

CENTRO UNIVERSITÁRIO CAMPO LIMPO PAULISTA
ENGENHARIA ELETRÔNICA

**DESENVOLVIMENTO DE UM FORNO DE REFUSÃO PARA EMPRESAS DE
PEQUENO PORTE**

RAFAEL CHIARETTI
SUELLEN OLIVEIRA FRANCISCO

CAMPO LIMPO PAULISTA - SP
DEZEMBRO – 2018

**RAFAEL CHIARETTI
SUELLEN OLIVEIRA FRANCISCO**

**DESENVOLVIMENTO DE UM FORNO DE REFUSÃO PARA EMPRESAS DE
PEQUENO PORTE**

*Trabalho de conclusão apresentado ao
Centro Universitário Campo Limpo
Paulista – UNIFACCAMP, como requisito
para a obtenção do grau de Bacharel em
Engenharia Eletrônica.*

**Orientador: Prof. Dr. Alexandre Martinewski
Prof. Francisco Coelho de Oliveira**

**CAMPO LIMPO PAULISTA - SP
DEZEMBRO – 2018**

**CENTRO UNIVERSITÁRIO CAMPO LIMPO PAULISTA
ENGENHARIA ELETRÔNICA**

**DESENVOLVIMENTO DE UM FORNO DE REFUSÃO PARA EMPRESAS DE
PEQUENO PORTE**

**RA 20279 - RAFAEL CHIARETTI
RA 20318 - SUELLEN OLIVEIRA FRANCISCO**

Orientador: Prof. Dr. Alexandre Martinewski

Banca Examinadora:

Prof.

Convidado

Prof. Dr. Alexandre Martinewski

Orientador

Prof. Dr. Alexandre Martinewski

Coordenador

**CAMPO LIMPO PAULISTA - SP
DEZEMBRO – 2018**

DEDICATÓRIA

Este trabalho é dedicado a todos aqueles que se interessa a aprender, e a todos integrantes do grupo e aos professores pelo grande apoio no desenvolvimento deste projeto.

AGRADECIMENTO

Agradecemos em primeiro lugar a DEUS autor de nossas vidas e a todos que colaborou para a realização deste trabalho, sem os quais não teria sido possível sua conclusão. Por isso gostaríamos de expressar toda nossa gratidão e apreço a todos aqueles que, direta ou indiretamente, contribuíram para que esta tarefa se tornasse uma realidade. A todos queremos manifestar os nossos sinceros agradecimentos.

EPÍGRAFE

*"Há homens que lutam um dia e são bons.
Há outros que lutam um ano e são melhores.
Há os que lutam muitos anos e são muito bons.
Porém, há os que lutam toda a vida.
Esses são os imprescindíveis."*

Bertolt Brecht.

RESUMO

Este projeto apresenta uma proposta de um sistema de forno de refusão para processo dedicado à soldagem de componentes eletrônicos, de baixo custo para empresas de pequeno porte. Essa soldagem é realizada pelo forno de refusão, no qual pertence ao processo de montagem de produtos eletrônicos com tecnologia de dispositivo de montagem em superfície SMD. Esse projeto, no caso estudado, é aplicado no ramo de desenvolvimento, o qual exige que o processo produtivo tenha características destacadas de receptibilidade, padronização, além da preservação do meio ambiente. Para atender essas características o mencionado processo emprega a tecnologia de solda sem chumbo na soldagem de componentes. Essa tecnologia exige que o perfil térmico possua uma média de temperatura maior sobre os elementos submetidos ao processo de soldagem, quando comparado com o processo tradicional estanho-chumbo. Para atender essa exigência o sistema de supervisão proposto neste trabalho adquire informações relacionadas a temperaturas das zonas do forno, integridade de resistência de aquecimento e sistema de insuflamento de ar quente. Essa aquisição é apresentada para o operador do sistema, por meio de uma Interface Homem-Máquina, com a meta de alertar sobre o atual estado do processo. Nessa interface é destacada a previsão de campos relacionados com registro contínuo das informações adquiridas, parametrização de informações de processo, alarmes visuais e sonoros.

Palavras Chaves: Refusão, SMD, Soldagem.

Abstract

This project presents a proposal of a reflow oven system for the process dedicated to the welding of electronic components, of low cost for small companies. This welding is performed by the reflow oven, in which it belongs to the assembly process of electronic products with SMD surface mount device technology. This project, in the case studied, is applied in the field of development, which requires that the productive process has outstanding characteristics of receptivity, standardization, and preservation of the environment. To meet these characteristics the mentioned process employs lead-free soldering technology in component welding. This technology requires that the thermal profile has a higher average temperature on the elements submitted to the welding process when compared to the traditional tin-lead process. To meet this requirement, the supervision system proposed in this work acquires information related to oven zone temperatures, heating resistance integrity and hot air insufflation system. This acquisition is presented to the system operator through a Man-Machine Interface, with the aim of alerting about the current state of the process. In this interface, it is highlighted the prediction of fields related to continuous recording of acquired information, parameterization of process information, visual and audible alarms.

Key Words: Reflow, SMD, Welding.

Lista de siglas

BGA - Ball Grid Array - Matriz de grade de bola.
Bit - Binary Digit – Dígito Binário.
Byte - 8 Bits
CAD - Computer Aided Design.
CI - Circuito integrado.
CLP - Controlador Logico Programável.
CLK –Serial Click Input.
CNC - Computer Numeric Control - Comando numérico computadorizado.
cm- Centímetro.
CS - Chip select.
GND - Graduated Neutral Density Fiter – Filtro de densidade neutra.
FAN -Resfriamento.
HIGH - Alto.
HEAT - Calor
LED - Light Emitting Diodo.
LCD - Liquid Crystal Display – Tela de cristal líquido.
mA - Miliampere.
Mhz - Megahertz.
mm- Milímetro.
NTC - Negative Temperature Coefficient.
PC - Personal Computer – Computador pessoal.
PCB - Printed Circuit Board - Placa de circuito impresso.
PCI - Placa de circuito impresso.
PT 100 - Termorresistência.
PTC - Positive Temperature Coefficient.
RTD - Resistance Temperature Detector – Termorresistência.
SMD - Surface Mounting Devide - Montagem de superfície Devide.
SMT - Surface Mount Technology - Tecnologia de montagem em superfície.
SPI - Serial Peripheral Interface.
Tc - Temperatura em Celsius.
Tf - Temperatura em Fahrenheit.
Tk - Temperatura em Kelvin.
THT - Through Hole Technology - Através da tecnologia buraco.
UART- Transmissor Receptor Assíncrono Universal.
USB - Universal Serial Bus – Porta universal.
VCC - Volts em corrente continua.
3D - Terceira dimensão
 Ω - Resistencia

Lista de figuras

Figura 1 - Processo atual.	20
Figura 2 - Aplicação de solda em pasta.	20
Figura 3 - Colocação de componentes.....	22
Figura 4 - Forno de refusão de solda.	22
Figura 5 - Perfil térmico estanho-chumbo.....	24
Figura 6 - (a) Apresentação dos resistores em carretel. (b) e (c) Exemplos de pré- formação.	27
Figura 7 - Exemplo de dispensadora. Região “a” destaca o reservatório de pasta de solda (podendo ser substituído por adesivo). Região “b” destaca o bico de injeção. 29	
Figura 8 - Exemplo de estêncil usado para deposição.	29
Figura 9 - Impressora semi-automática.	30
Figura 10 - Exemplo de máquina impressora SMT.	31
Figura 11 - Exemplo de forno de refusão.	32
Figura 12 - Máquina de solda automática.	33
Figura 13 - Aplicador de fluxo através de espuma.	34
Figura 14 - Exemplo idealizado de curva de aquecimento de forno de refusão.	35
Figura 15 - Perfil de solda ideal quando a deposição de material da liga.	36
Figura 16 - Sistema de soldagem de onda plana, também conhecido como onda simples ou onda lambda. (a) Foto e (b) Corte explicativo.	37
Figura 17 - Sistema Dupla Onda. (a) Foto e (b) Corte explicativo.....	37
Figura 18 - Sistema de solda seletiva.....	38
Figura 19 - Falhas comuns no processo de soldagem SMT.	39
Figura 20 - Escalas Termométricas.....	42
Figura 21 - Termopar.....	42
Figura 22 - Termo resistência.....	44
Figura 23 - Diagrama eletrônico da fonte.	48

Figura 24 - Diagrama eletrônico da placa de controle.....	49
Figura 25 - Diagrama eletrônico resfriamento.	50
Figura 26 - Diagrama eletrônico de aquecimento.....	51
Figura 27 - LCD Gráfico 128x64.....	52
Figura 28 - Diagrama eletrônico de aquecimento do LCD 128x64.....	54
Figura 29 - Layout e posicionamentos dos componentes.	56
Figura 30 - Máscara de solda.....	56
Figura 31 - Placa em 3D.....	56
Figura 32 - Ilustra a CNC que foi utilizada para criar a placa.	57
Figura 33 - Processo de usinagem da placa.	58
Figura 34 - PCI usinada e perfurada.	58
Figura 35 - Demonstra o Layout e posicionamentos dos componentes.....	59
Figura 36 - Layout da PCI.	60
Figura 37 - Processo de inserção e soldagem dos componentes.....	61
Figura 38 - Placa controladora do forno de refusão montada e pronta para uso	61
Figura 39 - Teste	62
Figura 40 - Montagem do painel.....	63
Figura 41 - Pintura do painel.	63
Figura 42 - Proteção do sensor de temperatura dentro do tubo de cobre.....	64
Figura 43 - Passagem de cabeamento.....	65
Figura 44 - Forno Montado e pronto para entrar em funcionamento.....	65
Figura 45 - Ilustra a CNC que foi utilizada para criar a placa.	66
Figura 46 - Teclado.	69
Figura 47 - Configuração do teclado.	69
Figura 48 - Tela inicial e execução.....	70
Figura 49 - Configurações da tela inicial e execução.	71

Figura 50 - Telas de seleções de perfil 1.....	71
Figura 51 - Telas de seleções de perfil 2.....	72
Figura 52 - Configuração e menu.....	72
Figura 53 - Menu e configuração.....	73
Figura 54 - Tela de calibração da resistência.....	73
Figura 55 - Configuração da tela de calibração da resistência.....	73
Figura 56 - Tela gráfico	74
Figura 57 - Configuração da tela gráfico.	74
Figura 58 - Descrição dos pinos do Arduino.....	75
Figura 59 - Resultado de soldagem obtido.....	77
Figura 60 - Orçamento Mercado livre.....	79
Figura 61 - Orçamento Empresa ALL Point.....	80
Figura 62 - Orçamento Empresa OMTEC	81
Figura 63 - Forno de Refusão empresa OMTEC.....	81

Lista de tabelas / Gráficos

Gráfico 1 - Auto alinhamento processo Estanho-Chumbo.....	25
Gráfico 2 - Auto alinhamento processo LeadFree.	25
Tabela 1 - Tipos de termopares.	43
Tabela 2 - Endereçamento entre LCD e arduino.	55
Tabela 3 - Valores dos componentes para o protótipo.....	76
Gráfico 3 - Comparações de preços.....	79
Tabela 4 - Salário e horas trabalhadas de um técnico em soldagem.....	82
Tabela 5 - Redução mão de obra com a implantação do forno.....	83

SUMÁRIO

1.	Introdução.....	16
1.1.	Objetivo geral	16
1.2.	Problema	17
1.3.	Justificativa.....	18
1.4.	Metodologia.....	18
2.	Referencial teórico.....	20
2.1.	Processo de refusão estanho chumbo	20
2.1.1.	Aplicação de solda em pasta.....	20
2.1.2.	Colocação de componentes	21
2.1.3.	Refusão de solda.....	22
2.1.4.	Controle Atual.....	23
2.2.	Novo processo de refusão de solda lead free	25
2.2.1.	Preparação	26
2.2.2.	Montagem.....	27
2.2.3.	Equipamentos.....	28
2.2.4.	Processos de soldagem	33
2.2.5.	Revisão e Pós-Processamento	38
2.3.	Temperatura.....	41
2.3.1.	Escalas Termométricas	41
2.3.1.1.	Escala Celsius	41
2.3.1.2.	Escala Fahrenheit.....	41
2.3.1.3.	Escala Kelvin	41
2.3.2.	Medidores.....	42
2.3.2.1.	Termopar	42
2.3.2.2.	Termo resistência ou RTD	44

2.3.3. Amplificador para Termopar	44
3. Materiais e Métodos	46
3.1. Projeto de Hardware.....	46
3.1.1. Escolha de circuitos e Definição de componentes	46
3.1.2. Esquemáticos	48
3.1.2.1. Fonte de alimentação.....	48
3.1.2.2. Placa de controle	49
3.1.2.3. Display	52
3.1.3. Principais Características:	52
3.1.4. Placa de circuito impresso.....	55
3.1.5. Roteamento e desenho da placa de circuito impresso	55
3.2. Montagem do equipamento.....	58
3.3. Projeto do Software.....	66
3.3.1. Programação.....	66
3.3.2. Comunicação.....	66
4. Resultados.....	67
4.1. Funcionamento do equipamento	67
4.2. Custo	75
4.3. Teste	76
5. Discussão	78
6. Considerações Finais	84
Referências bibliográficas	86
ANEXO I – DIAGRAMA ELÉTRICO	88
ANEXO II – LAYOUT DA PLACA.....	89
ANEXO III - CÓDIGO FONTE “FIRMWARE”	90
ANEXO IV - CÓDIGO FONTE “TECLADO”	100

ANEXO V - CÓDIGO FONTE “BUZZER”	121
ANEXO VI - CÓDIGO FONTE “CALIBRAÇÃO”	123
ANEXO VII - CÓDIGO FONTE “DISPLAY”	124
ANEXO VIII - CÓDIGO FONTE “GRAFICO”	149
ANEXO IX - CÓDIGO FONTE “MEMORIA”	150
ANEXO X - CÓDIGO FONTE “SERIAL_OUT”	166
ANEXO XI - CÓDIGO FONTE “START-STOP”	167
ANEXO XII - CÓDIGO FONTE “TIMER”	169

1. Introdução

Este trabalho de conclusão de curso apresenta uma proposta para um sistema de forno de refusão de processo dedicado à soldagem de componentes eletrônicos. Essa soldagem é realizada por forno de refusão de solda, o qual pertence ao processo de montagem de produtos eletrônicos com tecnologia de dispositivo de montagem em superfície SMD. Esses produtos, para o caso estudado, são aplicados no ramo automotivo, o qual exige que o processo produtivo tenha características destacadas de receptibilidade, padronização, além da preservação do meio ambiente. Para atender essas características o mencionado processo emprega a tecnologia de solda sem chumbo na soldagem de componentes.

Essa tecnologia exige que o perfil térmico possua uma média de temperatura maior sobre os elementos submetidos ao processo de soldagem, quando comparado com o processo tradicional estanho-chumbo. Para atender essa exigência o sistema de supervisão proposto neste trabalho adquire informações relacionadas a temperaturas das zonas do forno, integridade de resistência de aquecimento e sistema de insuflamento de ar quente. Essa aquisição é apresentada para o operador do sistema, por meio de uma Interface Homem-Máquina, com a meta de alertar sobre o atual estado do processo. Nessa interface é destacada a previsão de campos relacionados com registro contínuo das informações adquiridas, parametrização de informações de processo, alarmes visuais e sonoros.

1.1. Objetivo geral

Desenvolver uma máquina, que venha, controlar automaticamente o processo de refusão de solda de componentes SMD. Com a utilização de solda sem chumbo, criar uma necessidade de um processo, mas estável, com menos variações no processo, principalmente no perfil de temperatura usado. Todas as variáveis no processo serão comparadas e monitoradas pelo micro controlador num banco de dados adquiridos de forma que todos as variáveis de processo estejam dentro do limite do perfil de solda, garantindo assim a qualidade do produto final. Visando a utilização de equipamentos adequados que venha melhorar assim a eficiência do forno de refusão, utilizando os nossos conhecimentos adquiridos durante o curso e aperfeiçoá-los de acordo com a necessidade

1.2. Problema

A eletrônica tem possibilitado uma rápida evolução em todas as áreas do conhecimento. Medicina, mecânica, administração dentre outras fazem uso de aparelhos eletrônicos para desenvolvimento do conhecimento. A própria eletrônica vale-se dessa evolução. Componentes menores que consomem menos energia e que realizam suas funções em menor tempo possibilitam a criação de tecnologias cada vez melhores. E para que essas tecnologias sejam inseridas, difundidas e aproveitadas pelo mercado consumidor é necessário o aperfeiçoamento dos processos de montagem da indústria eletrônica. Processos de fabricação mais rápidos, baratos e eficientes são a chave para a difusão de qualquer tecnologia. A montagem de eletrônicos ganhou grande impulso na difusão com a adoção de processos automatizados de inserção e soldagem nos anos 60 e 70. Com o surgimento dos componentes SMD, no final da década de 70, a indústria modificou seus padrões e os processos ganharam ainda mais velocidade. A montagem de circuitos eletrônicos envolve várias etapas, podendo-se citar:

- Preparação de componentes.
- Inserção de componentes.
- Soldagem.
- Revisão e retrabalho.

Para que esses processos sejam bem-sucedidos é necessário o controle de várias condições, dentre elas estão a precisão e repetibilidade do posicionamento na inserção de componentes, a velocidade e a temperatura nas etapas de soldagem. O objetivo principal desse trabalho é o desenvolvimento de um equipamento para levantamento das curvas de temperatura nos fornos e máquinas de solda automatizadas no processo de soldagem em indústrias eletrônicas.

Notasse que as empresas de pequeno porte soldam componentes manualmente, que leva um longo período e nesse processo ocorre muitos danos por solda fria e terminais em curto, aumentando o prejuízo das empresas por retrabalho, prejudicando mais ainda quando essa placa chega aos clientes e brevemente apresentam falhas pelos mesmos motivos acima.

1.3. Justificativa

O desenvolvimento do forno de Refusão facilitara a mão de obra de pessoas que teria que fazer todo o processo de solda manualmente, é importante pois o tempo que é gasto no mesmo a empresa se beneficia em menos mão de obra para algo simples que só de colocar a placa dentro do forno com a pasta de solda em cada componente ela automaticamente solda várias placas em uma única vez.

1.4. Metodologia

Segundo Fontelles et al. (2008), um pesquisador que pretende desenvolver um estudo tem a necessidade de seguir uma sequência de raciocínio, “escolhendo dessa forma o tipo de pesquisa que esteja de acordo com o tema proposto”. A partir disto, podemos desenvolver diferentes maneiras de alcançar os objetivos.

Com base nos argumentos acima, podemos ver que um mesmo tipo de pesquisa pode ser executado de diferentes maneiras. Tendo como exemplo, um estudo observacional (tipo de pesquisa). Este tipo de estudo pode ser delineado como um estudo de corte ou como um estudo caso-controle, ambos definidos como delineamentos de características diferentes (FONTELLES et al., 2008).

Quanto à forma de abordagem, utilizaremos a qualitativa de natureza experimental, com relação aos procedimentos técnicos, iremos empregar o método de pesquisa bibliográfica. Neste caso, os objetivos serão abordados de forma exploratória, utilizando da pesquisa longitudinal para o desenvolvimento. Segundo Fontelles et al. (2008), uma pesquisa de natureza experimental, mostra o pesquisador atuando como um desenvolvedor, devido o mesmo altera os procedimentos ou dados contidos nesta pesquisa e intervém pela inclusão ou modificação de um determinado fator. No que se refere ao tipo de abordagem (FONTELLES et al., 2008) o autor afirma que, no caso de pesquisa qualitativa esse procedimento é para quem busca um entendimento em relação a fenômenos complexos, em profundidade de natureza. Com o problema desta pesquisa, “Como desenvolver um forno de refusão acessível que satisfaça os clientes e empresas de pequeno porte”, utilizar-se de referências bibliográficas. Sendo assim, os autores lançam mão do uso de material já publicado para assim, compor sua fundamentação teórica a partir de avaliação atenta de livros,

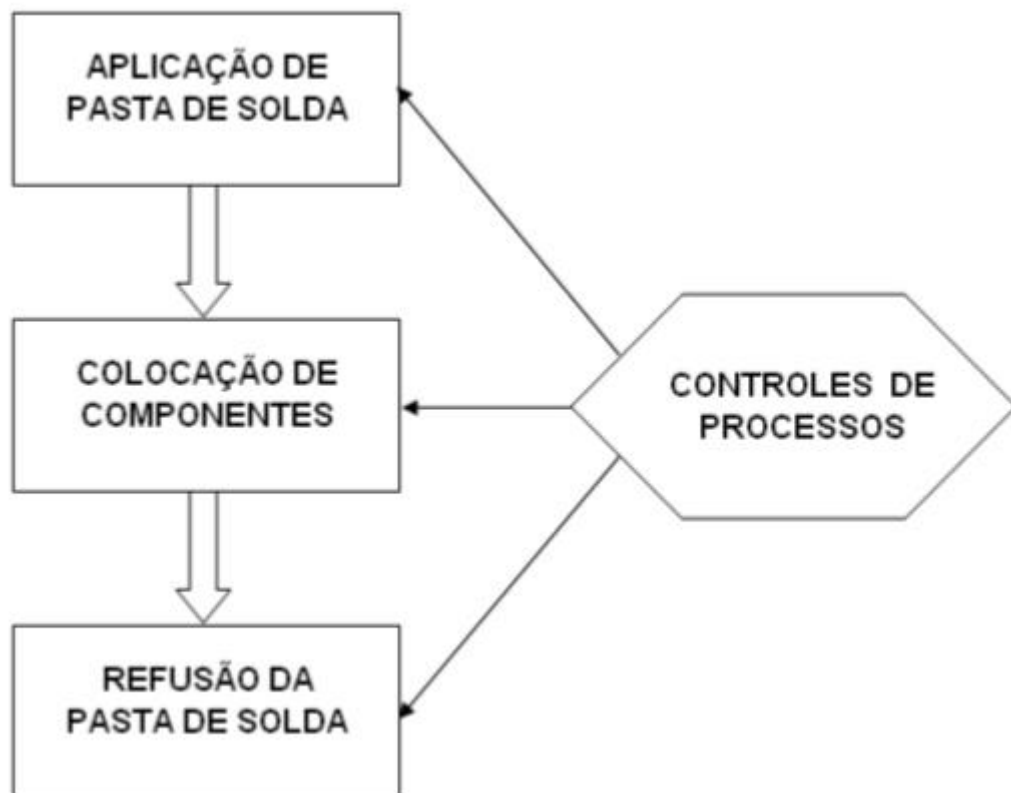
documentos, textos, mapas, fotos, manuscritos etc., em que estes materiais forneçam suporte a todas as fases de um protocolo de pesquisa (FONTELLES et al., 2008).

2. Referencial teórico

2.1. Processo de refusão estanho chumbo

A figura 1 mostra quais são as fases do processo atual de refusão estanho-chumbo.

Figura 1 - Processo atual.

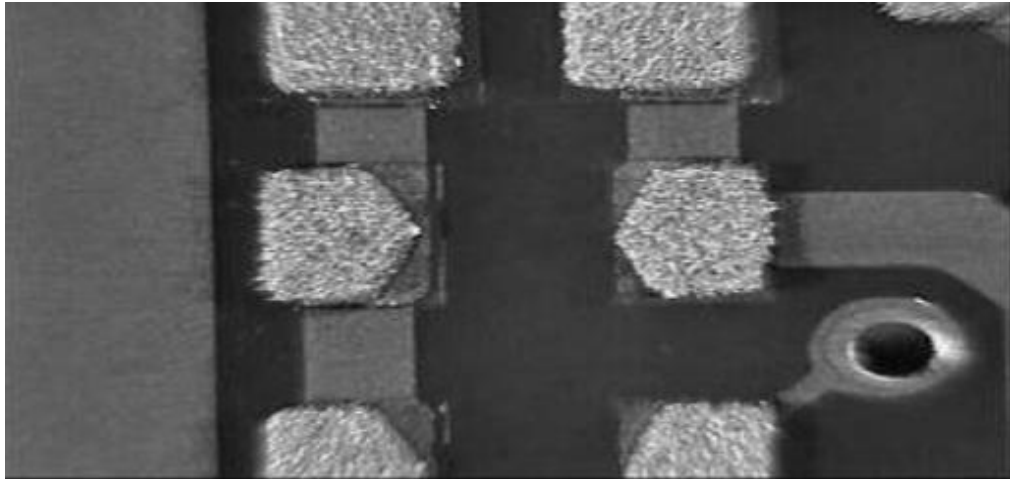


Fonte: Disponível (<http://www.bdttd.unitau.br>; acesso 27/04/18)

2.1.1. Aplicação de solda em pasta

Este processo é muito comum ao processo de silk-screen, que utiliza uma máscara metálica (chamada de Stencil), a aplicação é realizada manualmente ou automaticamente (screen printer), espalhando-se, no caso, a solda em pasta na máscara e a mesma é depositada sobre a placa de circuito impresso. Na figura 2 tem-se um exemplo de aplicação de solda em pasta. (JUSTI 2006).

Figura 2 - Aplicação de solda em pasta.



Fonte: Disponível (<http://www.btdtd.unitau.br>; acesso 27/04/18)

Ao usarmos solda em pasta ou creme de solda, muitas vezes não se tem o conhecimento de sua formulação, é importante ter uma exata noção de quais matérias envolvidos e suas propriedades. A solda em pasta deve ser aplicada por meio de suspensão ou maneira similares, com o apoio de uma espátula, lembrando que assim que a pasta for depositada na placa deve permanecer no mesmo local sem espalhar e deve ser suficiente pegajosa para possibilitar a fixação do componente a ser prensado de maneira que ele permaneça sua posição adequada durante o transporte até o destino da soldagem.

Quando a solda em pasta for utilizada em altas temperaturas, como, por exemplo, em uma *estação de retrabalho*, as partículas devem se fundir para formar uma única massa de solda e assim executar a fixação do componente eletrônico.

(Disponível: <http://btdtd.ibict.br>, acesso 27/04/2018)

2.1.2. Colocação de componentes

A colocação de componentes SMD é realizada por máquinas automáticas (figura 3). Basicamente é utilizado dois tipos diferentes, um para colocação dos componentes maiores (circuitos integrados, conectores, etc.), outro para colocação de componentes menores (resistores, capacitores, etc.).(JUSTI 2006).

Figura 3 - Colocação de componentes.



Fonte: Disponível (<http://www.btdt.unitau.br>; acesso 27/04/18)

(Disponível: <http://btdt.ibict.br>, acesso 27/04/2018)

2.1.3. Refusão de solda

A refusão de solda é feita em um forno industrial (figura 4) composto por várias pontos de aquecimento, que permitem o ajuste do perfil térmico desejado para o produto em produção. No processo estanho-chumbo o pico de temperatura está próximo de 225°C. (WASSINK 1989).

Figura 4 - Forno de refusão de solda.



Fonte: Disponível (<http://www.btdt.unitau.br>; acesso 27/04/18)

(Disponível: <http://btdt.ibict.br>, acesso 27/04/2018)

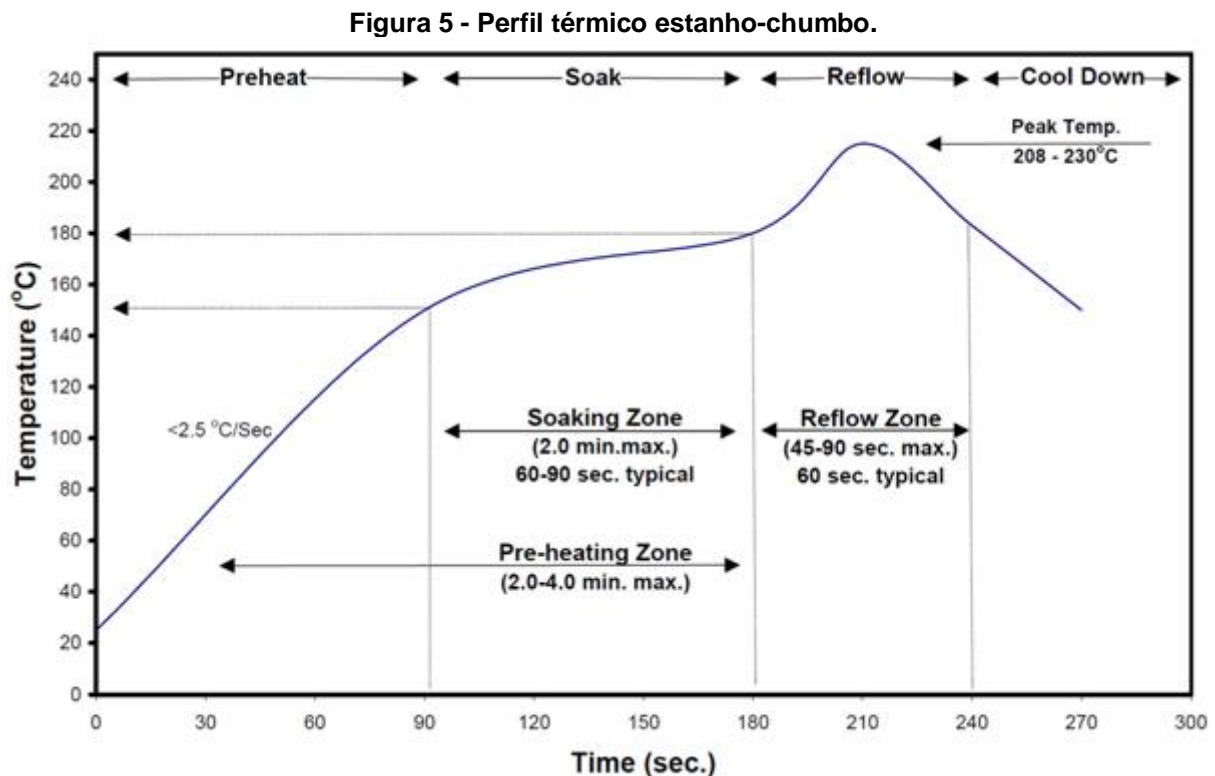
2.1.4. Controle Atual

Além das manutenções preventivas e preditivas realizadas no equipamento, na fase de aplicação de solda em pasta, têm-se basicamente dois tipos de controle de processo:

- Volume de solda aplicada nos pads (ilhas de contato) da placa de circuito impresso;
- Centralização da solda.

Com relação à colocação de componentes, o principal controle está na centralização do componente, que está relacionada ao sistema de coordenadas X-Y-Z do equipamento. O processo de refusão de solda não se baseia exclusivamente na temperatura aplicada nas zonas do forno industrial, pois existem outros fatores que são igualmente importantes para que se tenha o perfil térmico adequado ao produto, entre eles pode-se citar, velocidade da esteira transportadora de placas de circuito impresso, a integridade das resistências elétricas, além das condições dos insufladores de ar. (JUSTI 2006).

Atualmente para obter o perfil térmico, usa-se um equipamento coletor de dados, que é composto por uma placa com memória acoplada a vários sensores (termopar), que registra a temperatura ao longo do tempo. Estas informações são posteriormente descarregadas em um micro computador. A figura 5 é um exemplo de gráfico gerado a partir de dados coletados em um forno de refusão de solda. (JUSTI 2006).



Fonte: Disponível (<https://www.bga-reworking.co.uk>; acesso 27/04/18)

O problema com este tipo de controle é que ele é baseado na experiência e no tempo disponível do responsável pelo processo de manufatura, sendo que o espaço de tempo entre uma verificação e outra pode variar desde algumas horas, dias, semanas ou até meses. Caso algum problema ocorra entre uma verificação e outra, existe a probabilidade de que todos os produtos manufaturados neste período de tempo tenham algum problema de qualidade que só será observado no cliente final, gerando o desgaste da imagem da empresa. Mesmo que seja detectado durante a inspeção no final da linha de produção, também já é tarde, pois gerará reparos e possíveis refugos, aumentando o custo de produção. (JUSTI 2006).

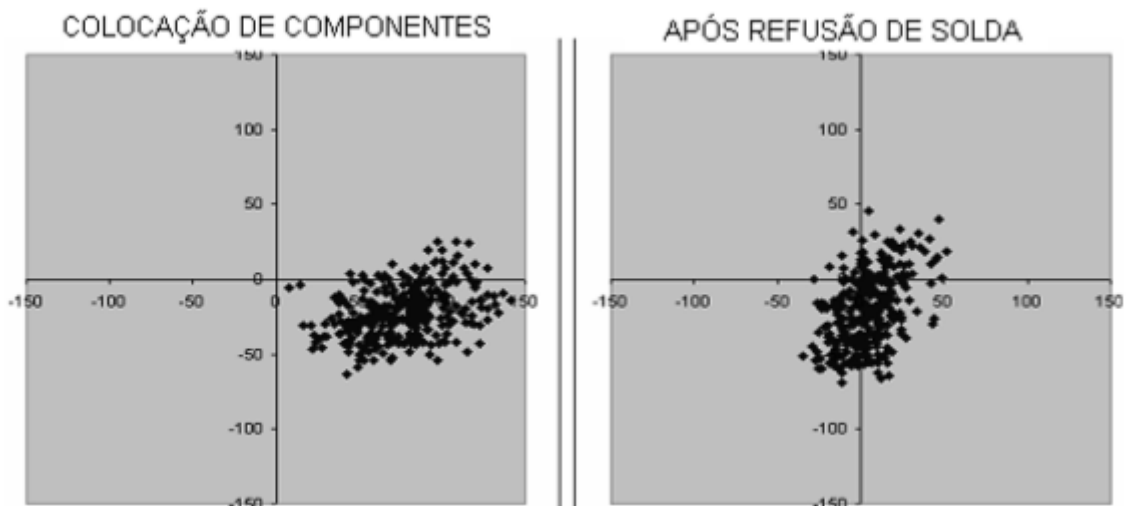
(Disponível: <http://btdt.ibict.br>, acesso 27/04/2018)

2.2. Novo processo de refusão de solda lead free

Assim como no processo estanho-chumbo, no lead free existem as mesmas três etapas de manufatura. Com relação às duas primeiras etapas a principal característica técnica que deve ser observada diz respeito à precisão, tanto da aplicação da solda, como da colocação de componentes. Pois na solda lead free não ocorre o efeito conhecido como “auto alinhamento” ou “auto ajuste” durante a refusão da solda, de forma que quando a pasta ou os componentes ficam mal posicionados pode se ter o efeito chamado tombstone.. (MORIYA 2006).

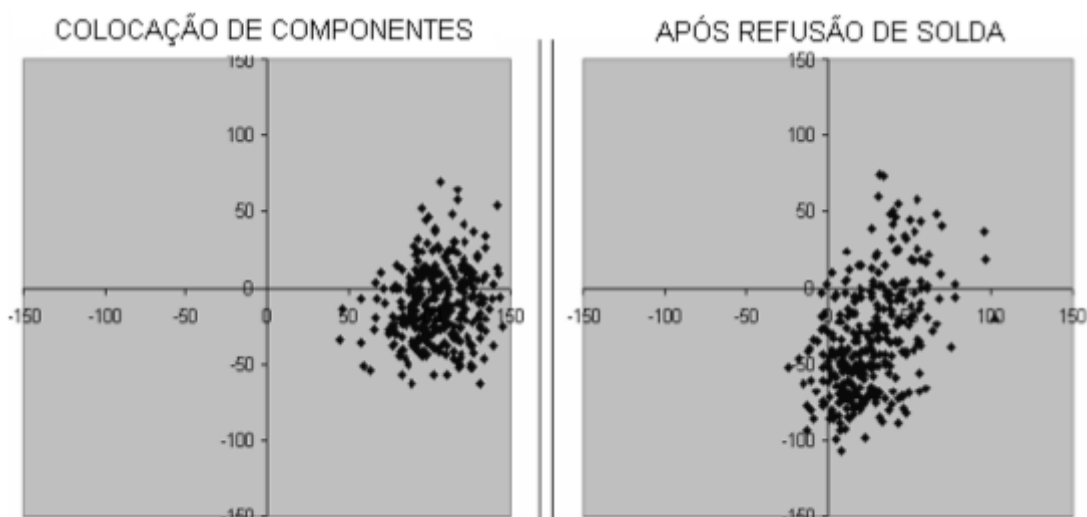
Pode-se verificar nos gráficos comparativos (gráfico 1 e gráfico 2) que no processo estanho-chumbo o efeito de auto alinhamento está potencializado, enquanto que no processo lead free este efeito ocorre de forma mais dispersa. Estes gráficos representam a precisão de colocação dos componentes (quanto mais próximo do zero, mais preciso foi a colocação do componente) e a posição final do componente após a soldagem (da mesma forma, quanto mais próximo do zero, mais preciso foi a posição final do componente). (MORIYA 2006).

Gráfico 1 - Auto alinhamento processo Estanho-Chumbo.



Fonte: Disponível (<http://www.abraci.org.br>; acesso 27/04/48)

Gráfico 2 - Auto alinhamento processo LeadFree.



Fonte: Disponível (<http://www.abraci.org.br>; acesso 27/04/48)

Na terceira etapa do processo, a refusão de solda, a principal característica que é alterada é o perfil térmico, que passou a ser mais agressivo, chegando a ter picos de temperaturas variando entre 235 e 255° C (figura 5), o que tornou esta etapa do processo muito delicada devido à sensibilidade dos componentes eletrônicos quanto à elevação da temperatura.

Até 245° C tem-se uma faixa segura para todos os tipos de componentes. De 245 a 250° C tem-se uma faixa crítica para os componentes ativos (circuitos integrados, etc.). De 250 a 260° C é uma faixa crítica para os componentes passivos (resistores, capacitores, etc.). O que torna a janela de processo extremamente estreita exigindo que haja um controle de processo constante e rigoroso. (MORIYA 2006).

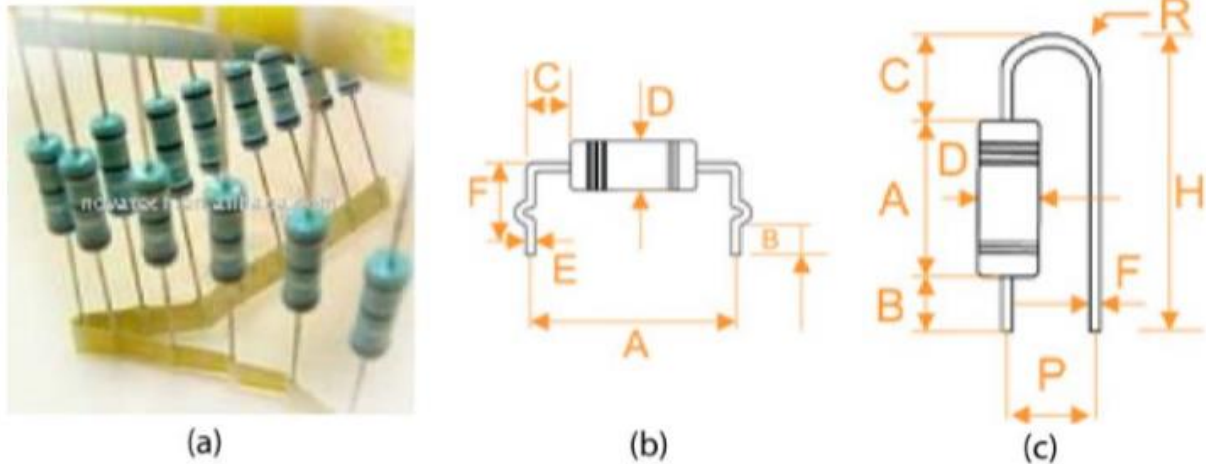
(Disponível: <http://btdt.ibict.br> acesso 27/04/2018)

2.2.1. Preparação

Antes que a soldagem dos componentes ocorra vários processos são realizados a fim de garantir sucesso na montagem como um todo. A separação de componentes e posicionamento na linha de montagem (seja automática ou manual) é comum em qualquer tecnologia utilizada (THT ou SMT). A diferença aparece na apresentação e acondicionamento dos componentes de cada tecnologia. Nas montagens THT manuais é usual os componentes serem pré-formados antes da apresentação na linha de montagem. Isso envolve a remoção dos componentes de seus carretéis de acondicionamento, conformação e ajuste do tamanho de seus

terminais. Na Figura 6a é a apresentação padrão de resistores THT e nas Figura 6b e 6c exemplos de pré-formações.

Figura 6 - (a) Apresentação dos resistores em carretel. (b) e (c) Exemplos de pré-formação.



Fonte: Disponível (professorpetry.com.br; acesso 05/05/18)

Essas pré-formações podem ser feitas com maquinário apropriado, no caso resistores, diodos, capacitores e demais componentes que sejam comercializados em fitas ou carretéis. Componentes comercializados a granel serão processados manualmente. Nas montagens THT automatizadas e SMT não são necessárias essas pré-formações previamente. No caso da THT, a própria inseridora realiza essa tarefa e no caso SMT realmente é desnecessário pois os componentes não possuem terminais e tem dimensões padronizadas, assumido por. Outro componente da montagem que necessita cuidados são as placas de circuito impresso (PCI ou do inglês PCB, Printed Circuit Board). Dependendo do local de armazenamento é necessária uma limpeza da placa para facilitar a ação do fluxo durante a soldagem.

(Disponível: <http://www.professorpetry.com.br>, acesso 05/05/2018)

2.2.2. Montagem

A montagem é caracterizada pela inserção e soldagem dos componentes em um substrato. A inserção pode ser manual ou automatizada, dependendo da capacidade produtiva desejada. Feita a inserção, deve ocorrer a soldagem dos componentes ao substrato. Essa soldagem também pode ser manual ou automatizada.

Aqui serão apenas tratados os casos que envolvam as partes automatizadas do processo, mesmo que etapas manuais estejam envolvidas em outro período.

Uma vez preparados para a inserção, os componentes THT são movimentados para sua posição na PCI. Após a inserção automatizada as placas seguem para a inserção manual de componentes maiores, fora da faixa de trabalho das máquinas, e finalmente para a máquina de soldagem automática.

As placas montadas com SMT primeiramente passam pelo processo de deposição de pasta de solda ou deposição de adesivo. Tanto a pasta quanto o adesivo servirão como fixação mecânica do componente à placa, porém a pasta de solda ainda realiza a conexão elétrica entre trilha e componente.

Após a etapa de deposição os componentes são inseridos também pelas máquinas e após isso todo o conjunto segue para o forno de refusão, calibrado para fusão da pasta de solda ou cura do adesivo.

Após o forno, essas placas podem voltar à etapa de deposição para que mais componentes sejam posicionados no outro lado da PCI, e o processo se repete, adentrar uma linha de inserção de componentes THT, ou serem enviadas para a máquina de solda automática. Tudo isso dependendo da densidade e variedade de componentes que povoam a placa. Finalmente, após a inserção e soldagem de todos os componentes a placa montada é enviada para revisão e retrabalho caso necessário.

Esses são os passos, em linhas gerais, nas linhas de montagens de placas eletrônicas. Os equipamentos utilizados serão discriminados na próxima seção. Tanto no processo THT quanto SMT, o fator temperatura é de vital importância. Seja no processo de cura do adesivo ou de fusão da pasta de solda em SMT, ou na rampa de aquecimento no processo de solda automatizada em THT é necessário o controle da quantidade de energia transferida em forma de calor.

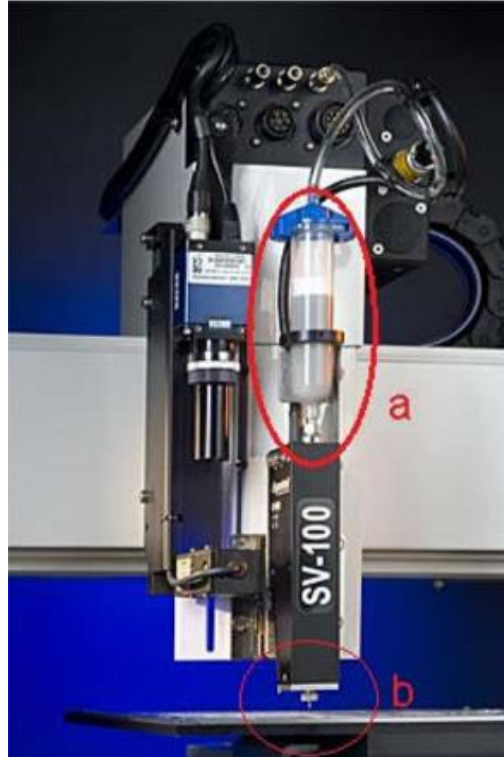
(Disponível: <http://www.professorpetry.com.br>, acesso 05/04/2018)

2.2.3. Equipamentos

Como já citado, a primeira etapa da montagem SMT é a deposição de pasta ou adesivo sobre a PCI. Essa deposição pode ser feita por dispensadoras, ou dosadoras, ou por transferência através de estêncil. As dispensadoras usualmente são acopladas como módulos auxiliares nas máquinas de inserção de componentes.

Na Figura 7 está um exemplo de dispensadora. Na região destacada “a” está o depósito de pasta de solda e em “b” o bico de injeção do mecanismo.

Figura 7 - Exemplo de dispensadora. Região “a” destaca o reservatório de pasta de solda (podendo ser substituído por adesivo). Região “b” destaca o bico de injeção.

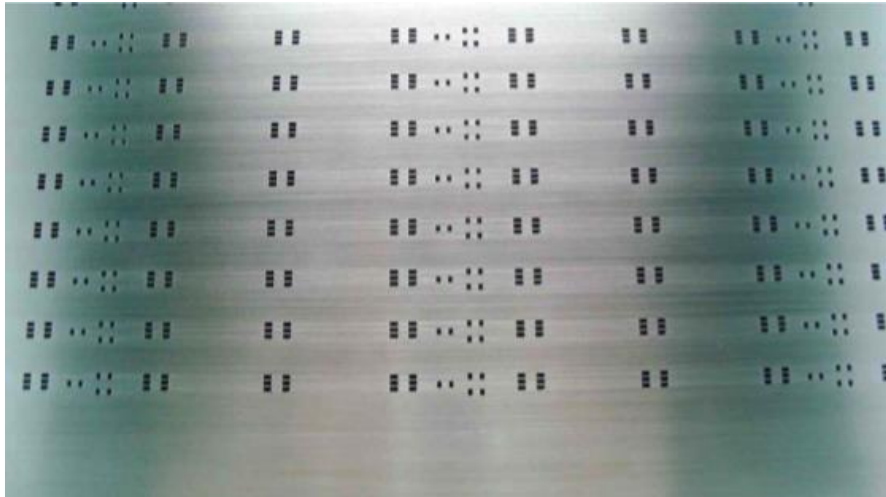


Fonte: Disponível (<http://www.smtnet.com>; acesso 10/05/18)

A maioria das máquinas de inserção trabalha baseada num sistema de eixos cartesianos onde são definidas as posições das ilhas (os contatos) de cada tipo de componente. De posse das coordenadas de cada ilha, a máquina deposita uma pequena quantidade de pasta, calibrada pelo operador da máquina, em cada ilha. Esse método tem pequenas desvantagens como a velocidade de deposição (cada ponto é depositado individualmente) e constante necessidade de renovação do reservatório (baixo volume disponível).

O método para deposição usando estêncil é comumente chamado de “stencil printer”. O processo ocorre quando a pasta de solda ou adesivo é pressionado através de pequenas aberturas feitas numa tela apropriada. As aberturas correspondem aos pontos onde existem ilhas na placa, ou posição para deposição de adesivo. A Figura 8 ilustra um estêncil para essa finalidade.

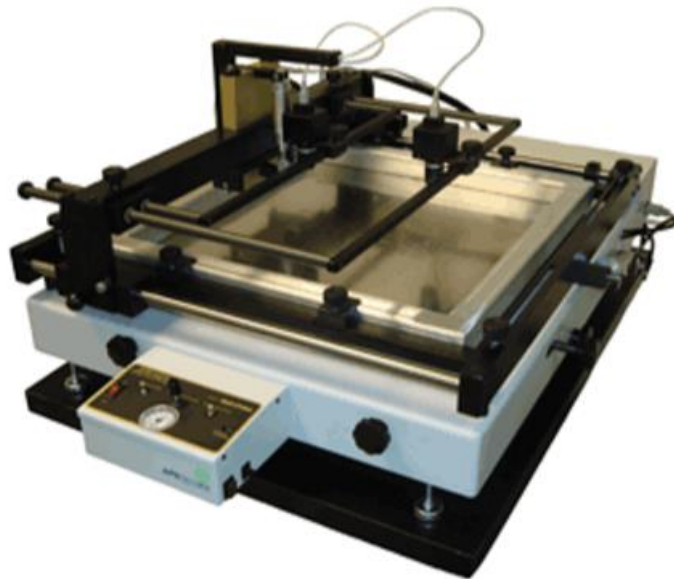
Figura 8 - Exemplo de estêncil usado para deposição.



Fonte: Disponível (<http://www.smtnet.com>; acesso 10/05/18)

O processo que faz uso do estêncil tem muitas semelhanças com as técnicas de “silk screen”, usadas para decoração de camisetas e outras peças de vestuário. Na Figura 9 é apresentada uma impressora semi-automática para uso com essa técnica de deposição.

Figura 9 - Impressora semi-automática.



Fonte: Disponível (<http://www.nei.com.br>; acesso 10/05/18)

Após a deposição de pasta ou de adesivo é feita a inserção dos componentes. As máquinas de inserção coletam os componentes de suas embalagens de acondicionamento e posicionam em lugares previamente indicados.

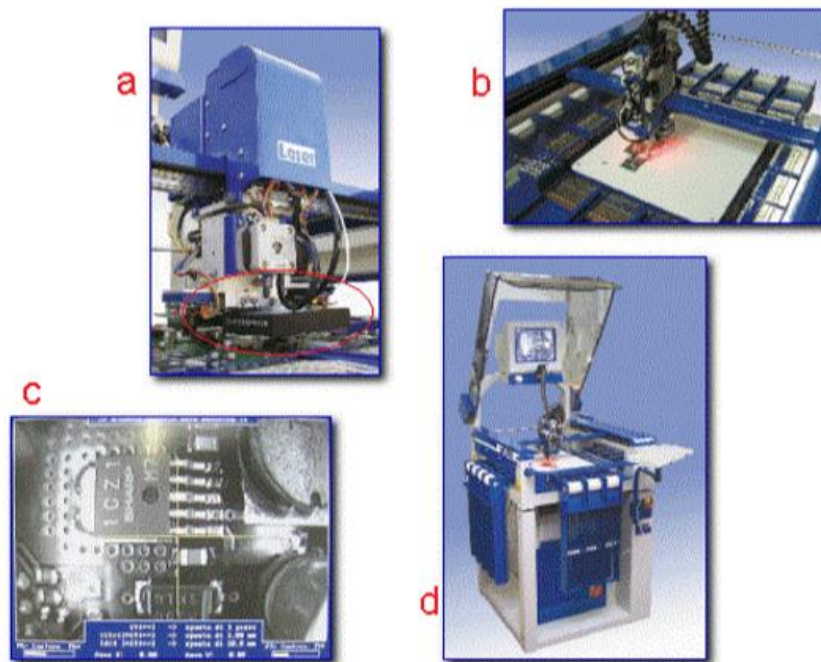
A remoção e inserção dos componentes são feitas com o uso de ações pneumáticas. Pressão negativa é gerada para remoção dos componentes de suas embalagens. Para seu posicionamento, essa pressão negativa é removida ou até

mesmo invertida para uma pequena pressão positiva, de modo que os terminais do componente sejam englobados pela pasta depositada anteriormente ou que o corpo do mesmo tenha contato suficiente com o adesivo.

As posições são dadas por coordenadas, como num plano cartesiano, e dependendo da máquina pode ter redundância através de um sistema ótico. O sistema ótico funciona através do reconhecimento de padrões previamente configurados na máquina.

A Figura 10 traz alguns detalhes de um modelo de inversora para componentes SMT. A figura 10a ilustra a cabeça de inserção e no detalhe a câmera usada no sistema de realimentação ótico. Na Figura 10b uma visão geral da mesa de inserção.

Figura 10 - Exemplo de máquina impressora SMT.



Fonte: Disponível (<http://www.nei.com.br>; acesso 10/05/18)

Na Figura 10c é mostrada a tela do sistema de realimentação ótica e finalmente na Figura 10d o aspecto da máquina.

Completa a inserção dos componentes na placa, o conjunto segue para o forno de refusão, para cura do adesivo ou fusão da pasta de solda. Os fornos de refusão possuem algumas tecnologias diferentes empregadas para a transmissão de energia. São elas emissão de Infravermelho ou convecção de calor.

Na emissão de infravermelho, lâmpadas especiais são posicionadas dentro das câmaras do forno e configuradas a emitir doses quantificadas de radiação. Ao atingir o conjunto (placa com componentes) a radiação é absorvida e transformada em calor. O problema dessa técnica consiste na variedade de materiais utilizados no conjunto. Cobre, fibras e silício entre outros tem diferentes taxas de absorção de radiação, fazendo assim com que existem diferenças de temperatura na placa gerando “pontos quentes” que levam a maior estresse térmico os componentes e mesmo o substrato da placa.

Os fornos que trabalham com convecção de calor usam elementos resistivos e ventiladores para a transmissão de energia. Em cada câmara do forno existe um elemento e ventiladores para geração de um fluxo de ar que atinge a placa. Na Figura 11 exemplifica um forno de refusão.

Figura 11 - Exemplo de forno de refusão.



Fonte: Disponível (<http://www.madelltech.com>; acesso 10/05/18)

Para evitar o estresse térmico que ocorre nos fornos infravermelhos, os fornos por convecção possuem diversas áreas de pré-aquecimento e uma área onde se atinge o pico de temperatura, que é a área onde ocorre a fusão da pasta de solda.

Após essa área, existem ainda zonas para resfriamento da placa. Tanto a recepção de muita energia (aquecimento) quanto à remoção da mesma (resfriamento) de forma muito abrupta gera o dito estresse térmico na placa. Por isso a necessidade de áreas de aquecimento e resfriamento nesse equipamento.

O processo também pode incluir algum componente THT, requisitando assim soldagem em outro tipo de maquinário. Assim, essa solda pode ser feita de modo manual, em tanque estático (semiautomático ou automático) ou máquina de onda. Na Figura 12 é mostrada uma máquina de solda de onda.

Figura 12 - Máquina de solda automática.



Fonte: Disponível (www.professorpetry.com.br; acesso 10/05/18)

(Disponível: <http://www.professorpetry.com.br>, acesso 10/05/2018)

2.2.4. Processos de soldagem

Apresentadas as etapas da montagem e o principal maquinário utilizado para o processo globalmente, volta-se o foco para o processo de soldagem propriamente dito.

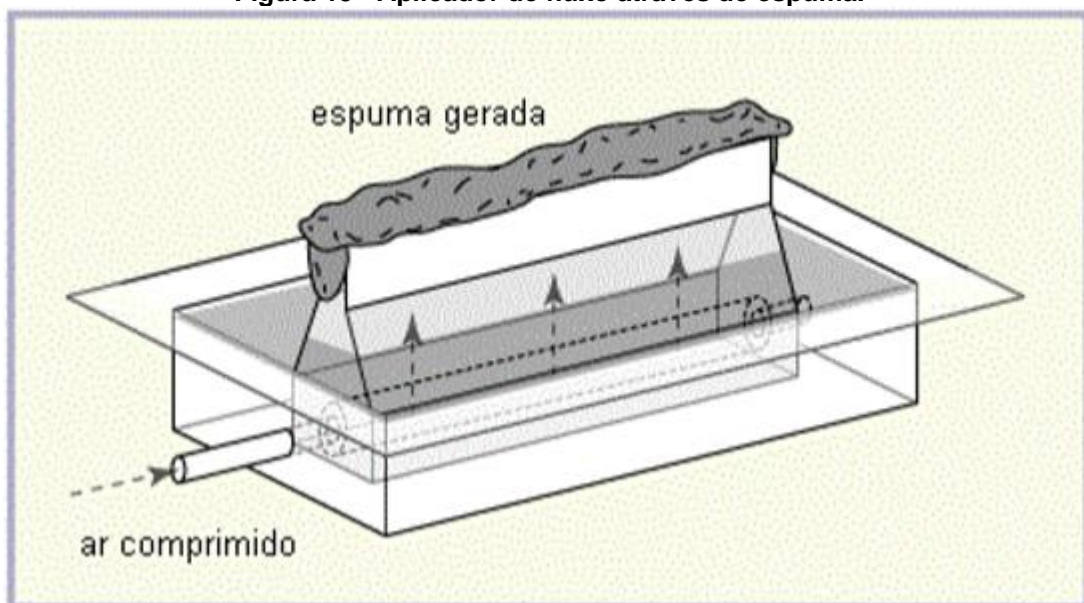
Como tratado no capítulo 2, a soldagem é um processo metalúrgico para união de dois ou mais metais fazendo uso de um terceiro elemento também metálico. Esse elemento pode ser uma liga ou metal puro. Para que o processo seja executado com sucesso é necessária a transmissão de energia suficiente para que o elemento que realizará a união possa criar camadas intermetálicas com os metais base (que são os metais a serem unidos) fortes o suficiente, comparáveis aos metais base.

A transmissão de energia ocorre através do aquecimento das partes envolvidas no processo, metais base e elemento de solda. O maquinário apresentado na sessão anterior necessita controlar a temperatura e velocidade com as quais o conjunto placa e componentes passa por cada uma de suas áreas. Tanto os fornos de refusão quanto as máquinas de solda possuem áreas de pré-aquecimento para que a energia seja transferida de forma gradual sem gerar estresse para a placa e para os componentes.

Nos processos SMT, a pasta de solda depositada previamente contém os elementos necessários para a soldagem. Os principais elementos são o fluxo e a liga metálica existente. Existem vários tipos de pasta, definidos pela granulação, composição da liga metálica e tipo de fluxo. Para cada tipo de pasta o fabricante define métodos de armazenagem e preparação para deposição. Em linhas gerais as pastas devem ser armazenadas refrigeradas e a preparação envolve reaquecimento lento (acima de 8 horas dependendo do fabricante) e a re-mixagem da pasta.

Na soldagem THT por máquina fluxo e liga estão separados. Ao adentrar a máquina de solda, a primeira estação aplica o fluxo. Essa aplicação pode ser por spray ou com espuma. Na aplicação por spray, pequenos bicos de injeção pulverizam o fluxo diretamente contra a placa. Na aplicação com espuma o fluxo é aerado a ponto de formar uma espuma densa o suficiente para espalhar o mesmo na placa. A Figura 13 traz uma ilustração da aplicação em espuma.

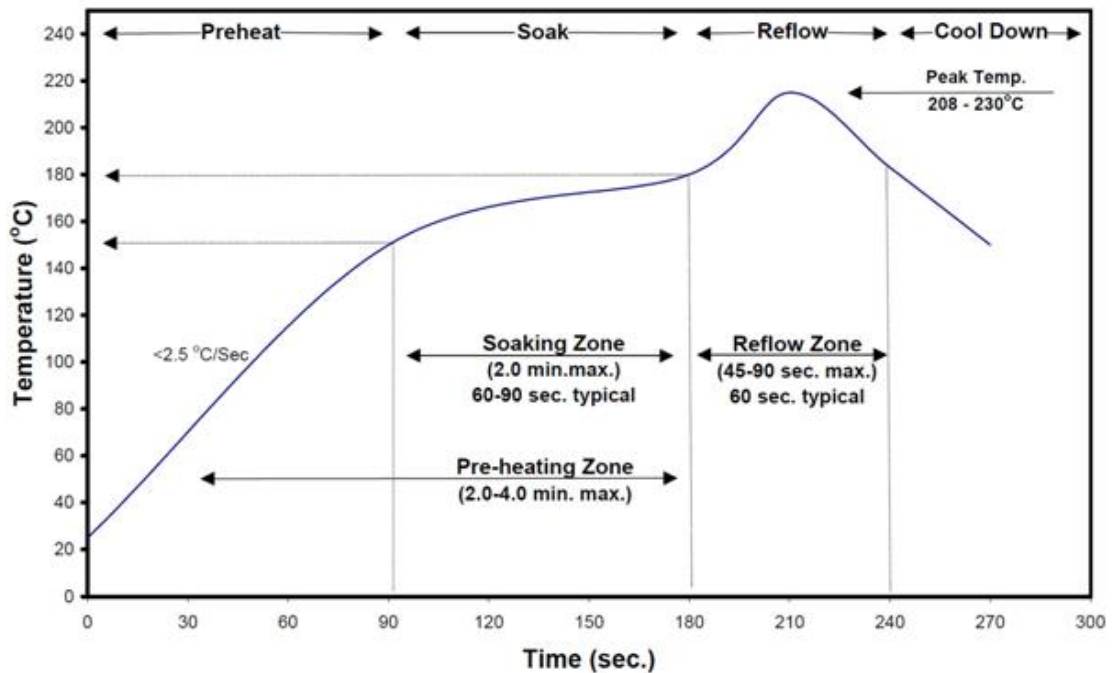
Figura 13 - Aplicador de fluxo através de espuma.



Fonte: Disponível (www.ami.ac.uk; acesso 13/05/18)

Ambas a tecnologias, SMT e THT, precisam de aquecimento gradual como já citado. Para isso os fornos de refusão e as máquinas de soldagem têm várias áreas de pré-aquecimento. Não só apenas a temperatura deve ser considerada, mas também o tempo de exposição da placa em cada área, configurado pela velocidade com a qual a mesma atravessa a área. Na Figura 14 é apresentado um modelo idealizado da curva de temperatura do forno de refusão.

Figura 14 - Exemplo idealizado de curva de aquecimento de forno de refusão.



Fonte: Disponível (<https://www.bga-reworking.co.uk/knowledge-base.html>; acesso 13/05/18)

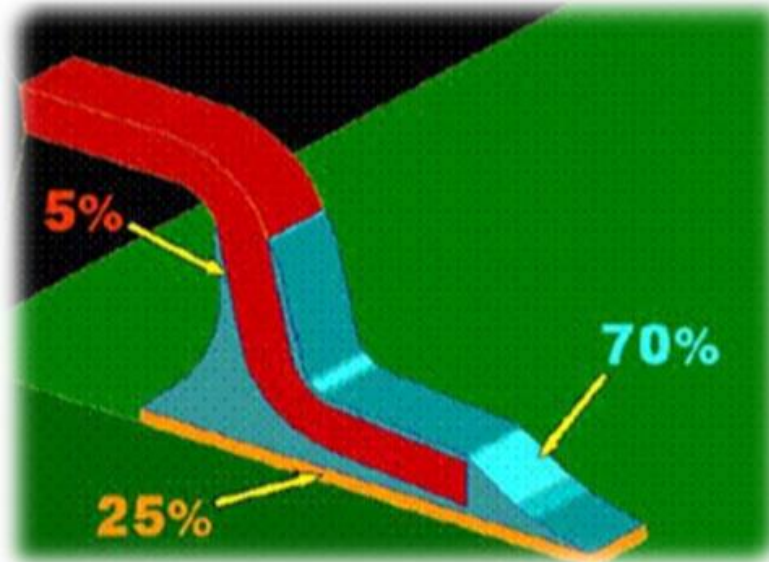
Para as máquinas de solda THT o perfil da curva é praticamente o mesmo, porém devem ser feitas leituras conforme a placa a ser soldada, uma vez que a densidade de componentes na placa tem influência direta na absorção de energia.

Cada uma das áreas tem suas funções e necessidades. A “Zona de pré-aquecimento” tem a função de transmitir a maior parte da energia necessária para o conjunto a ser soldado. Porém, sem excesso, pois também é necessário tempo para a atuação do fluxo e o mesmo evaporar. Caso isso não seja levado em consideração, o fluxo pode não ter tempo de limpar os contatos e facilitar a molhabilidade da solda, gerando soldas ruins. O fluxo também pode formar bolhas e explodir, deslocando os componentes.

A “Zona de Cura” tem esse nome pois é nessa área que ocorre a evaporação do fluxo ou o início da cura do adesivo, quando do uso desses. Um detalhe sobre o uso de adesivos é que sua curva é mais plana no topo. Para os adesivos não é necessário um pico de energia (calor), apenas que se mantenha um nível de temperatura por um período determinado pelo fabricante para a total cura do adesivo.

Considerando que o fluxo foi devidamente espalhando e ativado pelo calor, chega-se a “Zona de Refusão”. Aqui um pico de energia é dado para a pasta, fazendo com que a o metal da liga funda e forme as camadas intermetálicas, realizando a soldagem propriamente dita. Na Figura 15 é exibido um perfil de solda ideal.

Figura 15 - Perfil de solda ideal quando a deposição de material da liga.



Fonte: Disponível (<http://smd-on-line.com>; acesso 13/05/18)

A “Zona de Refusão” nos processos que envolvem THT ou soldagem por máquina de onda é a zona onde há o contato do conjunto placa e componentes com a liga já fundida. Esse contato pode se dar de três meios, onda plana, onda dupla ou soldagem seletiva.

O sistema de onda simples é exibido na Figura 16. Uma bomba é usada para impulsionar a liga já liquefeita para cima e para dentro de um bocal instalado no tanque de solda. Esse sistema gera uma onda controlada da liga fundida, comumente utilizado quando somente são utilizados componentes THT.

A parte da onda que se desloca contra o movimento da placa tende a ajudar a soldagem, forçando o contato entre os pontos de solda e a liga. A solda que flui no

mesmo sentido que a placa ajuda a remover excessos de solda dos terminais e ainda empurra as oxidações de volta para a superfície do tanque.

Figura 16 - Sistema de soldagem de onda plana, também conhecido como onda simples ou onda lambda. (a) Foto e (b) Corte explicativo.



Fonte: Disponível (<http://www.ami.ac.uk>; acesso 13/05/18)

O sistema de dupla onda funciona basicamente pelo mesmo princípio que o de onda plana. Bombas impelem a liga fundida para dentro de bocais na superfície do tanque. A Figura 17 apresenta o sistema. Esse sistema introduz uma onda com superfície turbulenta, ideal para aplicações onde os componentes SMT são colados no lado de solda da placa.

A onda turbulenta tem maior energia cinética que a onda plana, assim consegue impulsionar gotas de material em “pontos cegos” da onda plana. Essa por sua vez permanece nesse sistema com as mesmas funções do sistema de onda simples.

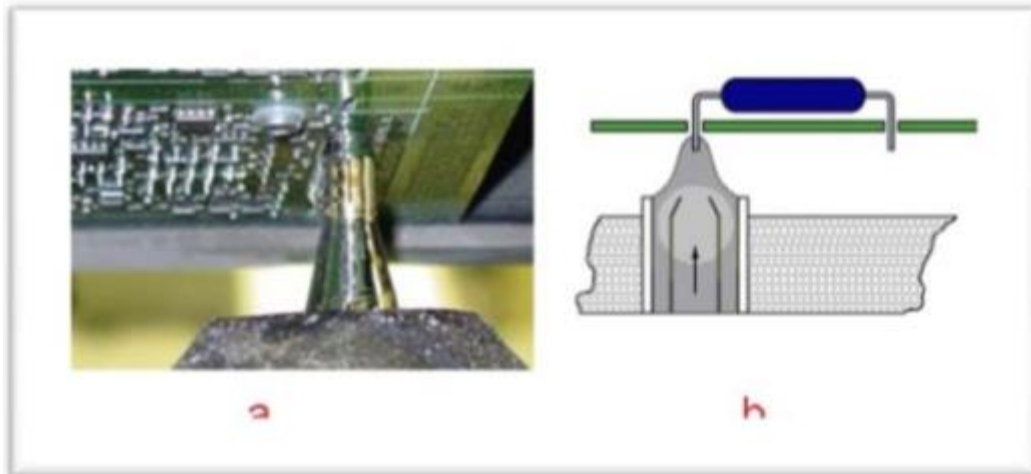
Figura 17 - Sistema Dupla Onda. (a) Foto e (b) Corte explicativo.



Fonte: Disponível (<http://www.ami.ac.uk>; acesso 13/05/18)

A soldagem seletiva é utilizada quando existem componentes sensíveis ao possível estresse térmico gerado pelos métodos anteriores e a quantidade de pontos que necessitam soldagem é pequena. A Figura 18 ilustra esse sistema.

Figura 18 - Sistema de solda seletiva.



Fonte: Disponível (<http://www.professorpetry.com.br>; acesso 13/04/18)

Nesse sistema um pequeno bocal se desloca, abaixo da placa, pelos pontos a serem soldados. A liga fundida é também impulsionada por bombas através do bocal. Na figura 18a uma máquina em atividade e na figura 18b um corte explicativo do sistema.

(Disponível: <http://www.professorpetry.com.br>, acesso 13/05/2018)

2.2.5. Revisão e Pós-Processamento

Após o processo de montagem é comum que exista uma etapa de revisão para combater possíveis falhas no processo. Os principais defeitos encontrados no processo com SMT são curtos entre terminais, falha na refusão, vazios de solda, solda granulada, componentes levantados, trincas ou fissuras na solda e aparecimento de bolas de solda. A figura 19 traz exemplos desses defeitos.

Os curtos entre terminais podem ter várias origens, tais como alinhamento deficiente entre estêncil e PCI, estêncil com resíduos de pasta de solda, parâmetros de Impressão incorretos, excesso de pasta de solda na abertura do estêncil, escorrimento da pasta de solda, terminais dos componentes desalinhados ou avariados, posicionamento do componente, ilhas subdimensionados.

As falhas na refusão são dadas principalmente por qualidade da pasta de solda comprometida, excesso de oxidação nas ilhas e componentes e perfil térmico incorreto.

Os vazios de solda têm principais causas perfil térmico incorreto, especialmente no pré-aquecimento e pré-refusão. Não ocorre uma evaporação correta do solvente do fluxo e seleção incorreta da pasta de solda.

O aparecimento de solda granulada após a refusão deve-se a má qualidade da PCI, movimentos durante o resfriamento (trepidação de esteira transportadora) ou quantidade insuficiente de fluxo.

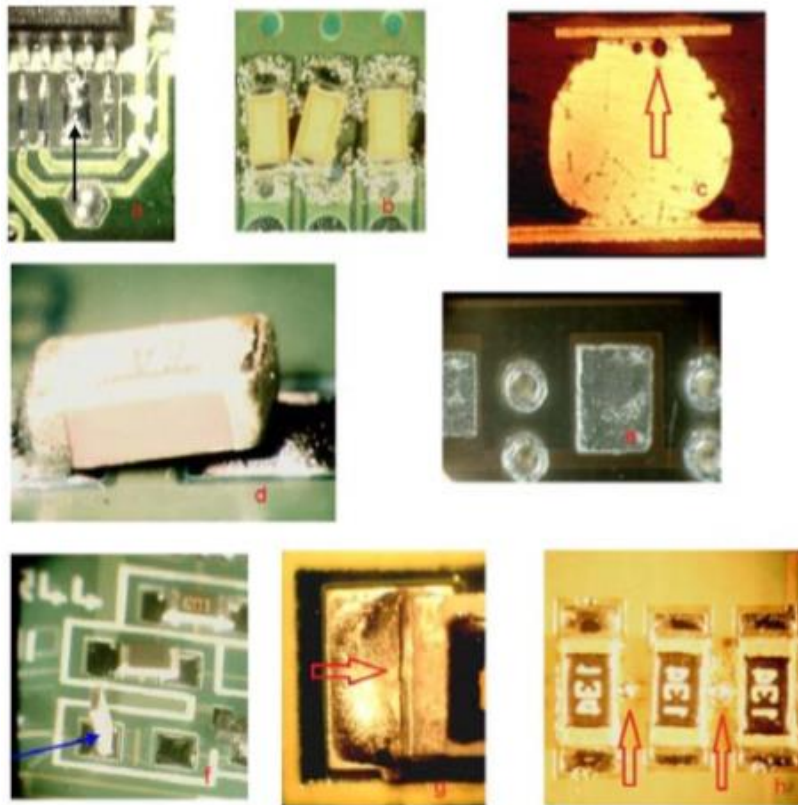
Os componentes levantados e Tombstoning são frequentes quando o depósito de pasta de solda é irregular, ou seja, existem variações da pressão aplicada durante impressão.

Outras causas comuns são vibração durante o resfriamento, tempo muito longo acima da fase líquida, durante a refusão, ilhas usadas como dissipadores térmicos, terminais contaminados ou oxidados, um lado do componente é soldado antes do outro, geometria desigual das ilhas, as ilhas e/ou terminais estão desnivelados.

A soldagem inconstante, a máscara de solda está avançando sobre a ilha - diminuindo assim a superfície de soldagem ou a máscara de solda está sob o componente também são motivos para aparecimento de componentes levantados.

As fissuras, ou trincas na solda, aparecem quando o perfil térmico é incorreto, a taxa de aquecimento do componente é muito íngreme, a temperatura do componente é muito alta, a taxa de resfriamento é muito longa ou ocorre crescimento exagerado da camada intermetálica.

Figura 19 - Falhas comuns no processo de soldagem SMT.



Fonte: Disponível (<http://smd-on-line.com>; acesso 15/05/18)

Na figura 19 - Falhas comuns no processo de soldagem SMT. (a) Curto entre terminais, (b) Falha na refusão, (c) Vazio de solda, (d) Componentes levantados, (e) Solda granulada, (f) Tombstoning – Componente levantado na horizontal, (g) Trincas e (h) Bolas de solda.

Os principais problemas nos processos envolvendo THT são a falha na solda por inexistência de solda, curtos entre terminais e levantamento dos componentes (o terminal sai do seu furo). Os principais motivos são o perfil térmico incorreto, oxidações severas nas ilhas e/ou terminais e projetos mal executados.

Todos esses problemas devem ser eliminados por ações corretivas em diversas etapas do processo. Porém uma pequena quantidade é aceita por norma, devendo então ser retrabalhada através de equipamentos adequados. O tradicional ferro de solda, acompanhado de estações de soldagem controlada, microscópios, lentes de aumento, estação de reballing, no caso de componentes BGA, e máquinas de soldagem seletiva são essenciais nessa tarefa.

(Disponível: <http://www.professorpetry.com.br>, acesso 15/05/2018)

2.3. Temperatura

Temperatura é a grandeza física que mede o estado de agitação das partículas de um corpo, caracterizando seu estado térmico. Sendo sem dúvida uma das variáveis mais importantes nos processos industriais.

2.3.1. Escalas Termométricas

São escalas que permitem medir certa temperatura. São mais conhecidas como Celsius, Fahrenheit e Kelvin.

$$\frac{T_c}{5} = \frac{T_f - 32}{9} = \frac{T_k - 273,15}{5}$$

Onde:

T_c = Temperatura em Celsius

T_f = Temperatura em Fahrenheit

T_k = Temperatura em Kelvin

Abaixo temos as principais escalas termométricas:

2.3.1.1. Escala Celsius

Na escala Celsius o ponto de gelo é 0 e o ponto de vapor é 100. Nessa escala, o intervalo entre os pontos fixos é dividido em 100 partes iguais, sendo que cada divisão corresponde a 1 grau.

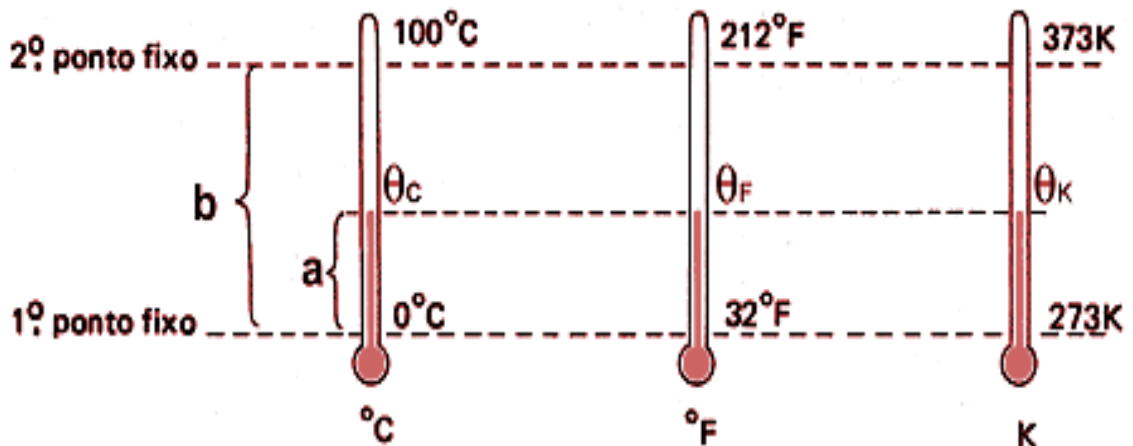
2.3.1.2. Escala Fahrenheit

Na escala Fahrenheit o ponto de gelo é 32 e o ponto de vapor é 212. Nessa escala, o intervalo entre os dois pontos fixos é dividido em 180 partes iguais, sendo que cada divisão corresponde a 1 grau fahrenheit (1°F).

2.3.1.3. Escala Kelvin

Na escala Kelvin o ponto de gelo é 273 e o ponto de vapor é 373. Nessa escala, o intervalo entre os dois pontos é dividido em 100 partes iguais, sendo que cada divisão corresponde a 1 kelvin (1k).

Figura 20 - Escalas Termométricas.



Fonte: Disponível (<https://www.resumoescolar.com.br/fisica/termometria>; acesso 29/05/18)

2.3.2. Medidores

Podemos dividir os medidores de temperatura em dois grandes grupos:

Contato Direto: Medidores nos quais o elemento sensível está em contato direto com o material cuja temperatura se deseja medir.

Contato Indireto: Medidores nos quais o elemento sensível não está em contato direto com o material cuja medida se deseja medir.

Os principais medidores de temperatura nas indústrias são o termopar e o Termo resistência.

2.3.2.1. Termopar

Os termopares ilustrados na figura 21 são dispositivos elétricos com larga aplicação, sendo um elemento primário para medição de temperatura, constituído por dois materiais diferentes ligados um ao outro, conforme tabela 1.

Figura 21 - Termopar



Fonte: Disponível (<http://www.mainzmaquinas.com.br/termopar-tipo-k>; acesso 17/05/18)

Tipos de termopares normalizados

Tabela 1 - Tipos de termopares.

Tipos	Constituição	Gama de Temperatura
B	Platina / 30% Ródio-Platina	0–1800 °C
C	Tung-5% Rênio/Tung-26% Rênio	0–2320 °C
E	Cromel / Constantan	-270–1000 °C
G	Tungstênio/ Tung-26% Rênio	0–2300 °C
J	Ferro / Constantan	-210–750 °C
K	Cromel / Alumel	-270–1370 °C
N	Nicrosil / Nisil	-270–1300 °C
R	Platina / 13%Ródio-Platina	-50–1750 °C
S	Platina / 10%Ródio-Platina	-50–1750 °C
T	Cobre / Constantan	-270–400 °C

Alumel = Níquel-Alumínio;

Constantan = Cobre-Níquel;

Cromel = Níquel-Crômio;

Nicrosil = Ni-Cr-Si

Nisil = Ni-Si-Mg.

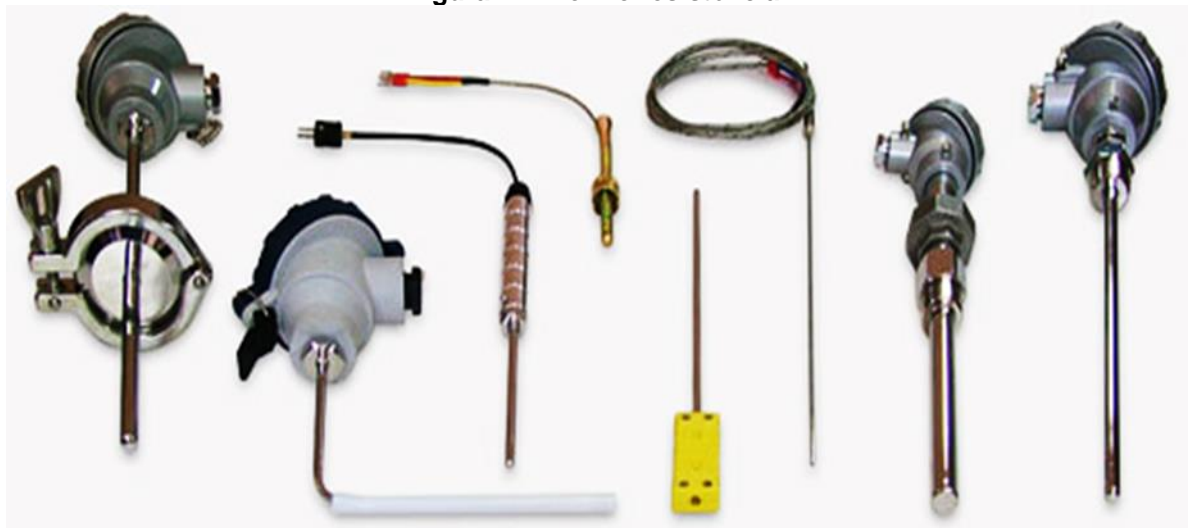
2.3.2.2. Termo resistência ou RTD

Termo resistência ou RTD são sensores que se baseiam no princípio de variação da resistência ôhmica em função da temperatura. “Aumenta a resistência com o aumento da temperatura”.

Sua característica é uma resistência de 100 Ω a 0°C. Por isso, convencionou chamá-la de PT-100 (Fios de platina com 100 Ω a 0°C). A platina é mais usada por apresentar uma ampla escala de temperatura, alta resistividade e maior sensibilidade. Sendo que existem outros materiais como o níquel e o cobre.

Faixa de Trabalho: -200 a 650°C

Figura 22 - Termo resistência.



Fonte: Disponível (<http://www.mainzmaquinas.com.br/termopar-tipo-k>; acesso 17/05/18)

2.3.3. Amplificador para Termopar

Como as tensões geradas pelos termopares são da ordem de mili-volts, faz-se necessária a utilização de algum meio para se amplificar o sinal gerado para um valor

que se possa manipular mais adequadamente pelo CLP. Desta forma foi escolhido como base para esta amplificação o circuito integrado da Analog Devices AD595, que é um amplificador de termopar com compensação de junção a frio. O AD595 foi projetado para utilizar especificamente os termopares do tipo K, de modo, que em sua saída tenha-se uma taxa de $10\text{mV}/^{\circ}\text{C}$.

3. Materiais e Métodos

3.1. Projeto de Hardware

Nesse capítulo encontra-se o foco do trabalho, a determinação e criação de um equipamento para soldagem de componentes SMD nas industriais de pequeno porte.

O requisito básico para esse projeto é a facilidade, praticidade e robustez do equipamento aliados a um custo compatível para o mercado a que se destina.

Tendo em vista equipamentos semelhantes importados, a variedade de configurações disponíveis é grande. Equipamentos com capacidade de avaliação da curva de temperatura diretamente no equipamento, transferência de dados *on-line* para um computador, possibilidade para até 25 sondas, capacidade para avaliação de períodos de até 24h são algumas características disponíveis.

O equipamento aqui apresentado conta com as seguintes características:

1. Capacidade para uma sonda de temperatura.
2. Capacidade de armazenamento de dados de temperaturas.
3. Amostragem de temperatura a cada 1 s com precisão de 1 °C.
4. Uso de sondas tipo Termopar K.
5. Capacidade de comunicação com um PC via interface USB.

3.1.1. Escolha de circuitos e Definição de componentes

Inicia-se a determinação dos circuitos através do tratamento dos sinais oriundos das sondas de temperatura. Conhecida a temperatura do ponto eutético da liga de estanho-chumbo usada no processo como 183 °C, limita-se o número de sondas disponíveis para essa aplicação.

Sondas baseadas em semicondutores (NTC, PTC e demais sensores, a exemplo LM35) serão eliminadas, pois nessa faixa de temperatura existe a degradação das junções dos semicondutores, gerando leituras incoerentes.

As sondas recomendadas para aferição de temperaturas nessa faixa são os termopares. Essas sondas baseiam-se no efeito Seebeck, o qual explica que uma pequena variação de tensão surge quando existe variação de temperatura em uma junção de metais (junção bimetálica). O termopar mais difundido é do tipo K (cromel/alumel).

Sabe-se que as sondas Termopar K têm faixa de atuação de -200 °C até 1200 °C são de baixo custo relativo e têm comportamento definido. A tensão em seus terminais varia na ordem de 48 $\mu\text{V}/^\circ\text{C}$, assim esse sinal precisa ser tratado e amplificado, de modo a conter a informação de temperatura correta.

Comparados a outros tipos de sensores os termopares tem uma grande faixa de atuação, porém sua exatidão é pequena. Na faixa de medição entre -200 °C e 800 °C a exatidão é de apenas 1°C. Acima dessa temperatura pode-se ter erros de até 7,5 °C. Apesar da pequena faixa de operação (-55 °C a 150°C) o sensor LM35 tem exatidão de no mínimo 0,4 °C. Ainda há o efeito de junção fria, ligado ao efeito bimetálico entre conectores, trilhas e conectores do termopar na placa.

Para compensar esse efeito e facilitar a leitura das temperaturas será usado o circuito integrado MAX6675 da Maxis®, cujas funções incluem a leitura, discretização e digitalização da temperatura das sondas com precisão de 0,25 °C, compensação de junção fria e tratativa dos dados serial (SPI). Apesar do alto custo desses CI, sua utilização facilita o roteamento da placa, a tratativa de hardware e software para compensação da junção fria e praticamente equivale ao custo quando do uso de amplificadores de instrumentação, a exemplo INA129, para tratamento do sinal dos termopares ou de CI similares, tal como o AD595.

De acordo com os requisitos de hardware listados para o armazenamento dos dados de quatro sondas no tempo máximo de 10 minutos, sendo que cada temperatura coletada é descrita por um byte, são necessários 4800 endereços de um byte cada. Para armazenamento dos dados recolhidos será usada uma memória com comunicação SPI de 128K x 8 bits, ATmega328P da Atmel®.

O controlador utilizado será o ATmega328P, também da Atmel®. Esse controlador de 8bits tem os recursos necessários para a aplicação (capacidade de

processamento, comunicação SPI, comunicação UART), aliado a um custo baixo e grande disponibilidade de bibliotecas de programação.

3.1.2. Esquemáticos

O esquemático do projeto foi feito na suíte de desenvolvimento PROTEUS, que engloba o gerador de esquemáticos ISIS e o “CAD” ARES. O diagrama elétrico está disponível no anexo I.

.Tendo em vista 3 esquemáticos distintos:

- Fonte de alimentação 5vcc ou 0 á 24vcc.
- Placa de controle.
- Display 126x64

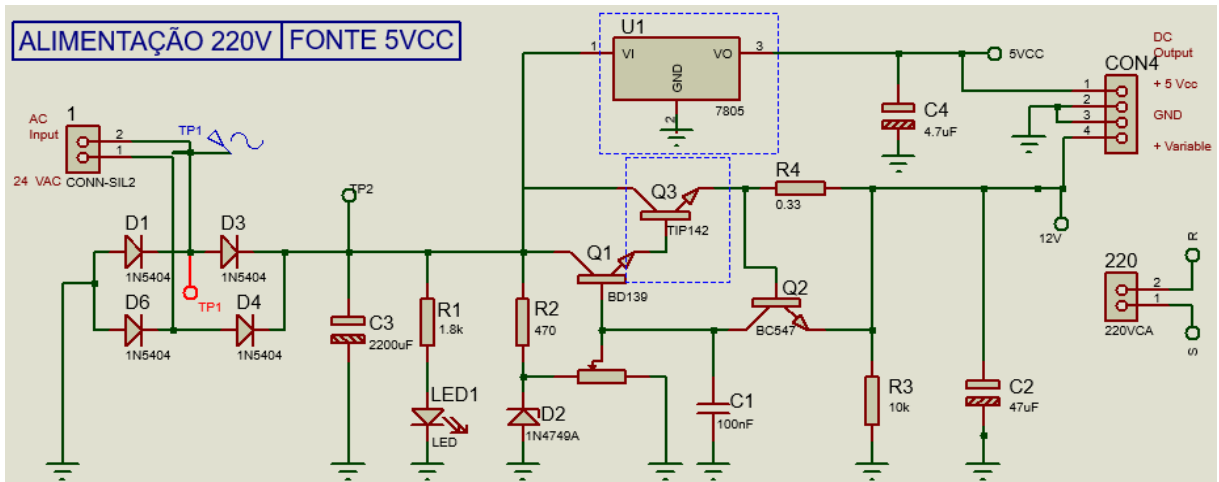
3.1.2.1. Fonte de alimentação

Os diodos D1, D2, D3 e D4 retificam a tensão, e C3 é o filtro. R1 e Led1 são indicadores de fonte ligada ou desligada.

O CI 7829 é um regulador de tensão serie, de até 800 mA, o qual alimenta o Arduino. C4 é o capacitor exigido pelo fabricante desse CI. Já R2e o Zener D2 (1N4749), junto com P1, Q1 e Q3 formam um regulador de tensão série de alta corrente de saída, e 24 Vcc de tensão. Q1 e Q3 estão ligados na configuração Darlington para reduzir a corrente da base de Q1.. Q2, R4 e R3 formam o regulador “SHUNT” certas variações de tensão de acordo com a variação da carga.

C2 é o capacitor de filtro de circuito “SHUNT”, conforme a Figura 23.

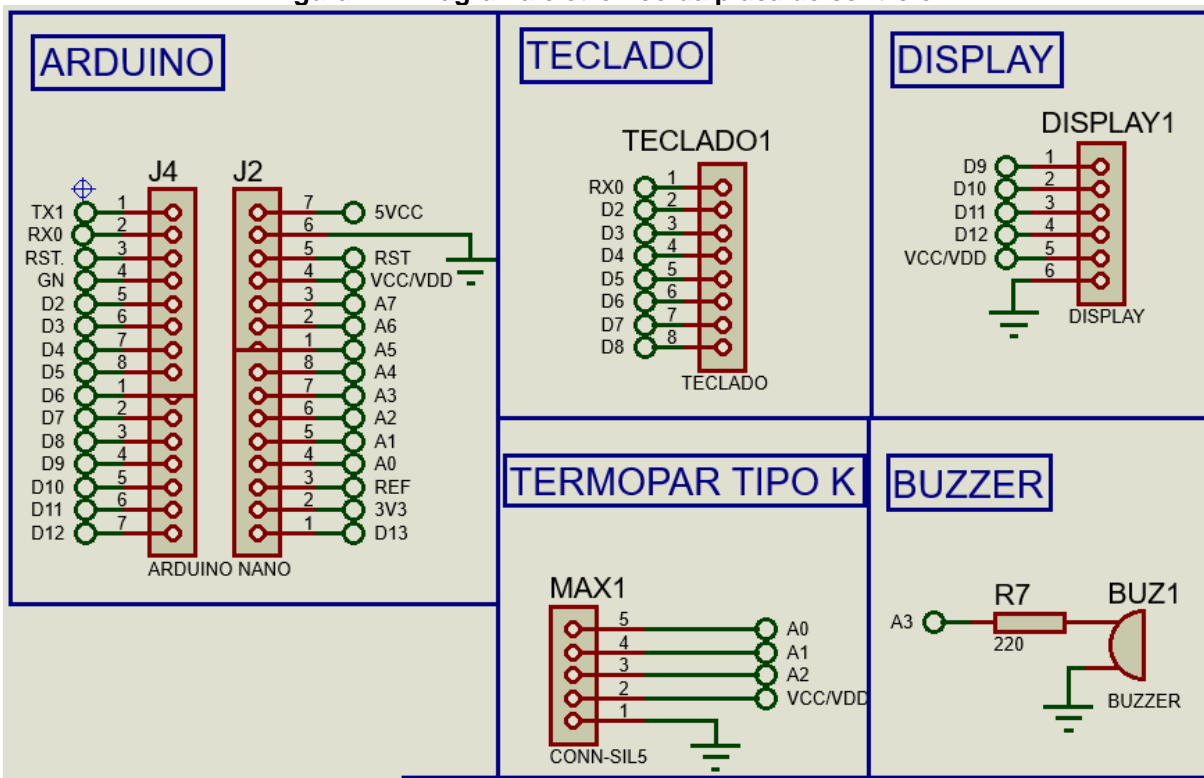
Figura 23 - Diagrama eletrônico da fonte.



3.1.2.2. Placa de controle

A figura 24 ilustra a comunicação do arduino Nano com os periféricos.

Figura 24 - Diagrama eletrônico da placa de controle.



ARDUINO

Na figura 24 demonstra o diagrama eletrônico da placa de controle, demonstrando as entradas e saídas do arduino Nano,

ENTRADAS:

Pinos (D0, D2-D8) = Teclado;

Pinos (A0-A2) = MAX 6675

SAÍDAS:

Pinos (D9-D12) = Display;

Pino (A4) = Saída FAN

Pino (A5) = Saída HEAT

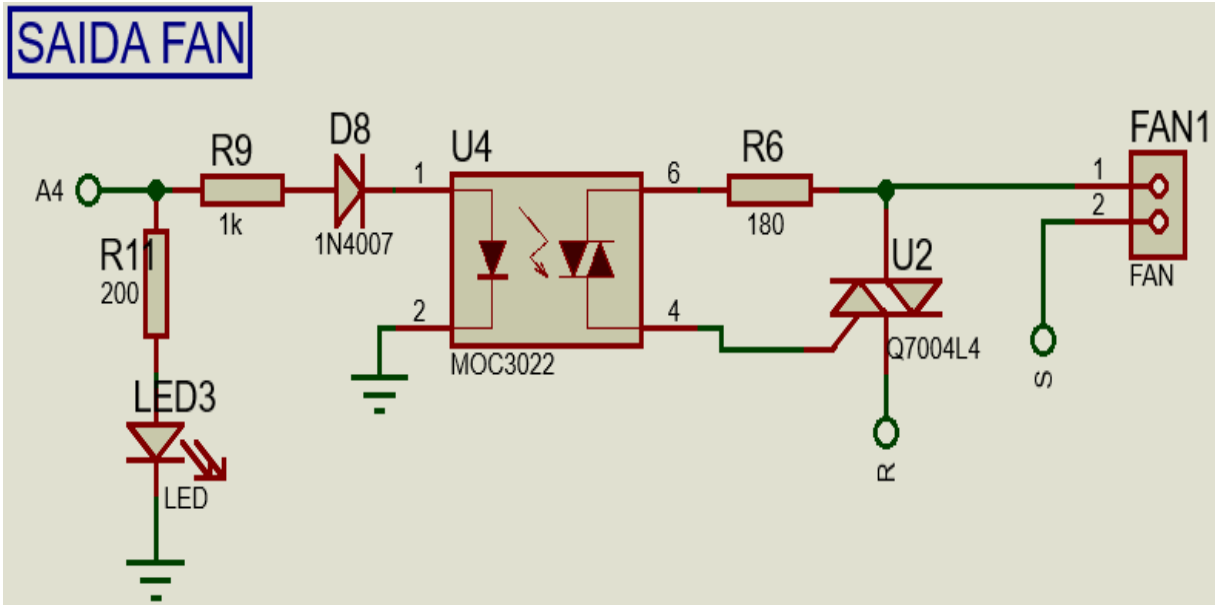
Pino (A3) = Buzzer

Há muita necessidade de controlar cargas mais elevadas na saída, separando o arduino fisicamente do circuito. Podem usar-se Transistor ligado ao acoplador que execute e suporte a corrente necessária.

SAÍDA DE RESFRIAMENTO

O diodo D1 protege a entrada do CI de inversão de tensão. Ao ser aplicada uma tensão no pino 1 do foto-acoplador MOC3022, o diodo interno emite luz (no interior) levando o coletor do Triac a uma tensão de praticamente 0V, assim polarizando a base do Triac no pino 8, chaveando o 220V a saída. O Led2 e R2 podem ser omitidos, servem apenas como ajuda visual. O controle de resfriamento foi montado na mesma placa, conforme o esquema mostrado na figura 25.

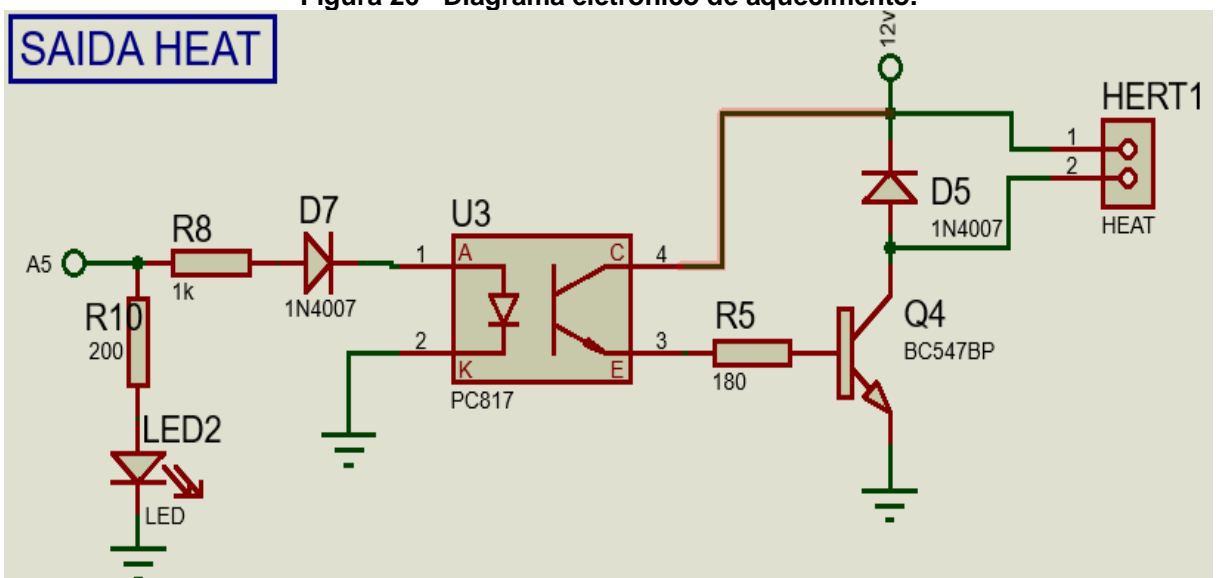
Figura 25 - Diagrama eletrônico resfriamento.



SAÍDA DE AQUECIMENTO

O díodo D3 protege a entrada do CI de inversão de tensão. Ao ser aplicada uma tensão no pino 1 do foto-acoplador PC817, o diodo interno emite luz (no interior) levando o coletor do Transistor a uma tensão de praticamente 0V, assim polarizando a base do Transistor BC547BP (Q3) no pino 3, chaveando o 24V a saída. O Led3 e R5 podem ser omitidos, servem apenas como ajuda visual. O controle de aquecimento foi montado na mesma placa, conforme o esquema mostrado na figura 26.

Figura 26 - Diagrama eletrônico de aquecimento.



3.1.2.3. Display

O Display LCD 128 x 64 são baseados na tecnologia LCD (monocromática), mas oferecem recursos gráficos interessantes, pois cada pixel é individualmente endereçável. Esses monitores são mais complicados de programar do que os monitores LCD 'comuns' 16 x 2 ou 20 x 4 que apresentam duas ou quatro fileiras de posições de caracteres fixos. O resultado é uma apresentação com flexibilidade considerável graças à existência de caracteres incorporados e, em software para o Arduino, uma extensa biblioteca gráfica u8g, ou seja, a mesma biblioteca que suporta exibição de gráficos em displays OLED. Aqui nós discutimos a fiação de uma tela gráfica 128x64 LCD de 128 x 64 baseada no ST7920.

3.1.3. Principais Características:

O display tem 20 pinos, numerados da direita para a esquerda, conforme podemos ver na figura 27.

- Controlador interno: ST7920;
- Tensão de trabalho: 3,3V ~ 5,5V;
- Corrente do backlight: 200mA ~ 360mA;
- Clock: 2Mhz;
- Backlight AZUL e escrita BRANCA;
- Dimensão do visor: 7,0 x 3,88 cm (C x L);
- Dimensão total: 9,3 x 7,0 x 1,4 cm (C x L x A);

Figura 27 - LCD Gráfico 128x64.



A placa de fuga do display LCD possui 20 pinos para conectividade, denominada GND, VCC, V0, RS, R / W, E, DB0, DB1, DB3, DB4, DB5, DB6, DB7, PSB, NC, RST, VOUT, BLA e BLK. A Figura 28 mostra a conectividade básica via interface SPI (SPI significa Serial Peripheral Interface). GND está ligado ao GND do Arduino, 5V a 5V, V0 ao limpador de um potenciômetro de 10k Ω (isto é para adaptar o contraste entre caracteres ou gráficos do LCD em segundo plano), RS para o pino 9, R / W para fixar 10, E para o pino 11 e RST para o pino 12, conforme tabela 2. Os pinos DB0 a DB7 da placa de breakout precisam ser conectados somente se o modo paralelo for considerado. O PSB está conectado ao terra, enquanto o BLA precisa de 3,3V (luz de LED de fundo do monitor LCD) e o BLK precisa de conexão com o GND. O BLA pode ser conectado ao pino 5V do Arduino com um resistor de 220 Ω em série para reduzir a tensão, ou pode ser conectado diretamente ao pino 3.3V do Arduino.

Embora os protocolos SPI funcionem com pinos marcados com MISO, MOSI, CLK e CS, a placa de expansão ST7920 possui designações de pinos completamente diferentes, enquanto suporta o protocolo SPI. MISO e MOSI são necessários para o modo mestre / escravo e não importam na presente construção. CLK (clock) e CS (chip select) são necessários, assim como o pino de dados (R / W) e o pino marcado PSB (este último precisa ser conectado ao GND - pino definido continuamente LOW para informar ao chip controlador do display que ele deve funcionar em modo serial (PSB HIGH significa modo paralelo)).

Ele pode ser utilizado para se comunicar com o Arduino tanto de forma serial como paralela. Onde a velocidade de atualização do LCD não é um ponto crítico, vamos utilizar a comunicação serial, ocupando apenas 4 portas do Arduino conforme a figura 28.

A biblioteca utilizada, é a U8Glib , é uma biblioteca bem completa e complexa, com muitos exemplos de utilização. Recomendo uma visita à página da u8glib , onde você vai encontrar instruções sobre o funcionamento da biblioteca e também informações detalhadas sobre a utilização de cada comando.

Figura 28 - Diagrama eletrônico de aquecimento do LCD 128x64.

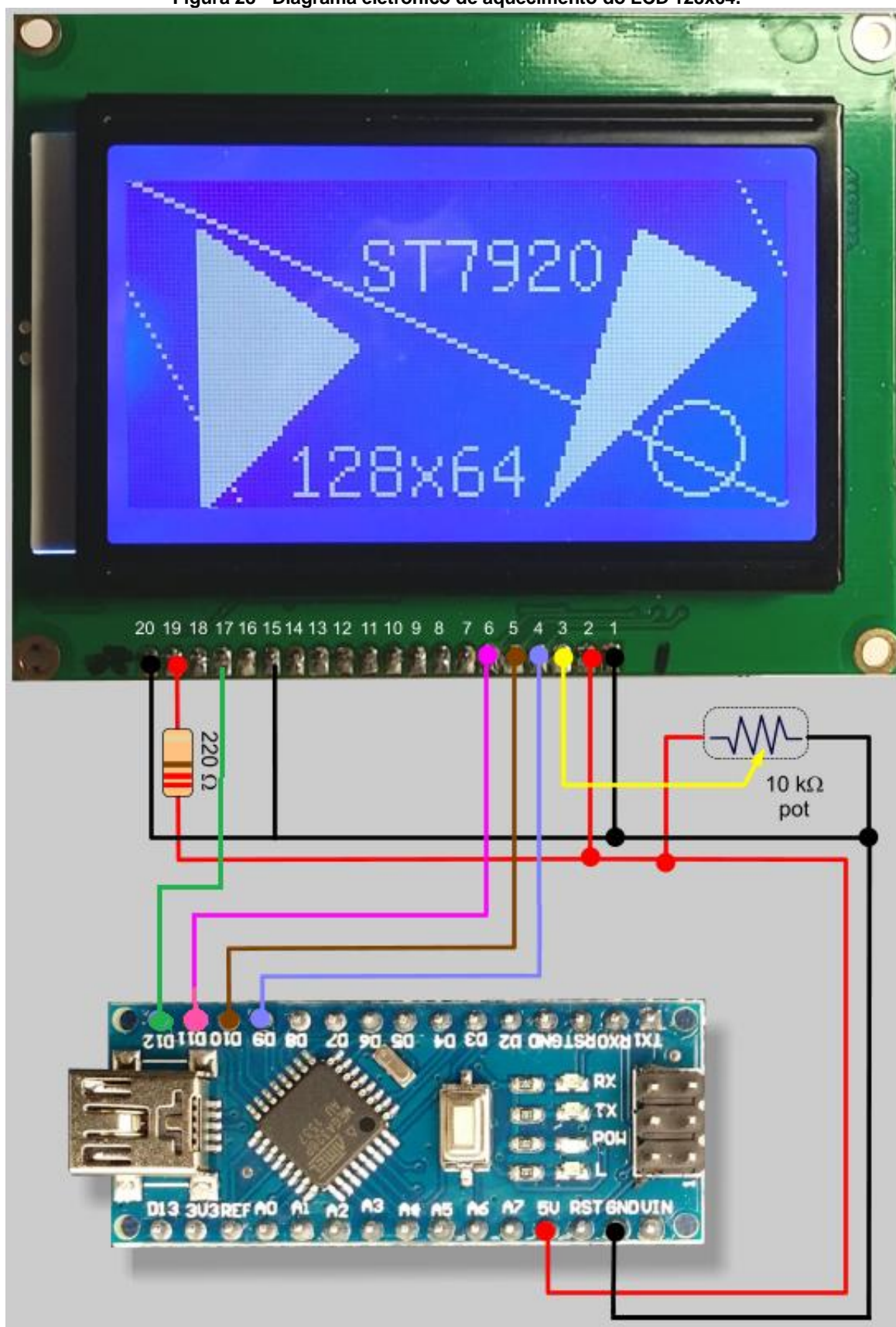


Tabela 2 - Endereçamento entre LCD e arduino.

Pinos LCD	Função	Conexão
1	GND	GND
2	Vcc	5 volts
3	Ajuste de contraste	Pino central do Arduino
4	RS	Porta digital 9 Arduino
5	RW	Porta digital 10 Arduino
6	Enable	Porta digital 11 Arduino
15	Comun. Paralela (HIGH) / Serial (LOW)	GND
17	Reset	Porta digital 12 Arduino
19	Backlight +	3.3 V (ou 5 volts)
20	Backlight -	GND

3.1.4. Placa de circuito impresso

O esquemático do projeto foi feito na suíte de desenvolvimento PROTEUS, que engloba o gerador de esquemáticos ISIS e o “CAD” ARES para desenvolvimento de PCI’s, está disponível no anexo II. Foram levados em consideração vários aspectos industriais na confecção do protótipo, como custo de inserção de componentes, posicionamento de componentes para facilitar a revisão da solda, dimensões da PCI entre outros.

3.1.5. Roteamento e desenho da placa de circuito impresso

Desenvolvido na plataforma ARES o Layout e o posicionamentos dos componentes para melhor praticidade. Desenvolvido a máscara de solda e ilustração em 3D, conforme figuras 29, 30 e 31.

Figura 29 - Layout e posicionamentos dos componentes.

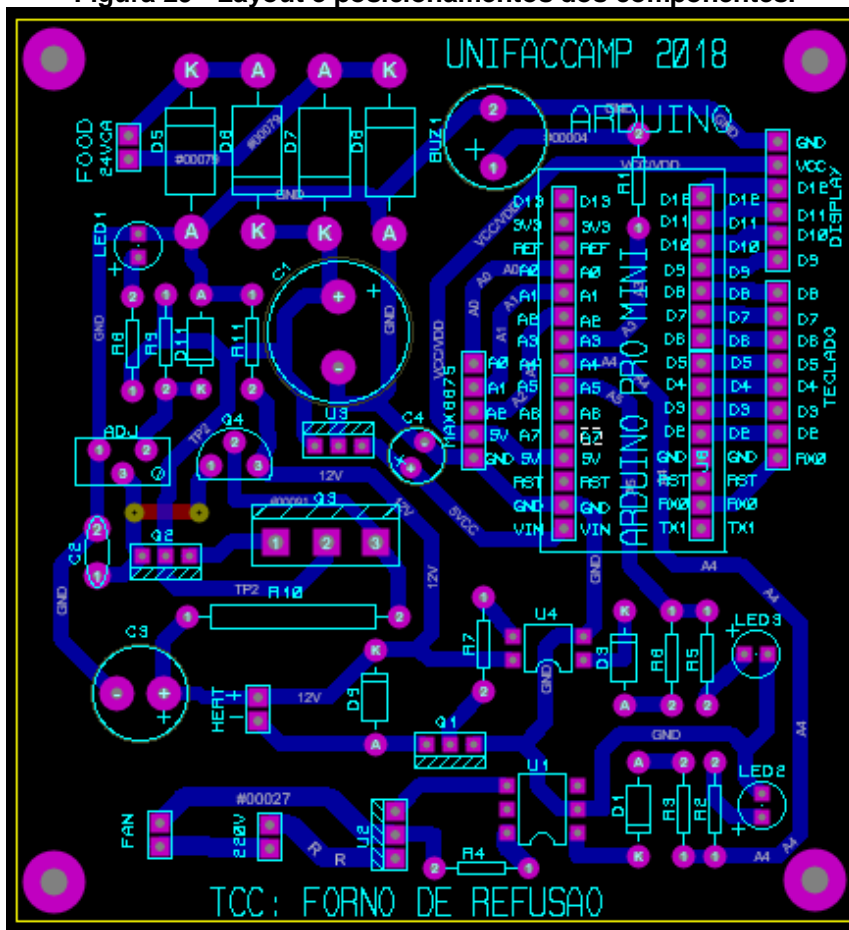


Figura 30 - Máscara de solda.

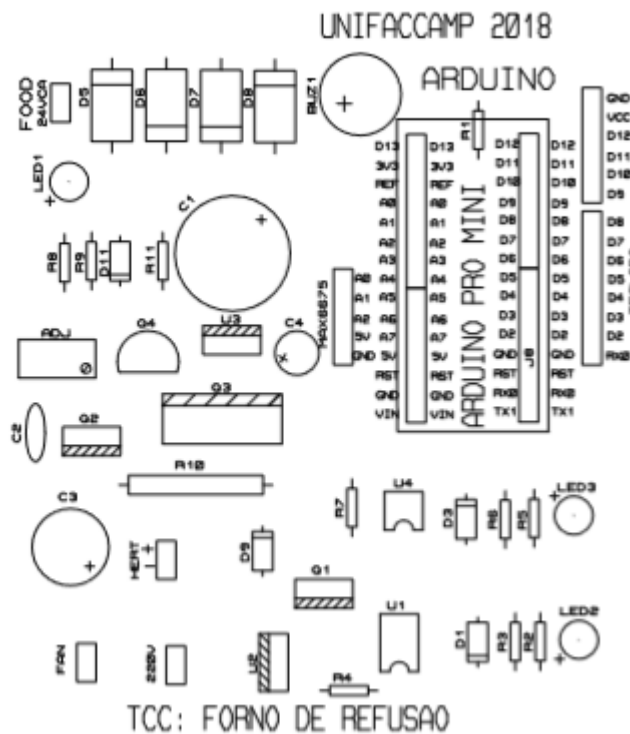


Figura 31 - Placa em 3D.

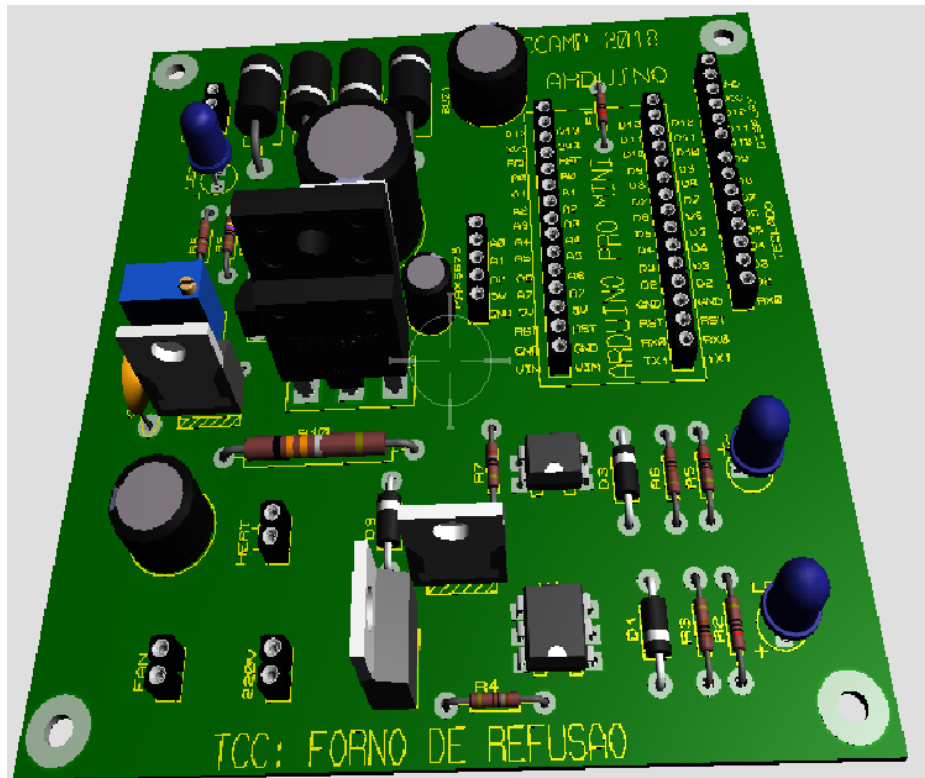
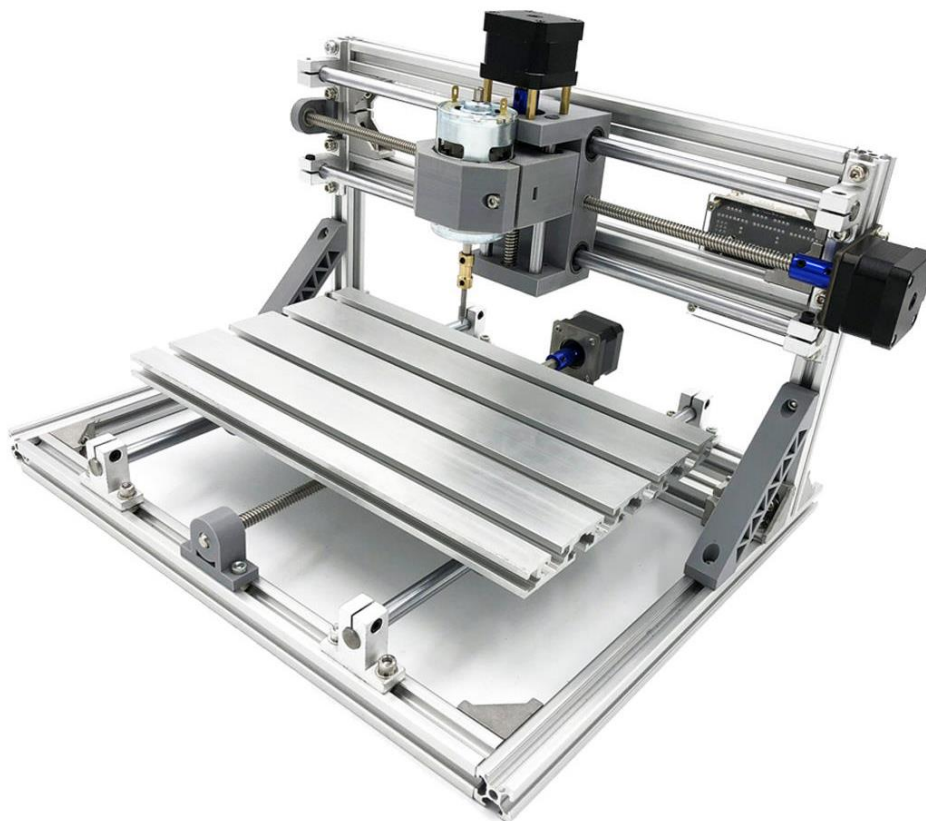


Figura 32 - Ilustra a CNC que foi utilizada para criar a placa.



3.2. Montagem do equipamento

O circuito foi usinado na placa de PCI de cobre pela CNC 3018 é perfurada pela broca de 0,01mm, em seguida retirado as rebarbas com palhas de aço, conforme mostrado na figura 33 e 34.

Figura 33 - Processo de usinagem da placa.

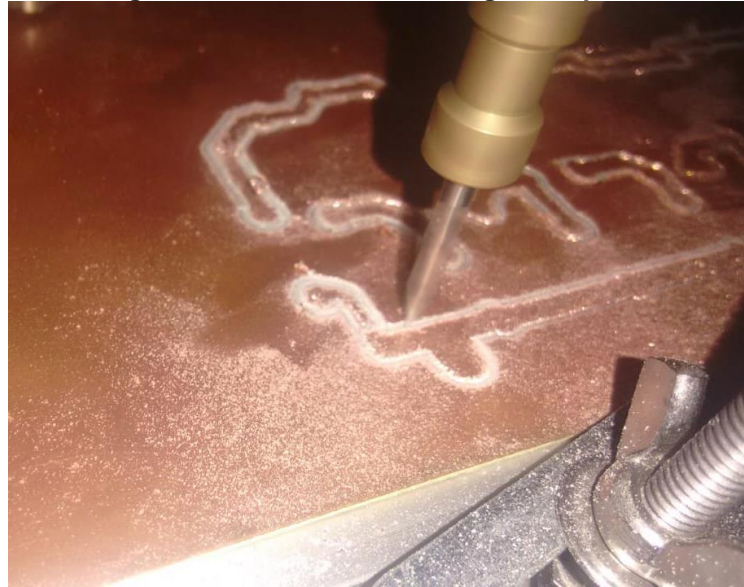
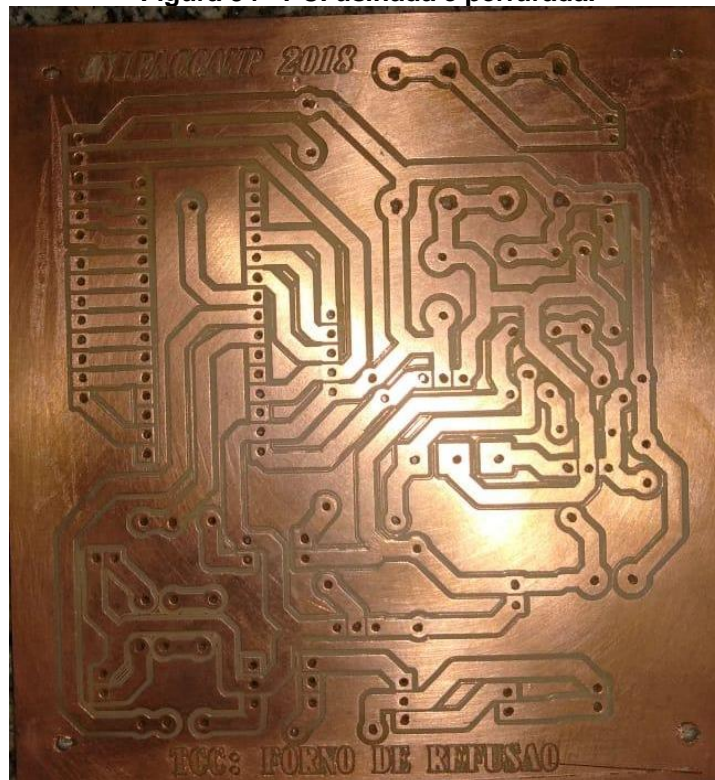


Figura 34 - PCI usinada e perfurada.



Para melhor identificação dos componentes imprimindo o circuito do programa Proteus na folha couche A4 impresso em uma impressora com tinta tonner para melhor fixação da tinta na placa.

Alinhando o papel com o layout sobre a PCI, em seguida realizar a transferência do toner é necessário aquecer o conjunto placa + desenho. O ferro a uma temperatura de 150°C passando levemente até fixar o circuito na placa totalmente. De maneira artesanal, um ferro de passar roupas pode ser usado para esse fim. Aqueça-o a uma temperatura média (pouco calor não derrete o toner, calor excessivo pode deformar o papel couche (poliéster), derreter a fita entortar a placa se ela for de fenolite, por exemplo).

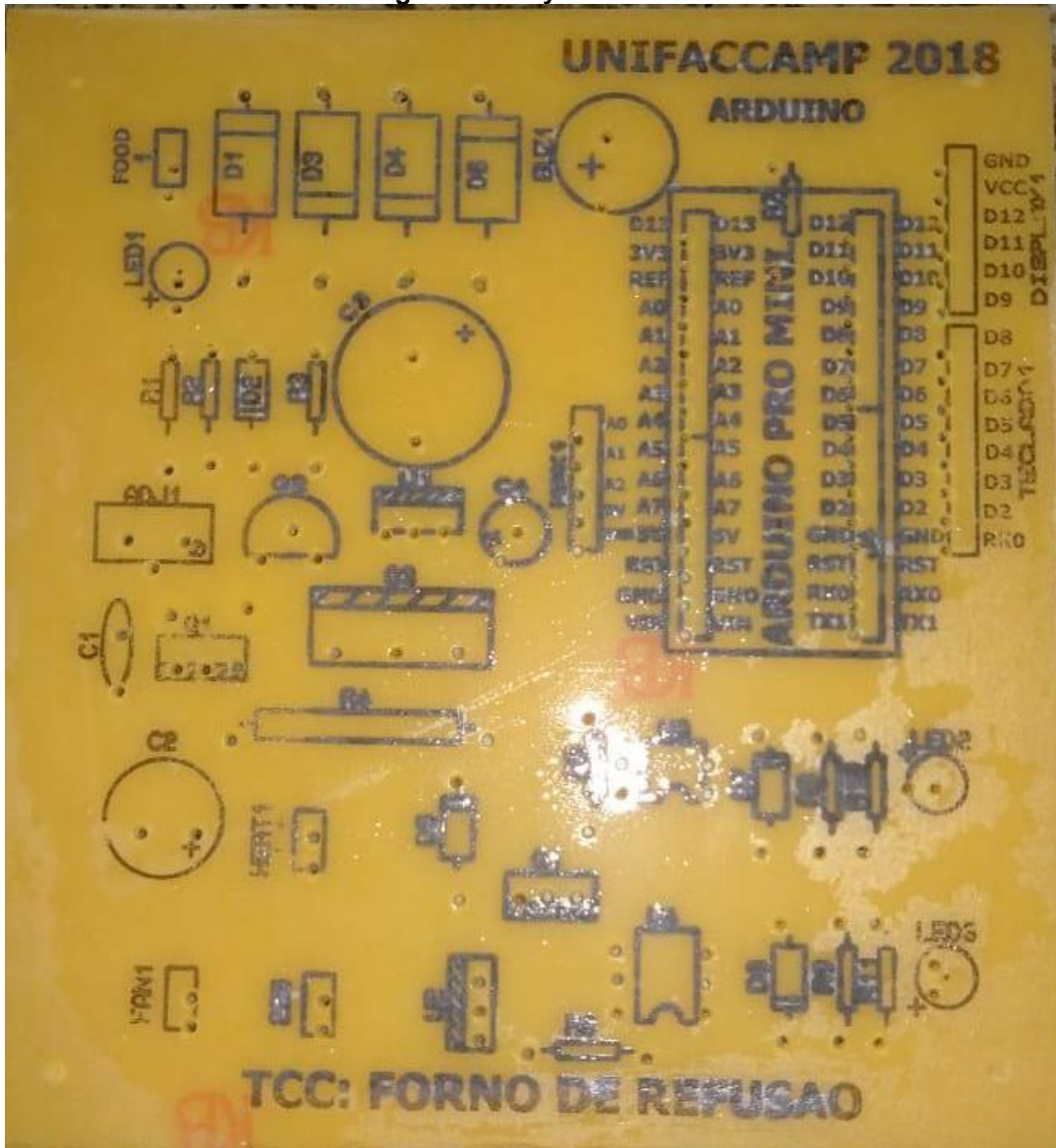
Aplique pressão igualmente sobre o conjunto, de forma a aquece-lo de maneira uniforme, por algum tempo. Pressão, temperatura e tempo devem ser experimentados até se obter um bom resultado conforme figura 35.

Figura 35 - Demonstra o Layout e posicionamentos dos componentes.



Para efetuar a limpeza do produto, a PCI foi lavada em água corrente, e em seguida a placa foi polida com palha de aço. O processo de polimento visa à remoção do papel couche, aderido à placa no processo de transferência, como discutido anteriormente. Após a retirada do torne, a placa está pronta para ser perfurada. A Figura 36 ilustra a placa com polimento parcial.

Figura 36 - Layout da PCI.



Em seguida, com o esquema elétrico da placa em mãos e com os componentes devidamente corretos, iniciou-se o processo de montagem final da placa controladora da MPA com a inserção e soldagem dos componentes. (Figura 37).

Figura 37 - Processo de inserção e soldagem dos componentes.



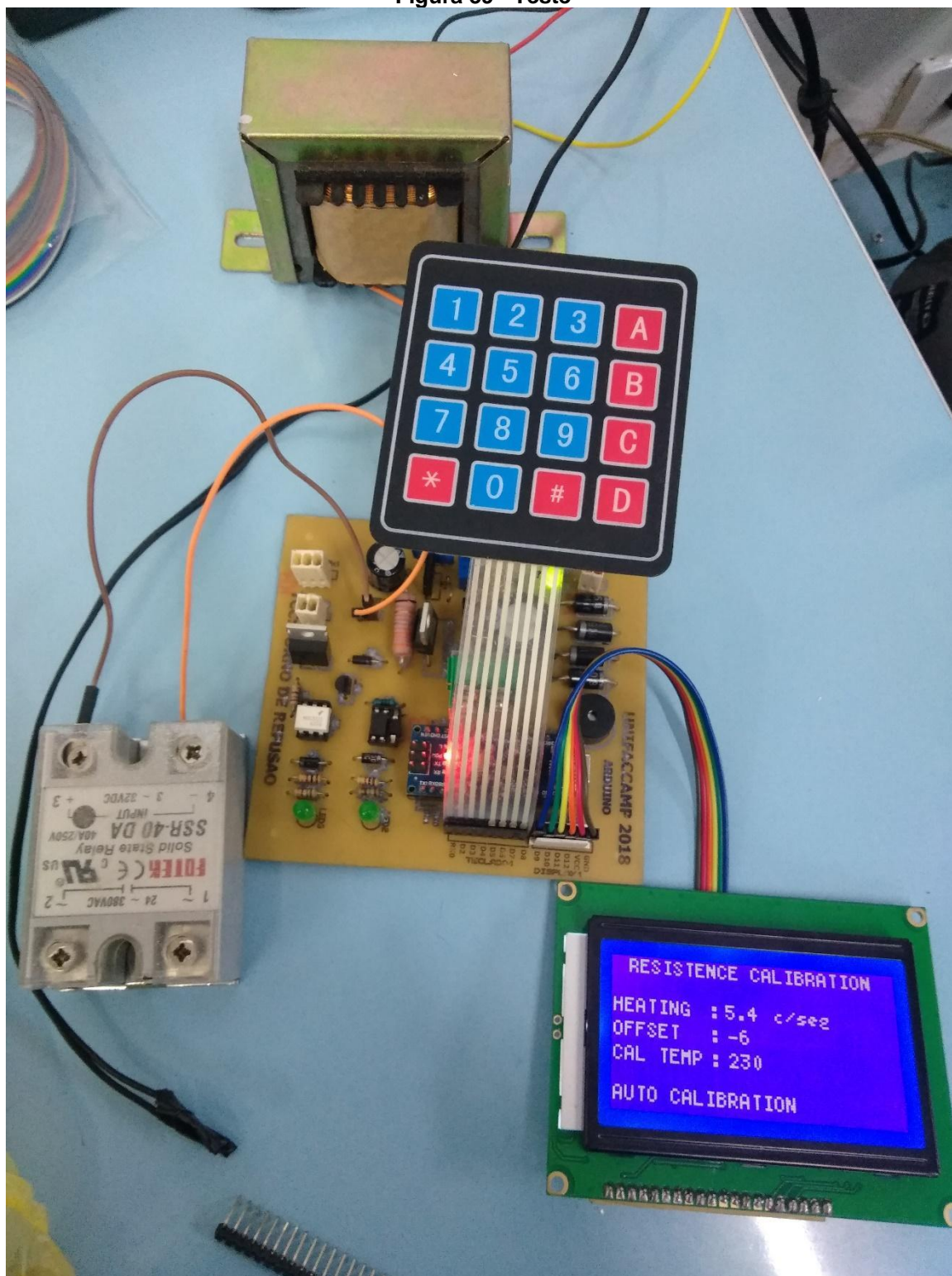
A figura 38 mostra todos os componentes soldados a placa está completamente montada e pronta para teste e verificação da comunicação com os periféricos.

Figura 38 - Placa controladora do forno de refusão montada e pronta para uso.



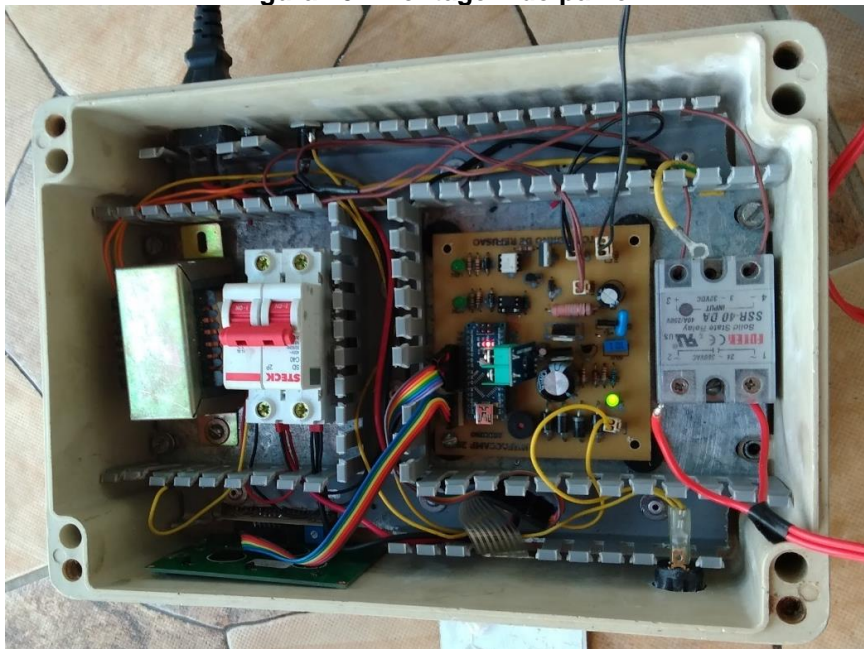
Para proteção do forno foi projetado alguns sistemas de segurança para proteger o forno de superaquecimento e curto circuito. O relé de estado sólido ele protege as resistências, conforme figura 39.

Figura 39 - Teste



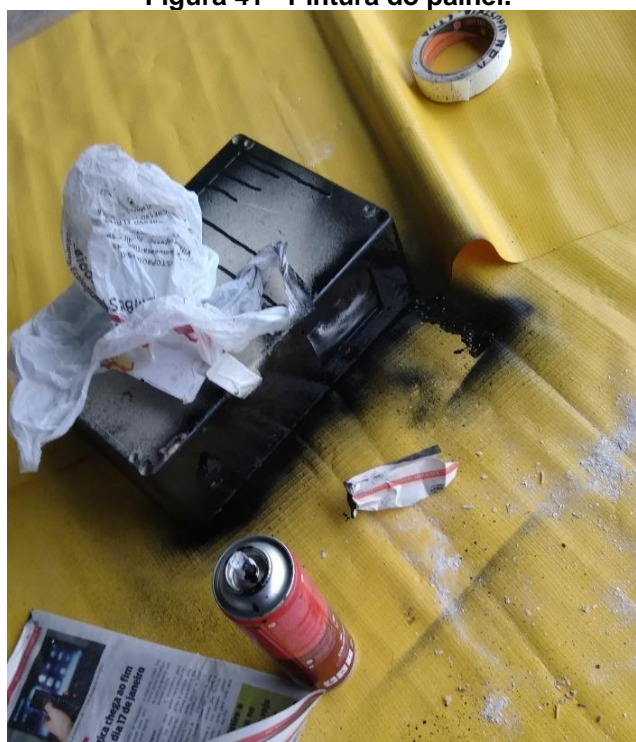
A figura 40 ilustra a finalização da montagem do circuito dentro de um painel para melhor organização e proteção da placa de controle.

Figura 40 - Montagem do painel.



A figura 41 ilustra a pintura do forno que foi utilizado a tinta Coral de alta temperatura suportando até 350°C.

Figura 41 - Pintura do painel.



Um sensor termopar tipo K pode ser visto na figura 42, dentro do forno para indicar a temperatura do mesmo e controlar a curva de solda. Para proteção do sensor de temperatura adaptado em um tubo de cobre para não danificar o cabo.

Figura 42 - Proteção do sensor de temperatura dentro do tubo de cobre.



Foi interligado as resistências, motor de resfriamento, sensor de temperatura e comunicação entre forno e painel de controle para realizar o teste final. Conforme figuras 43 e 44.

Figura 43 - Passagem de cabeamento.



Figura 44 - Forno Montado e pronto para entrar em funcionamento.



3.3. Projeto do Software.

Para o desenvolvimento das placas, foi utilizado o software Arduino. Este software tem sido muito utilizado pelos programadores, projetista e principalmente por estudantes devido a facilidade de utilização. Desenvolvido pela empresa Australiana ARDUINO, ele está disponível para download no site do fabricante, nos idiomas inglês, português e etc. É uma versão gratuita, possui um número grande de bibliotecas pré-instaladas, abrangendo um número elevado de programação.

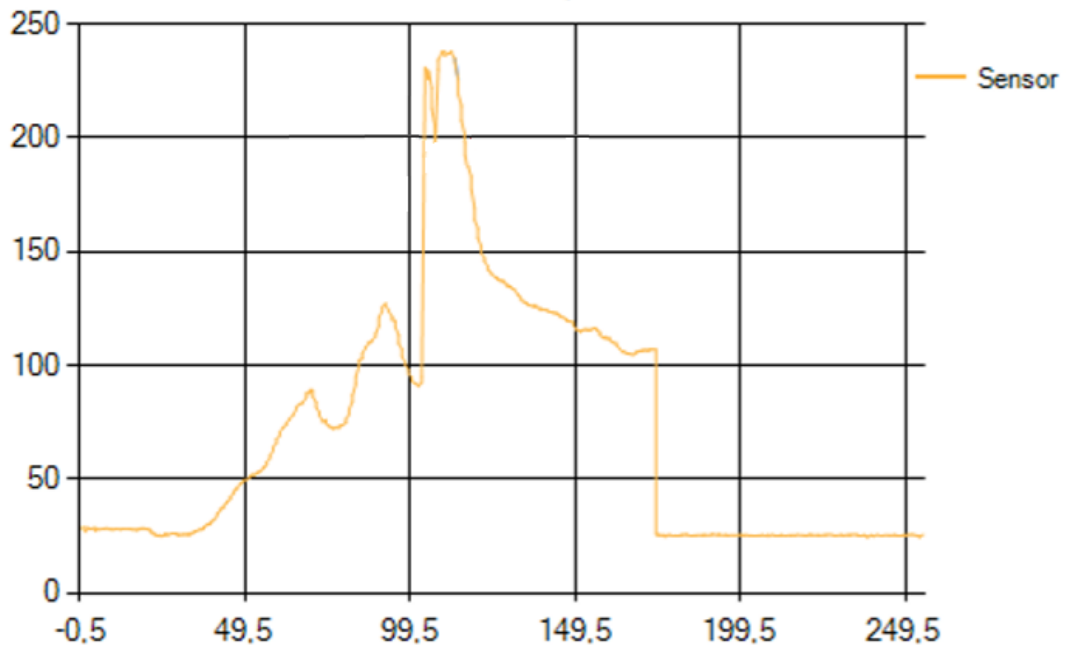
3.3.1. Programação

O código está disponível no anexo III ao anexo XII. Várias partes do código são originadas em exemplos de códigos disponibilizadas pela fabricante do micro controlador.

3.3.2. Comunicação

O projeto fornece dados em tempo real do perfil da temperatura exportando os resultados em forma de gráfico no excel, conforme a figura 45.

Figura 45 - Ilustra a CNC que foi utilizada para criar a placa.
Medidas de Temperatura



4. Resultados

4.1. Funcionamento do equipamento

CONTROLADOR DE TEMPERATURA COM 4 RAMPAS E 9 MEMORIAS

Manual de uso:

Pressione a tecla "#" para avançar uma casa ou posição de ajuste ou a tecla "*" para retroceder uma casa ou posição de ajuste para os valores de tempo e temperatura das rampas 1 a 4 ou para menu de configuração.

Digite os valores desejados para os tempos em segundos ou temperatura em graus Célsius sempre com 3 dígitos. Se for usar um valor de 2 dígitos, digitar zero antes.

São aceitos valores entre 0 e 255.

Manter pressionada a tecla "C" na tela inicial para gravar na memória os ajustes atuais de tempo e temperatura, logo após, selecionar a posição da memória através das teclas numéricas de 1 a 9, mantendo a mesma pressionada. Será emitido um beep longo para confirmar a gravação. Selecionando uma posição de memória já ocupada, a mesma será substituída pelos novos valores. Nos nomes dos perfis gravados são pré-definidos não podem ser alterados.

Manter pressionada a tecla "D" durante a tela inicial para carregar um perfil pré-gravado na memória, logo após selecione o perfil através das teclas numéricas de 1 a 9. O perfil será carregado e o número e nome exibido no lado direito do display. Pressione a tecla # para avançar para a página 2 com as memórias de 5 a 8.

Manter pressionada a tecla "A" inicia o processo. Se durante a execução do processo for preciso aumentar o tempo, cada pressionamento de tecla "C", adicionará +30 segundos ao tempo da rampa 4, ou pressionando a tecla "D" diminui 30s. Pressionar "A" rapidamente encerra o processo e zera o contador de tempo.

Pressionar a tecla "B" mostra a tela do gráfico, outro toque em "B" retorna a tela inicial.

Manter pressionada a tecla "1" quando o processo não estiver sendo executado, liga o FAN, um toque curto na mesma tecla desliga o FAN.

Mantenha pressionada a tecla "2" durante a tela inicial para entrar no menu de configuração. Manter a tecla zero "0" pressionada para sair. Dentro do menu de configuração, pressionar as teclas "#" e "*" para avançar e retroceder entre os itens do menu, e as teclas "C"(+) e "D"(-) para alterar os valores.

Resumo:

Tecla "A" -> liga e desliga o processo;

Tecla "B" -> alterna entre tela inicial e gráfico;

Tecla "C" -> grava memória ou função mais no menu de ajuste;

Tecla "D" -> lê memória ou função menos no menu de ajuste;

Tecla "*" -> retrocede posição/item;

Tecla "#" -> avança posição/item;

Teclas numéricas -> utilizadas para entrar com os tempos e temperaturas/
Acessa ou grava as posições de memória;

Tecla "1" -> Liga e desliga o FAN manualmente;

Tecla "2" -> entra no menu de configuração;

Tecla "3" -> entra no menu de calibração da resistência;

Os dados da temperatura lida também são enviados via saída serial/USB para o monitor serial da IDE do arduíno, com taxa de atualização de 1 vez por segundo, podendo ser usados posteriormente para os fins desejados.

Para calibração da resistência utilizada:

Certifique-se de que a resistência esteja totalmente fria, na temperatura ambiente, e anote essa temperatura. Coloque o sensor de temperatura no ponto de medição na resistência, ligue a resistência diretamente na energia elétrica (sem passar pelo relê do equipamento) e com um relógio ou cronômetro meça o tempo para atingir a temperatura máxima de trabalho, em torno de 230°C. Ao término dessa etapa, subtraia o valor da temperatura máxima lida, da temperatura inicial e dividida pelo tempo total de aquecimento em segundos. O valor obtido deverá ser informado no menu de calibração da resistência, e indica quantos graus por segundo a sua resistência aquece.

Também pode ser usada a opção de AUTO CALIBRATION no menu RESISTENCE CALIBRATION, onde o sistema fará a calibração da resistência automaticamente.

Segue ilustrações das telas do LCD, conforme figuras 46 ... 58.

Figura 46 - Teclado.



Figura 47 - Configuração do teclado.

TECLADO

```

                MENU CONFIG
                |
                |
FAN --> 1 | 2 | 3 | A --> INICIAR/ENCERRA
        4 | 5 | 6 | B --> MOSTRA GRÁFICO
        7 | 8 | 9 | C --> AUMENTA/GRAVA MEMÓRIA
        * | 0 | * | D --> DIMINUI/LER MEMORIA
        ^           ^
        |           |
        RETROCEDE   AVANÇA

```

Figura 48 - Tela inicial e execução.



Figura 49 - Configurações da tela inicial e execução.

```

TELA INICIAL/EXECUÇÃO

RAMPA 1 RAMPA 2 RAMPA 3 RAMPA 4

RUN/STOP -----> | STOP | 1ST | 2ST | 3ST | 4ST |
|-----|
TEMPERATURA ALVO ----> | TEMP | 70 | 120 | 170 | 225 |
TEMPO DE PERMANENCIA NA --> | TIME | 80 | 60 | 50 | 40 |
TEMPERATURA ALVO
RAMPA ATUAL -----> | 0st | | | | | <----- TEMPO RESTANTE POR RAMPA
|-----|
GRAVAÇÃO DE MEMÓRIA ----> | STORE: | | | | 1:LEADED | <----- PERFIL SELECIONADO
| | | | | TEMP | TIME |
HEATING/COOLING -----> | HEATING | 23c | 0:00 | | <----- TEMPO TOTAL
INDICA AQUECIMENTO OU
RESFRIAMENTO
^
|
MOSTRA TEMPERATURA
LIDA EM TEMPO REAL
    
```

Figura 50 - Telas de seleções de perfil 1.



Figura 51 - Telas de seleções de perfil 2.

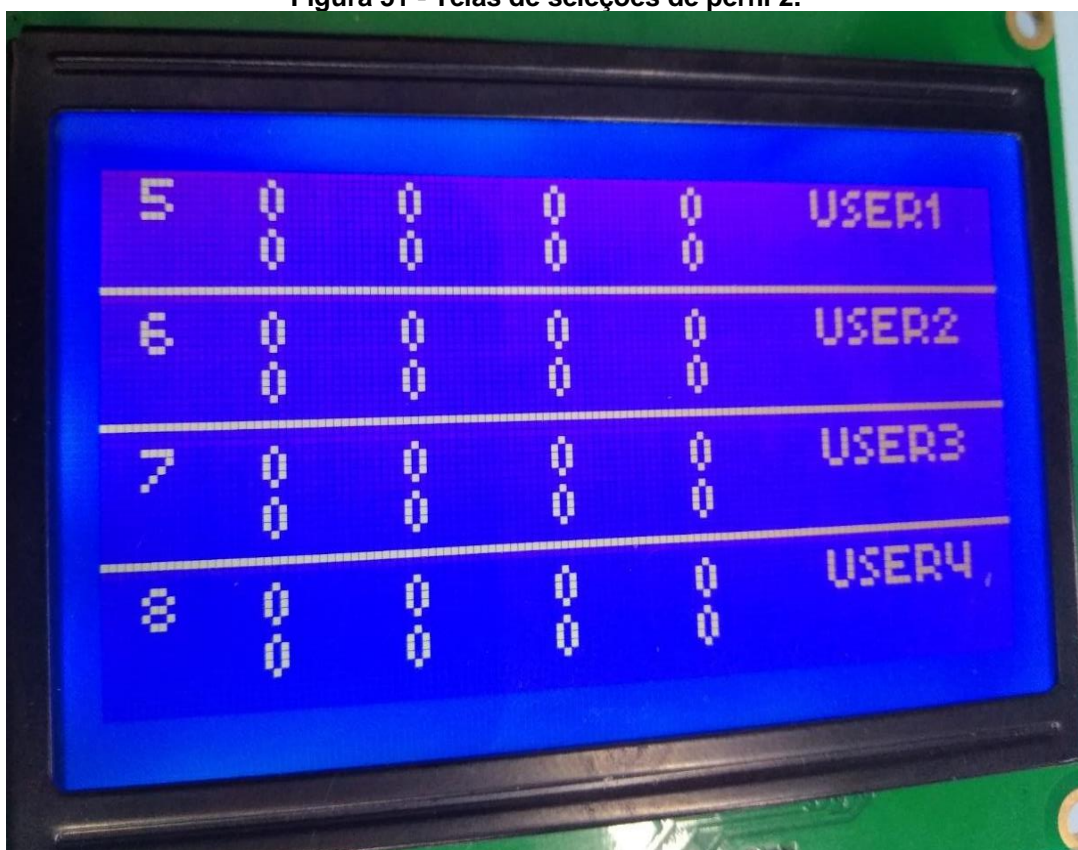


Figura 52 - Configuração e menu.



Figura 53 - Menu e configuração.

```
MENU CONFIG  
  
KEY BEEP      : ON   -> liga/desliga o beep das teclas  
FAN MODE     : AUTO -> fan liga automaticamente ao final do processo  
TEMP FAN     : 40  -> temperatura de desligamento do fan quando em modo AUTO  
BEEP TIME    : 10  -> segundos restantes para final do processo em que começa  
                a emitir beeps  
LOAD DEFAULT: NO   -> carrega os ajustes padrões de fábrica
```

Figura 54 - Tela de calibração da resistência.

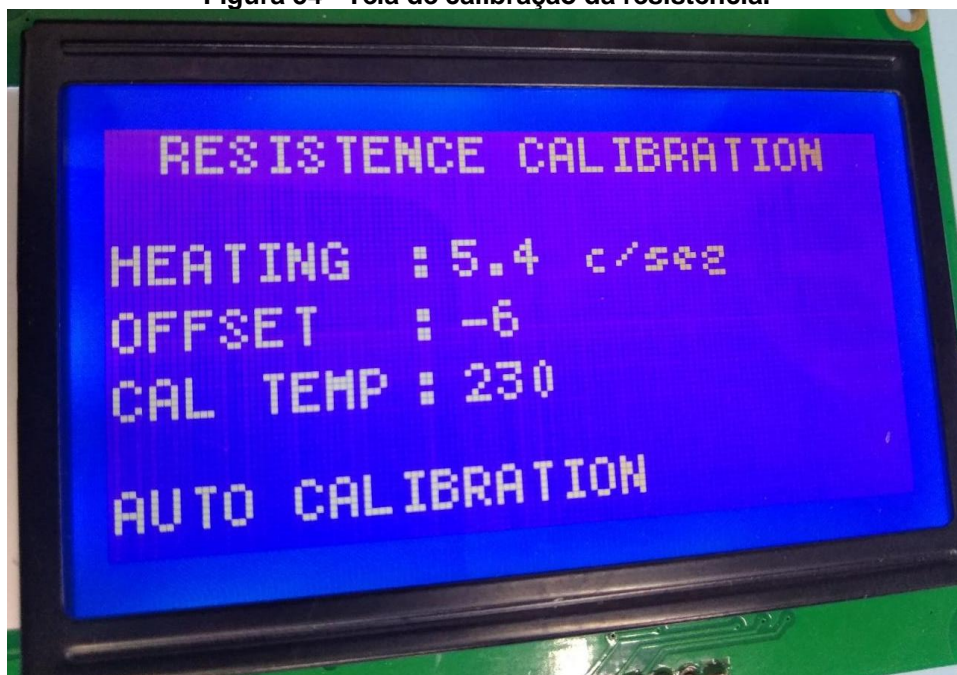


Figura 55 - Configuração da tela de calibração da resistência.

```
RESISTENCE CALIBRATION  
  
HEATING : 0.5 °c/seg -> tempo de aquecimento da resistência  
OFFSET   : 0         -> offset de desligamento da resistência em relação a temperatura alvo  
CAL TEMP : 230      -> define a temperatura de calibração da resistência  
AUTO CALIBRATION -> executa o processo de calibragem da resistência automaticamente.  
                Pressione a tecla 'C' para iniciar o processo. O Equipamento liga  
                a resistência e mantém ligada até atingir a temperatura de 230 graus  
                celsius, sem seguida ira calcular automaticamente o tempo de aquecimento  
                e definirá o valor "HEATING". Certifique-se de colocar o sensor de  
                temperatura adequadamente antes de iniciar o processo.
```


Figura 58 - Descrição dos pinos do Arduino.

```
* ----CONEXÕES----  
*  
* A0 - SO MAX6675  
* A1 - CS MAX6675  
* A2 - SCK MAX6675  
* A3 - SAÍDA BUZZER  
* A4 - SAÍDA FAN  
* A5 - SAÍDA RELE HEAT  
* A6 - NA  
* A7 - NA  
*  
* D0 - KEYPAD 0  
* D1 - NA  
* D2 - KEYPAD 2  
* D3 - KEYPAD 3  
* D4 - KEYPAD 4  
* D5 - KEYPAD 5  
* D6 - KEYPAD 6  
* D7 - KEYPAD 7  
* D8 - KEYPAD 8  
* D9 - DISPLAY RS  
* D10 - DISPLAY R/W  
* D11 - DISPLAY E  
* D12 - DISPLAY RST  
* D13 - NA
```

4.2. Custo

Para o desenvolvimento do projeto foi necessário a compra de alguns componentes e equipamentos. Todo equipamento citado abaixo foi realizado a compra nos sites mercado livre e Aliexpress. Menos o forno que foi comprado em descarte de sucatas.

A tabela 2 exibe os valores para os componentes utilizados no protótipo, sendo os valores obtidos através de consultas “online” em distribuidor.

QUANTIDADE	COMPONENTE	VALOR (R\$)
1	FORNO	300,00
1	VENTOINHA	200,00
1	TRANSFORMADOR	45,00
1	RELÉ DE ESTADO SOLIDO	40,00
1	PAINEL	70,00
1	PLACA	20,00
1	DISPALY 128x64	100,00
15 Metros	CABOS	15,00
1	DISJUNTOR BIPOLAR 32 A	40,00
	OUTROS	50,00
Total		R\$ 880,00

Tabela 3 - Valores dos componentes para o protótipo

4.3. Teste

Para a obtenção dos resultados experimentais o protótipo foi posto em funcionamento para obter os dados de temperatura e levantamento do perfil térmico em condições reais. A base da experimentação foi uma placa com os componentes SMD

É recomendado que os valores de temperatura das zonas de pré-aquecimento não sejam muito grandes. A transferência de muita energia (aquecimento muito rápido) pode fazer com que o substrato da placa de circuito sofra deformação fazendo com que os componentes não tenham a fixação mecânica ideal e a conexão elétrica fique prejudicada.

Outro fator para que o pré-aquecimento seja controlado é a ativação do fluxo. Parte da energia transferida nesses períodos é absorvida pelo fluxo que usa esse calor para executar a limpeza dos terminais e ilhas onde a solda irá se depositar e para evaporar o solvente do fluxo.

Do mesmo modo que a energia das zonas de pré-aquecimento é transferida para a placa de circuito impresso e para o fluxo a fim de que os processos que envolvem esses componentes (soldagem e limpeza de substrato) sejam corretamente realizados, também é necessária a remoção dessa energia após a etapa de deposição da liga de solda no substrato. Assim, também nessa etapa deve ser controlada a temperatura, porém de modo inverso.

O resfriamento incorreto da placa também acarreta danos ao substrato, gerando deformações, e as soldas que caso forem resfriadas muito rapidamente podem não formar as camadas intermetálicas corretamente.

Nesse teste foi usada apenas uma placa de cobre, posicionada no centro do forno.

Na Figura 59 é exibida a imagem da placa finalização, após o processo de soldagem no forno RS TEC.

Figura 59 - Resultado de soldagem obtido.



5. Discussão

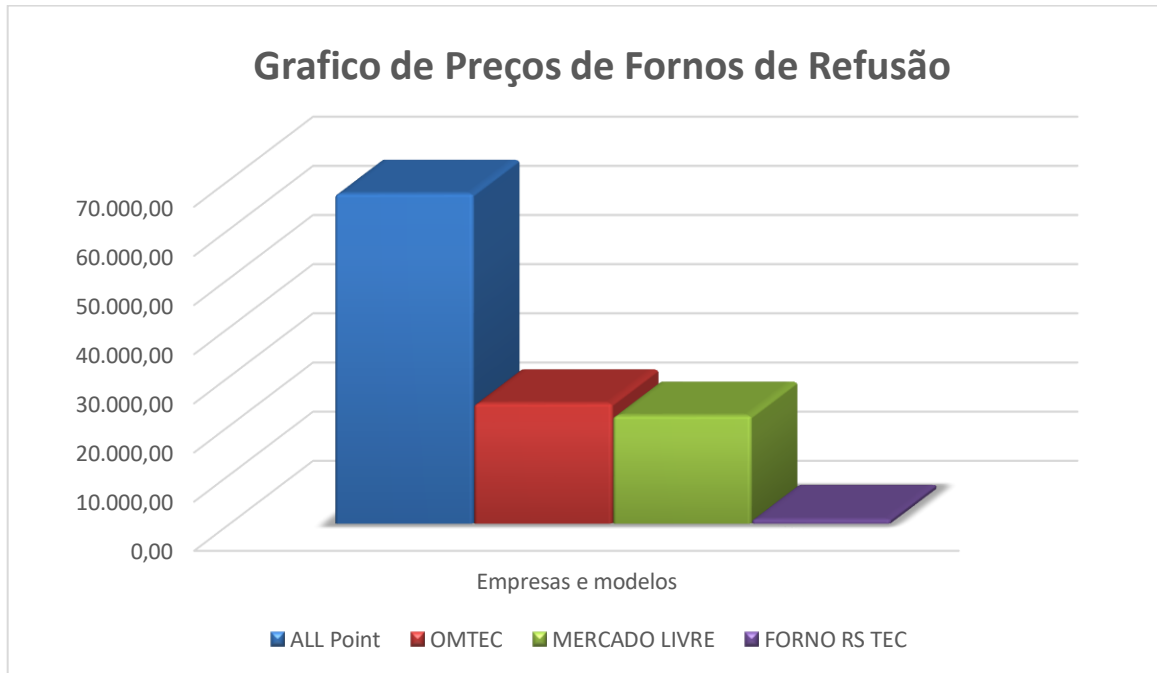
O forno de refusão RS TEC atingiu os objetivos proposto. Com a sua utilização, criou-se um novo processo para substituir o processo manual e caseiro, geralmente empregado por empresas de pequeno porte que não tem acesso a metodologia industrial devido ao seu alto custo, este novo processo obteve placas soldadas de forma mais rápida e com qualidade superior.

Quanto à qualidade deve-se ressaltar que, quando as placas são soldadas de modo artesanal, as trilhas podem ocorrer solda fria prejudicando seu funcionamento. Ao se utilizar o forno RS TEC, as trilhas são soldadas corretamente minimizando o risco de reparo.

De acordo com pesquisas realizadas em contato com empresas fabricantes e fornecedoras de fornos indústrias de refusão SMD, foi constatado que o investimento para aquisição dessas máquinas pode variar de R\$ 70.000,00 até R\$20.000,00. Sendo que as respectivas máquinas são destinadas à um elevado nível de produção, atendendo melhor as expectativas de empresas de grande porte e, portanto, representando um melhor custo-benefício. O intuito do projeto, é o desenvolvimento de uma máquina que possa atender baixas demandas de produção em um custo acessível, tornando extremamente viável para clientes de pequeno porte. Exemplo: Microempresas do segmento eletrônico; oficinas de manutenção; assistências técnicas; faculdades; universidades; escolas técnicas; etc.

Solicitado orçamentos foi possível desenvolver o gráfico 3, comparativo de preços de fornos de refusão SMD.

Gráfico 3 - Comparações de preços



Orçamentos conforme figuras 60, 61, 62 e 63.

Figura 60 - Orçamento Mercado livre.

mercado livre

Buscar produtos, marcas e muito mais...

Baixe grátis o app do Mercado Livre!

Categorias Ofertas da semana Descontos Mercado Pontos Seu histórico Suas compras Vender Crie a sua conta Entre Contato

Você também pode gostar: conversor digital - smart tv - receptor cinebox - mini system - placa amplificador

Voltar à lista | Eletrônicos, Áudio e Vídeo > Outros Eletrônicos Compartilhar | Vender um igual

Novo

Forno De Refusão Smd 5 Zonas

R\$ 22.000

12x R\$ 2.155⁸¹

VISA Mastercard Boleto

Mais informações

Envio para todo o país

Saiba os prazos de entrega e as formas de envio.

Calculador de frete

Único disponível!

Comprar agora

Compra Garantida, receba o produto que está esperando ou devolvemos o dinheiro.

Figura 61 - Orçamento Empresa ALL Point

Forno de Refusão		
Forno de refusão Configuração ideal zonas TOP zonas BOTTON		Valor do equipamento Pode variar dependendo de acessórios. US\$ 32.000,00 a US\$ 36.000,00 EXW
Forno de refusão Configuração mínima ideal 06 zonas TOP 06 zonas BOTTON		Valores de cada modelo pode variar. US\$ 18.000,00 EXW  

Estou à disposição para outras informações ou esclarecimentos de eventuais dúvidas.

Atenciosamente / Best Regards,

Carlos Almir Clemente



carlos.almir@allpoint.com.br

 almir.allpoint

Rua Inglês de Souza, 285 Aclimação,
01546-010, São Paulo-SP - Brasil

 (11)-2062-6900 -  (11) 99249-6595

www.allpoint.com.br

Figura 62 - Orçamento Empresa OMTEC



OMTEC INDÚSTRIA E COMÉRCIO LTDA
 Rua Conselheiro Moreira de Barros, 3244 – Lauzane Paulista – São Paulo – SP
 Cep: 02430-002 – Fone: 55 11 2255-6844 - 55 11 2256-8143
www.omtec.com.br - e-mail: comercial@omtec.com.br - nfe@omtec.com.br
 CNPJ: 53.668.455/0001-46 I.E. 111.272.062.113

Proposta N° /2018

São Paulo,

Cliente:	NATURA CAJAMAR	Tel:	(11) 9 4491-6337
A/C:	Suelen Oliveira	Cel:	
e-mail:	suelen.oliveirafr@gmail.com	Depto:	
Cidade/UF:	Cajamar - SP		
Pagamento:	À vista, 28DDL ou entrada, 28/56 DDL sem taxa financeira		
Transporte:	Por conta do destinatário		

It.	Qt.	Descrição dos Produtos	Unitário	Total	Prazo de entrega salvo venda prévia
01	01	FR-17 – FORNO DE REFUSÃO PARA SMD, POSSUI 6 ZONAS, SEIS QUENTES E UMA FRIA COM SISTEMA ANALÓGICO.	R\$ 24.500,00	R\$ 24.500,00	15 DIAS
02	01	BANCADA EM PERFIL DE ALUMÍNIO FR-17 (OPCIONAL).	R\$ 1.600,00	R\$ 1.600,00	IMEDIATO
03	01	MC-15 - EQUIPAMENTO PARA CONTAGEM DE COMPONENTES AXIAIS, RADIAIS E SMD COM POSSIBILIDADE DE CONTAGEM EM AMBAS AS DIREÇÕES PROGRESSIVA E REGRESSIVA.	R\$ 7.100,00	R\$ 7.100,00	IMEDIATO
04	01	PRINTER – MÁQUINA PARA APLICAR PASTA DE SOLDA EM PLACAS DE SMD. POSSUI AJUSTE X, Y NA MESA FEITOS ATRAVÉS DE KNOBS, POSSUI GARRAS AJUSTÁVEIS PARA FIXAÇÃO DA PLACA.	R\$ 7.800,00	R\$ 7.800,00	IMEDIATO

Figura 63 - Forno de Refusão empresa OMTEC.



O custo final da montagem do forno RS TEC, foi muito baixo se comparados a outros fornos semelhantes. O custo total do projeto gira em torno de R\$880,00, considerado que o forno foi desenvolvido a partir de materiais sucateados e que os principais gastos foram gerados com o material utilizado na fabricação da estrutura e com os componentes para a placa de controle.

Uma microempresa que fabrica placas de circuitos eletrônicos com pouco capital para investimentos, teria que recorrer a máquinas ultrapassadas, leiloadas por empresas maiores. Ou investimento pesado em trabalho manual, o que acarretaria em um custo elevado com mão-de-obra qualificada no ramo de soldagem e/ou microeletrônica.

A Tabela 3 apresenta a média de salário, e quantidade de horas trabalhadas de um técnico em soldagem.

Tabela 4 - Salário e horas trabalhadas de um técnico em soldagem.

Quantidade (Técnico)	Salário R\$	Horas mensais (t)
1	R\$1.833,00	200 horas
5	R\$ 9.165,00	1000 horas

Com a aquisição do forno de fusão RS TEC, o investimento seria significativamente menor e o retorno expressivamente maior.

Com o exemplo, a tabela 4 apresenta o retorno em média de R\$3.666,00 mensal.

Tabela 5 - Redução mão de obra com a implantação do forno.

Quantidade (Forno)	Preço R\$	Anos (t) / vida útil
1	R\$ 880,00	10
Quantidade (Técnico)	Salário R\$	Horas mensais (t)
3	R\$ 5.499,00	600 horas

6. Considerações Finais

A indústria eletrônica cresceu muito movida principalmente pela grande demanda por bens de consumo como eletrodomésticos, carros e equipamentos diversos do nosso dia-a-dia. Essa crescente demanda fez com que a indústria melhorasse seus processos para que pudesse continuar competitiva.

Melhoria de processos resume-se a redução de desperdícios levando a aumento da eficiência e maior competitividade. Também essa competitividade fez com que a velocidade dos processos aumentasse, não somente sua eficiência, fazendo com que etapas da produção como a soldagem se tornasse críticas.

Nesse âmbito esse trabalho traz um estudo básico sobre a metalurgia envolvida no processo de soldagem, a apresentação dos equipamentos mais comuns nos processos e apresenta uma solução para o controle de um parâmetro fundamental no processo de soldagem, a temperatura.

O experimento mostrou-se válido pois através do protótipo e dos dados obtidos foi possível a avaliação e interpretação de uma curva de temperatura. De posse dos dados verifica-se que a máquina na qual as medições foram feitas está com um perfil térmico adaptado para a placa proposta.

Os testes finais mostraram que houve um incremento na qualidade de uma redução do tempo gasto nas soldagens de componentes SMD, se comparados aos métodos convencionais, caseiros, geralmente empregados para esse fim.

O uso de termopares, especificamente os tipo K, atende aos requisitos do projeto plenamente. Sua faixa de operação de temperatura é compatível com o uso e seu custo é baixo quando comparado a outros tipos de sondas de temperatura.

Após aproximadamente 12 meses de trabalho, o projeto de desenvolver um forno de refusão para componentes SMD, de baixo custo chega ao seu final. Onde todos os objetivos propostos foram atingidos.

Como possíveis melhorias e trabalhos futuros propõem-se o aumento do número de sondas para aferição da temperatura, disponibilização da troca de alguns

parâmetros como tempo entre amostras, ou número de amostras por segundo, a avaliação de outros meios para transferência dos dados coletados (Bluetooth, zig-bee, wi-fi dentre outros) para a parte física.

Este projeto mostrou-se bem útil e flexível, podendo ser utilizado em escolas técnicas, universidades, colaborando de alguma forma para o aprendizado dos alunos ou até mesmo em uma pequena empresa de soldagem de SMD.

Referências bibliográficas

FONTELLAS, M. J.; SIMÕES, M. G.; FARIAS, S. H.; FONTELLAS, R. G. S. (2018) Metodologia da pesquisa científica: diretrizes para a elaboração de um protocolo de pesquisa. Trabalho realizado no Núcleo de Bioestatística Aplicado à pesquisa da Universidade da Amazônia. Acesso em 05 de abril de 2018.

JUSTI, M.A. / LOMBARDI, J.C. / BIZARRIA, F.C.P. (2006) – Automatização do processo de refusão para soldagem Lead Free – SAE BRASIL Acesso em 08 de abril de 2018.

MORIYA, J. (2006) – Apostila de treinamento Lead Free – TecSolda consultoria. Acesso em 15 de abril de 2018.

WASSINK, R.J.K. (1989) – Soldering in electronics- 2nd edition – Electrochemical Publication Limited. Acesso em 25 de abril de 2018.

http://bdtd.ibict.br/vufind/Record/ITAU_4798a1513fa60acbafd62706f4796905. Acesso em 25 de abril de 2018.

<http://www.brainy-bits.com/connect-and-use-a-spi-oled-display>. Acesso em 25 de abril de 2018.

<http://www.bdtd.unitau.br>. Acesso em 26 de abril de 2018.

http://www.professorpetry.com.br/Ensino/Defesas_Pos_Graduacao/Defesa_16_Daniel_Korbes_Tracador_de_Perfil_Termico_para_Uso_na_Industria_Eletronica.pdf. Acesso em 27 de abril de 2018.

<https://www.bga-reworking.co.uk/knowledge-base.html>. Acesso em 27 de abril de 2018.

<http://www.abraci.org.br>. Acesso em 01 de maio de 2018.

<http://www.smtnet.com>. Acesso em 01 de maio de 2018.

<http://www.nei.com.br>. Acesso em 04 de maio de 2018.

<http://www.madelltech.com>. Acesso em 04 de maio de 2018.

<http://www.ami.ac.uk>. Acesso em 04 de maio de 2018.

<http://www.bga-reworking.co.uk/knowledge-base.html>. Acesso em 07 de maio de 2018.

<http://smd-on-line.com>. Acesso em 07 de maio de 2018.

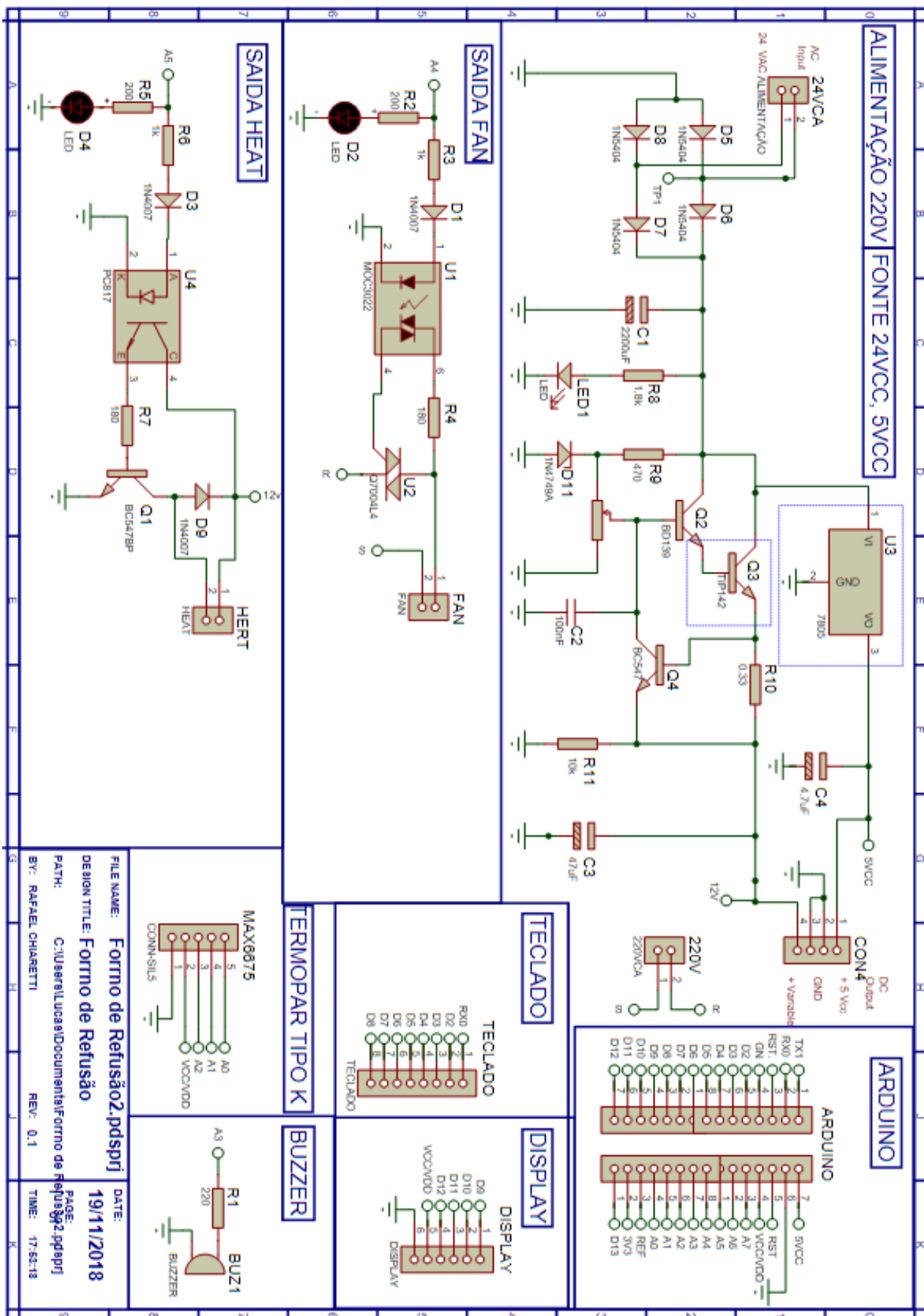
<http://www.ami.ac.uk>. Acesso em 10 de maio de 2018.

<http://www.resumoescolar.com.br/fisica/termometria>. Acesso em 10 de maio de 2018.

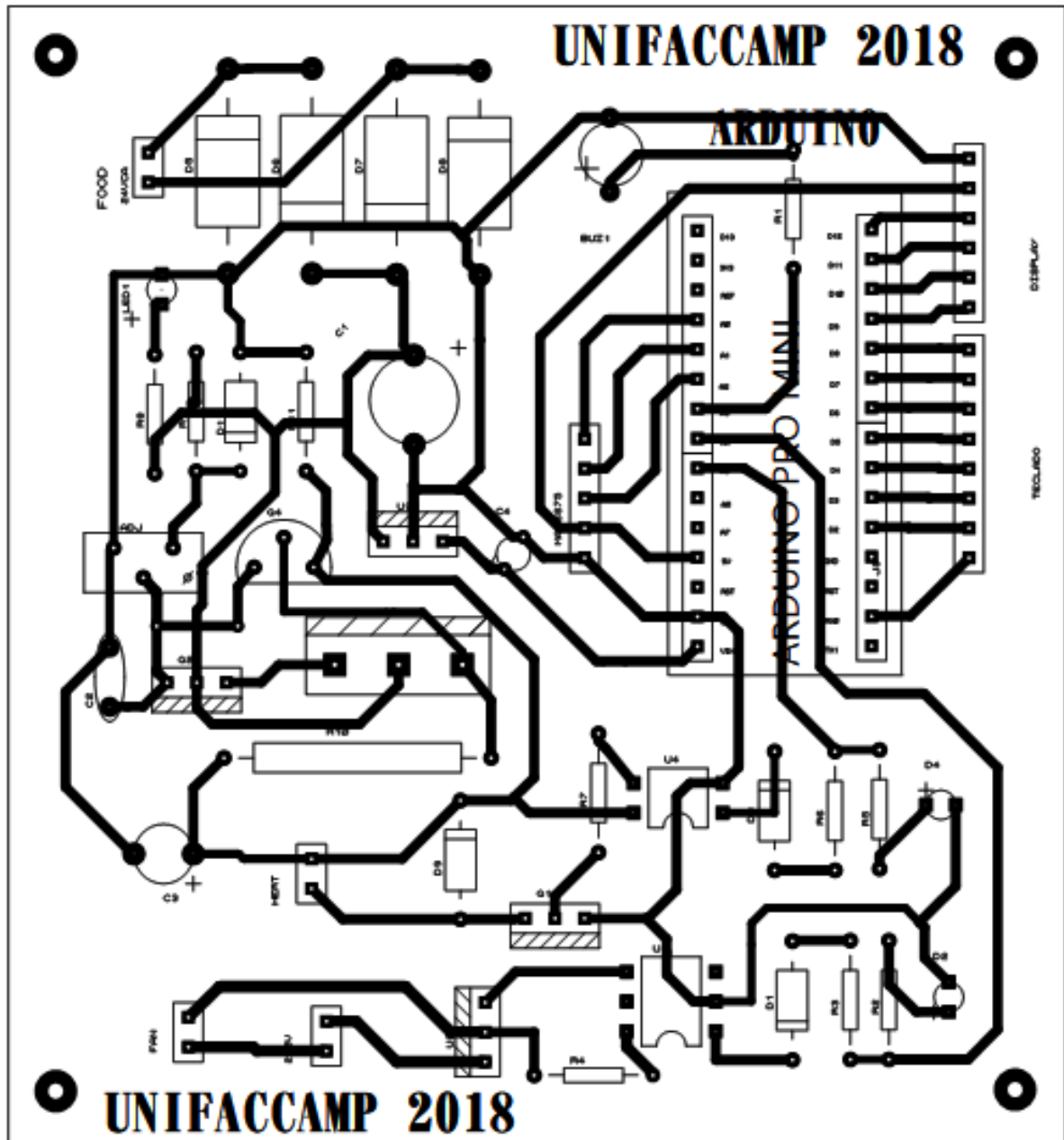
<http://www.mainzmaquinas.com.br/termopar-tipo-k>. Acesso em 15 de maio de 2018.

http://www.abraci.org.br/arquivos/desafios_tecnicos.pdf. Acesso em 15 de maio de 2018.

ANEXO I – DIAGRAMA ELÉTRICO



ANEXO II – LAYOUT DA PLACA



ANEXO III - CÓDIGO FONTE “FIRMWARE”

```
//CONTROLADOR DE TEMPERATURA COM 4 RAMPAS E 9 MEMORIAS
```

```
#include <max6675.h> //biblioteca do sensor de temperatura
```

```
#include <EEPROM.h> //biblioteca para EEPROM
```

```
#include <Keypad.h> //biblioteca para o teclado
```

```
#include "U8glib.h" //biblioteca para display gráfico
```

```
U8GLIB_ST7920_128X64_1X u8g(11,10,9,12); //Enable, RW, RS, RESET
```

```
//define os pinos físicos de conexão
```

```
#define buzzer A3
```

```
#define fan A4
```

```
#define heat A5
```

```
const byte ROWS = 4; //quatro linhas
```

```
const byte COLS = 4; //quatro colunas
```

```
char keys[ROWS][COLS] = {
```

```
{'1','2','3','A'},
```

```
{'4','5','6','B'},
```

```
{'7','8','9','C'},
```

```
{'*', '0', '#', 'D'}
```

```
};
```

```

byte rowPins[ROWS] = {0,2,3,4}; //pinos referentes as linhas

byte colPins[COLS] = {5,6,7,8}; //pinos referentes as colunas

Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS,
COLS);

//pinos de conexão do sensor de temperatura MAX6675
(thermoCLK,thermoCS,thermoDO)

MAX6675 thermocouple(A2,A1,A0);

unsigned int tempo = 0;

//unsigned int ult_tempo = 0;

unsigned int ult_tempo1 = 0;

unsigned int ult_tempo2 = 0;

unsigned int ult_tempo3 = 0;

unsigned int ult_tempo4 = 0;

unsigned int mostra_tempo;

unsigned int seg = 0;

unsigned int min = 0;

byte temporestante1=0;

byte temporestante2=0;

```

```

byte temporestante3=0;

byte temporestante4=0;

byte temp_1;

byte time_1;

byte temp_2;

byte time_2;

byte temp_3;

byte time_3;

byte temp_4;

byte time_4;

byte cal_temp;           //temperatura de calibração da resistência

float heating;           //tempo de aquecimento da resistência em °C/s

int temp_inicial;        //armazena a temperatura inicial

int set_temp;            //armazena temporariamente a temperatura do
processo atual

int read_temp;           //armazena a temperatura lida tempo real

int refresh;             //taxa de atualização para o gráfico

int Xg[128];             //tempo mapeado

int Yg[128];             //temperatura mapeada

int i;                   //ponteiro do FOR para desenho do gráfico

```

```

unsigned long previousMillis = 0; //contador para a função read_sensor

unsigned long previousMillis1 = 0; //contador para o buzzer2

unsigned long previousMillis2 = 0; //contador para o timer

unsigned long previousMillis3 = 0; //contador para o grafico

unsigned char heat_indicator = 0; //indica resistencia liga/desligada

unsigned char id_display = 0; //flag da tela atual

char key; //armazena a última tecla pressionada

char keypressed; //tecla pressionada

char memory; //leitura e gravação da memoria

char perfil; //nome do perfil

char grava_memoria; //número do perfil onde será gravado

byte flag_calibration = 0;

byte flag_cursor = 0; //controle de posicao do cursor

byte flag_start = 0; //controle de start/stop

byte flag_timer = 0; //controle dos timer 1st/2st/2st/4st

byte flag_temp = 0; //controle do timer por temperatura

byte flag_fan = 0; //indica de o cooler esta ligado ou desligado

byte key_beep = 0; //liga ou desliga o beep das teclas

```

```

byte fan_mode = 0;           //controle do fan

byte temp_fan = 0;          //temperatura de desligamento automatico do
fan

byte beep_time = 0;         //tempo de aviso do final de operação em
segundos

int temp_offset = 0;        //off set de controle de temperatura

byte offset = 0;           //indica se temp_offset é positivo(1) ou negativo(0)

byte set_default = 1;       //restaura padrões de fábrica se diferente de 1

String string;             //armazena os números digitados

void setup() {

// Serial.begin(9600);      // inicializa a saída serial

pinMode(heat, OUTPUT);     // SAÍDA PARA ACIONAR A RESISTENCIA

pinMode(buzzer, OUTPUT);   // SAIDA PARA BUZZER

pinMode(fan, OUTPUT);      // SAIDA PARA FAN

if(EEPROM.read(50)!=1)write_memory_first_time(); //se a posição de
memória 50 for diferente de 1, grava os padrões de fábrica.

fan_mode = EEPROM.read(8);

key_beep = EEPROM.read(9);

temp_fan = EEPROM.read(10);

beep_time = EEPROM.read(11);

offset = EEPROM.read(12);

```

```

if(offset==0)temp_offset = EEPROM.read(13)-256;

if(offset==1)temp_offset = EEPROM.read(14);

perfil    = EEPROM.read(15);

heating=float(EEPROM.read(17))+float(EEPROM.read(18))/100;

cal_temp  = EEPROM.read(19);

read_memory();

digitalWrite(buzzer, HIGH); // beep curto ao ligar o equipamento

delay(150);

digitalWrite(buzzer, LOW);

}

```

//função de leitura do sensor de temperatura, com delay de 250ms

```

void read_sensor(){

    unsigned long currentMillis = millis();

    if (currentMillis - previousMillis >= 250) {

        previousMillis = currentMillis;

        read_temp=(thermocouple.readCelsius());

    }

}

```

```

void fan1(){

```

```

if(fan_mode==0){

    digitalWrite(fan, LOW); //desliga fan

    flag_fan=0;

}

if(fan_mode==2){ //modo automatico

    if(read_temp>temp_fan && flag_timer==5){

        digitalWrite(fan, HIGH);

        flag_fan=1;

    }else if(read_temp<temp_fan && flag_timer==5){

        digitalWrite(fan, LOW);

        flag_timer=6;

        flag_fan=0;

    }

}

}

```

```

void loop() {

    u8g.firstPage();

    do{

        if(id_display==0){

```



```

display1();

}else if(id_display==1){

    display2();

}else if(id_display==2 && flag_cursor==0){

    display3();

}else if(id_display==2 && flag_cursor==1){

    display4();

}else if(id_display==3){

    display5();

}else if(id_display==4){

    display6();

}

}while(u8g.nextPage());

read_sensor();

keypressed = keypad.getKey();

fan1();

if(keypressed){

    buzzer0();

    key = keypressed;

```

```

    buttons();

}

if(keypad.getState()==HOLD){

    buttons();

}

if(flag_calibration==1)calibration();

if(flag_start==1){ //liga as funções ao iniciar o processo

    control_temp();

    timer();

    if(refresh<128)grafico();

    //aumenta ou diminui o tempo 4 durante a execução

    if(keypressed=='C'){ //botao +

        temporestante4=temporestante4+30;

        if(temporestante4>255){

            temporestante4=255;

        }

    }

    if(keypressed=='D'){ //botao -

        temporestante4=temporestante4-30;

```

```
        if(temporestante4<0){  
            temporestante4=0;  
        }  
    }  
  
}else if(flag_start==0){ //desliga as funções ao parar o processo  
  
    flag_temp=0;  
  
    set_temp=0;  
  
    heat_indicator=0;  
  
    digitalWrite(heat,LOW);  
  
    digitalWrite(buzzer,LOW);  
  
}  
  
} //void loop
```

ANEXO IV - CÓDIGO FONTE “TECLADO”

```
void buttons(){

//mater pressionada a tecla 3 para calibrar resistencia

if(keypad.getState()==HOLD && id_display==0 && flag_start==0){

    if(key=='3'){

        if(flag_start==0){

            id_display=4;

        }

    }

}

//mater pressionada a tecla 2 para entrar no menu

if(keypad.getState()==HOLD && id_display==0){

    if(key=='2'){

        if(flag_start==0){

            id_display=1;

            flag_cursor=0;

        }

    }

}
```

```

//sai do menu e grava os dados

if(keypad.getState()===HOLD){

    if(key=='0'){

        string="";

        id_display=0;

        flag_cursor=0;

        memory=0;

        grava_config();

    }

}

if(keypad.getState()===HOLD){

    if(key=='D' && id_display==0 && flag_cursor==0 && flag_start==0){

        memory=1; //leitura de memoria

        id_display=2;

    }

}

if(keypad.getState()===HOLD){

    if(key=='C' && flag_cursor==0 && flag_start==0){

        memory=2; //escrita de memoria

    }

}

```

```

}

if(keypressed=='B'){

    if(id_display==0)id_display=3;

    else if(id_display==3)id_display=0;

}

//liga o fan ao manter pressionada a tecla 1

if(keypad.getState()==HOLD){

    if(key=='1' && flag_start==0 && memory !=2){

        digitalWrite(fan, HIGH); //liga fan

        flag_fan=1;

    }

}

if(keypressed=='1' && flag_start==0){

    digitalWrite(fan, LOW); //desliga fan

    flag_fan=0;

}

//gravação e leitura das memórias 1~9

switch(memory){

```

```
case (1):  
  
if(keypressed=='1'){  
  
    perfil=1;  
  
    read_memory();  
  
    EEPROM.update(15, perfil);  
  
    id_display=0;  
  
    memory=0;  
  
    flag_cursor=0;  
  
}  
  
if(keypressed=='2'){  
  
    perfil=2;  
  
    read_memory();  
  
    EEPROM.update(15, perfil);  
  
    id_display=0;  
  
    memory=0;  
  
    flag_cursor=0;  
  
}  
  
if(keypressed=='3'){  
  
    perfil=3;  
  
    read_memory();  
  
    EEPROM.update(15, perfil);
```

```
id_display=0;

memory=0;

flag_cursor=0;

}

if(keypressed=='4'){

    perfil=4;

    read_memory();

    EEPROM.update(15, perfil);

    id_display=0;

    memory=0;

    flag_cursor=0;

}

if(keypressed=='5'){

    perfil=5;

    read_memory();

    EEPROM.update(15, perfil);

    id_display=0;

    memory=0;

    flag_cursor=0;

}

if(keypressed=='6'){
```



```
    perfil=6;

    read_memory();

    EEPROM.update(15, perfil);

    id_display=0;

    memory=0;

    flag_cursor=0;
}

if(keypressed=='7'){

    perfil=7;

    read_memory();

    EEPROM.update(15, perfil);

    id_display=0;

    memory=0;

    flag_cursor=0;
}

if(keypressed=='8'){

    perfil=8;

    read_memory();

    EEPROM.update(15, perfil);

    id_display=0;

    memory=0;
```

```
    flag_cursor=0;

}

if(keypressed=='9'){

    perfil=9;

    read_memory();

    EEPROM.update(15, perfil);

    id_display=0;

    memory=0;

    flag_cursor=0;

}

break;

case (2):

if(keypad.getState()===HOLD){

    if(key=='1'){

        grava_memoria=1;

        write_memory();

        memory=0;

        buzzer1();

    }

    if(key=='2'){
```

```
grava_memoria=2;

write_memory();

memory=0;

buzzer1();

}

if(key=='3'){

grava_memoria=3;

write_memory();

memory=0;

buzzer1();

}

if(key=='4'){

grava_memoria=4;

write_memory();

memory=0;

buzzer1();

}

if(key=='5'){

grava_memoria=5;

write_memory();

memory=0;
```

```
buzzer1();  
  
}  
  
if(key=='6'){  
  
grava_memoria=6;  
  
write_memory();  
  
memory=0;  
  
buzzer1();  
  
}  
  
if(key=='7'){  
  
grava_memoria=7;  
  
write_memory();  
  
memory=0;  
  
buzzer1();  
  
}  
  
if(key=='8'){  
  
grava_memoria=8;  
  
write_memory();  
  
memory=0;  
  
buzzer1();  
  
}  
  
if(key=='9'){
```

```

    grava_memoria=9;

    write_memory();

    memory=0;

    buzzer1();

}

}

break;

}

switch(id_display){ //alterna a funções dos botões de acordo com a tela

    case (0): //display_1

//botao start-stop

if(keypad.getState()==HOLD){

    if(key=='A'){

        buzzer1();

        start();

    }

}

if(keypressed=='A'){

    if(flag_start==1){

        stop();

```

```
id_display=0;

}

}

//botao avanca-retrocede

if(key=='#'){ //avança casa

string="";

if(flag_start==0){

flag_cursor++;

if(flag_cursor>8){

flag_cursor=0;

}

}

}

if(key=='*'){ //volta casa

string="";

flag_cursor--;

if(flag_cursor<1){

flag_cursor=8;

}

}
```

```
//botao aumenta-diminui

if(flag_cursor==1){

    if(keypressed!='*' && keypressed!='#'){

        string.concat(keypressed);

        if(string.length()==3){

            temp_1=string.toInt();

            string="";

        }

    }

}

if(flag_cursor==2){

    if(keypressed!='*' && keypressed!='#'){

        string.concat(keypressed);

        if(string.length()==3){

            time_1=string.toInt();

            string="";

        }

    }

}
```

```
if(flag_cursor==3){  
    if(keypressed!='*' && keypressed!='#'){  
        string.concat(keypressed);  
        if(string.length()==3){  
            temp_2=string.toInt();  
            string="";  
        }  
    }  
}
```

```
if(flag_cursor==4){  
    if(keypressed!='*' && keypressed!='#'){  
        string.concat(keypressed);  
        if(string.length()==3){  
            time_2=string.toInt();  
            string="";  
        }  
    }  
}
```



```
if(flag_cursor==5){  
  
    if(keypressed!='*' && keypressed!='#'){  
  
        string.concat(keypressed);  
  
        if(string.length()==3){  
  
            temp_3=string.toInt();  
  
            string="";  
  
        }  
  
    }  
  
}
```

```
if(flag_cursor==6){  
  
    if(keypressed!='*' && keypressed!='#'){  
  
        string.concat(keypressed);  
  
        if(string.length()==3){  
  
            time_3=string.toInt();  
  
            string="";  
  
        }  
  
    }  
  
}
```

```
if(flag_cursor==7){
```

```
if(keypressed!='*' && keypressed!='#'){  
    string.concat(keypressed);  
    if(string.length()==3){  
        temp_4=string.toInt();  
        string="";  
    }  
}  
}
```

```
if(flag_cursor==8){  
    if(keypressed!='*' && keypressed!='#'){  
        string.concat(keypressed);  
        if(string.length()==3){  
            time_4=string.toInt();  
            string="";  
        }  
    }  
}
```

break;

case (1): //display_2

```

//botao avanca-retrocede

if(key=='#'){ //avança casa

    if(flag_start==0){

        flag_cursor++;

        if(flag_cursor>5){

            flag_cursor=0;

        }

    }

}

if(key=='*'){ //volta casa

    flag_cursor--;

    if(flag_cursor<1){

        flag_cursor=5;

    }

}

if(flag_cursor==1){ // beep das teclas

    if(key=='C')key_beep=1;

    else if(key=='D')key_beep=0;

}

```

```
if(flag_cursor==2){ // fan mode

    if(key=='C')fan_mode=2;

    else if(key=='D')fan_mode=0;

}
```

```
if(flag_cursor==3){ // temp fan

    if(key=='C')temp_fan++;

    else if (key=='D')temp_fan--;

}
```

```
if(flag_cursor==4){ // end_ beep_time

    if(key=='C')beep_time++;

    else if(key=='D')beep_time--;

}
```

```
if(flag_cursor==5){ //restore default

    if(key=='C')set_default=0;

    else if(key=='D')set_default=1;

}
```

```
break;
```

```
case (2): //display_3 e display_4
```

```
//botao avanca-retrocede
```

```
if(key=='#'){ //avança casa
```

```
if(flag_start==0){
```

```
flag_cursor=1;
```

```
}
```

```
}
```

```
if(key=='*'){ //volta casa
```

```
flag_cursor=0;
```

```
}
```

```
break;
```

```
case (3)://display_5
```

```
if(keypad.getState()==HOLD){
```

```
if(key=='A'){
```

```
buzzer1();
```

```
start();
```

```
}
```

```
}
```

```
if(keypressed=='A'){
```

```
stop();
```

```
}
```

```
break;
```

```
case (4)://display_6 - resistence calibration
```

```
//botao avanca-retrocede
```

```
if(key=='#'){ //avança casa
```

```
if(flag_start==0){
```

```
flag_cursor++;
```

```
if(flag_cursor>4){
```

```
flag_cursor=0;
```

```
}
```

```
}
```

```
}
```

```
if(key=='*'){ //volta casa
```

```
flag_cursor--;
```

```
if(flag_cursor<1){
```

```
flag_cursor=4;
```

```
}
```

```
}
```

```
if(flag_cursor==1){
```

```

if(key=='C'){
    heating=heating+0.1;
    if(heating>9.9)heating=9.9;
}

if(key=='D'){
    heating=heating-0.1;
    if(heating<0.1)heating=0.1;
}
}

if(flag_cursor==2){ // offset temperatura

    if(key=='C')temp_offset++;

    else if(key=='D')temp_offset--;

}

if(flag_cursor==3){

    if(key=='C')cal_temp=cal_temp+5;

    if(cal_temp>250)cal_temp=250;

    if(key=='D')cal_temp=cal_temp-5;

    if(cal_temp<0)cal_temp=0;

}

```

```
if(flag_cursor==4){  
    if(key=='C'){  
        temp_inicial=read_temp;  
        flag_calibration=1;  
        flag_start=2;  
    }  
    if(key=='D'){  
        flag_calibration=0;  
        flag_start=0;  
        min=0;  
        seg=0;  
    }  
}  
  
break;  
  
} //switch  
  
} //buttons
```


ANEXO V - CÓDIGO FONTE “BUZZER”

```
//beep curto
```

```
void buzzer0(){  
  
    if(key_beep==1){  
  
        digitalWrite(buzzer, HIGH);  
  
        delay(80);  
  
        digitalWrite(buzzer, LOW);  
  
    }else{}  
  
}
```

```
//beep longo
```

```
void buzzer1(){  
  
    digitalWrite(buzzer, HIGH);  
  
    delay(1000);  
  
    digitalWrite(buzzer, LOW);  
  
}
```

```
//beep curto intermitente
```

```
void buzzer2(){  
  
    unsigned long currentMillis1 = millis();  
  
    digitalWrite(buzzer, HIGH);
```

```
if (currentMillis1 - previousMillis1 >= 800) {  
  
    previousMillis1 = currentMillis1;  
  
    digitalWrite(buzzer, LOW);  
  
}  
  
}
```

ANEXO VI - CÓDIGO FONTE “CALIBRAÇÃO”

```
void calibration(){  
  
    if(read_temp<cal_temp+1){  
  
        digitalWrite(heat, HIGH);  
  
        heat_indicator=1;  
  
        run_time();  
  
    }else{  
  
        digitalWrite(heat, LOW);  
  
        heat_indicator=0;  
  
        flag_calibration=0;  
  
        flag_start=0;  
  
        flag_cursor=0;  
  
        heating=((float)cal_temp-temp_inicial) / (min*60+seg);  
  
        min=0;  
  
        seg=0;  
  
        if(heating>9.9)heating=9.9;  
  
        if(heating<0.1)heating=0.1;  
  
        buzzer1();  
  
    }  
  
}
```

ANEXO VII - CÓDIGO FONTE “DISPLAY”

```
// tela inicial e execução
```

```
void display1() { //id_display=0
```

```
    u8g.setFont(u8g_font_5x7);
```

```
        if(perfil==1)u8g.drawStr(78,47, " 1:LEADED");
```

```
    else if(perfil==2)u8g.drawStr(78,47, "2:LEADFREE");
```

```
    else if(perfil==3)u8g.drawStr(78,47, " 3:REFLOW");
```

```
    else if(perfil==4)u8g.drawStr(78,47, " 4:SOLDER");
```

```
    else if(perfil==5)u8g.drawStr(78,47, " 5:USER1");
```

```
    else if(perfil==6)u8g.drawStr(78,47, " 6:USER2");
```

```
    else if(perfil==7)u8g.drawStr(78,47, " 7:USER3");
```

```
    else if(perfil==8)u8g.drawStr(78,47, " 8:USER4");
```

```
if(memory==2){
```

```
    u8g.setFont(u8g_font_5x7);
```

```
    u8g.drawStr( 2, 47, "STORE:");
```

```
    u8g.setPrintPos(35,47);
```

```
    if(key!='C')u8g.print(key);
```

```
}
```

```
u8g.setPrintPos(56,47);
```

```
u8g.print(string);

u8g.setFont(u8g_font_5x7);

u8g.drawLine(0,0,128,0); //traço horizontal 1

u8g.drawLine(0,0,0,38); //traço vertical 1

u8g.drawLine(26,0,26,38); //traço vertical 2

u8g.drawStr( 32, 8, "1ST");

u8g.drawLine(51,0,51,38); //traço vertical 3

u8g.drawStr( 57, 8, "2ST");

u8g.drawLine(76,0,76,38); //traço vertical 4

u8g.drawStr( 82, 8, "3ST");

u8g.drawLine(101,0,101,38); //traço vertical 5

u8g.drawStr( 108, 8, "4ST");

u8g.drawLine(127,0,127,38); //traço vertical 6

u8g.drawLine(0,9,128,9); //traço horizontal 2

u8g.drawStr( 2, 18, "TEMP");

u8g.drawStr( 2, 26, "TIME");

u8g.drawLine(0,28,128,28); //traço horizontal 3

u8g.drawLine(0,38,128,38); //traço horizontal 4
```

```

u8g.drawStr(107,57,"TIME");

if(flag_start==1){

    u8g.drawStr(5,8,"RUN");

        u8g.setPrintPos(33,36);

u8g.print(temporestante1);

        u8g.setPrintPos(57,36);

u8g.print(temporestante2);

        u8g.setPrintPos(82,36);

u8g.print(temporestante3);

        u8g.setPrintPos(108,36);

u8g.print(temporestante4);

}else{

    u8g.drawStr(3,8,"STOP");

}

```

```
//mostra tempo total
```

```

if(flag_timer>=1){

    if(min>10){

        u8g.setPrintPos(102,64);

        u8g.print(min);

    }else{

```

```

    u8g.setPrintPos(108,64);

    u8g.print(min);

}

    u8g.drawStr(112,64, ".");

if(seg<10){

    u8g.drawStr(115,64,"0");

    u8g.setPrintPos(120,64);

    u8g.print(seg);

}else{

    u8g.drawStr(115,64," ");

    u8g.setPrintPos(116,64);

    u8g.print(seg);

}

}

}

//tempo total

//1st

if(flag_cursor==1){

    u8g.setFont(u8g_font_courB08);

}else{}

u8g.setPrintPos(33,18);

u8g.print(temp_1);

```

```
u8g.setFont(u8g_font_5x7);

if(flag_cursor==2){

    u8g.setFont(u8g_font_courB08);

}else{}

u8g.setPrintPos(33,26);

u8g.print(time_1);

u8g.setFont(u8g_font_5x7);

//2st

if(flag_cursor==3){

    u8g.setFont(u8g_font_courB08);

}else{}

u8g.setPrintPos(57,18);

u8g.print(temp_2);

u8g.setFont(u8g_font_5x7);

if(flag_cursor==4){

    u8g.setFont(u8g_font_courB08);

}else{}

u8g.setPrintPos(57,26);

u8g.print(time_2);

u8g.setFont(u8g_font_5x7);
```



```
//3st

if(flag_cursor==5){

    u8g.setFont(u8g_font_courB08);

}

}else{}

u8g.setPrintPos(82,18);

u8g.print(temp_3);

u8g.setFont(u8g_font_5x7);

if(flag_cursor==6){

    u8g.setFont(u8g_font_courB08);

}

}else{}

u8g.setPrintPos(82,26);

u8g.print(time_3);

u8g.setFont(u8g_font_5x7);

//4st

if(flag_cursor==7){

    u8g.setFont(u8g_font_courB08);

}

}else{}

u8g.setPrintPos(108,18);

u8g.print(temp_4);
```

```
u8g.setFont(u8g_font_5x7);

if(flag_cursor==8){

    u8g.setFont(u8g_font_courB08);

}else{}

u8g.setPrintPos(108,26);

u8g.print(time_4);

u8g.setFont(u8g_font_5x7);

u8g.setPrintPos(2,36);

if(flag_timer<5){

u8g.print(flag_timer);

u8g.drawStr(9,36,"st");

}else{

    u8g.drawStr(3,36,"END ");

}

u8g.drawStr(56,57,"TEMP");

if(read_temp>99)u8g.setPrintPos(56,64);

else u8g.setPrintPos(60,64);

u8g.print(read_temp);

u8g.drawStr(72,64,"c");
```

```

u8g.setFont(u8g_font_5x7);

if(heat_indicator==1)u8g.drawStr(1,64,"HEAT");

else if (flag_fan==1)u8g.drawStr(1,64,"FAN");

}

//display1

// tela menu de configuração

void display2(){//id_display=1

u8g.setFont(u8g_font_5x8);

u8g.drawStr( 38,7, "CONFIG MENU");

u8g.setFont(u8g_font_5x7);

u8g.drawStr( 3,18, "KEY BEEP");

u8g.drawStr( 59,18, ":");

if(flag_cursor==1)u8g.setFont(u8g_font_courB08);

if(key_beep==0)u8g.drawStr(65,18, "OFF");

else if(key_beep==1)u8g.drawStr(65,18, "ON");

u8g.setFont(u8g_font_5x7);

u8g.drawStr( 2,27, "FAN MODE");

u8g.drawStr( 59,27, ":");

```

```
if(flag_cursor==2)u8g.setFont(u8g_font_courB08);

u8g.setPrintPos(65,27);

if(fan_mode==0)u8g.drawStr( 65,27, "OFF");

else if(fan_mode==1)u8g.drawStr( 65,27, "ON");

else if(fan_mode==2)u8g.drawStr( 65,27, "AUTO");

u8g.setFont(u8g_font_5x7);

u8g.drawStr( 1,36, "TEMP FAN");

u8g.drawStr( 59,36, ":");

if(flag_cursor==3)u8g.setFont(u8g_font_courB08);

u8g.setPrintPos(65,36);

u8g.print(temp_fan);

u8g.setFont(u8g_font_5x7);

u8g.drawStr( 2,45, "BEEP TIME");

u8g.drawStr( 59,45, ":");

if(flag_cursor==4)u8g.setFont(u8g_font_courB08);

u8g.setPrintPos(65,45);

u8g.print(beep_time);

u8g.setFont(u8g_font_5x7);
```

```

u8g.drawStr( 1,63, "LOAD DEFAULT");

u8g.drawStr( 61,63, ":");

if(flag_cursor==5)u8g.setFont(u8g_font_courB08);

    if(set_default !=1){

        u8g.drawStr(67,63,"YES");

    }else{u8g.drawStr(67,63,"NO");

    }

}

```

//tela das memorias gravadas 1 a 4

```
void display3(){//id_display=2
```

```
//coluna 1
```

```
u8g.setFont(u8g_font_baby);
```

```
u8g.drawStr(5,6, "1");
```

```
u8g.drawStr(5,22, "2");
```

```
u8g.drawStr(5,38, "3");
```

```
u8g.drawStr(5,54, "4");
```

```
u8g.drawLine(0,14,128,14); //traço horizontal 1
```

```
u8g.drawLine(0,30,128,30); //traço horizontal 2
```

```
u8g.drawLine(0,46,128,46); //traço horizontal 3
```

```
u8g.drawStr(98,6,"LEADED");

u8g.drawStr(98,22,"FREE");

u8g.drawStr(98,38,"REFLOW");

u8g.drawStr(98,54,"SOLDER");

u8g.setPrintPos(20,6); //memory 1
u8g.print(EEPROM.read(200)); //temp_1
u8g.setPrintPos(20,12);
u8g.print(EEPROM.read(201)); //time_1

u8g.setPrintPos(20,22); //memory 2
u8g.print(EEPROM.read(208)); //temp_1
u8g.setPrintPos(20,28);
u8g.print(EEPROM.read(209)); //time_1

u8g.setPrintPos(20,38); //memory 3
u8g.print(EEPROM.read(216)); //temp_1
u8g.setPrintPos(20,44);
u8g.print(EEPROM.read(217)); //time_1
```

```
u8g.setPrintPos(20,54); //memory 4  
  
u8g.print(EEPROM.read(224)); //temp_1  
  
u8g.setPrintPos(20,60);  
  
u8g.print(EEPROM.read(225)); //time_1
```

```
//coluna 2
```

```
u8g.setPrintPos(38,6); //memory 1  
  
u8g.print(EEPROM.read(202)); //temp_2  
  
u8g.setPrintPos(38,12);  
  
u8g.print(EEPROM.read(203)); //time_2
```

```
u8g.setPrintPos(38,22); //memory 2  
  
u8g.print(EEPROM.read(210)); //temp_2  
  
u8g.setPrintPos(38,28);  
  
u8g.print(EEPROM.read(211)); //time_2
```

```
u8g.setPrintPos(38,38); //memory 3  
  
u8g.print(EEPROM.read(218)); //temp_2  
  
u8g.setPrintPos(38,44);  
  
u8g.print(EEPROM.read(219)); //time_2
```

```
u8g.setPrintPos(38,54); //memory 4  
  
u8g.print(EEPROM.read(226)); //temp_2  
  
u8g.setPrintPos(38,60);  
  
u8g.print(EEPROM.read(227)); //time_2
```

```
//coluna 3
```

```
u8g.setPrintPos(58,6); //memory 1  
  
u8g.print(EEPROM.read(204)); //temp_3  
  
u8g.setPrintPos(58,12);  
  
u8g.print(EEPROM.read(205)); //time_3
```

```
u8g.setPrintPos(58,22); //memory 2  
  
u8g.print(EEPROM.read(212)); //temp_3  
  
u8g.setPrintPos(58,28);  
  
u8g.print(EEPROM.read(213)); //time_3
```

```
u8g.setPrintPos(58,38); //memory 3  
  
u8g.print(EEPROM.read(220)); //temp_3  
  
u8g.setPrintPos(58,44);  
  
u8g.print(EEPROM.read(221)); //time_3
```



```
u8g.setPrintPos(58,54); //memory 4  
  
u8g.print(EEPROM.read(228)); //temp_3  
  
u8g.setPrintPos(58,60);  
  
u8g.print(EEPROM.read(229)); //time_3
```

```
//coluna 4
```

```
u8g.setPrintPos(78,6); //memory 1  
  
u8g.print(EEPROM.read(206)); //temp_4  
  
u8g.setPrintPos(78,12);  
  
u8g.print(EEPROM.read(207)); //time_4
```

```
u8g.setPrintPos(78,22); //memory 2  
  
u8g.print(EEPROM.read(214)); //temp_4  
  
u8g.setPrintPos(78,28);  
  
u8g.print(EEPROM.read(215)); //time_4
```

```
u8g.setPrintPos(78,38); //memory 3  
  
u8g.print(EEPROM.read(222)); //temp_4  
  
u8g.setPrintPos(78,44);  
  
u8g.print(EEPROM.read(223)); //time_4
```

```

u8g.setPrintPos(78,54); //memory 4

u8g.print(EEPROM.read(230)); //temp_4

u8g.setPrintPos(78,60);

u8g.print(EEPROM.read(231)); //time_4

}

//tela das memorias gravadas 4 a 8

void display4(){//id_display=2

//coluna 1

u8g.setFont(u8g_font_baby);

u8g.drawStr(5,6, "5");

u8g.drawStr(5,22, "6");

u8g.drawStr(5,38, "7");

u8g.drawStr(5,54, "8");

u8g.drawLine(0,14,128,14); //traço horizontal 1

u8g.drawLine(0,30,128,30); //traço horizontal 2

u8g.drawLine(0,46,128,46); //traço horizontal 3

u8g.drawStr(98,6,"USER1");

```

```
u8g.drawStr(98,22,"USER2");

u8g.drawStr(98,38,"USER3");

u8g.drawStr(98,54,"USER4");

u8g.setPrintPos(20,6); //memory 5

u8g.print(EEPROM.read(232)); //temp_1

u8g.setPrintPos(20,12);

u8g.print(EEPROM.read(233)); //time_1

u8g.setPrintPos(20,22); //memory 6

u8g.print(EEPROM.read(240)); //temp_1

u8g.setPrintPos(20,28);

u8g.print(EEPROM.read(241)); //time_1

u8g.setPrintPos(20,38); //memory 7

u8g.print(EEPROM.read(248)); //temp_1

u8g.setPrintPos(20,44);

u8g.print(EEPROM.read(249)); //time_1

u8g.setPrintPos(20,54); //memory 8

u8g.print(EEPROM.read(256)); //temp_1
```

```
u8g.setPrintPos(20,60);

u8g.print(EEPROM.read(257)); //time_1

//coluna 2

u8g.setPrintPos(38,6); //memory 5

u8g.print(EEPROM.read(234)); //temp_2

u8g.setPrintPos(38,12);

u8g.print(EEPROM.read(235)); //time_2

u8g.setPrintPos(38,22); //memory 6

u8g.print(EEPROM.read(242)); //temp_2

u8g.setPrintPos(38,28);

u8g.print(EEPROM.read(243)); //time_2

u8g.setPrintPos(38,38); //memory 7

u8g.print(EEPROM.read(250)); //temp_2

u8g.setPrintPos(38,44);

u8g.print(EEPROM.read(251)); //time_2

u8g.setPrintPos(38,54); //memory 8

u8g.print(EEPROM.read(258)); //temp_2
```

```
u8g.setPrintPos(38,60);

u8g.print(EEPROM.read(259)); //time_2

//coluna 3

u8g.setPrintPos(58,6); //memory 5

u8g.print(EEPROM.read(236)); //temp_3

u8g.setPrintPos(58,12);

u8g.print(EEPROM.read(237)); //time_3

u8g.setPrintPos(58,22); //memory 6

u8g.print(EEPROM.read(244)); //temp_3

u8g.setPrintPos(58,28);

u8g.print(EEPROM.read(245)); //time_3

u8g.setPrintPos(58,38); //memory 7

u8g.print(EEPROM.read(252)); //temp_3

u8g.setPrintPos(58,44);

u8g.print(EEPROM.read(253)); //time_3

u8g.setPrintPos(58,54); //memory 8

u8g.print(EEPROM.read(260)); //temp_3
```

```
u8g.setPrintPos(58,60);

u8g.print(EEPROM.read(261)); //time_3

//coluna 4

u8g.setPrintPos(78,6); //memory 5

u8g.print(EEPROM.read(238)); //temp_4

u8g.setPrintPos(78,12);

u8g.print(EEPROM.read(239)); //time_4

u8g.setPrintPos(78,22); //memory 6

u8g.print(EEPROM.read(246)); //temp_4

u8g.setPrintPos(78,28);

u8g.print(EEPROM.read(247)); //time_4

u8g.setPrintPos(78,38); //memory 7

u8g.print(EEPROM.read(254)); //temp_4

u8g.setPrintPos(78,44);

u8g.print(EEPROM.read(255)); //time_4

u8g.setPrintPos(78,54); //memory 8

u8g.print(EEPROM.read(262)); //temp_4
```

```

u8g.setPrintPos(78,60);

u8g.print(EEPROM.read(263)); //time_4

}

//tela do gráfico de temperatura

void display5(){//id_display=3

if(flag_start==1){

    u8g.drawStr(5,7,"RUN");

}else{

    u8g.drawStr(3,7,"STOP");

}

if(flag_timer<5){

u8g.setPrintPos(25,7);

u8g.print(flag_timer);

u8g.drawStr(30,7,"ST");

}else{

    u8g.drawStr(25,7,"END ");

}

u8g.drawLine(0,0,0,63); //desenha a linha vertical

u8g.drawLine(0,63,127,63); //desenha a linha horizontal

```

```
if(read_temp>99)u8g.setPrintPos(45,7);

else u8g.setPrintPos(49,7);

u8g.print(read_temp);

u8g.drawStr(61,7,"c");

if(heat_indicator==1)u8g.drawStr(75,7,"HEAT");

else if (flag_fan==1)u8g.drawStr(75,7,"FAN");

if(flag_timer>=1){

    if(min>10){

        u8g.setPrintPos(102,7);

        u8g.print(min);

    }else{

        u8g.setPrintPos(108,7);

        u8g.print(min);

    }

    u8g.drawStr(112,7, ":");

if(seg<10){

    u8g.drawStr(115,7,"0");

    u8g.setPrintPos(120,7);

    u8g.print(seg);
```



```
}else{

    u8g.drawStr(115,7," ");

    u8g.setPrintPos(116,7);

    u8g.print(seg);

}

} //tempo total

for(i=0; i<=refresh; i++){

    u8g.drawLine(Xg[i],Yg[i],Xg[i],Yg[i]);

}

} //fecha display5

//tela de calibração da resistência

void display6(){ //id_display=4

    u8g.setFont(u8g_font_5x8);

    u8g.drawStr(8,7,"RESISTENCE CALIBRATION");

    if(flag_calibration==1){

        u8g.setFont(u8g_font_courB08);

        u8g.drawStr(16,40,"WAIT CALIBRATION");

    }

}
```

```
if(min>10){  
  
    u8g.setPrintPos(101,64);  
  
    u8g.print(min);  
  
}else{  
  
    u8g.setPrintPos(105,64);  
  
    u8g.print(min);  
  
}  
  
    u8g.drawStr(111,64, ":");  
  
if(seg<10){  
  
    u8g.drawStr(115,64,"0");  
  
    u8g.setPrintPos(121,64);  
  
    u8g.print(seg);  
  
}else{  
  
    u8g.drawStr(115,64," ");  
  
    u8g.setPrintPos(116,64);  
  
    u8g.print(seg);  
  
}  
  
    u8g.setPrintPos(1,64);  
  
    u8g.print(read_temp);  
  
    u8g.drawStr(20,64,"c");  
  
}else{
```

```
u8g.setFont(u8g_font_5x7);

u8g.drawStr(1,24,"HEATING");

u8g.drawStr(44,24,":");

if(flag_cursor==1)u8g.setFont(u8g_font_courB08);

u8g.setPrintPos(51,24);

u8g.print(heating,1);

u8g.setFont(u8g_font_5x7);

u8g.drawStr(73,24,"c/seg");

u8g.setFont(u8g_font_5x7);

u8g.drawStr(1,34,"OFFSET");

u8g.drawStr(44,34,":");

if(flag_cursor==2)u8g.setFont(u8g_font_courB08);

u8g.setPrintPos(52,34);

u8g.print(temp_offset);

u8g.setFont(u8g_font_5x7);

u8g.drawStr(1,44,"CAL TEMP");

u8g.drawStr(44,44,":");

if(flag_cursor==3)u8g.setFont(u8g_font_courB08);
```

```
u8g.setPrintPos(52,44);

u8g.print(cal_temp);

u8g.setFont(u8g_font_5x7);

if(flag_cursor==4)u8g.setFont(u8g_font_courB08);

u8g.drawStr(1,60,"AUTO CALIBRATION");

u8g.setFont(u8g_font_5x7);

}

} //fecha display6
```

ANEXO VIII - CÓDIGO FONTE “GRAFICO”

```
void grafico(){

    unsigned long currentMillis = millis();

    int tempo_fixo=time_1+time_2+time_3+time_4;

    float tempo_variavel=(temp_4-temp_inicial)/heating;

    if(currentMillis - previousMillis3 >=
((tempo_fixo+tempo_variavel)/127)*1000){

        previousMillis3 = currentMillis;

        refresh++;

        Xg[i] = refresh;           // eixo do tempo

        Yg[i] = map(read_temp,0,254,63,10); // eixo da temperatura

        display5();

    }

} //fecha grafico
```

ANEXO IX - CÓDIGO FONTE “MEMORIA”

```
void write_memory_first_time(){  
  
    //grava os padrões de fábrica  
  
    EEPROM.update(8,2); //FAN MODE  
  
    EEPROM.update(9,1); //KEY BEEP  
  
    EEPROM.update(10,40); //TEMP FAN  
  
    EEPROM.update(11,10); //BEEP TIME  
  
    EEPROM.update(12,1); //OFFSET  
  
    EEPROM.update(14,0); //TEMP OFFSET NEGATIVO  
  
    EEPROM.update(13,0); //TEMP OFFSET POSITIVO  
  
    heating=0.5;  
  
    EEPROM.update(17,int(heating));  
  
    EEPROM.update(18,int((heating-int(heating))*100));  
  
    EEPROM.update(19,230); //CAL_TEMP  
  
  
    //LEADED  
  
    EEPROM.update(200, 70); //TEMP1  
  
    EEPROM.update(201, 60); //TIME1  
  
    EEPROM.update(202, 120); //TEMP2  
  
    EEPROM.update(203, 50); //TIME2
```

EEPROM.update(204, 160); //TEMP3

EEPROM.update(205, 45); //TIME3

EEPROM.update(206, 225); //TEMP4

EEPROM.update(207, 40); //TIME4

//LEAD FREE

EEPROM.update(208, 70);

EEPROM.update(209, 60);

EEPROM.update(210, 110);

EEPROM.update(211, 50);

EEPROM.update(212, 150);

EEPROM.update(213, 45);

EEPROM.update(214, 185);

EEPROM.update(215, 40);

//REFLOW

EEPROM.update(216, 60);

EEPROM.update(217, 45);

EEPROM.update(218, 110);

EEPROM.update(219, 45);

EEPROM.update(220, 150);

```
EEPROM.update(221, 40);  
  
EEPROM.update(222, 185);  
  
EEPROM.update(223, 30);
```

```
//SOLDER
```

```
EEPROM.update(224, 60);  
  
EEPROM.update(225, 50);  
  
EEPROM.update(226, 110);  
  
EEPROM.update(227, 50);  
  
EEPROM.update(228, 155);  
  
EEPROM.update(229, 40);  
  
EEPROM.update(230, 190);  
  
EEPROM.update(231, 40);
```

```
//USER1
```

```
EEPROM.update(232, 0);  
  
EEPROM.update(233, 0);  
  
EEPROM.update(234, 0);  
  
EEPROM.update(235, 0);  
  
EEPROM.update(236, 0);  
  
EEPROM.update(237, 0);
```



```
EEPROM.update(238, 0);
```

```
EEPROM.update(239, 0);
```

```
//USER2
```

```
EEPROM.update(240, 0);
```

```
EEPROM.update(241, 0);
```

```
EEPROM.update(242, 0);
```

```
EEPROM.update(243, 0);
```

```
EEPROM.update(244, 0);
```

```
EEPROM.update(245, 0);
```

```
EEPROM.update(246, 0);
```

```
EEPROM.update(247, 0);
```

```
//USER3
```

```
EEPROM.update(248, 0);
```

```
EEPROM.update(249, 0);
```

```
EEPROM.update(250, 0);
```

```
EEPROM.update(251, 0);
```

```
EEPROM.update(252, 0);
```

```
EEPROM.update(253, 0);
```

```
EEPROM.update(254, 0);
```

```
EEPROM.update(255, 0);
```

```
//USER4
```

```
EEPROM.update(256, 0);
```

```
EEPROM.update(257, 0);
```

```
EEPROM.update(258, 0);
```

```
EEPROM.update(259, 0);
```

```
EEPROM.update(260, 0);
```

```
EEPROM.update(261, 0);
```

```
EEPROM.update(262, 0);
```

```
EEPROM.update(263, 0);
```

```
//OTHER
```

```
EEPROM.update(264, 0);
```

```
EEPROM.update(265, 0);
```

```
EEPROM.update(266, 0);
```

```
EEPROM.update(267, 0);
```

```
EEPROM.update(268, 0);
```

```
EEPROM.update(269, 0);
```

```
EEPROM.update(270, 0);
```

```
EEPROM.update(271, 0);
```

```
EEPROM.update(50,1); //grava 1 na memoria 50 para indicar que foi gravado os padrões de fábrica
```

```
EEPROM.update(15,1); //grava perfil 1 como padrao
```

```
}
```

```
void read_memory(){
```

```
switch(perfil){
```

```
case (1):
```

```
temp_1 = EEPROM.read(200);
```

```
time_1 = EEPROM.read(201);
```

```
temp_2 = EEPROM.read(202);
```

```
time_2 = EEPROM.read(203);
```

```
temp_3 = EEPROM.read(204);
```

```
time_3 = EEPROM.read(205);
```

```
temp_4 = EEPROM.read(206);
```

```
time_4 = EEPROM.read(207);
```

```
break;
```

```
case(2):
```

```
temp_1 = EEPROM.read(208);
```

```
time_1 = EEPROM.read(209);
```

```
temp_2 = EEPROM.read(210);

time_2 = EEPROM.read(211);

temp_3 = EEPROM.read(212);

time_3 = EEPROM.read(213);

temp_4 = EEPROM.read(214);

time_4 = EEPROM.read(215);

break;

case(3):

temp_1 = EEPROM.read(216);

time_1 = EEPROM.read(217);

temp_2 = EEPROM.read(218);

time_2 = EEPROM.read(219);

temp_3 = EEPROM.read(220);

time_3 = EEPROM.read(221);

temp_4 = EEPROM.read(222);

time_4 = EEPROM.read(223);

break;

case(4):

temp_1 = EEPROM.read(224);

time_1 = EEPROM.read(225);

temp_2 = EEPROM.read(226);
```

```
time_2 = EEPROM.read(227);

temp_3 = EEPROM.read(228);

time_3 = EEPROM.read(229);

temp_4 = EEPROM.read(230);

time_4 = EEPROM.read(231);

break;

case(5):

temp_1 = EEPROM.read(232);

time_1 = EEPROM.read(233);

temp_2 = EEPROM.read(234);

time_2 = EEPROM.read(235);

temp_3 = EEPROM.read(236);

time_3 = EEPROM.read(237);

temp_4 = EEPROM.read(238);

time_4 = EEPROM.read(239);

break;

case(6):

temp_1 = EEPROM.read(240);

time_1 = EEPROM.read(241);

temp_2 = EEPROM.read(242);

time_2 = EEPROM.read(243);
```

```
temp_3 = EEPROM.read(244);  
  
time_3 = EEPROM.read(245);  
  
temp_4 = EEPROM.read(246);  
  
time_4 = EEPROM.read(247);  
  
break;
```

```
case(7):
```

```
temp_1 = EEPROM.read(248);  
  
time_1 = EEPROM.read(249);  
  
temp_2 = EEPROM.read(250);  
  
time_2 = EEPROM.read(251);  
  
temp_3 = EEPROM.read(252);  
  
time_3 = EEPROM.read(253);  
  
temp_4 = EEPROM.read(254);  
  
time_4 = EEPROM.read(255);  
  
break;
```

```
case(8):
```

```
temp_1 = EEPROM.read(256);  
  
time_1 = EEPROM.read(257);  
  
temp_2 = EEPROM.read(258);  
  
time_2 = EEPROM.read(259);  
  
temp_3 = EEPROM.read(260);
```

```

time_3 = EEPROM.read(261);

temp_4 = EEPROM.read(262);

time_4 = EEPROM.read(263);

break;

case(9):

temp_1 = EEPROM.read(264);

time_1 = EEPROM.read(265);

temp_2 = EEPROM.read(266);

time_2 = EEPROM.read(267);

temp_3 = EEPROM.read(268);

time_3 = EEPROM.read(269);

temp_4 = EEPROM.read(270);

time_4 = EEPROM.read(271);

break;

}

}

void write_memory(){

switch(grava_memoria){

case(1):

EEPROM.update(200, temp_1);

```

```
EEPROM.update(201, time_1);

EEPROM.update(202, temp_2);

EEPROM.update(203, time_2);

EEPROM.update(204, temp_3);

EEPROM.update(205, time_3);

EEPROM.update(206, temp_4);

EEPROM.update(207, time_4);

perfil=1;

read_memory();

break;

case(2):

EEPROM.update(208, temp_1);

EEPROM.update(209, time_1);

EEPROM.update(210, temp_2);

EEPROM.update(211, time_2);

EEPROM.update(212, temp_3);

EEPROM.update(213, time_3);

EEPROM.update(214, temp_4);

EEPROM.update(215, time_4);

perfil=2;

read_memory();
```



```
break;
```

```
case(3):
```

```
EEPROM.update(216, temp_1);
```

```
EEPROM.update(217, time_1);
```

```
EEPROM.update(218, temp_2);
```

```
EEPROM.update(219, time_2);
```

```
EEPROM.update(220, temp_3);
```

```
EEPROM.update(221, time_3);
```

```
EEPROM.update(222, temp_4);
```

```
EEPROM.update(223, time_4);
```

```
perfil=3;
```

```
read_memory();
```

```
break;
```

```
case(4):
```

```
EEPROM.update(224, temp_1);
```

```
EEPROM.update(225, time_1);
```

```
EEPROM.update(226, temp_2);
```

```
EEPROM.update(227, time_2);
```

```
EEPROM.update(228, temp_3);
```

```
EEPROM.update(229, time_3);
```

```
EEPROM.update(230, temp_4);
```

```
EEPROM.update(231, time_4);

perfil=4;

read_memory();

break;

case(5):

EEPROM.update(232, temp_1);

EEPROM.update(233, time_1);

EEPROM.update(234, temp_2);

EEPROM.update(235, time_2);

EEPROM.update(236, temp_3);

EEPROM.update(237, time_3);

EEPROM.update(238, temp_4);

EEPROM.update(239, time_4);

perfil=5;

read_memory();

break;

case(6):

EEPROM.update(240, temp_1);

EEPROM.update(241, time_1);

EEPROM.update(242, temp_2);

EEPROM.update(243, time_2);
```

```
EEPROM.update(244, temp_3);

EEPROM.update(245, time_3);

EEPROM.update(246, temp_4);

EEPROM.update(247, time_4);

perfil=6;

read_memory();

break;

case(7):

    EEPROM.update(248, temp_1);

    EEPROM.update(249, time_1);

    EEPROM.update(250, temp_2);

    EEPROM.update(251, time_2);

    EEPROM.update(252, temp_3);

    EEPROM.update(253, time_3);

    EEPROM.update(254, temp_4);

    EEPROM.update(255, time_4);

    perfil=7;

    read_memory();

    break;

case(8):

    EEPROM.update(256, temp_1);
```

```
EEPROM.update(257, time_1);

EEPROM.update(258, temp_2);

EEPROM.update(259, time_2);

EEPROM.update(260, temp_3);

EEPROM.update(261, time_3);

EEPROM.update(262, temp_4);

EEPROM.update(263, time_4);

perfil=8;

read_memory();

break;

case(9):

EEPROM.update(264, temp_1);

EEPROM.update(265, time_1);

EEPROM.update(266, temp_2);

EEPROM.update(267, time_2);

EEPROM.update(268, temp_3);

EEPROM.update(269, time_3);

EEPROM.update(270, temp_4);

EEPROM.update(271, time_4);

perfil=9;

read_memory();
```

```

        break;
    }
}

void grava_config(){

    EEPROM.update(8,fan_mode);

    EEPROM.update(9,key_beep);

    EEPROM.update(10,temp_fan);

    EEPROM.update(11,beep_time);

    if(temp_offset>=0){

        offset=1;//positivo

        EEPROM.update(14,temp_offset);

    }else {

        offset=0;//negativo

        EEPROM.update(13,temp_offset);

    }

    EEPROM.update(12,offset);

    EEPROM.update(17, int(heating));

    EEPROM.update(18, int((heating-int(heating))*100));

    EEPROM.update(19,cal_temp);

    EEPROM.update(50,set_default);

}

```

ANEXO X - CÓDIGO FONTE “SERIAL_OUT”

```
void serial_out(){  
  
    Serial.println(read_temp);  
  
}
```

ANEXO XI - CÓDIGO FONTE “START-STOP”

```
void start(){

    flag_start=1; //start

    flag_cursor=0; //desabilita os ajustes

    flag_temp=0;

    flag_timer=1;

    seg=0;

    min=0;

    refresh=0;

    i=0;

    temp_inicial=read_temp;

    temporestante1=time_1;

    temporestante2=time_2;

    temporestante3=time_3;

    temporestante4=time_4;

    set_temp=temp_1;

    id_display=3;

    flag_fan=0;

    digitalWrite(fan, LOW); //desliga fan se estiver ligado

}
```

```
void stop(){  
  
    flag_start=0; //stop  
  
    flag_timer=0; //zera as etapas  
  
    flag_temp=0; //zera controle  
  
    set_temp=0;  
  
    heat_indicator=0;  
  
    digitalWrite(heat,LOW); //desliga aquecimento  
  
}
```


ANEXO XII - CÓDIGO FONTE “TIMER”

```
void control_temp(){

    if(flag_start>0 && flag_start<5){ //só liga a resistencia se as etapas de 1 a 4
estiver rodando

        if(read_temp<set_temp+temp_offset){

            digitalWrite(heat, HIGH);

            heat_indicator=1;

        }else{

            digitalWrite(heat,LOW);

            heat_indicator=0;

        }

    }

//-----

    if(read_temp>=temp_1 && flag_temp==0){

        flag_temp=1;

    }

    if(read_temp>=temp_2 && flag_temp==1){

        flag_temp=2;

    }

    if(read_temp>=temp_3 && flag_temp==2){

        flag_temp=3;

    }

}
```

```

if(read_temp>=temp_4 && flag_temp==3){

    flag_temp=4;

}

//-----

if(temporestante1==0 && flag_temp==1){

    set_temp=temp_2;

}

if(temporestante2==0 && flag_temp==2){

    set_temp=temp_3;

}

if(temporestante3==0 && flag_temp==3){

    set_temp=temp_4;

}

if(temporestante4==0 && flag_temp==4){

    set_temp=0;

    digitalWrite(heat,LOW);

}

} //control_temp

void timer(){

run_time();

```

```

tempo = millis();

//-----

if(flag_temp==1){

    if(temporestante1>0){

        if(tempo - ult_tempo1 >= 950){

            ult_tempo1 = tempo;

            temporestante1--;

        }flag_timer=1;

        set_temp=temp_1;

    }

}

if(flag_temp==2 && temporestante1==0){

    if(temporestante2>0 && temporestante1==0){

        if(tempo - ult_tempo2 >= 950){

            ult_tempo2 = tempo;

            temporestante2--;

        }flag_timer=2;

        set_temp=temp_2;

    }

}

if(flag_temp==3 && temporestante2==0){

```

```

if(temporestante3>0 && temporestante2==0){

    if(tempo - ult_tempo3 >= 950){

        ult_tempo3 = tempo;

        temporestante3--;

    }flag_timer=3;

    set_temp=temp_3;

}

}

if(flag_temp==4 && temporestante3==0){

    if(temporestante4>0 && temporestante3==0){

        if(tempo - ult_tempo4 >= 950){

            ult_tempo4 = tempo;

            temporestante4--;

        }flag_timer=4;

        set_temp=temp_4;

    }

}

if(flag_timer==4 && temporestante4==0){

    flag_start=0;

    flag_timer=5;

    set_temp=0;

```


}

}