuário e a variedade de ingredientes que podem ser utilizados.

Após a configuração do pré-aquecimento e pressionado o botão Ok, será exibida a tela mostrada na FIGURA 32. Nesta interface o usuário pode acompanhar o aquecimento do processo, visualizando a temperatura atual do sistema e a temperatura a ser atingida, esta última já somada com o pré-aquecimento escolhido. No canto inferior direito da tela pode-se verificar a potência do atuador variando conforme a ação do controlador, bem como o estado do misturador, se ligado ou desligado. Caso seja necessário, o usuário tem a opção de retornar e redefinir os parâmetros das telas anteriores.



FIGURA 32 – Aquecendo

Quando o sistema atinge a temperatura definida para o pré-aquecimento, é exibida, de forma intermitente, a mensagem "Arriar malte", termo comumente utilizado pelos fabricantes de cerveja, que significa despejar o malte. Neste momento é emitido um alarme sonoro para alertar o usuário. O botão Continuar permite a sequência do processo, conforme FIGURA 33.



FIGURA 33 – Pré-aquecimento concluído

Concluído o pré-aquecimento, o sistema inicia a receita configurada e o usuário pode acompanhar o progresso através da interface mostrada na FIGURA 34.



FIGURA 34 – Executando a mosturação

Esta tela exibe os parâmetros de todas as etapas da mosturação, das quais aquela em execução destaca-se na cor azul. Pode-se verificar o tempo decorrido na etapa atual e também o tempo total da receita. A temperatura atual do sistema

e a porcentagem de atuação da resistência de aquecimento são exibidas no canto inferior direito.

Ao término de cada etapa os tempos de rampa são exibidos nos campos correspondentes e, nos períodos de transição, onde é necessário elevar a temperatura do sistema, é indicado a rampa correspondente no campo antes destinado ao tempo da etapa.

Caso seja necessário, no canto inferior esquerdo desta tela há botões para pausar e cancelar a receita em execução. Se cancelado, o sistema desliga o aquecimento e o misturador, voltando a exibir a tela inicial. Na opção de pause, o tempo da etapa é interrompido, porém o tempo total da receita continua a ser contabilizado e a temperatura é mantida estabilizada para aquela etapa.

Durante a execução da receita de mosturação, ou ao término dela, o usuário pode visualizar um gráfico Temperatura *versus* Tempo pressionando o botão correspondente localizado na parte inferior da tela. Este recurso, atualizado simultaneamente conforme o decorrer do processo, exibe a curva de temperatura programada para a receita na cor amarela e a temperatura real atingida em vermelho, conforme a FIGURA 35.



FIGURA 35 – Gráfico da mosturação

Esta ferramenta é importante para que o usuário possa avaliar o desempenho da receita e visualizar eventuais distúrbios ou alterações, haja vista que não há necessidade da presença de

um operador. Pressionando o botão Voltar, os dados do processo e da receita são novamente exibidos.

Ao final da última etapa da mosturação, é exibida a tela da FIGURA 36, informando que este processo foi concluído. Novamente um alarme sonoro é emitido solicitando a confirmação do usuário.



FIGURA 36 – Mosturação concluída

Na etapa seguinte, chamada de clarificação, faz-se a circulação do mosto pela panela a fim de retirar cascas e grãos, além de extrair açúcares ainda retidos no malte. Nesta tela, apresentada na FIGURA 37 a temperatura atual do sistema e o tempo total da receita continuam a ser contabilizados e exibidos para o usuário. Este processo é feito manualmente e quando concluído o usuário deve pressionar o botão na tela para seguir para a próxima etapa. Durante este procedimento, tanto a resistência de aquecimento quanto o misturador não atuam do sistema, haja vista que neste momento estes componentes são retirados da panela para que seja feita a circulação do mosto.

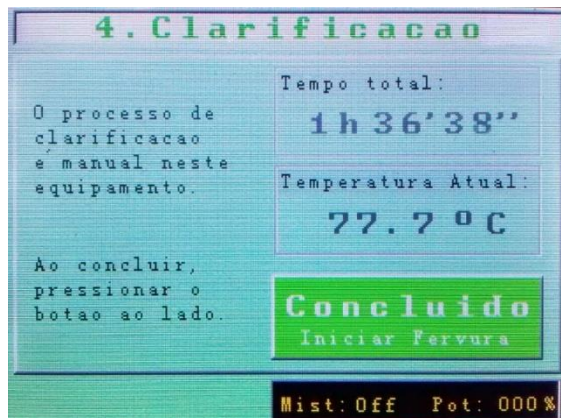


FIGURA 37 – Clarificação

O processo seguinte, chamada de fervura ou lupulagem, inicialmente apresenta a tela exibida na FIGURA 38, onde o objetivo é ferver o mosto.

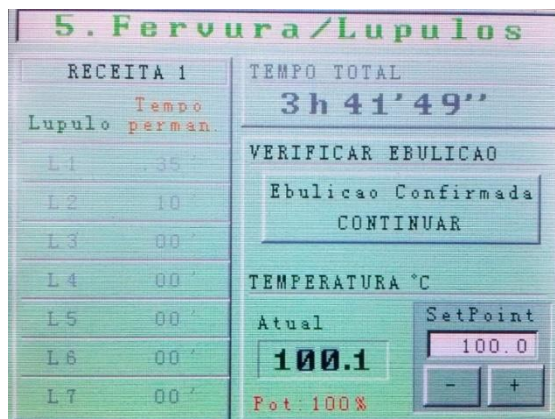


FIGURA 38 – Iniciando a fervura

Como a temperatura de ebulição pode variar de uma receita para outra, devido aos ingredientes utilizados ou devido a opção do cervejeiro por uma fervura mais intensa ou mais branda, o usuário deve indicar uma temperatura de ebulição inicial para o sistema e, caso necessário, pode fazer ajustes

neste *setpoint* durante o aquecimento. Ao observar a ebulição da solução o usuário deve pressionar o botão na tela para continuar o processo de lupulagem. Após a ebulição do mosto, inicia-se a contagem de tempo para adição dos lúpulos, conforme a FIGURA 39.

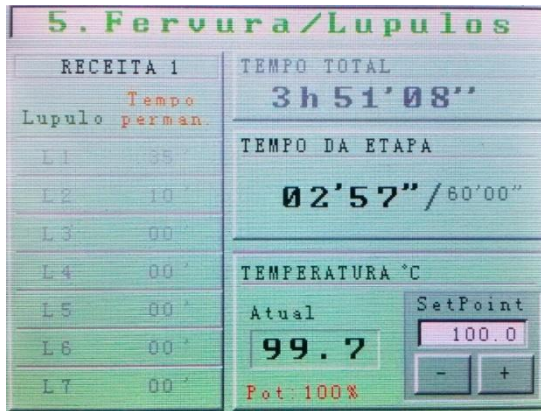


FIGURA 39 – Tempo para adição de lúpulos

Ao atingir o tempo determinado para cada adição de lúpulo, é exibida a caixa de mensagem vista na FIGURA 40. Um alarme sonoro é emitido até que o usuário confirme o acréscimo do ingrediente.

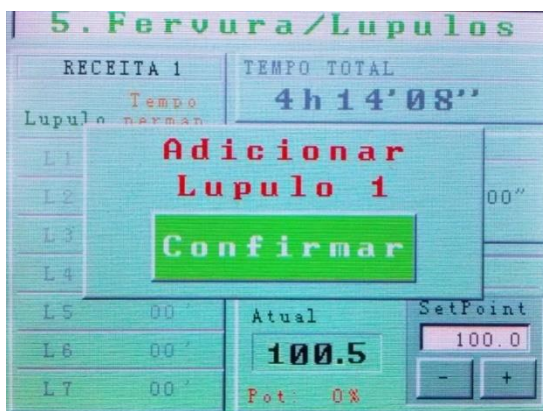


FIGURA 40 – Adicionar lúpulos

Ao concluir a última etapa de lupulagem, o sistema exibe a tela da FIGURA 41, indicando o fim do processo. Novamente um alarme sonoro solicita a confirmação do usuário.

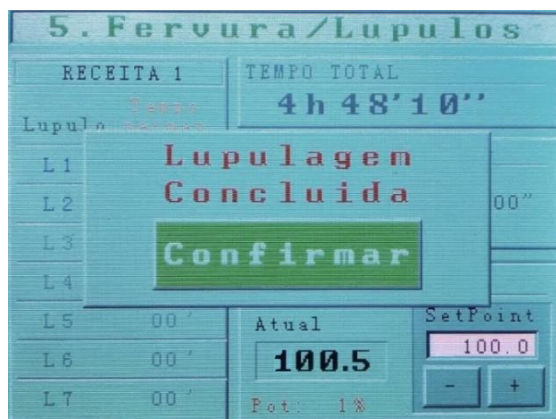


FIGURA 41 – Lupulagem concluída

Confirmando a conclusão da lupulagem, a receita está terminada e o sistema exibe a tela apresentada na FIGURA 42. Nesta interface pode-se verificar o tempo total da receita e o tempo despendido em cada processo. Com a receita concluída o sistema é desligado e deixa de controlar a temperatura do mosto.

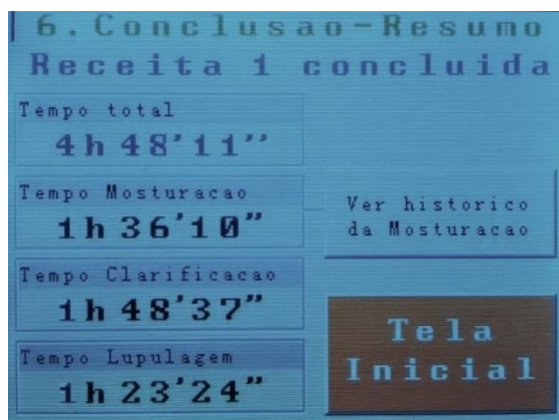


FIGURA 42 – Receita concluída

Ao pressionar o botão Ver histórico da Mosturação, é exibido novamente o gráfico desta etapa da receita. Pressionando o botão Tela Inicial, o usuário pode configurar e executar uma nova receita.

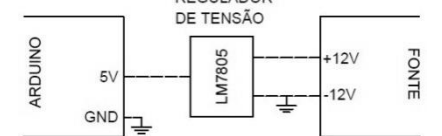
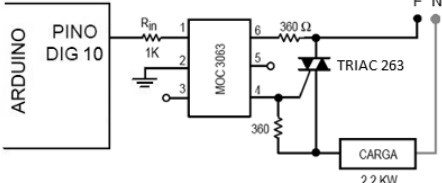
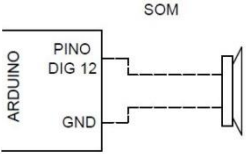
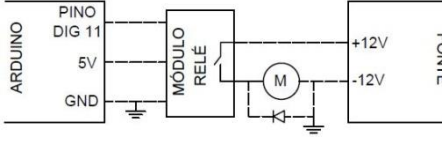
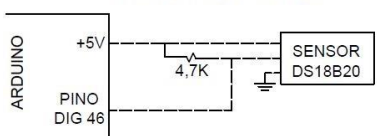
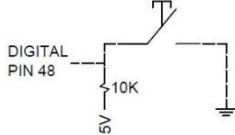
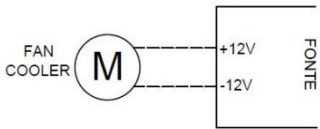
3.6 Circuito de comando

O circuito de comando é composto pela plataforma Arduino Mega 2560, tela LCD sensível ao toque com seu respectivo *shield* e o *shield* principal, este último responsável pela alimentação e comunicação dos componentes periféricos.

O *shield* principal é uma placa de circuito impresso desenvolvida para este equipamento cujo objetivo é possibilitar o acoplamento simultâneo da tela LCD e dos demais componentes ao Arduino, além de regular a tensão de alimentação deste. Esta placa é necessária para estender as conexões do Arduino para os componentes periféricos, pois o *shield* da tela LCD utiliza todos os pinos de entrada/saída da plataforma. Desta forma, o *shield* do *display* é acoplado sobre o *shield* principal e este, por sua vez, é conectado sobre o Arduino.

Esta placa possui os circuitos de comando dos componentes periféricos do equipamento. Estes circuitos foram projetados separadamente, consultando a bibliografia técnica, e posteriormente foram inseridos em conjunto no projeto da placa de circuito impresso do *shield* principal. O QUADRO 6 a seguir apresenta o nome dos circuitos com suas respectivas funções e esquemas elétricos.

QUADRO 6 – Esquema elétrico

Circuito / Função	Esquema elétrico
<p>Regulador de tensão</p> <p>Regula a tensão, com o componente LM7805, em 5Vcc para alimentação do Arduino.</p>	<p>REGULADOR DE TENSÃO</p> 
<p>Comando do aquecedor</p> <p>Controla a potência do aquecedor em CA. Descrito no tópico 3.3.2</p>	
<p>Comando do Alto falante</p> <p>Emite os avisos sonoros durante a receita. O pino digital 12 do Arduino envia um sinal em PWM conforme a frequência do som definida no programa.</p>	<p>SOM</p> 
<p>Comando acionamento misturador</p> <p>Comanda o relé que aciona o motor do misturador com pino digital 11 do arduino.</p>	
<p>Sensor de temperatura</p> <p>Aumenta o sinal digital, vindo do sensor, com um resistor de <i>pull-up</i> para que seja feita a leitura pelo Arduino através do pino digital 46.</p>	<p>SENSOR DE TEMPERATURA</p> 
<p>Botão liga / desliga</p> <p>Emite um sinal lógico alto, para o pino digital 48 do Arduino, quando pressionado o botão liga / desliga.</p>	
<p>Alimentação dos <i>collers</i></p> <p>Faz a ligação dos <i>collers</i> com a fonte 12V com o uso de conectores tipo kk.</p>	

A partir dos circuitos desenvolvidos, tendo o conhecimento de todas as entradas e saídas para a conexão dos periféricos ao *shield* principal, iniciou-se o projeto da placa de circuito impresso no *software* TraxMaker. O resultado é conferido na FIGURA 43.

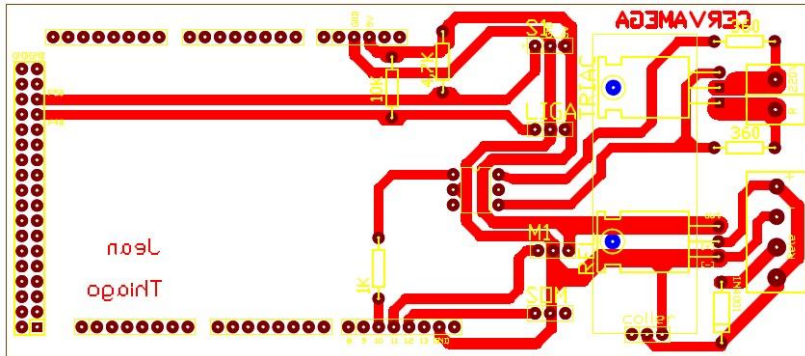


FIGURA 43 – Projeto do *shield* principal

O *shield* principal possui alimentação em corrente alternada para o comando do aquecedor do líquido e em corrente contínua para alimentação do Arduino e dos *coolers*. Para as entradas de alimentação foram utilizados conectores do tipo borne e para os componentes periféricos foram empregados conectores do tipo kk. A FIGURA 44 abaixo mostra a placa montada.

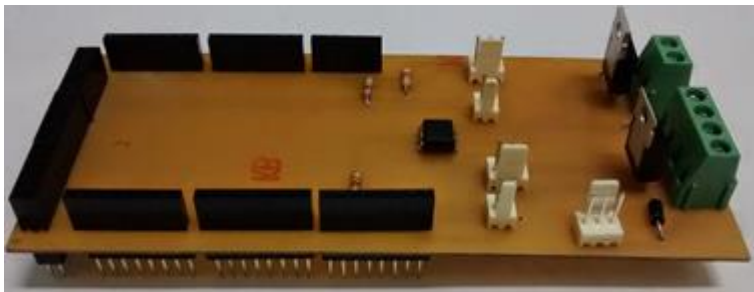


FIGURA 44 – *Shield* principal

Após a montagem do *shield* principal, foram feitos os testes com todos os componentes do equipamento a fim de averiguar o comportamento do sistema.

3.7 Painel elétrico

Utilizou-se uma Caixa Light do fabricante Steck, dimensões de 300 x 220 x 120mm, para montagem do painel. Na FIGURA 45 é apresentada uma foto do modelo de caixa empregado.



FIGURA 45 – Caixa Light utilizada para montagem

Fonte: <http://catalogo.steck.com.br/viewitems/caixas-light/caixa-light-300-x-220-x-120?>. Acesso em: 10 mai. 2015.

Em posse da caixa apresentada, iniciou-se o desenvolvimento do esquema elétrico do painel de controle para fazer a montagem dos componentes. A FIGURA 46 apresenta o esquema elétrico do painel.

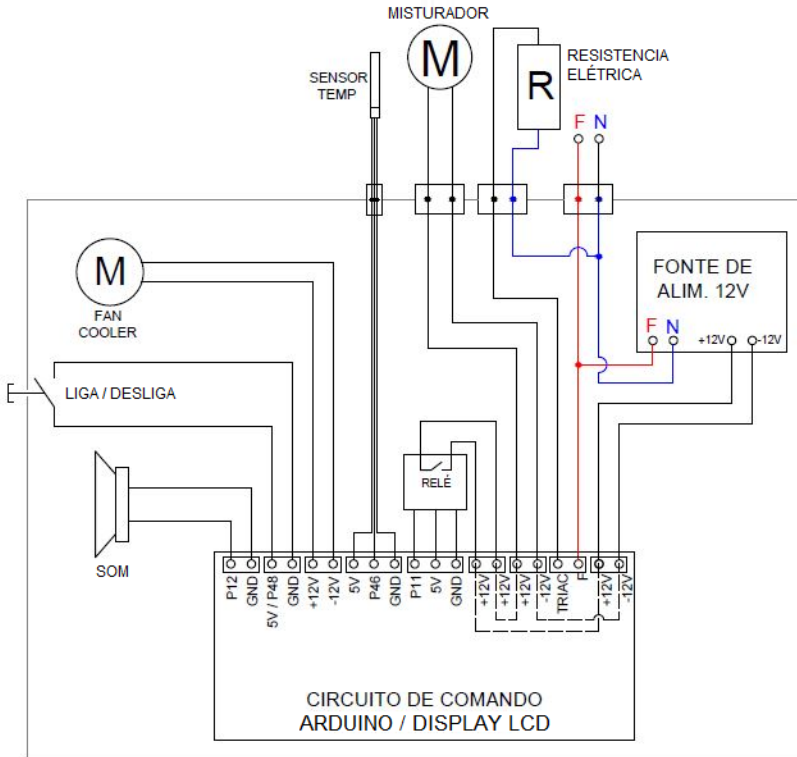


FIGURA 46 – Esquema elétrico do painel de controle

Com base neste esquema, os componentes foram conectados e dispostos no interior da caixa light para realização dos testes finais antes da montagem, conforme visto na FIGURA 47.

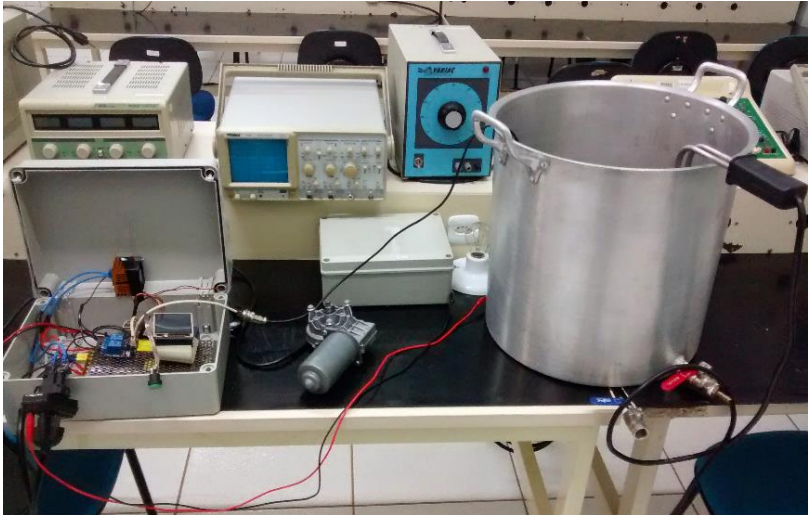


FIGURA 47 – Testes antes da montagem

Após os testes e confirmado o correto funcionamento de todo o circuito, iniciou-se a montagem do painel elétrico.

Com o auxílio de um disco de corte foram feitos recortes na caixa para posicionamento da tela sensível ao toque, fixação de conectores, passagem de cabos de alimentação e para ventilação.

No interior da caixa, todos os componentes foram fixados por parafusos em uma placa, conforme visto na FIGURA 48. Para conectar o *display* LCD ao Arduino foi utilizado um cabo *flat*.

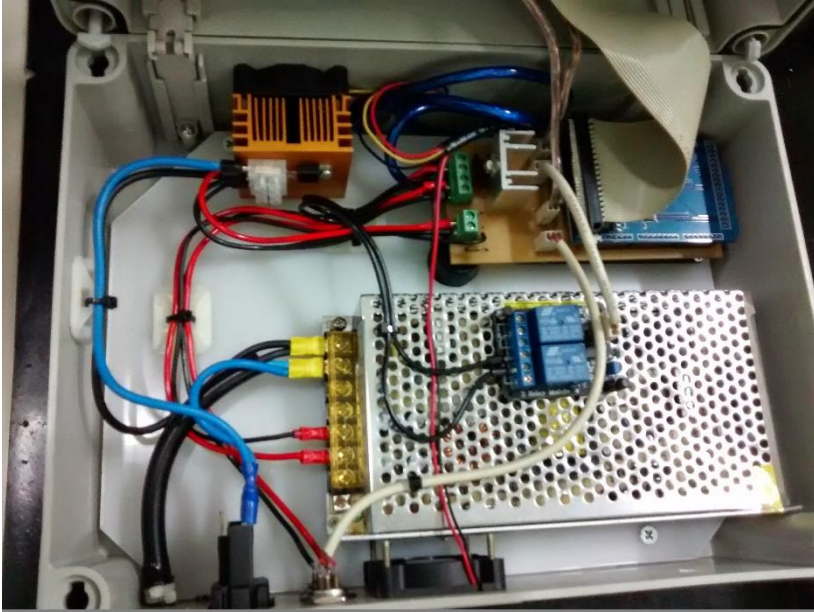


FIGURA 48 – Interior do painel elétrico

Algumas peças em metal foram confeccionadas através de corte a laser para um melhor acabamento. Estas peças foram empregadas como moldura da tela e fixação da mesma no interior da caixa, proteção dos *coolers* e para fixação de conectores. A FIGURA 49 mostra o desenho de uma destas peças feito no *software SolidWorks*.

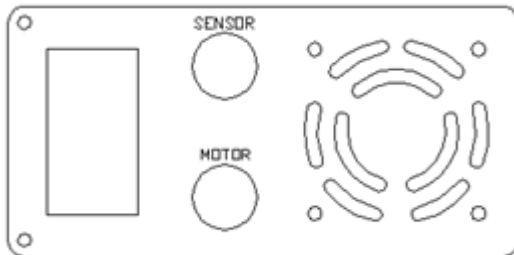


FIGURA 49 – Chapa lateral

Foram utilizados conectores do tipo *mike* para a ligação do sensor de temperatura e do motor do misturador. Na FIGURA 50 observa-se o interior do painel em outro ângulo, detalhando a chapa lateral apresentada anteriormente em desenho.

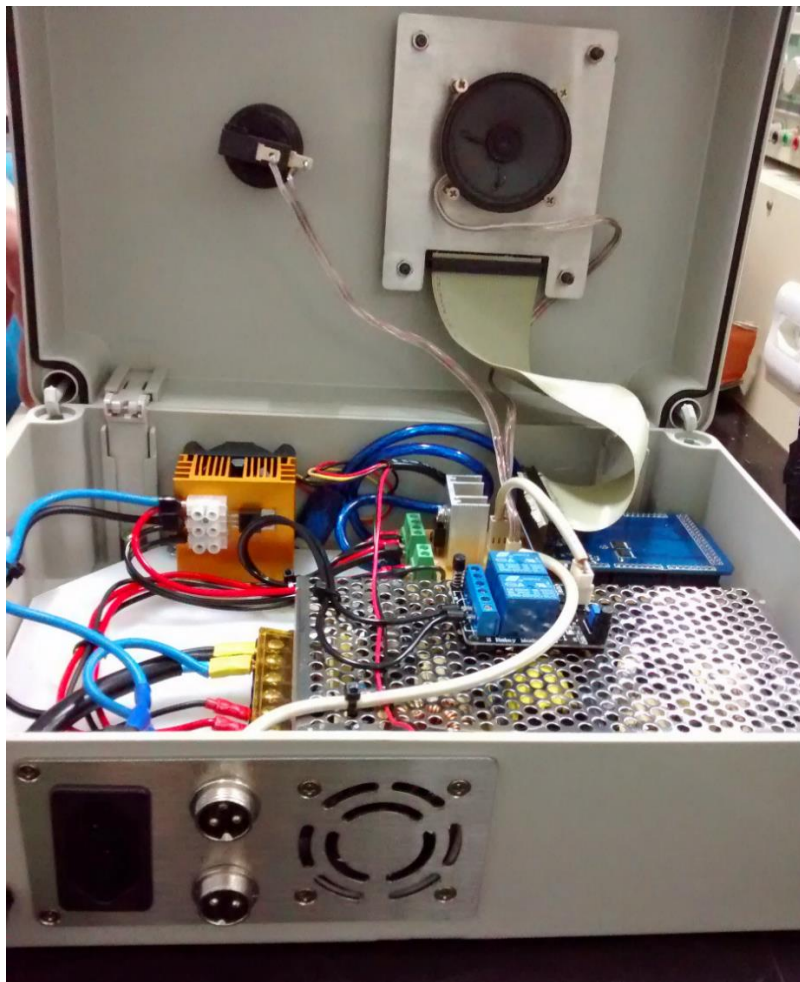


FIGURA 50 – Interior do painel elétrico com vista lateral

Na tampa da caixa, pelo lado externo, foi fixado um botão para ligar e desligar o *display* LCD quando este não estiver em

uso, sendo possível realizar este comando nos momentos em que o sistema está apresentado a tela inicial ou durante as edições de uma receita. Na FIGURA 51 observa-se a vista frontal do painel elétrico fechado, destacando a moldura do *display*.



FIGURA 51 – Vista frontal do painel elétrico fechado

4 RESULTADOS OBTIDOS

Concluído o desenvolvimento e montagem do equipamento, o teste definitivo foi realizado em parceria com o cervejeiro artesanal que inspirou este trabalho, para assim avaliar o protótipo no âmbito para o qual foi desenvolvido: a fabricação de cerveja.

Por escolha do cervejeiro, foi produzida uma receita do tipo *Belgian Blond Ale*, que caracteriza uma cerveja de alta fermentação e teor alcoólico em torno de 7%.

Assim, no início do processo configurou-se a receita desta cerveja no equipamento, através da tela de escolha ou edição de receitas, programando os processos de mosturação e fervura conforme as figuras 29 e 30 respectivamente. A receita foi configurada rapidamente pelo cervejeiro, sem a necessidade de muitas instruções dos desenvolvedores.

A FIGURA 52 mostra o início do processo de fabricação da cerveja citada.



FIGURA 52 – Início da fabricação de cerveja

Durante a mosturação, o equipamento manteve a temperatura de acordo com a receita configurada. Conforme o tempo de cada etapa era atingido, o controlador assumia o novo *setpoint* e realizava aquecimento necessário.

Apesar do controlador de temperatura apresentar certas oscilações durante o processo, a variação observada em relação à temperatura desejada foi pouco relevante para o resultado final da cerveja.

Ao término do processo de mosturação, retirou-se a resistência da panela e verificou-se que houve queima de malte, porém em pouca quantidade, ficando os grãos aderidos à resistência de aquecimento, conforme visto na FIGURA 53.



FIGURA 53 – Malte queimado aderido à resistência

Este fato ocorreu devido à alta dissipação superficial de potência na resistência de aquecimento, ocasionado pelo mau dimensionamento do projeto deste componente.

Posteriormente, a etapa de clarificação foi realizada manualmente. Conforme projetado, durante esta operação o equipamento aguardou a ação do cervejeiro até que o procedimento fosse concluído.

Durante a etapa de fervura e lupulagem, onde foi realizada a ebulição do líquido, o sistema contabilizou corretamente os tempos para adição de lúpulos, emitindo os avisos para o cervejeiro conforme programado.

Ao final deste processo, o sistema exibiu os tempos gastos em cada procedimento, bem como o tempo total despendido, indicando que a receita estava concluída.

Após estes processos iniciais foram feitos os demais procedimentos necessários para a produção da cerveja, como resfriamento, fermentação, maturação, filtração e envase. Estes processos foram todos realizados manualmente, sem a atuação do equipamento desenvolvido. A FIGURA 54 mostra o resultado final da cerveja produzida.



FIGURA 54 – Fabricação de cerveja

5 CONCLUSÕES

Neste trabalho de conclusão de curso foi desenvolvido um equipamento automatizado para mosturação e fervura da produção de cerveja. O objetivo principal foi auxiliar o fabricante artesanal nestes processos que, quando feitos manualmente, exigem um trabalho cansativo e demorado, sem garantir total confiabilidade no processo realizado.

No decorrer das etapas que culminaram neste equipamento, iniciado pela pesquisa acerca da produção de cerveja, seguida pela pesquisa por componentes e técnicas a serem empregados, desenvolvimento da IHM e do controlador automático, finalizando com a montagem do painel elétrico, muitos conhecimentos adquiridos de diversas áreas do curso de mecatrônica industrial foram utilizados, além de proporcionar ainda mais conhecimentos através da teoria e da prática.

5.1 Desenvolvimento

As pesquisas realizadas a respeito da fabricação de cerveja e da automação possibilitaram vislumbrar como deveria ser o equipamento e como alcançar as soluções para os objetivos. Durante o desenvolvimento, problemas diversos e situações não planejadas exigiram pesquisas mais aprofundadas sobre os temas intrínsecos, conforme descrito no apêndice A, propiciando maior experiência aos autores, sobretudo no campo da eletrônica e programação. O desenvolvimento do equipamento só foi possível com as pesquisas realizadas e os conhecimentos obtidos durante o curso.

Durante o projeto, foram desenvolvidos um *hardware* e um *software* para possibilitar o controle automático do processo e também a interação do usuário com o equipamento.

5.1.1 Pesquisas na área da cerveja

No decorrer deste trabalho verificou-se que a cerveja requer um controle rigoroso dos seus processos de fabricação para que se obtenha um produto de qualidade. Dentre estes processos, as etapas de mosturação e fervura podem ser apontadas como críticas e determinantes para o resultado final da bebida.

Na mosturação ocorre a conversão do amido em açúcares através da ação de enzimas. Durante esta etapa, o controle preciso das temperaturas e tempos programados é de vital importância, pois cada tipo de enzima atua em determinada faixa de temperatura. É este o processo que determina grande parte das características finais da cerveja, como por exemplo o teor alcoólico.

Durante a etapa de fervura ocorre a adição dos lúpulos à cerveja. Este processo determina o sabor e o aroma final da bebida. Para tanto, se faz necessário um controle rigoroso dos tempos de permanência de cada tipo de lúpulo.

Assim, visto a precisão necessária nas variáveis do processo, a automação se mostrou ideal para esta aplicação. Com as técnicas empregadas, garantiu-se um controle condizente ao requisitado, eliminando eventuais falhas que possam ocorrer em processos manuais, além de reduzir o trabalho feito pelo cervejeiro.

5.1.2 Pesquisas na área de automação

A pesquisa na área de automação foi iniciada com base nos conhecimentos já obtidos no curso, sendo que alguns foram, neste trabalho, muito mais aprofundados. Esta pesquisa possibilitou visualizar a grande abrangência do assunto, notando-se que a literatura possui em síntese as informações necessárias para se iniciar o desenvolvimento de um projeto como este. Além disso, possibilitou analisar os equipamentos já existentes no mercado, que atendem a objetivos análogos aos deste trabalho, verificando as deficiências e as vantagens existentes em muitos deles, possibilitando assim o desenvolvimento de um equipamento, em certos pontos, mais aperfeiçoado.

Através das pesquisas sobre o controle automático, definiu-se a elaboração do controlador PID para temperatura, bem como o método para retirar os dados da planta e assim definir os parâmetros deste controlador. Também possibilitou desenvolver o cálculo de controle na programação do *software* do equipamento com o uso da biblioteca PID, disponível gratuitamente para o Arduino, além de direcionar os ajustes necessários para alcançar um controle mais preciso.

As pesquisas também conduziram a escolha do método de controle de potência CA de ciclos inteiros para a resistência elétrica de aquecimento, ou seja, o atuador da planta, para que fosse possível a aplicação da ação de controle calculada pelo controlador PID. Todos os detalhes, como aplicabilidade, cálculos, e componentes necessários para desenvolver o *hardware* e o *software* de controle de potência foram obtidos através desta pesquisa.

O desenvolvimento da programação das telas da IHM se fez através das pesquisas sobre as bibliotecas empregadas para desenhar os *layouts*. Da mesma forma, a programação na IDE do Arduino também necessitou de pesquisas de funções, bibliotecas e métodos para se alcançar a solução final.

Para desenvolver o *hardware* de acionamento do misturador do equipamento, recorreu-se à literatura, buscando métodos e componentes para este fim, possibilitando verificar a viabilidade, relação custo/benefício e segurança de operação.

Toda a pesquisa realizada agregou conhecimentos novos e mais aprofundados sobre diversos temas e possibilitou alcançar os objetivos propostos pelo trabalho.

5.1.3 Desenvolvimento do *software*

O desenvolvimento do *software*, que faz o controle de todas as operações do equipamento, foi a etapa mais longa do projeto e a que exigiu maior empenho para chegar ao resultado. Este *software* é processado pela plataforma Arduino Mega 2560 e foi programado no IDE Arduino.

O *software* programado executa a geração das telas da IHM, que possuem os botões, campos de dados, textos e variáveis, juntamente com a interpretação dos toques na tela

LCD, e direciona as ações através destes. Também é responsável por contar o tempo decorrido e iniciar funções repetidamente, a cada intervalo de tempo determinado, como: interpretar a leitura de temperatura do sensor, executar o cálculo do controlador PID, enviar os sinais para acionamento da resistência elétrica através do método de controle de *potência* CA de ciclos inteiros, executar as funções de verificação de conclusão de tempos de permanência e etapas do processo conforme a receita selecionada, além das plotagens de tempo decorrido e temperatura atual em determinadas telas do processo.

Todas estas funções foram elaboradas em um programa único, gerando um código extenso, haja vista a necessidade de telas específicas para cada etapa da receita, ocupando 45% do espaço de armazenamento para programas do Arduino Mega 2560. Visto o tamanho do programa, verificou-se o acerto na escolha pelo modelo Mega para esta aplicação.

A programação do *software* contendo tantas funções permitiu visualizar a grande capacidade de processamento do Arduino Mega, verificando a possibilidade de agregar muitas funções à um único microcontrolador e assim reduzir gastos com a automação.

Diversas dificuldades surgiram durante o desenvolvimento do *software*, sendo necessário recorrer à literatura para buscar possíveis soluções. Porém, muitas dúvidas sobre a programação dos *layouts*, contendo telas e botões, não puderam ser sanadas desta forma, sendo contornadas através da experiência adquirida durante o desenvolvimento.

Com o equipamento finalizado, observou-se o perfeito funcionamento do *software* programado, sem apresentar falhas durante a operação do equipamento. A IHM, formada principalmente pelos *layouts* exibidos na tela LCD através da execução do *software*, se mostrou bastante completa e personalizada para o processo, possuindo um *design* sofisticado se comparado às IHM's dos equipamentos existentes.

5.1.4 Desenvolvimento do *hardware*

Através das pesquisas realizadas no início deste trabalho, definiu-se quais componentes seriam utilizados para alcançar o objetivo proposto. Após a conclusão do protótipo e dos testes realizados, verificou-se a correta aplicação e dimensionamento dos mesmos.

Obteve-se um ótimo custo/benefício com a utilização do Arduino, visto que a plataforma efetua com êxito todo o processamento das informações e controle do sistema com um relativo baixo custo.

A aplicação de um *display* sensível ao toque proporcionou uma interface intuitiva e com alto grau de personalização para o processo de produção de cerveja, possibilitando assim a fácil operação do equipamento.

O *shield* principal, desenvolvido especificamente para este projeto, possibilitou a conexão simultânea do *display* e dos componentes periféricos ao Arduino. Certificou-se a correta aplicação dos circuitos eletrônicos apresentados no Quadro 6, assim como os seus respectivos componentes foram adequadamente selecionados. Ainda assim, esta placa pode ser revisada para se obter algumas melhorias, conforme descrito no item 5.4.

A fonte empregada para alimentar o equipamento em corrente contínua se mostrou adequada e bem dimensionada para os objetivos propostos.

Por fim, o painel elétrico desenvolvido apresentou uma boa aparência. Os conectores utilizados foram de fácil operação e atenderam as necessidades do projeto. Os recortes feitos na caixa obtiveram bom resultado e as molduras com chapas metálicas conferiram um bom acabamento ao painel.

5.2 Baixo custo de implementação

Um dos objetivos propostos para este projeto foi a implementação com baixo custo. Para alcançá-lo escolheu-se reduzir o custo na área de controle lógico, uma das mais impactantes do custo total desta automação, utilizando-se a

plataforma Arduino, ao invés de um controlador lógico programável (CLP). Além disso, dois outros fatores contribuíram para alcançar este objetivo. Um deles foi a escolha por desenvolver um controlador PID utilizando o Arduino, eliminando assim o custo da aquisição de um controlador comercial. O outro foi o desenvolvimento da IHM, utilizando um *display LCD touch screen*, programada inteiramente por códigos em linguagem *processing* no Arduino, ao invés de adquirir uma IHM industrial que pode ser programada graficamente, o que facilitaria o trabalho.

Para comprovar a concretização deste objetivo comparou-se o controle lógico deste projeto com um possível projeto que utiliza um CLP de pequeno porte, marca Delta, modelo DVP14SS211T, mais a IHM DOPB03S211, mais o controlador PID, marca Coel, modelo K48E. A FIGURA 55 ilustra o aspecto físico dos componentes comparados.



FIGURA 55 – Comparação: aspecto físico Arduino x CLP

O QUADRO 7 a seguir faz o comparativo entre o controle lógico desenvolvido neste projeto com Arduino modelo Mega, e o controle lógico em que se utilizaria o CLP DVP14SS211T.

QUADRO 7 – Comparativo de custo: Arduino x CLP

	Arduino	CLP
Tempo de desenvolvimento da programação de controle lógico + IHM	6 meses	1 mês
Custo de aquisição	Arduino Mega R\$72,00	Kit CLP + IHM 4,3" + Software marca Delta R\$ 1410,00
	LCD 3,5" touchscreen R\$125,00	
	Shield p/ LCD R\$24,10	Controlador PID marca Coel R\$ 210,00
	Total: R\$221,00	Total: R\$ 1620,00

Neste quadro observa-se a grande diferença de valores entre as duas formas de controle lógico, comprovando assim que o objetivo de baixo custo de implementação foi alcançado. Como houve tempo hábil para o desenvolvimento da IHM na programação do Arduino, esta escolha tornou-se muito vantajosa. Os demais componentes desta automação como sensor de temperatura, relés, conectores e circuitos, não foram comparados, pois em ambos os casos, estes devem ser adquiridos, o que não causaria diferença nos custos.

5.3 Discussão dos resultados

A partir dos testes realizados, tanto no desenvolvimento dos componentes quanto na brassagem de um lote de cerveja utilizando o equipamento deste projeto, levantaram-se os resultados para verificar o êxito dos objetivos inicialmente propostos.

O controlador de temperatura apresentou certas oscilações durante o processo, com um sobressinal em torno de 0,7°C, porém considerado aceitável e que não prejudicou o resultado final da cerveja. O controle se mostrou eficiente nos momentos de queda da temperatura, atuando antecipadamente de forma a

não permitir uma queda maior que 0,3°C em relação ao valor desejado.

A queima de malte, provocada pela alta dissipação de potência em pouca superfície de contato da resistência, constatada no teste, não teve grande influência no resultado final da cerveja produzida. Porém, este fato deixa margem para um produto com baixo teor alcoólico, visto que os grãos perdidos reduzem a proporção de açúcar convertido em álcool durante o processo de fermentação. Além disso, a limpeza da resistência com os grãos aderidos torna-se demasiado trabalhosa.

A interface para operação e monitoramento dos processos superou as expectativas iniciais idealizadas no projeto, apresentando ao usuário um ambiente intuitivo e expondo as informações de forma bem organizada. Os dados do sistema são exibidos na tela nos momentos necessários aos processos, proporcionando ao usuário o total monitoramento da receita, garantindo a exata integração do equipamento com o objetivo do trabalho.

Em síntese, o equipamento se mostrou muito eficaz durante os testes realizados. Na brassagem de teste efetuada, os processos de mosturação, clarificação e fervura duraram cerca de cinco horas, tempo em que o equipamento operou de forma ininterrupta, certificando a correta aplicação de todos os componentes empregados.

Todos os requisitos de projeto definidos inicialmente foram cumpridos com êxito, resultando em um equipamento de relativo baixo custo se comparado àqueles comercializados atualmente.

5.4 Oportunidades de melhoria

Ao término do protótipo, algumas melhorias puderam ser previstas para futuras implementações. Estas não foram aplicadas neste momento devido ao tempo reduzido e ao prazo para apresentação do trabalho.

Verificou-se que o painel elétrico pode ter seu tamanho reduzido, caso alguns testes adicionais comprovem a empregabilidade de componentes mais compactos, como a fonte de alimentação e o *shield* principal, caso seja feita uma revisão no projeto da placa. Pode-se, também, inserir o relé utilizado

para o acionamento do misturador na mesma placa de circuito dos demais componentes.

Todas as melhorias citadas serão implementadas pelos autores deste trabalho após a conclusão do curso, pois constatou-se que este protótipo pode se tornar um produto, sendo comercializado aos produtores artesanais de cerveja.

REFERÊNCIAS

ACERVA Paulista. **Controlando a temperatura do processo**. Disponível em: <http://www.acervapaulista.com.br/index.php/artigos/77-controlando-a-temperatura-do-processo>. Acesso em: 14 out. 2013.

AHMED, Ashfaq; **Eletrônica de potência**. São Paulo: Pearson Prentice Hall, 2000. 479 p.

ARAÚJO, João C. D. A. **Sistema de aquisição e monitorização de dados para indústria alimentar e laboratorial**. 2012. 90 f. Dissertação (Mestrado em Engenharia Eletrotécnica) – Instituto Politécnico de Viseu, Portugal.

BANZI, Massimo. **Primeiros passos com o Arduino**. 1. ed. São Paulo: Novatec, 2011. 152 p.

BAZANELLA, Alexandre S.; SILVA JUNIOR, João M. G. **Ajuste de controladores PID**. 1. ed. Porto Alegre: Universidade Federal do Rio grande do Sul, 2005. Disponível em: <http://www.ece.ufrgs.br/~jmgomes/pid/Apostila/apostila/apostila.html>. Acesso em: 18 mai. 2015.

BEGA, Egídio A. **Instrumentação industrial**. 2. ed. Rio de Janeiro: Interciência, 2006.

BRAGA, Newton C. **Eletrônica básica para mecatrônica**. 1. ed. São Paulo: Editora Saber, 2005. 160 p.

BRAGA, Newton C. **Fontes de alimentação**. 1. ed. São Paulo: Editora Saber, 2005. 240 p.

CERVEJANTES. **Panela elétrica BeerBot 60 (Cozinha completa para cerveja artesanal)**. Disponível em: <http://www.cervejantes.com.br/panela-eletrica-beerbot-50>. Acesso em: 01 jun. 2015.

CIRIACO, Douglas. **A história das telas touchscreen**. Disponível em

<http://www.tecmundo.com.br/touchscreen/42036-a-historia-das-telas-touchscreen.htm>. Acesso em: 01 out. 2014.

Congresso Brasileiro de Educação em Engenharia. 35, 2007, Curitiba. **Instrumentação de uma coluna de destilação didática**. Maringá: Universidade Estadual de Maringá, 2007.14 p.

COSTA, Felipe C. **Processo de produção de cerveja artesanal**. Disponível em <http://www.missaosommelier.com.br/processo-de-producao-de-cerveja-artesanal/>. Acesso em: 27 set. 2014.

COSTA, Luiz A. **Especificando sistemas de automação industrial**. São Paulo: Biblioteca24horas, 2011. 210 p.

CYBIS, Walter A. **Engenharia de usabilidade: uma abordagem ergonômica**. Florianópolis: Universidade Federal de Santa Catarina, 2003.

DIAS, Felipe S. B. **Controle PID**. Ceará: Instituto Federal do Ceará, 2012. 8 p.

EVANS, Martins; NOBLE, Joshua; HOCHENBAUM, Jordan. **Arduino em ação**. 1. ed. São Paulo: Novatec. 2013. 423 p.

FRANKLIN, Gene F.; POWELL, David; NAEINI, Abbas E. **Feedback Control of dynamic systems**. 6. ed. Prentice Hall, 2009.

GOMES, Victor H. **Processo de produção de cerveja**. Disponível em <http://www.engenhariadacerveja.com.br/2013/processo-de-producao-de-cerveja>. Acesso em: 27 set. 2014.

JHONAS BIER. **Cervejaria de 100 Litros – JB**. Disponível em: <http://jhonasbier.webnode.com.br/products/cervejaria-de-100-litros-jb/>. Acesso em: 08 jun. 2015.

MANEDE FILHO, João. **Instalações elétricas industriais**. 8. ed. Rio de Janeiro: LTC. 2011. 667 p.

MCROBERTS, Michael. **Arduino básico**. 1. ed. São Paulo: Novatec, 2011. 456 p.

MONTEIRO, Luiz H. A. **Sistemas dinâmicos**. 2. Ed. São Paulo: Livraria da Física, 2006. 625 p.

MORADO, Ronaldo. **Larousse da cerveja**. São Paulo: Larousse do Brasil, 2009. 357 p.

MORAES, Cícero C.; CASTRUCCI, Plínio L. **Engenharia de automação industrial**. 2. ed. Rio de Janeiro: LTC, 2013. 347 p.

MUNHOZ, André. **Tecnologia touch screen**. Disponível em <http://partners.bematech.com.br/2011/02/edicao-151-tecnologia-touch-screen/>. Acesso em: 01 out. 2014.

NATIONAL INSTRUMENTS. **Explicando a teoria PID**. Disponível em: <http://www.ni.com/white-paper/3782/pt/>. Acesso em: 28 set. 2014.

NISE, Norman. **Engenharia de sistemas de controle**. LTC, 2009. 706 p.

OGATA, Katsuhiko. **Engenharia de controle moderno**. 4. ed. São Paulo: Practice Hall, 2003. 929p.

ROSÁRIO, João M. **Automação industrial**. São Paulo: Baraúna, 2009. 515 p.

SILVEIRA, João A. **Experimentos com o Arduino**. São Paulo: Ensino Profissional, 2011. 200 p.

SILVEIRA, Paulo R.; SANTOS, Winderson E. **Automação e controle discreto**. 9. ed. São Paulo: Érica, 1998. 229 p.

SOUZA, David J. **Desbravando o PIC: ampliado e atualizado para PIC 16F628A**. 12. ed. São Paulo: Érica, 2008. 268 p.

TSCHOPE, Egon C. **Micro cervejarias e Cervejarias: A História, a Arte e a Tecnologia**. 1. ed. São Paulo: Aden, 2001. 223 p.

APÊNDICE A – Dificuldades encontradas

Durante o desenvolvimento deste trabalho, alguns imprevistos precisaram ser superados. Os problemas enfrentados propiciaram maior conhecimento acerca do assunto pertinente, visto as pesquisas complementares que foram realizadas.

No início do projeto definiu-se que a alimentação em corrente contínua do equipamento seria proveniente de uma fonte de 12 Volts. O Arduino possui em sua placa o regulador de tensão AMS1117 que lhe permite receber uma tensão contínua máxima de 20 Volts para alimentação, pois este componente converte a tensão de entrada para os 5 Volts necessários para o correto funcionamento da plataforma. Porém, nos testes iniciais verificou-se um aquecimento considerável na placa ao alimentá-la com 12V, pois apenas o *display* LCD necessita de uma corrente relativamente alta, em torno de 0,2A, resultando em uma alta potência a ser dissipada no regulador.

A potência dissipada pode ser calculada conforme a equação abaixo.

$$P_D = (V_{in} - V_{out}) \cdot I$$

Onde:

P_D = Potência dissipada

V_{in} = Tensão de entrada (fonte de alimentação)

V_{out} = Tensão de saída (Arduino)

I = Corrente consumida

Assim, para este projeto, tem-se:

$$P_D = (12 - 5) \cdot 0,2$$

$$P_D = 1,4 \text{ Watts}$$

Conforme o *datasheet* do regulador de tensão AMS1117, a máxima potência dissipada para o correto funcionamento do componente é de 1,2W. Assim, como neste projeto esta potência excede o valor recomendado pelo fabricante, decidiu-se alimentar o Arduino através de um regulador de tensão LM7805 acoplado a um dissipador de calor para maior segurança, visto

que no painel elétrico a placa ficaria em um espaço confinado e com pouca circulação de ar.

Um superaquecimento também foi observado do tiristor TRIAC, responsável pelo chaveamento do aquecedor do líquido. Este problema estava ocasionando o mal funcionamento do componente, prejudicando o controle de temperatura do sistema. Após algumas tentativas de solucionar o problema que não surtiram grande efeito, verificou-se que o optoacoplador não estava identificando corretamente a passagem de tensão por 0V, gerando o chaveamento irregular no tiristor. Após substituir o optoacoplador, resolveu-se retirar o TRIAC do *shield* principal para acoplá-lo a um dissipador de calor de maior capacidade fixado a um *cooler*, visto o espaço reduzido destinado ao tiristor na placa, garantindo assim o resfriamento forçado do componente. Esta alternativa apresentou-se como uma boa solução, permitindo o correto funcionamento do sistema.

