

UNIVERSIDADE SÃO FRANCISCO

Engenharia Elétrica

RAFAEL DE CASTRO ROCHA

**TRABALHO DE CONCLUSÃO DE CURSO:
PROJETO DE CONTROLADOR DE TEMPERATURA PARA MANTAS
TERMOELÉTRICAS**

Itatiba

2012

RAFAEL DE CASTRO ROCHA – R.A. 002200800430

**TRABALHO DE CONCLUSÃO DE CURSO:
PROJETO DE CONTROLADOR DE TEMPERATURA PARA MANTAS
TERMOELÉTRICAS**

Monografia apresentada ao curso de Engenharia Elétrica da Universidade São Francisco, como requisito parcial para obtenção do título de Bacharel em Engenharia Elétrica.

Orientador: Profº André Renato Bakalereskis

Itatiba

2012

RAFAEL DE CASTRO ROCHA

**TRABALHO DE CONCLUSÃO DE CURSO:
PROJETO DE CONTROLADOR DE TEMPERATURA PARA MANTAS
TERMOELÉTRICAS**

Monografia apresentada ao curso de Engenharia Elétrica da Universidade São Francisco, como requisito parcial para obtenção do título de Bacharel em Engenharia Elétrica.

Data da aprovação: ____ / ____ / ____

Banca examinadora:

Profº André Renato Bakalereskis (Orientador)
Universidade São Francisco

Profª Ms Débora Meyhofer Ferreira (Examinadora)
Universidade São Francisco

Profº João Alex Franciscón Vaz (Examinador)
Universidade São Francisco

Para meus pais Maria e Edevaldo,
minha irmã Daniela, e para Suellen,
minha companheira de todas as horas

AGRADECIMENTOS

Agradeço em primeiro lugar a minha família pela ajuda e incentivo de agora e sempre e a minha querida Suellen pelo carinho e compreensão dedicados nos momentos em que mais precisei. Aos meus colegas de curso Edson, Dorival e Maicon pelo conhecimento trocado, por todos os trabalhos realizados e pela amizade e companheirismo presente ao longo destes anos.

Agradeço aos professores que fizeram parte desta jornada, pela contribuição profissional e pessoal que cada um me concedeu.

Por fim agradeço a Deus por tornar os meus sonhos possíveis e por abrir caminhos na minha vida.

RESUMO

As propriedades do calor estão sendo cada vez mais exploradas por diversos tipos de tratamentos, sejam eles médicos, estéticos ou fisioterápicos. O controlador de temperatura é o dispositivo central de qualquer sistema de aquecimento, uma vez que suas características podem definir a eficácia do processo como um todo. O objetivo deste trabalho é desenvolver um controlador de temperatura microcontrolado para aquecimento de mantas termoelétricas. Serão abordadas todas as etapas do processo construtivo do mesmo como a fonte de alimentação, circuito do microcontrolador, interface com o usuário, eletrônica de potência e aquisição de dados do sensor. O projeto contará também com um programa supervisor, para que a manta termoelétrica possa ser controlada à distância através de um computador. Todos os estágios do controlador de temperatura serão desenvolvidos pelo autor, desde o circuito eletrônico, o programa do microcontrolador, o programa supervisor, até a confecção da placa de circuito impresso e montagem final. Ao término do trabalho, pretende-se obter um equipamento otimizado para a aplicação proposta, que seja estável e que apresente diversos parâmetros que possam ser configurados pelo usuário. Todos os assuntos abordados terão o devido embasamento teórico necessário para sua compreensão.

Palavras-chave: controlador. temperatura. PIC. supervisor.

ABSTRACT

The properties of heat are being increasingly exploited by a variety of treatments, whether medical, aesthetic or physical therapy. The temperature controller is the central device of any heating system, since its characteristics may determine the effectiveness of the process as a whole. The objective of this work is to develop a microcontroller temperature controller for thermoelectric heating blankets. It will address all stages of the construction process the same as the power source, the microcontroller circuit, user interface, power electronics and sensor data acquisition. The project will also include a program supervisor for the thermal blanket can be controlled remotely by a computer. All stages of the temperature controller will be developed by the author, since the electronic circuit, the microcontroller program, the supervisor program until the confection of the printed circuit board and final *assembly*. Upon completion of the work, it is intended to obtain a product optimized for the proposed application, which is stable and which presents a number of parameters that can be configured by the user. All matters discussed will have the proper theoretical background needed to understand.

Key words: controller. temperature. PIC. supervisor.

LISTA DE ILUSTRAÇÕES

Figura 1 – Diagrama de blocos do projeto	3
Figura 2 – Modelo de Von Neumann para o computador	7
Figura 3 – Acesso a memória pela arquitetura Von Neumann	8
Figura 4 – Acesso a memória pela arquitetura Harvard	8
Figura 5 – Terminais do PIC 18F4520 e suas respectivas funções	10
Figura 6 – Diagrama de blocos do PIC 18F4520	11
Figura 7 – Estágios de uma fonte de alimentação linear	14
Figura 8 – Retificador de meia onda	16
Figura 9 – Sinal retificado em meia onda	16
Figura 10 – Retificador de onda completa	17
Figura 11 – Sinal retificado em onda completa	17
Figura 12 – Retificador de onda completa em ponte	18
Figura 13 – Sinal retificado em onda completa em ponte	18
Figura 14 – Sinal filtrado para o retificador de onda completa e meia onda	20
Figura 15 – Regulador de tensão	21
Figura 16 – Circuito regulador de tensão	21
Figura 17 – Sensor LM 35	24
Figura 18 – Amplificador inversor	25
Figura 19 – Amplificador não inversor	26
Figura 20 – Circuito integrado LM324	27
Figura 21 – Terminais de um triac	28
Figura 22 – Optoacoplador MOC3020	29
Figura 23 – Conector DB9	31
Figura 24 – Pinagem do circuito integrado MAX232	32
Figura 25 – Esquema elétrico da fonte de alimentação (fonte própria)	37
Figura 26 – Esquema elétrico do regulador de tensão	38
Figura 27 – Esquema elétrico do microcontrolador	40
Figura 28 – Botões de configuração	41
Figura 29 – Tela inicial do <i>display</i> LCD	42
Figura 30 – Tela de ajuste do <i>set point</i>	43
Figura 31 – Tela de ajuste da temperatura máxima	43
Figura 32 – Tela de ajuste da temperatura mínima	43
Figura 33 – Esquema elétrico do <i>display</i> LCD	45
Figura 34 – Esquema de ligação do sensor LM35	48

Figura 35 – Amplificador de sinal do sensor LM35	50
Figura 36 – Circuito de acionamento da carga	52
Figura 37 – Pinagem do triac TIC246	53
Figura 38 – Interface do software supervisorio	54
Figura 39 – Ajuste do valor do <i>set point</i>	55
Figura 40 – Ajuste da temperatura mínima	56
Figura 41 – Ajuste da temperatura máxima	56
Figura 42 – Conversão RS-232 para TTL	58
Figura 43 – Esquema eletrônico do controlador de temperatura	59
Figura 44 – Roteamento das trilhas de cobre	60

LISTA DE TABELAS

Tabela 1 – Frequência do conversor A/D	13
Tabela 2 – Parâmetros do LED do CI MOC3020	29
Tabela 3 – Parâmetros do optotriac do CI MOC3020	29
Tabela 4 – Pinagem do conector DB9	31
Tabela 5 – Pinagem do <i>display</i> LCD	44
Tabela 6 – Conjunto de instruções do <i>display</i> LCD	46
Tabela 7 – Comandos utilizados para a inicialização do <i>display</i> LCD	47
Tabela 8 – Endereços de um <i>display</i> 16x2	47
Tabela 9 – Especificações da carga	52
Tabela 10 – Configuração da porta serial	57
Tabela 11 – Teste na condição inicial do controlador	61
Tabela 12 – Teste de elevação do SP	61
Tabela 13 – Teste de redução do SP	61

LISTA DE FÓRMULAS

Fórmula 1 – Resolução de um conversor A/D	13
Fórmula 2 – Relações de um transformador	14
Fórmula 3 – Potências de um transformador	15
Fórmula 4 – Tensão e corrente média em um retificador de meia onda	16
Fórmula 5 – Tensão e corrente média em um retificador de onda completa	18
Fórmula 6 – Tensão e corrente média em um retificador de onda completa em ponte	19
Fórmula 7 – Tensão de ripple	20
Fórmula 8 – Cálculo de V_0 para um amplificador inversor	25
Fórmula 9 – Cálculo de V_0 para um amplificador não inversor	26
Fórmula 10 – Cálculo do <i>baud rate</i>	30

LISTA DE EQUAÇÕES

Equação 1 – Cálculo de V_e	35
Equação 2 – Tensão de pico do transformador	35
Equação 3 – Tensão média após retificação	36
Equação 4 – Valor da carga do circuito	36
Equação 5 – Valor do capacitor de filtro	36
Equação 6 – Tensão no secundário de um transformador	37
Equação 7 – Cálculo do valor de saída do amplificador	50
Equação 8 – Cálculo do resistor para o optoacoplador	54

LISTA DE ABREVIATURAS E SIGLAS

CI – Circuito Integrado

CISC – Complex Instruction Set Computer

CPU – Central Processing Unit

EEPROM – Electrically Erasable Programmable Read Only Memory

GPR – General Purpose Registers

LED – Light Emitting Diode

NTC – Negative Temperature Coefficient

OTP – One Time Programmeble

PTC – Positive Temperature Coefficient

RAM – Random Access Memory

RISC – Reduced Instruction Set Computer

RMS – Root Mean Square

RTD – Resistance Temperature Detectors

SFR – Special Function Registers

TTL – Transistor Transistor Logic

ULA – Unidade Lógica e Aritmética

USART – Universal Synchronous Asynchronous Receiver Transmitter

SUMÁRIO

1 INTRODUÇÃO	1
2 REVISÃO BIBLIOGRÁFICA	3
2.1 MICROCONTROLADORES	3
2.1.1 Características construtivas dos microcontroladores	4
2.1.2 Unidade central de processamento	4
2.1.3 Núcleo CISC e RISC	5
2.1.4 Memórias	5
2.1.5 Arquitetura Von Neumann e Harvard	7
2.2 A FAMÍLIA PIC DA MICROCHIP	8
2.2.1 O PIC 18F4520	9
2.2.2 O conversor analógico/digital	12
2.3 FONTE DE ALIMENTAÇÃO	14
2.3.1 Transformador	14
2.3.2 Retificador	15
2.3.3 Filtro Capacitivo	19
2.3.4 Regulador de tensão	20
2.4 MEDIÇÃO DE TEMPERATURA	22
2.4.1 Sensores de temperatura	22
2.4.2 Sensor LM35	24
2.4.3 Amplificador operacional	24
2.4.4 Circuito integrado LM324	26
2.5 ELETRÔNICA DE POTÊNCIA	27
2.5.1 Triac	28
2.5.2 Optoisolador	28
2.6 SUPERVIÓRIO	30
2.6.1 Comunicação serial	30
2.6.2 Interface	31
3 METODOLOGIA	33
4 DESENVOLVIMENTO	35
4.1 FONTE DE ALIMENTAÇÃO	35
4.2 MICROCONTROLADOR	38
4.2.1 Geração de clock	40
4.2.2 Botões de configuração	41
4.2.3 <i>Display</i> LCD	42

4.3 LEITURA DE TEMPERATURA	47
4.3.1 Sensor LM35	48
4.3.2 Condicionamento de sinal do sensor	49
4.4 ELETRÔNICA DE POTÊNCIA	51
4.4.1 Carga do controlador	51
4.4.2 Circuito de acionamento da carga	52
4.5 SUPERVISÓRIO	54
4.5.1 Interface	57
4.6 Placa de circuito impresso	58
5 RESULTADOS	61
6 CONCLUSÃO	62
REFERÊNCIAS BIBLIOGRÁFICAS	64
ANEXOS	67

1 INTRODUÇÃO

A temperatura é uma grandeza física com grande importância em nossas vidas. Muitas de nossas atividades são influenciadas, direta ou indiretamente, pela ação da temperatura. Sua aplicação está voltada para os mais diversos fins, sejam eles domésticos, médicos, terapêuticos ou em processos industriais. Dentre estas aplicações, gostaria de enfatizar as áreas médica e fisioterápica. Atualmente existem diversas pesquisas sobre a ação exercida no corpo humano perante a aplicação de calor controlado. Segundo Zaidan (2012), “o calor promove a vasodilatação, o que melhora a nutrição e elasticidade dos tecidos (tendões e músculos), diminui a rigidez articular, relaxa a musculatura e reduz a dor”. Neste contexto, a medição e o controle da temperatura são de fundamental importância para a otimização dos tratamentos.

O presente trabalho aborda o desenvolvimento de um controlador de temperatura destinado à utilização em conjunto com mantas termoelétricas para uso médico, estético e fisioterápico. As mantas foram desenvolvidas com o objetivo de fornecer aquecimento localizado para diversas partes do corpo. Na área médica são utilizadas principalmente no pós-cirúrgico, momento em que o paciente sofre de hipotermia devido à perda de sangue durante a cirurgia. No segmento estético o calor potencializa a ação dos cosméticos, oferecendo melhores resultados. As mantas termoelétricas são também muito utilizadas em tratamentos de fisioterapia devido aos benefícios da aplicação do calor localizado descritos anteriormente. Entretanto, para que todos esses benefícios possam ser aproveitados, é imprescindível o controle correto da intensidade do calor aplicado.

O objetivo deste trabalho é desenvolver um controlador de temperatura que possa atender as exigências das aplicações acima citadas. Para tanto, ele deve possuir algumas características encontradas em diversos controladores de temperatura vendidos comercialmente. Dentre estas características, destaca-se a necessidade de o controlador permitir que alguns parâmetros relacionados ao seu funcionamento sejam facilmente configuráveis pelo usuário. É muito importante que o usuário possa ajustar o valor do *set point* da temperatura, para assim dosar corretamente a intensidade do calor que será aplicado. Outro ponto importante é a configuração de temperatura máxima e mínima, para que o usuário seja alertado caso o valor da temperatura saia da faixa segura de aplicação. Outra característica importante, principalmente para a área médica, é que o controlador de temperatura possa ser ajustado e monitorado a distância através de um programa supervisor, pois desta forma o tratamento de vários pacientes pode ser acompanhado pelo médico através de um único terminal.

A escolha do tema deste projeto se justifica pela importância do controle de temperatura nas aplicações propostas, assim como pela necessidade de se desenvolver um equipamento específico para atender às necessidades de tais aplicações. Controladores de temperatura vendidos comercialmente não são desenvolvidos para atender a propósitos específicos. Os equipamentos que possuem todos os recursos necessários para atender as aplicações citadas normalmente são caros, tornando inviável a sua utilização.

Neste trabalho serão abordadas todas as etapas do desenvolvimento do controlador. Inicialmente será feita uma revisão de toda a teoria aplicada. Na sequência todas as etapas do projeto do controlador serão detalhadas, tendo sempre como embasamento a teoria estudada. Ao final do projeto, pretende-se obter um controlador robusto, confiável e que atenda aos propósitos para os quais foi desenvolvido.

2 REVISÃO BIBLIOGRÁFICA

O desenvolvimento deste projeto foi dividido em diversos estágios. O diagrama da figura 1 apresenta estes estágios e a maneira com que eles se relacionam. O microcontrolador é o principal estágio do projeto. Ele é responsável por adquirir a informação de temperatura através do circuito de aquisição de dados, realizar o acionamento da carga, efetuar a interface com o usuário através de botões e *display* LCD e realizar a comunicação serial com um computador. A fonte de alimentação fornece 16 Vcc para o estágio de amplificação de sinal do sensor e fornece 5 Vcc para os demais circuitos eletrônicos.

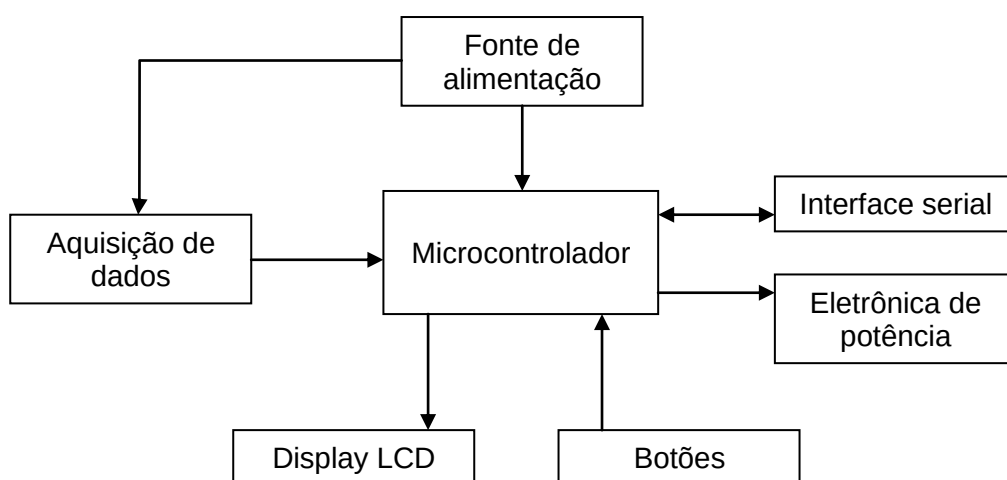


Figura 1 – Diagrama de blocos do projeto (fonte própria)

2.1 Microcontroladores

Os microcontroladores são dispositivos amplamente utilizados em circuitos eletrônicos. Por serem dispositivos programáveis, seu uso não se limita a aplicações específicas, podendo ser empregado em praticamente qualquer circuito que demande algum grau de processamento de informações. Seu uso simplifica os projetos eletrônicos, pois reduz significativamente a quantidade de componentes necessários, uma vez que toda a lógica é realizada via programação. Atualmente os microcontroladores estão integrando uma série de periféricos com o objetivo de centralizar nele a maior parte das funções dos circuitos, tornando-os ainda muito mais versáteis.

Existem diversos modelos e fabricantes de microcontroladores no mercado, dentre eles se destacam os microcontroladores da família PIC da Microchip e o microcontrolador

8051 (lançado pela Intel, mas atualmente várias empresas fabricam variantes deste modelo). Os microcontroladores da Holtek também estão ganhando seu espaço no mercado. Segundo Souza (2012), o baixo custo e a alta performance destes microcontroladores os tornam ideais para empresas que possuem alta produção de circuitos microcontrolados, onde o custo deste tenha um forte impacto no custo geral do projeto. Microcontroladores de 18 pinos da Holtek podem ser encontrados com preços na faixa de R\$2,90.

Os microcontroladores são constituídos basicamente por uma unidade central de processamento (CPU), a qual executa instruções lógicas e aritméticas; as memórias semicondutoras, que armazenam o programa a ser executado e os registradores de uso geral e específico; e dispositivos de I/O, que realizam a interface com o circuito eletrônico externo e adicionam funcionalidades ao microcontrolador.

2.1.1 Características construtivas dos microcontroladores

Segundo Zanco (2005), os microcontroladores podem ser considerados como um computador em um único chip, pois agrega em seu encapsulamento um processador, memórias e periféricos, ou seja, as principais partes que compõem um computador pessoal. A evolução dos sistemas informatizados trouxe consigo uma série de mudanças na arquitetura dos microprocessadores e memórias. Tais mudanças também se refletiram no universo dos microcontroladores, que passaram a ter diversas configurações de hardware distintas, variando principalmente entre os diversos fabricantes existentes no mercado.

2.1.2 Unidade central de processamento

A unidade central de processamento de um microcontrolador, conhecida como CPU, é um processador de pequeno porte responsável pelo controle e processamento dos dados e instruções. Segundo Zanco (2005), a CPU coordena o tráfego de dados e executa o programa. Os programas são constituídos por uma série de instruções, as quais são reconhecidas e executadas no interior da CPU. Esta utiliza a memória interna e registradores para manipulação dos dados. É constituída por três módulos: a ULA, a unidade de instrução e controle e a rede de registradores.

A ULA é responsável por executar os cálculos matemáticos, operações lógicas, incremento, decremento, setar bit, resetar bit e rotação de dados. A unidade de instrução e

controle gera os sinais necessários para a execução das instruções e a rede de registradores são alocações de memória que podem ser utilizadas para armazenamento temporário de dados e também para o uso específico de configuração do microcontrolador.

2.1.3 Núcleo CISC e RISC

Um conceito importante referente a um microprocessador são as arquiteturas CISC e RISC. Segundo Song (2012), na arquitetura CISC existe uma grande quantidade de instruções de programa que o processador está apto a reconhecer. Para realizar este trabalho, a CPU conta com uma unidade que interpreta estas instruções, convertendo instruções complexas em uma série de instruções mais simples que a ULA pode executar. O uso de um núcleo CISC se justificava na época em que o acesso às instruções do programa era prejudicado devido à lentidão das memórias. Utilizando instruções complexas se reduzia o número de acesso à memória, pois uma instrução complexa substituíva várias instruções simples. Entretanto, o uso de instruções complexas onera o processamento como um todo, pois estas demandam um tempo maior para serem decodificadas e executadas. Com os avanços na tecnologia de fabricação das memórias, o acesso às mesmas deixou de ser lento como era no passado, não justificando mais o uso de um núcleo CISC por parte dos processadores.

No núcleo RISC existe um número reduzido de instruções, dispensando assim a necessidade de um módulo de interpretação e conversão de instruções. Assim as instruções são executadas diretamente pelo hardware do processador, sem nenhum tratamento prévio da mesma, fazendo com que elas sejam executadas mais rapidamente. As instruções são carregadas da memória externa para registradores internos da CPU, aumentando assim a velocidade do processamento.

2.1.4 Memórias

Memórias semicondutoras constituem um dispositivo elementar em qualquer sistema informatizado. Em sistemas microprocessados, as memórias de dados e de programa são externas ao processador, devendo ser acessada por este através de um barramento de dados e endereço. Em sistemas microcontrolados, ambos os tipos de memória estão presen-

tes no mesmo encapsulamento do processador, o que simplifica significativamente o circuito eletrônico. Existem vários tipos de memória, que se distinguem principalmente quanto ao acesso, a volatilidade, a troca de dados e ao tipo de armazenamento.

Segundo Idoeta e Capuano (2003), as memórias podem ser de acesso sequencial ou aleatório. Nas memórias de acesso sequencial, para se acessar uma determinada localidade, todas as posições de memória anteriores ao endereço estipulado deverão ser percorridas. Desta forma, o tempo de acesso a um determinado dado está diretamente relacionado à posição em que este dado está escrito na memória. Um exemplo de memórias sequenciais são as fitas magnéticas. Já nas memórias de acesso aleatório, qualquer endereço pode ser acessado diretamente, sem que para isto tenha que se acessarem endereços anteriores. Este é o tipo de memória empregada nos microcontroladores.

As memórias também podem ser voláteis e não voláteis. Memórias voláteis são aquelas que perdem o dado armazenado quando sua alimentação é interrompida. É o caso da memória RAM, utilizada nos microcontroladores como registradores de propósito geral (GPR) e registradores com finalidade específica (SFR). As memórias não voláteis mantêm o dado armazenado mesmo quando a alimentação é interrompida. É o caso da memória flash, utilizada nos microcontroladores para o armazenamento do programa, e da memória EEPROM, utilizada para armazenamento de dados diversos.

Em relação à troca de dados com os outros componentes do sistema, as memórias podem ser de escrita/leitura ou apenas de leitura. As memórias de escrita/leitura permite que seja escrito um dado em qualquer endereço, assim como todos os dados escritos podem posteriormente serem acessados. As memórias de apenas leitura permite que os dados possam ser gravados uma única vez, podendo ser posteriormente lidos, mas não modificados. Todas as memórias existentes na maioria dos microcontrolador são do tipo escrita/leitura. Apenas alguns microcontroladores de baixo custo utilizam memória OTP para armazenamento do programa, podendo esta ser programada uma única vez. Os microcontroladores da família PIC com esta característica possuem a letra “C” em seu nome, por exemplo “16C75”. Os microcontroladores PIC que possuem memória de programa do tipo Flash são caracterizados pela letra F, por exemplo, “16F628”.

As memórias também variam quanto ao tipo de armazenamento, podendo ser estáticas ou dinâmicas. Nas memórias estáticas, o dado, uma vez armazenado, permanece lá até a alimentação ser suspensa. Já as memórias dinâmicas, o dado precisa ser periodicamente reinserido para que não seja perdido. Esta ação de atualização do dado gravado é conhecida como “*refresh*”. Nos microcontroladores as memórias são do tipo estática, que são mais fáceis de se utilizar que as memórias dinâmicas, pois não necessitam do *refresh*. Memória do tipo dinâmica se justifica apenas para capacidades de armazenamento elevadas, pois

utilizam menos componentes que as memórias estáticas, resultando assim em um custo mais baixo.

2.1.5 Arquitetura Von Neumann e Harvard

A organização interna de um microcomputador ou microcontrolador se divide basicamente em dois tipos de arquiteturas distintas, a arquitetura Von Neumann e Harvard. Segundo Borges e Silva (2012), John Von Neumann foi o responsável por introduzir o conceito de memória, um dispositivo de armazenamento temporário, onde o programa e os dados seriam carregados a partir de um dispositivo de entrada para posteriormente serem processados pela ULA. A unidade de controle é responsável pela coordenação de todo este tráfego de dados.

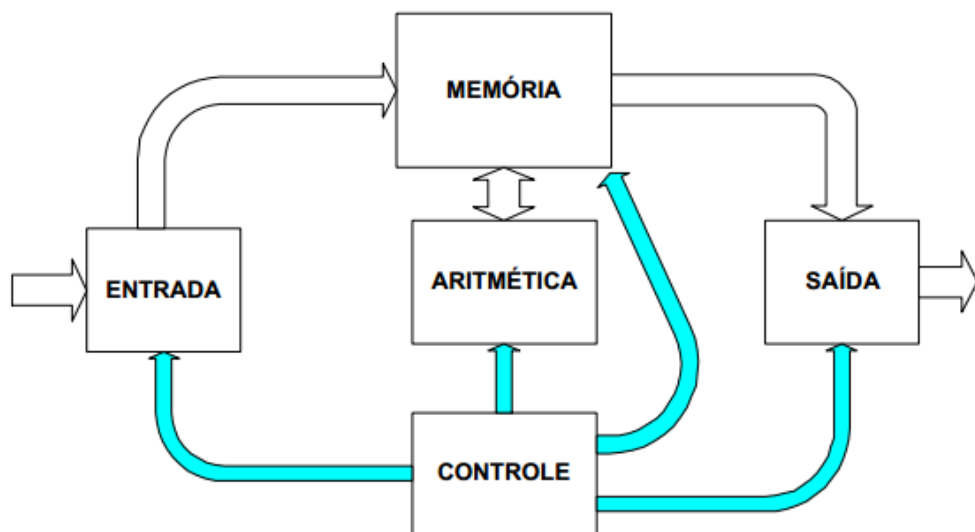


Figura 2 – Modelo de Von Neumann para o computador [5]

Uma das características do modelo desenvolvido por Von Neumann é que os dados e o programa a ser executado são armazenados em um mesmo dispositivo de memória, contando com apenas um barramento de dados e endereços. Desta forma, não é possível para a CPU acessar um dado e uma instrução simultaneamente.

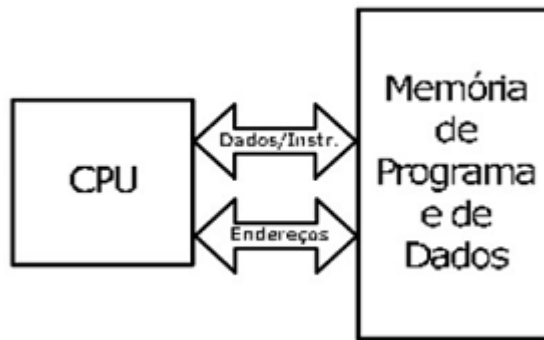


Figura 3 – Acesso a memória pela arquitetura Von Neumann [6]

A arquitetura Harvard apresenta uma evolução em relação a arquitetura Von Neumann. Segundo Zanco (2005), na arquitetura Harvard temos memórias distintas de dados e programa, cada uma contendo o seu próprio barramento de dados e endereços. Desta forma, um dado e uma instrução podem ser acessados simultaneamente, o que aumenta a capacidade de processamento.

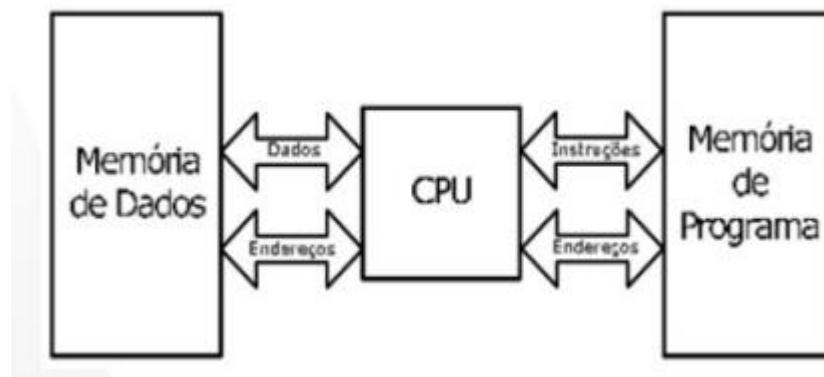


Figura 4 – Acesso a memória pela arquitetura Harvard [6]

Os microcontroladores PIC da Microchip são fabricados utilizando a arquitetura Harvard.

2.2 A família PIC da Microchip

Dentre os diversos microcontroladores disponíveis foi escolhida a família PIC da Microchip para a realização deste projeto. Sua escolha se deve a sua versatilidade. A Micro-

chip elaborou chips com as mais diversas configurações, oferecendo uma ampla gama de opções. Desta forma podemos escolher qual o microcontrolador se adapta melhor ao nosso projeto.

Existem PIC's de 8 bits, 16 bits e 32 bits. Destes, os chips de 8 bits são os mais baratos e fáceis de encontrar no mercado, pois são suficientes para a grande maioria das aplicações. Registradores de 16 bits e de 32 bits facilitam no tratamento de números muito grandes ou com ponto flutuante, mas, mesmo nestes casos, pode-se utilizar registradores de 8 bits empregando determinadas técnicas de programação, que simula registradores maiores concatenando registradores de 8 bits.

A linha de PIC's de 8 bits é dividida nas famílias 10F, 12F, 16F e 18F. A família 10F é a linha de baixo custo. É constituída de chips pequenos (6 terminais) com o mínimo de periféricos integrados. As famílias 12F (chips com 8 terminais) e 16F (chips de diversos tamanhos, de 14 a 64 terminais) possuem o núcleo de processamento mais desenvolvido, contando com um conjunto de instruções maior que a família 10F. Eles possuem também uma grande quantidade de periféricos integrados, como conversor analógico/digital, módulo PWM, referência de tensão, entre outros. Da mesma forma, a família 18F é uma evolução da família 16F. Emprega melhorias em sua unidade de processamento, contando com um set maior de instruções e apresentando mais periféricos integrados, como interface USB e ethernet.

2.2.1 O PIC 18F4520

O 18F4520 é um dos microcontroladores mais avançados da família PIC. Segundo Sousa, Souza e Lavinia (2010), este microcontrolador possui um elevado número de periféricos integrados. Suas principais características são:

- Microcontrolador de 40 pinos, o que possibilita a montagem de um hardware complexo e capaz de interagir com diversos recursos e funções ao mesmo tempo;
- 75 instruções (83 instruções contando o set de instruções estendido);
- 36 portas configuráveis como entrada ou saída;
- 20 interrupções disponíveis;
- Memória de programação E2PROM FLASH;
- Memória de programa com 32 kbytes, com capacidade de escrita e leitura pelo próprio código interno;

- Memória E2PROM (não volátil) interna com 256 bytes;
- Memória RAM com 1536 bytes;
- Quatro timers (1x8 bits e 3x16 bits);
- Comunicações seriais: SPI, I²C e USART;
- Conversores analógicos de 10 bits (13x) e comparadores analógicos (2x);
- Dois módulos CCP: Capture, Compare e PWM;
- Programação in-circuit (alta e baixa tensão);
- Power-on Reset (POR) interno;
- Brown-out Reset (BOR) interno;
- Diversos recursos adicionais.

O PIC 18F4520 possui 40 pinos, contando com 36 terminais de I/O, que podem ser configurados como entrada ou saída de dados. A exceção é o terminal 1 (RE3), o qual pode apenas ser configurado como entrada. Todos os terminais possuem diversas funções multiplexadas, correspondentes aos periféricos.

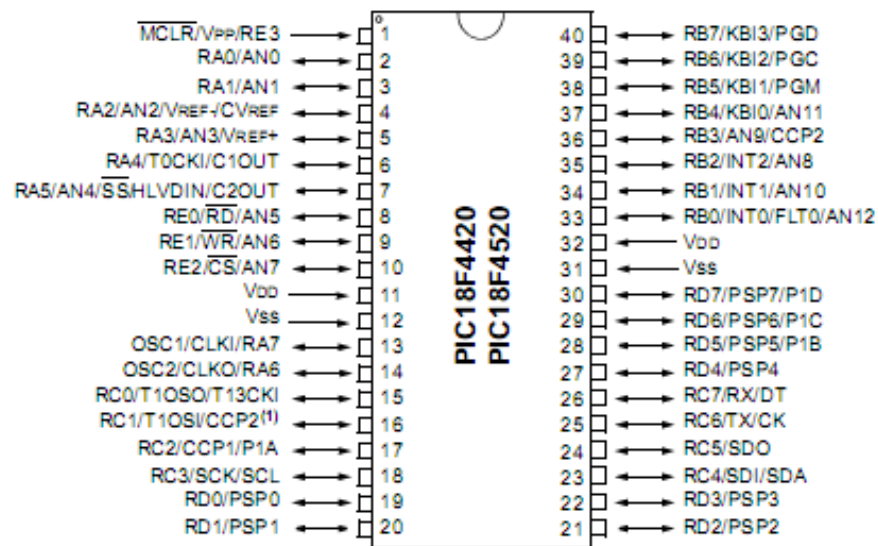


Figura 5 – Terminais do PIC 18F4520 e suas respectivas funções [13]

Através de seu diagrama de blocos podemos ter uma melhor visão de suas características.

já foi dito anteriormente, a Microchip possui PICs de 8, 16 e 32 bits. A família 18F possui um núcleo CISC, mas conta com um conjunto de instruções estendido. Duas destas instruções são a MULLW e MULWF, que realizam, respectivamente, a multiplicação de um literal com o registro W e realiza a multiplicação de W com outro registro qualquer. O resultado desta multiplicação é armazenado nos registros PRODH e PRODL que estão localizados próximos a ULA, no diagrama. Na região central do diagrama podemos ver também alguns parâmetros de configuração, os quais são ajustados através dos registros CONFIG.

Na parte superior esquerda está localizada a memória de programa e alguns registros de acesso a ela. Este modelo de PIC possui um recurso denominado *Table Pointer*. Segundo Sousa, Souza e Lavinia (2010), este recurso permite que a memória de programa seja acessada via programação para operações de leitura e escrita. Na região superior direita do diagrama está localizada a memória de dados e seus registros associados. Podemos observar que a memória de dados e de programa são separadas, cada uma contando com seu próprio barramento de dado e endereço, o que caracteriza a arquitetura Harvard.

2.2.2 O conversor analógico/digital

O PIC 18F4520 conta com um módulo dedicado para realizar conversão de valores analógicos em digitais. Segundo Sousa, Souza e Lavinia (2010), as características deste conversor são:

- Conversor interno de 10 bits, dando um total de 1.024 pontos;
- Até 13 canais de conversão, com diversas configurações entre analógicas e digitais;
- Quatro tipos de referência: V_{oo} (interna), V_{ss} (interna), VREF+ (externa) e VREF- (externa);
- Frequência de conversão baseada no clock da máquina ou em RC dedicado, possibilitando o funcionamento em modo SLEEP;
- Três ajustes de frequência (divisores) para o clock de máquina;
- Dois tipos de justificação para o resultado da conversão: direita e esquerda;
- Ajuste do tempo de carga do capacitor interno;
- Uma interrupção para término da conversão.

A Microchip fabrica PICs com conversores de 8, 10 e 12 bits. Este é um parâmetro muito importante na escolha do modelo de PIC que será utilizado em um projeto, pois está diretamente relacionada com o valor de resolução da conversão. A resolução pode ser obtida através da fórmula 1.

$$\text{Resolução} = V_{\text{ref}} / 2^n$$

Fórmula 1 – Resolução de um conversor A/D [7]

Onde n é o valor de bits do conversor e V_{ref} é a faixa de valores que o sinal analógico irá variar. O PIC 18F4520 oferece duas possibilidades para V_{ref} . A referência pode ser obtida pela tensão de alimentação, ou seja, V_{ss} e V_{dd} ; ou pode ser obtida por uma tensão aplicada nos terminais RA2 e RA3. Considerando a tensão de alimentação como referência e se tratando de um conversor de 10 bits, o conversor A/D do PIC 18F4520 nos dá uma resolução de $5 / 2^{10}$, que equivale a 4,88 mV.

Podemos utilizar até 13 canais de conversão, ou seja, podemos ter até 13 fontes de sinal analógico. No momento de iniciar a conversão, escolhemos o canal que iremos utilizar através de um registro de controle do conversor.

Um parâmetro importante que deve ser observado quando trabalhamos com o conversor A/D é o período de conversão, ou T_{AD} , que corresponde ao tempo de conversão de 1 bit. Como o conversor é de 10 bits, o tempo total necessário para conversão é de $10 T_{\text{AD}} + 2 T_{\text{AD}}$, ou seja, $12 T_{\text{AD}}$. Os $2 T_{\text{AD}}$ adicionais são necessários para o conversor religar seu capacitor interno e se preparar para uma nova conversão. O valor de T_{AD} equivale ao inverso da frequência de trabalho do conversor. Esta frequência de trabalho é determinada através de três bits de configuração, conforme a tabela 1.

ADCS2	ADCS1	ADCS0	Frequência
0	0	0	$F_{\text{osc}} / 2$
0	0	1	$F_{\text{osc}} / 8$
0	1	0	$F_{\text{osc}} / 32$
0	1	1	RC Interno
1	0	0	$F_{\text{osc}} / 4$
1	0	1	$F_{\text{osc}} / 16$
1	1	0	$F_{\text{osc}} / 64$
1	1	1	RC Interno

Tabela 1 – Frequência do conversor A/D [7]

Para o PIC 18F4520, o valor do T_{AD} deve ser maior que $0,7 \mu\text{s}$. É recomendável que seu valor máximo não ultrapasse $25 \mu\text{s}$. Desta forma, o valor da frequência de trabalho do conversor deve ser escolhido respeitando estes limites.

2.3 Fonte de alimentação

A função básica de uma fonte de alimentação é converter o sinal senoidal da rede elétrica para o sinal contínuo com um nível de tensão ideal para alimentar os circuitos eletrônicos. Existem diversos modelos de fontes que realizam este propósito. Os principais tipos são as fontes de alimentação linear e chaveada. Neste projeto foi utilizada uma fonte de alimentação linear. Seus estágios podem ser observados na figura 7.

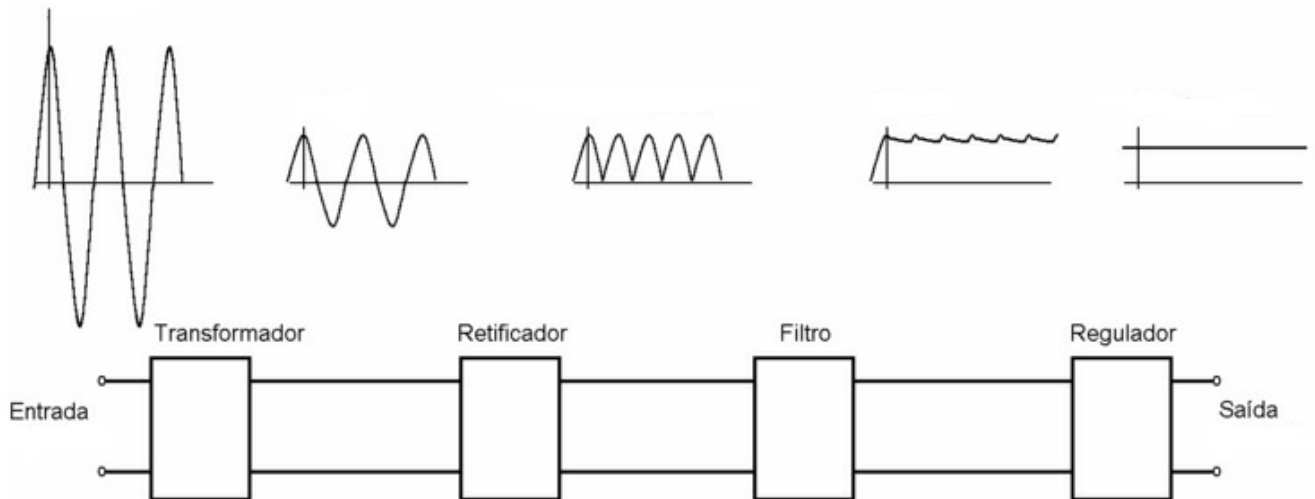


Figura 7 – Estágios de uma fonte de alimentação linear [8]

2.3.1 Transformador

A função do transformador é diminuir a amplitude do sinal senoidal da rede elétrica para um nível apropriado para alimentar os circuitos eletrônicos. Para esta finalidade utiliza-se um transformador abaixador de tensão. Segundo Marques, Cruz e Júnior (2002), o transformador é constituído por um enrolamento primário, conectado à rede elétrica; e um enrolamento secundário, no qual é ligada a carga a ser alimentada. Há uma relação de proporcionalidade entre o número de espiras, a corrente e a tensão nos enrolamentos primário e secundário. Esta relação é mostrada na fórmula 2.

$$\frac{V_1}{V_2} = \frac{N_1}{N_2} \quad \frac{I_2}{I_1} = \frac{N_1}{N_2}$$

Fórmula 2 – Relações de um transformador [9]

Sendo V_1 , N_1 e I_1 , respectivamente, a tensão, o número de espiras e a corrente no enrolamento primário. Da mesma forma, V_2 , N_2 e I_2 são, respectivamente, a tensão, o número de espiras e a corrente no enrolamento secundário. Substituindo N_1/N_2 da segunda equação na primeira, temos que:

$$V_1 \cdot I_1 = V_2 \cdot I_2$$

Fórmula 3 – Potências de um transformador [9]

Ou seja, a potência absorvida da rede pelo enrolamento primário, em um transformador ideal, é igual a potência entregue a carga pelo enrolamento secundário, existindo assim a máxima transferência de potência. Em um transformador real não é isso que acontece, pois existem perdas por efeito Joule devido à resistência do fio dos enrolamentos; há correntes parasitas induzidas no núcleo (correntes de *Foucault*) e também ocorre dispersão das linhas de campo magnético; fazendo com que a potência no secundário seja menor que a do primário.

Um tipo de transformador muito utilizado em fontes lineares é o transformador com derivação central no secundário (*Center Tap*). Ele funciona como se tivesse dois enrolamentos secundários, com as terminações centrais dos enrolamentos interligadas. Desta forma, as relações de tensão, corrente e número de espiras se mantêm da mesma forma que um transformador comum, porém suas tensões de saída são defasadas em 180° . Esta característica o torna ideal para ser utilizado com um circuito retificador em onda completa.

2.3.2 Retificador

A função do circuito retificador é converter a tensão alternada, composta por uma onda senoidal, em um sinal contínuo, no qual a corrente flua em apenas um sentido. Para realizar esta tarefa utiliza-se o diodo, pois este semicondutor possui a característica de deixar passar corrente em apenas um sentido, bloqueando quando a polaridade é invertida. Segundo Marques, Cruz e Júnior (2002), as fontes lineares utilizam basicamente três tipos de circuitos retificadores, o retificador de meia onda, o retificador de onda completa e o retificador de onda completa em ponte.

O retificador de meia onda é o mais simples, utilizando apenas um diodo. O sinal gerado por este tipo de retificador possui uma variação (ripple) muito elevada, pois o semiciclo negativo do sinal senoidal é cortado e o tempo entre um pulso e outro é longo. Em um sinal

senoidal o valor médio de tensão é nulo, pois o semiciclo negativo cancela o positivo. Como o retificador bloqueia o semiciclo negativo, o valor médio de tensão que é entregue a carga deixa de ser nulo. Em um retificador de meia onda, o valor da tensão e corrente média pode ser calculado pelas expressões:

$$V_m = \frac{V_{2P} - V_Y}{\pi}$$

$$I_m = \frac{V_m}{R_L}$$

Fórmula 4 – Tensão e corrente média em um retificador de meia onda [9]

Sendo V_{2P} a tensão pico no secundário do transformador, V_Y a barreira de potencial do diodo e R_L a resistência da carga. Neste caso, existem dois limites do diodo que devem ser respeitados. A corrente direta máxima que o diodo suporta deverá ser maior que a corrente média do circuito (I_m) e a tensão reversa máxima do diodo deverá ser maior que a tensão de pico (V_{2P}). O circuito retificador de meia onda e o sinal sobre a carga gerado por este podem ser observados nas figuras 8 e 9.

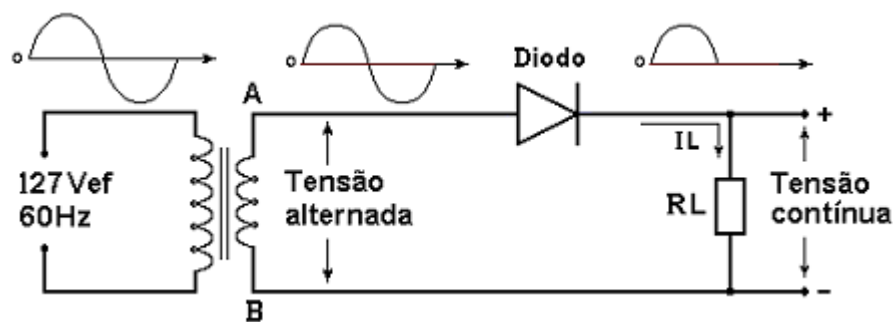


Figura 8 – Retificador de meia onda [10]

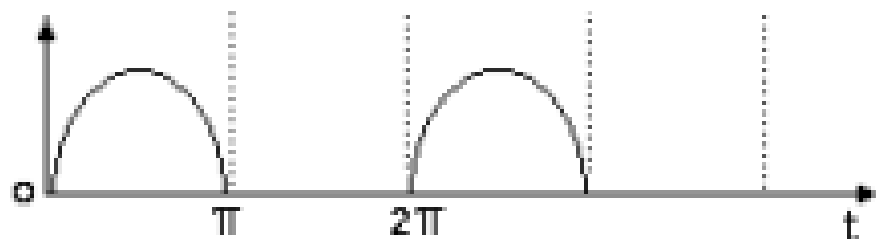


Figura 9 – Sinal retificado em meia onda [10]

Já o retificador de onda completa emprega dois diodos, um operando no semiciclo positivo e o outro no semiciclo negativo. Desta forma, segundo Marques, Cruz e Júnior (2002), este tipo de retificador “faz com que tanto o semiciclo positivo quanto o negativo caiam sobre a carga sempre com a mesma polaridade”. Sua utilização é propiciada com o uso de um transformador com derivação central. O circuito retificador de onda completa e o sinal sobre a carga gerado por este podem ser observados nas figuras 10 e 11.

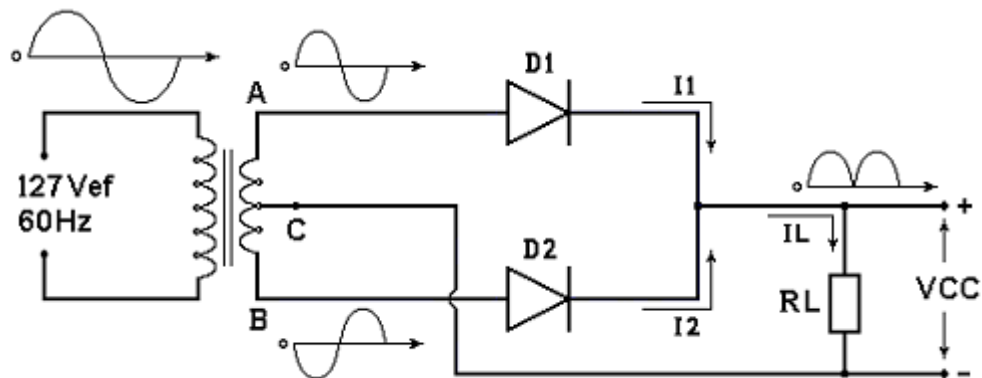


Figura 10 – Retificador de onda completa [10]

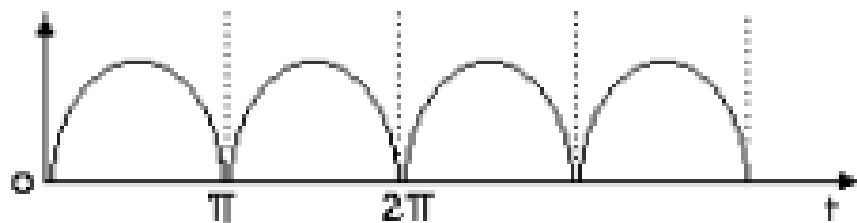


Figura 11 – Sinal retificado em onda completa [10]

Como pode ser observada na figura 11, a frequência da tensão sobre a carga no retificador de onda completa é o dobro da frequência da tensão gerada pelo retificador de meia onda, o que dobraria também o valor da tensão média. Entretanto, utilizando-se um transformador com derivação central, a tensão de pico no secundário é a metade do valor da tensão de pico que teríamos ao se utilizar um transformador comum. Desta forma, o aumento da frequência do sinal gerado pelo retificador em onda completa não corresponde a um aumento na tensão média na carga.

O valor da tensão e corrente médias de um retificador de onda completa podem ser calculadas pela fórmula 5.

$$V_m = \frac{V_{2P} - 2.V_Y}{\pi}$$

$$I_m = \frac{V_m}{R_L}$$

Fórmula 5 – Tensão e corrente média em um retificador de onda completa [9]

Como podemos observar na fórmula 5, para este tipo de retificador consideramos a barreira de potencial dos dois diodos. O retificador de onda completa apresenta duas vantagens em relação ao retificador de meia onda. Em primeiro lugar, com o dobro da frequência do sinal o tempo entre um pico e outro cai pela metade, o que facilita a redução do ripple no estágio de filtragem. Em segundo lugar, como “cada diodo conduz corrente em apenas um semiciclo, a corrente que eles devem suportar corresponde à metade da corrente média na carga”, segundo Marques, Cruz e Júnior (2002). Entretanto, a tensão reversa máxima que cada diodo deve suportar continua sendo a tensão de pico do transformador, igual ao retificador de meia onda.

O retificador de onda completa em ponte é similar ao retificador de onda completa, porém temos dois diodos conduzindo o semiciclo positivo e outros dois diodos conduzindo o semiciclo negativo. A vantagem deste tipo de retificador é que a tensão média na carga é o dobro comparado com os outros retificadores, pois trabalha em ambos os semiciclos. Neste caso não é necessário utilizar um transformador com tap central. O circuito retificador de onda completa em ponte e o sinal retificado gerado por ela podem ser observados nas figuras 12 e 13.

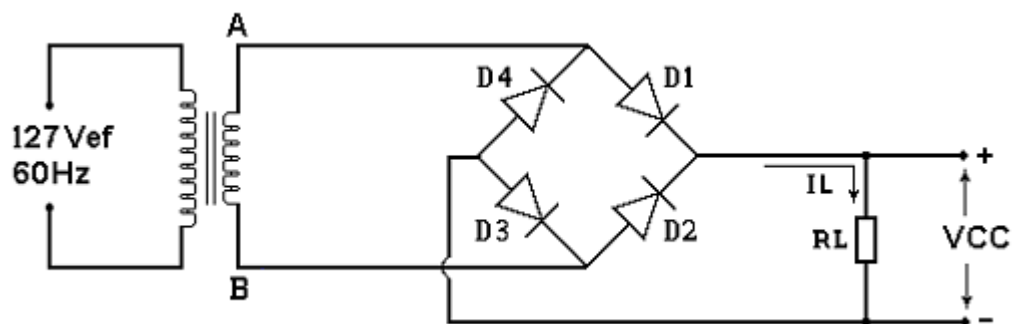


Figura 12 – Retificador de onda completa em ponte [10]

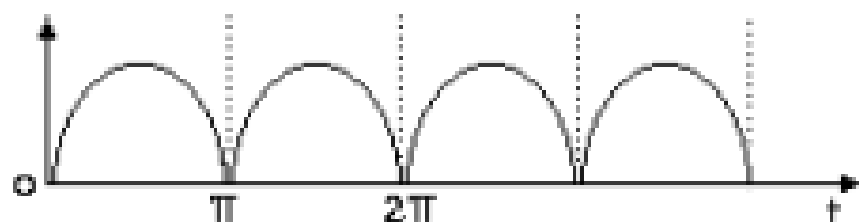


Figura 13 – Sinal retificado em onda completa em ponte [10]

A tensão e corrente médias para o retificador de onda completa em ponte podem ser calculadas pela fórmula 6.

$$V_m = \frac{2 \cdot (V_{2P} - 2 \cdot V_Y)}{\pi} \quad I_m = \frac{V_m}{R_L}$$

Fórmula 6 – Tensão e corrente média em um retificador de onda completa em ponte [9]

Em relação às especificações dos diodos, este modelo de retificador é semelhante ao retificador de onda completa, devendo cada diodo suportar uma corrente direta máxima de no mínimo metade da corrente média na carga e uma tensão reversa máxima maior que a tensão de pico do secundário do transformador.

2.3.3 Filtro Capacitivo

Com o objetivo de eliminar a oscilação da tensão proveniente do sinal retificado, utiliza-se um filtro capacitivo, o qual consiste em um capacitor conectado em paralelo com a carga. Segundo Marques, Cruz e Júnior (2002), no primeiro semiciclo do sinal retificado, os diodos carregam o capacitor até a tensão de pico. Quando a tensão retificada começa a cair, os diodos ficam reversamente polarizados e o capacitor se descarrega através da carga. Esta descarga do capacitor é lenta, não permitindo que a tensão sobre a carga caia para um valor muito baixo. No próximo semiciclo, quando a tensão retificada torna-se novamente maior que a tensão do capacitor, este volta a se carregar, repetindo o ciclo.

Este processo gera uma ondulação no sinal denominada ripple. O sinal retificado após ser submetido ao filtro capacitivo pode ser observado na figura 14.

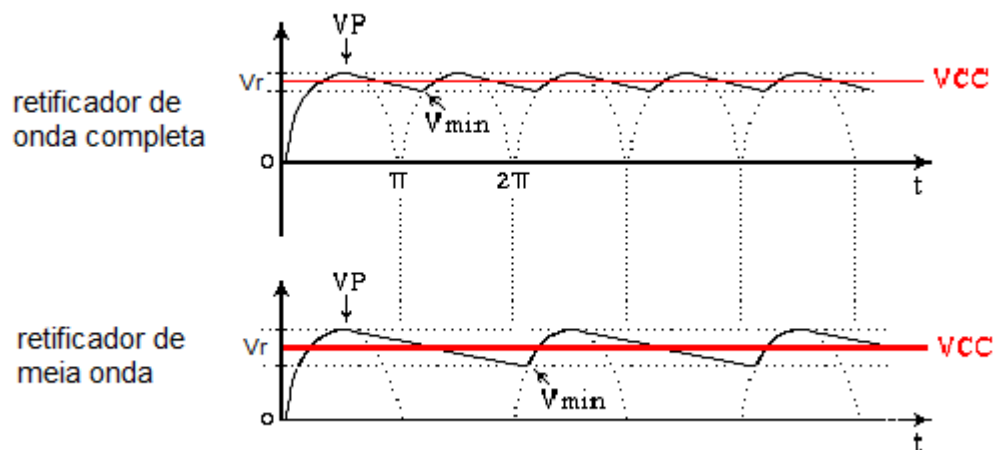


Figura 14 – Sinal filtrado para o retificador de onda completa e meia onda [10]

Como pode ser observado na figura 14, o filtro capacitivo mantém o sinal retificado próximo ao sinal perfeitamente contínuo (V_{CC}). A tensão de ripple (V_r) corresponde a variação entre a tensão de pico (V_P) e a tensão mínima (V_{min}). A figura 14 nos mostra também que o aumento da frequência do sinal resultante da retificação em onda completa diminui o valor da tensão de ripple.

Desta forma, a tensão de ripple é inversamente proporcional à frequência do sinal e também ao tempo de descarga do capacitor, o qual depende do valor da capacitância e da resistência da carga. Assim, podemos calcular a tensão de ripple através da fórmula 7.

$$V_r = \frac{V_{mf}}{f \cdot R(\text{carga}) \cdot C}$$

Fórmula 7 – Tensão de ripple [9]

Onde V_{mf} é a tensão média na carga após a filtragem, f é a frequência do sinal, $R(\text{carga})$ é a resistência de carga e C o valor da capacitância.

2.3.4 Regulador de tensão

Após todas as etapas da fonte de alimentação descritas anteriormente, temos agora uma tensão contínua e com uma baixa variação, porém ainda muito suscetível as oscilações da rede elétrica. Para estabilizar a tensão e deixá-la com o valor que o circuito necessita utiliza-se os reguladores de tensão. Existem diversos circuitos discretos compostos de tran-

sistor, diodo zener e amplificador operacional que realizam a regulação de tensão, porém torna-se mais prático utilizar circuitos integrados desenvolvidos para esta finalidade, os quais incorporam, em um único encapsulamento, os estágios de fonte de referência, amplificador comparador, dispositivo de controle e proteção contra sobrecarga (Boylestad e Nashelsky, 2004).

Sua função básica é receber uma tensão contínua não regulada em seu terminal V_i e fornecer uma tensão regulada e estabilizada em seu terminal V_o . A série de reguladores 78xx (sendo “xx” o valor da tensão V_o) é amplamente utilizada para esta finalidade e fornece tensões reguladas fixas de 5V a 24V.

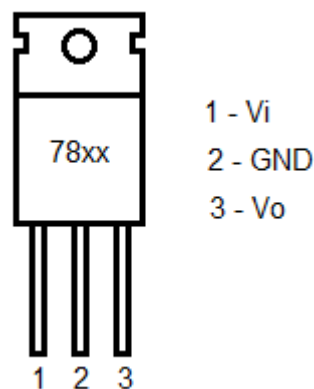


Figura 15 – Regulador de tensão [12]

O circuito recomendado no datasheet do fabricante é mostrado na figura 16.

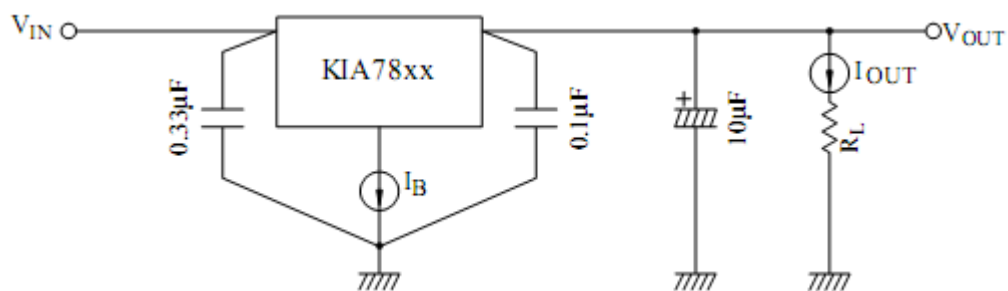


Figura 16 – Circuito regulador de tensão [12]

O capacitor de $0.33\mu\text{F}$ filtra o sinal de entrada, o capacitor de $0.1\mu\text{F}$ filtra o sinal de saída (principalmente para ruídos de alta frequência) e o capacitor de $10\mu\text{F}$ supre os picos de corrente do circuito.

Os reguladores possuem algumas especificações que devem ser observadas. Será levado em consideração o regulador 7805, o qual foi utilizado no projeto.

- Tensão de saída: as especificações do regulador 7805 nos mostram que a tensão de saída é, normalmente, +5V, mas tem como limite inferior 4.8V e limite superior de 5.2V;
- Corrente de saída em curto-circuito: a corrente de saída é limitada em 0.35A se a saída entrar em curto;
- Corrente de pico na saída: a corrente máxima nominal para esta série de reguladores é de 1.5A, entretanto a corrente de pico pode ser de até 2.2A por um breve período de tempo.
- Tensão de desoperação: é a tensão mínima que deve existir entre os terminais de entrada e saída para que o CI opere como regulador. Seu valor para o regulador 7805 é 2.3V. Dessa forma, a tensão mínima de entrada deverá ser 7.3V ($V_o + 2.3V$).
- Tensão máxima de entrada: a máxima tensão que poderá ser aplicada ao terminal V_i é 35V.
- Potência máxima dissipada: 1.9W (sem dissipador de calor) e até 30W (considerando um dissipador de calor ideal).

2.4 Medição de temperatura

2.4.1 Sensores de temperatura

Existem diversos tipos de sensores que convertem valores de temperatura em grandezas que podem ser visualizadas ou utilizadas por um sistema de controle. Segundo Balbinot e Brusamarello (2006), os principais tipos de sensores de temperatura são os termômetros que utilizam efeito mecânico, os termômetros de resistência elétrica, os termopares e os sensores semicondutores.

Os termômetros que utilizam efeito mecânico tem como princípio de funcionamento a expansão de líquidos em um bulbo de vidro. Este tipo de termômetro baseia-se no coeficiente de dilatação térmica do líquido. O aumento de temperatura provoca uma expansão do líquido, o qual passa por um capilar no interior de um bulbo de vidro graduado. O valor de temperatura pode ser medido observando-se o nível do líquido no interior do bulbo. Os sensores bimetalicos também se enquadram neste grupo, pois seu princípio de operação baseia-se na deformação causada pela temperatura quando dois metais com coeficientes de

dilatação diferentes são unidos. Estes sensores são conhecidos como termostatos e sua aplicação limita-se a sistemas de controle liga/desliga.

Os termômetros de resistência elétrica são baseados na variação da resistência elétrica provocada pela temperatura. Podem ser termômetros do tipo metálico (RTDs) ou semicondutores, conhecidos como termistores. Os termômetros metálicos são constituídos por um fio ou enrolamento de metal de alto grau de pureza, geralmente cobre platina ou níquel. A platina é o metal mais recomendado, pois é um metal quimicamente inerte, conservando assim suas características a altas temperaturas. Os RTDs de platina são descritos como PT-(resistência à temperatura de 0°C). Os mais comuns são o PT-25.5, PT-100, PT-120, PT-130 e o PT-500, sendo o PT-100 o mais conhecido e utilizado industrialmente. Os termistores também tem sua resistência alterada como efeito direto da temperatura, porém são constituídos de semicondutores cerâmicos e possuem um coeficiente de variação maior que os RTDs. Podem ser de dois tipos, PTC e NTC. Os termistores PTC possuem um coeficiente de temperatura positivo, ou seja, seu valor de resistência aumenta com a elevação da temperatura. Por sua vez, os termistores do tipo NTC possuem um coeficiente de temperatura negativo, ou seja, seu valor de resistência diminui com o aumento da temperatura.

Os termopares são sensores do tipo ativo, pois geram um sinal elétrico a partir da grandeza que está sendo medida. Este é o caso também dos sensores piezoelétricos, piroelétricos, fotovoltaicos e eletroquímicos. Os termopares são constituídos de dois metais diferentes unidos em uma de suas extremidades. Quando dois metais diferentes são unidos, há um fluxo de corrente elétrica entre ambos que pode aumentar ou diminuir com a variação da temperatura. Desta forma, cria-se uma diferença de potencial entre seus terminais. Os termopares podem ser do tipo B, C, E, J, K, N, R, S e T, podendo variar seu range, fundo de escala e precisão de acordo com o seu tipo.

Os dispositivos semicondutores, como os diodos e transistores, são sensíveis à temperatura e, assim, podem ser utilizados como sensores. Existem comercialmente sensores de estado sólido que utilizam as características dos semicondutores para realizar medições de temperatura. Normalmente estes sensores são alimentados por uma tensão contínua e disponibilizam em sua saída um nível de tensão ou corrente proporcional à temperatura medida. Sua grande vantagem é a resposta linear que apresentam, facilitando muito o tratamento da informação proveniente do sensor.

2.4.2 Sensor LM35

O sensor LM35 se enquadra no grupo dos sensores semicondutores. Segundo o Datasheet LM35 (2012), este circuito integrado é um sensor de temperatura que apresenta sua tensão de saída linearmente proporcional à temperatura medida em graus Celsius, apresentando uma escala de $10\text{mV}/^{\circ}\text{C}$. Sua precisão é de $0,5^{\circ}\text{C}$ e a faixa de temperatura em que ele pode operar varia de -55°C a $+150^{\circ}\text{C}$. O LM35 deverá ser alimentado por uma tensão contínua na faixa de 4 a 30 Vcc. A pinagem do sensor LM35 para o encapsulamento TO-92 pode ser observada na figura 17.

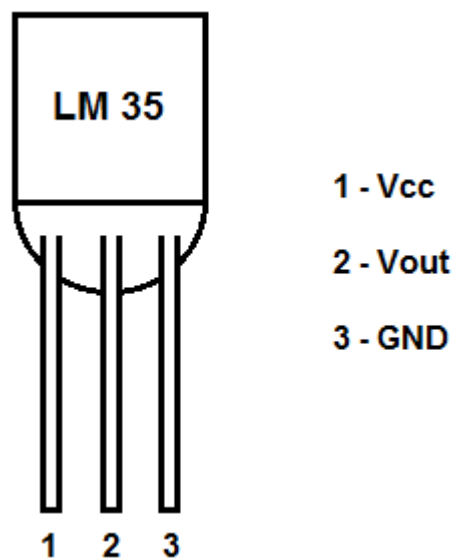


Figura 17 – Sensor LM 35 [19]

2.4.3 Amplificador operacional

Segundo Boylestad e Nashelsky (2004), “um amplificador operacional, ou ampop, é um amplificador diferencial de ganho muito alto com impedância de entrada muito elevada e impedância de saída baixa”. Ele é composto por um terminal inversor (-), um terminal não inversor (+) e o terminal de saída (V_o). Basicamente este componente amplifica a diferença de potencial existente entre os terminais inversor e não inversor por um ganho determinado por um circuito externo. Se o sinal for aplicado em sua entrada não inversora o sinal de saída terá a mesma polaridade do sinal de entrada. Caso o sinal seja aplicado em sua entrada inversora, o sinal de saída terá uma polaridade inversa em relação ao sinal de entrada.

Existem diversas configurações e circuito que utilizam as propriedades do amplificador operacional. Entretanto, para este projeto será abordado apenas o circuito básico do amplificador operacional, que é o multiplicador com ganho constante. Este pode se apresentar como amplificador inversor e amplificador não inversor. Segue na figura 18 o circuito do amplificador inversor.

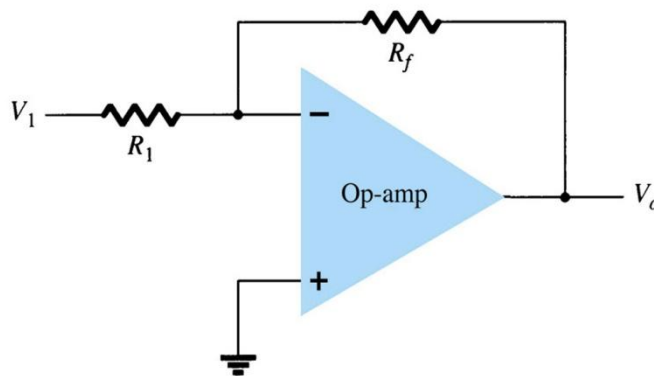


Figura 18 – Amplificador inversor [11]

Nesta configuração, o sinal (V_1) é aplicado na entrada inversora, sendo então amplificado por um ganho equivalente a razão entre os resistores R_f e R_1 . O sinal amplificado é enviado para a saída V_0 com sua polaridade invertida. O valor de V_0 pode ser calculado pela fórmula 8.

$$V_o = - \frac{R_f}{R_1} V_1$$

Fórmula 8 – Cálculo de V_0 para um amplificador inversor [11]

Já no amplificador não inversor, o sinal (V_1) é aplicado na entrada não inversora, sendo então amplificado por um ganho equivalente a razão entre os resistores R_f e R_1 acrescida de uma unidade. O sinal amplificado é enviado para a saída V_0 com a mesma polaridade do sinal de entrada. Segue na figura 19 o circuito do amplificador não inversor.

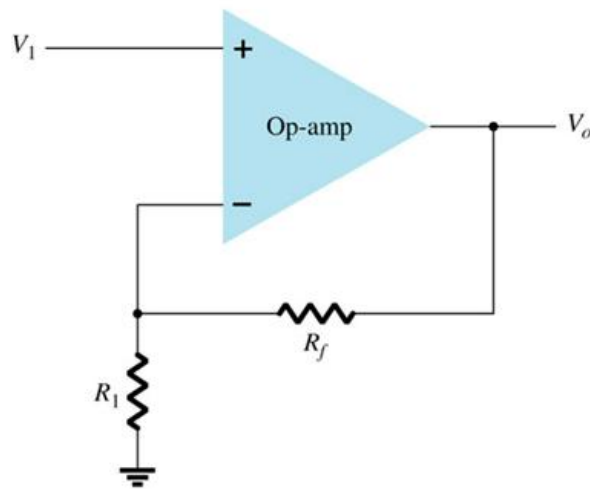


Figura 19 – Amplificador não inversor [11]

O valor de V_o pode ser calculado pela fórmula 9.

$$V_o = \left(1 + \frac{R_f}{R_1}\right) V_1$$

Fórmula 9 – Cálculo de V_o para um amplificador não inversor [11]

2.4.4 Circuito integrado LM324

O circuito integrado LM324 possui, em um mesmo encapsulamento, quatro amplificadores operacionais independentes. Segundo o Datasheet do LM324 (2012), este pode ser alimentado com uma tensão contínua na faixa de +3 a +30 Vcc ou $\pm 1,5$ a ± 15 Vcc. Seu diagrama de pinos pode ser observado na figura 20.

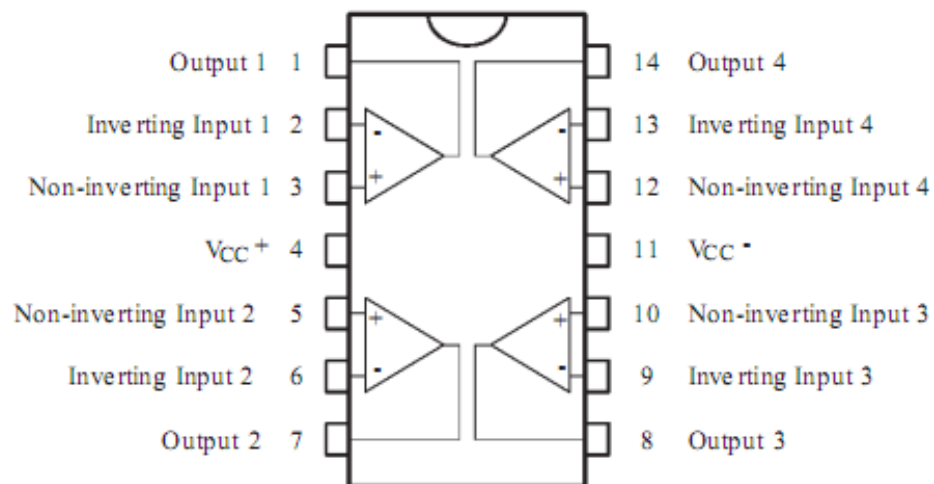


Figura 20 – Circuito integrado LM324 [25]

2.5 Eletrônica de potência

Os terminais de entrada e saída de um microcontrolador comumente possuem baixa capacidade de fornecimento de corrente. Segundo o Datasheet PIC 18F4520 (2012), este modelo de PIC possui as seguintes limitações para o fornecimento de corrente:

- Corrente máxima por pino de I/O: 25mA
- Corrente máxima por port: 200mA
- Corrente máxima no pino Vdd: 250mA
- Corrente máxima no pino Vss: 300mA

Estas limitações não são problema quando trabalhamos com acionamento de pequenas cargas. Entretanto, quando trabalhamos com cargas que demandam um fornecimento de corrente mais elevado, o microcontrolador não será capaz de alimentá-las diretamente através de seus terminais de I/O. Nestes casos, utiliza-se a eletrônica de potência como interface entre o sistema microcontrolado e a carga.

Os tiristores são dispositivos muito empregados na eletrônica de potência. Segundo Boylestad e Nashelsky (2004), são denominados tiristores os dispositivos semicondutores de 4 camadas (pnpn) que possuem um mecanismo de controle. Os principais tiristores são o SCR (retificador controlado de silício), o SCS (chave controlada de silício), o UJT (transistor de unijunção), o diac e o triac.

2.5.1 Triac

O triac é um tiristor projetado para chavear corrente alternada. Segundo Electrónica (2012), o triac possui três terminais: MT1 (anodo 1), MT2 (anodo 2) e o terminal de controle (gate). Seu símbolo pode ser visto na figura 21.

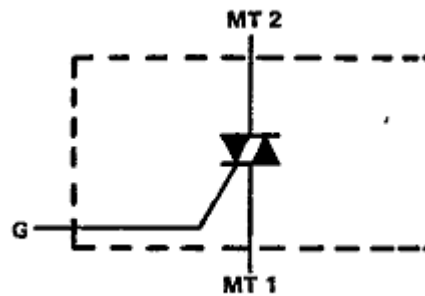


Figura 21 – Terminais de um triac [23]

Em seu funcionamento básico, o triac conduz corrente alternada entre seus terminais MT1 e MT2 quando uma tensão positiva ou negativa é aplicada em seu terminal de controle. O triac apresenta inicialmente um estado de bloqueio, passando para o estado de condução quando se atinge a tensão de ruptura. Esta tensão pode ser controlada aplicando-se um impulso positivo ou negativo no gate. Conforme a amplitude do impulso aumenta, diminui o valor da tensão de ruptura.

Uma vez que o triac entrou no estado de condução, ele permanecerá assim mesmo que o impulso deixe de ser aplicado em seu terminal de controle. Ele entrará em estado de bloqueio novamente quando a tensão entre seus terminais MT1 e MT2 cair abaixo da tensão de corte. Para que o triac possa conduzir novamente, deverá ser aplicado um novo impulso no gate.

2.5.2 Optoisolador

Segundo Boylestad e Nashelsky (2004), o optoisolador é constituído de uma capsula que contém um LED infravermelho e um fotodetector, como um diodo de silício, um par de transistores na configuração Darlington ou tiristores. Sua função é realizar o isolamento elétrico entre dois circuitos, sendo a comunicação entre eles realizada através de feixe de luz.

Este projeto utiliza o optoacoplador MOC3020. O optotriac interno do CI é acionado por um sinal de luz emitido em seu gate. O diagrama de pinos do MOC3020 pode ser observado na figura 22.

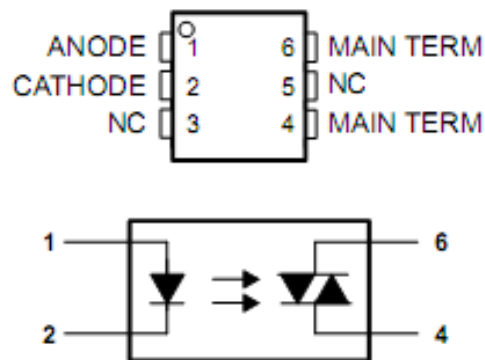


Figura 22 – Optoacoplador MOC3020 [21]

Nas tabelas 2 e 3 estão descritos os principais parâmetros do LED e do optotriac do MOC3020.

VD	Tensão direta sobre o led	1,2V - típico , 1,5V -máximo (ID=10mA)
VR	Tensão inversa máxima sobre o led	3 V
IDmáx.	Corrente direta máxima no led	50mA
PDmáx.	Potência máxima de dissipação	100mW
IFT	Corrente do diodo para disparo do triac	15mA (típ.) , 30mA (máx.)

Tabela 2 – Parâmetros do LED do CI MOC3020 [22]

VTM	Queda de tensão em ambos sentidos	1,4V (típico) , 3,0 V (máx.)
Iop	Corrente de saída para um único pulso em estado ligado	1.2A para período de 10mS
Iorms	Corrente de saída rms em estado ligado para 50 / 60Hz, ciclo completo	100mA (25°C) , 50mA (70°C)
PDmáx.	Potência de dissipação máxima contínua	300mW

Tabela 3 – Parâmetros do optotriac do CI MOC3020 [22]

2.6 Supervisório

Segundo Silva e Salvador (2012), “os sistemas supervisórios permitem que sejam monitoradas e rastreadas informações de um processo produtivo ou instalação física. Tais informações são coletadas através de equipamentos de aquisição de dados e, em seguida, manipuladas, analisadas, armazenadas e posteriormente, apresentadas ao usuário”.

Programas supervisórios são muito utilizados na indústria para se monitorar e controlar processos à distância. A partir de um único terminal, o operador tem a possibilidade de visualizar as variáveis do processo como um todo e interagir com o mesmo sempre que necessário. O *software* desenvolvido neste projeto segue este mesmo princípio, permitir que a manta termoeletrica seja monitorada e controlada à distância.

2.6.1 Comunicação serial

O PIC 18F4520 possui um módulo específico para comunicação serial. Segundo Sousa, Souza e Lavinia (2010), o módulo USART (Universal Synchronous Asynchronous Receiver Trasnmmitter) deste microcontrolador pode opera em dois modos de funcionamento, o modo síncrono e o modo assíncrono. O modo síncrono utiliza duas vias de comunicação, uma destinada ao tráfego de dados e outra para o sinal de *clock*, o qual realiza o sincronismo entre o emissor e o receptor. Como os dados trafegam por apenas uma via, a comunicação neste modo de operação é *half duplex*.

O modo assíncrono também utiliza duas vias para comunicação, uma para transmissão (TX) e outra para recepção (RX). Neste caso, como a transmissão e a recepção de dados são realizadas por vias distintas, a comunicação pode ser *full duplex*. Neste modo de comunicação não temos o sinal do *clock* para realizar o sincronismo. Para que possa ocorrer a comunicação, os dados devem ser enviados em intervalos de tempo definidos. Tanto o emissor quanto o receptor devem ser configurados com a mesma taxa de transmissão, para que o dado possa ser lido no momento certo. Esta taxa de transmissão é denominada *baud rate*. O *baud rate* pode ser calculado pela fórmula 10.

SYNC	BRGH=0	BRGH=1
0	$BR = F_{osc}/(64x(SPBRG+1))$	$BR = F_{osc}/(16x(SPBRG+1))$
1	$BR = F_{osc}/(4x(SPBRG+1))$	Não válido

Fórmula 10 – Cálculo do *baud rate* [7]

Sendo que o *bit* de configuração “SYNC” define o modo de funcionamento (0 para modo assíncrono e 1 para modo síncrono), o *bit* de configuração “BRGH” define o ajuste da velocidade de transmissão (0 para baixa velocidade e 1 para alta velocidade), F_{OSC} é a frequência do oscilador e “SPBRG” é o registro no qual se deve carregar o valor necessário para se obter o *baud rate* desejado.

2.6.2 Interface

A comunicação entre o computador e o controlador de temperatura será realizada serialmente através da interface RS-232. Este tipo de comunicação utiliza um conector de 9 pinos, chamado de DB9. O conector e sua pinagem podem ser observados na figura 23 e na tabela 4.



Figura 23 – Conector DB9 [29]

Pino	Descrição
1	DCD
2	RX
3	TX
4	DTR
5	GND
6	DSR
7	RTS
8	CTS
9	NC

Tabela 4 – Pinagem do conector DB9 [28]

Segundo Comunicação [...] (2012), os pinos 2 e 3 são utilizados para realizar a comunicação e o pino 5 deverá ser interligado ao GND do equipamento. Entretanto os níveis de tensões utilizados no padrão RS-232 são diferentes dos níveis do padrão TTL utilizado pelo microcontrolador PIC. No padrão RS-232 o nível baixo equivale a uma tensão de 3V a 18V e o nível alto equivale a uma tensão de -3V a -18V. Já no padrão TTL, o nível baixo equivale a uma tensão de 0V e o nível alto equivale a uma tensão de 5V. O circuito integrado MAX232 realiza a interface entre o padrão RS-232 e o padrão TTL. Sua pinagem pode ser observada na figura 24.

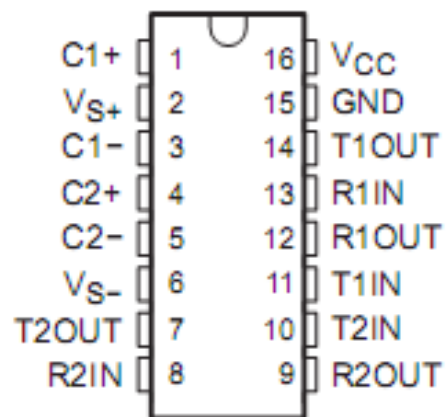


Figura 24 – Pinagem do circuito integrado MAX232 [30]

3 METODOLOGIA

Este projeto irá desenvolver um controlador de temperatura microcontrolado para o acionamento de mantas termoelétricas, as quais são constituídas por cargas resistivas. A primeira etapa do projeto será o desenvolvimento da fonte de alimentação, a qual será do tipo linear e irá prover alimentação de 5 Vcc para o circuito do microcontrolador e 16 Vcc para o amplificador operacional que amplifica o sinal proveniente do sensor. A fonte será constituída por um transformador abaixador, o circuito de retificação de onda completa em ponte, o filtro capacitivo e o estágio de regulação.

A próxima etapa do projeto é o desenvolvimento do circuito microcontrolado. Será utilizado o microcontrolador PIC 18F4520, o qual possui todos os recursos necessários para o projeto. Sua programação será feita utilizando-se a linguagem *assembly*, pois, além de gerar um código fonte menor e mais otimizado, esta linguagem proporciona ao programador o controle completo do *hardware* utilizado. O programa fará periodicamente a leitura do sinal proveniente de sua entrada analógica e realizará uma média aritmética das últimas 60 leituras. Este procedimento é necessário para diminuir a interferência no sinal medido. Então este valor da média será convertido em um valor de temperatura. Esta informação será mostrada em um *display* LCD com 2 linhas e 16 colunas. O *display* também mostrará informações referentes ao *set point*, valor máximo de temperatura e valor mínimo de temperatura, parâmetros estes que poderão ser configurados pelo usuário.

O microcontrolador também estará preparado para receber informação proveniente de quatro botões pulsantes. Um botão para incremento de parâmetro, um botão para decremento de parâmetro, um botão para avançar pelas telas de configuração e um botão para voltar para a tela inicial. Por fim, o microcontrolador realizará o acionamento da carga através de um circuito de potência. Este circuito será composto por um optotriac, que será responsável pelo isolamento elétrico entre o circuito microcontrolado e a tensão da rede; e um triac de potência, que será responsável pelo fornecimento de corrente para a carga. Quando a temperatura medida ficar acima do valor definido pelo *set point*, o microcontrolador enviará um sinal para que a carga seja desligada. Quando o valor medido da temperatura ficar abaixo do valor definido no *set point*, o microcontrolador enviará um sinal para que a carga seja acionada. O valor do *set point* poderá ser configurado pelo usuário de 0 °C a 99,9 °C com incrementos de 0,1 °C. Entretanto este valor estará limitado dentro da faixa correspondente aos parâmetros “temperatura máxima” e “temperatura mínima”, os quais também poderão ser configurados pelo usuário.

A temperatura da manta termoelétrica será medida através de um sensor LM35 instalado em seu interior. O sensor será responsável por converter o valor medido de tempera-

tura em um valor correspondente de tensão. Esta tensão passará por um estágio de amplificação para que fique na faixa de 0 a 5 Vcc. Após a amplificação, o sinal será enviado até uma entrada analógica do microcontrolador para que possa ser tratado.

O projeto também contará com um *software* supervisor que será executado em um computador comum. O mesmo será desenvolvido utilizando-se a linguagem de alto nível *Visual Basic*. O supervisor terá uma interface similar ao *layout* do controlador físico e contará com as mesmas funcionalidades deste, como o fornecimento de informações via *display* e os quatro botões de ação. A atualização das informações será realizada em tempo real, sendo a comunicação entre o computador e o controlador realizada de forma serial (padrão RS-232) através do circuito integrado MAX-232.

Este trabalho é multidisciplinar, envolvendo conceitos de várias áreas distintas. Para o projeto do controlador de temperatura foram estudados os conceitos envolvendo a programação de baixo nível para microcontroladores, eletrônica básica e fontes de alimentação, programação em alto nível para computadores, os conceitos de instrumentação no tratamento de informações provenientes de sensores e a eletrônica de potência envolvida no acionamento de cargas que demandam correntes mais elevadas. Todos estes assuntos tiveram o devido embasamento teórico necessário, assim como um estudo mais aplicado ao projeto desenvolvido.

4 DESENVOLVIMENTO

4.1 Fonte de alimentação

A fonte de alimentação foi a primeira etapa a ser desenvolvida no projeto. Ela será responsável por alimentar o microcontrolador, o *display* LCD e o sensor de temperatura LM35 com a tensão de 5 Vcc, assim como também alimentará o amplificador operacional LM324N com a tensão de 16 Vcc.

A escolha do transformador foi baseada na tensão de saída do secundário e na corrente máxima suportada. Segundo Marques, Cruz e Júnior (2002, p. 159), “um dado de ordem prática, é que a tensão V_e deve ser pelo menos 50% maior que a tensão V_s desejada”. A tensão V_e corresponde a tensão após a retificação e V_s é a tensão de saída da fonte. No projeto, V_s equivale à tensão de 5 Vcc que alimentará os circuitos. Assim:

$$V_e = 1.5 * 5 \qquad V_e = 7.5 \text{ V}$$

Equação 1 – Cálculo de V_e

A tensão de pico do secundário do transformador deve ter um valor próximo ao valor de V_e . Entretanto, esta tensão V_e não será suficiente para alimentar o amplificador operacional com uma tensão de 16 Vcc. Portanto foi escolhido um transformador com tensão rms de 12 Vca, o qual terá uma tensão de pico de:

$$V_p = 12 * \sqrt{2} \qquad V_p = 16,97 \text{ Vca}$$

Equação 2 – Tensão de pico do transformador

A corrente do circuito foi medida e vale 50 mA. Portanto, foi escolhido um transformador com tensão do primário de 127 / 220 Vca, tensão do secundário de 12 Vca e corrente máxima de 500 mA. Desta forma o transformador trabalhará com folga.

A forma de retificação escolhida foi a retificação em onda completa em ponte, pois apresenta diversas vantagens. Desta forma, considerando a tensão de pico no secundário do transformador de 16,97 Vca, a barreira de potencial dos diodos de 0,7 V e utilizando-se a fórmula 6, podemos calcular a tensão média após o estágio de retificação.

$$V_m = \frac{2 \cdot (16.97 - 2 \cdot 0.7)}{\pi} \quad V_m = 9,91 \text{ Vcc}$$

Equação 3 – Tensão média após retificação

Podemos verificar que a tensão média ficou com um baixo valor, pois mesmo que o retificador em onda completa em ponte dobre a frequência do sinal, este ainda apresenta uma grande oscilação. Para resolver este problema foi utilizado um filtro capacitivo. Para calcularmos o valor do capacitor, precisamos estimar qual o valor da carga do circuito. Para obter este valor, foi medida a corrente elétrica do circuito, alimentando-o com 16 Vcc. O valor obtido foi de 50 mA. Utilizando-se a Lei de Ohm, podemos calcular o valor da carga RL.

$$R_L = \frac{16}{0.05} \quad R_L = 320\Omega$$

Equação 4 – Valor da carga do circuito

Considerando-se o valor de RL calculado, adotando-se o valor da tensão média após a filtragem de 16 Vcc e considerando uma tensão de ripple aceitável de 0.5V, calcula-se o valor do capacitor utilizando-se a fórmula 7.

$$C = \frac{16}{120 \cdot 320 \cdot 0,5} \quad C = 833 \mu\text{F}$$

Equação 5 – Valor do capacitor de filtro

Como 833 μF não é um valor comercial foi adotado o valor de 1000 μF . A tensão sobre este capacitor será de 16 Vcc, portanto foi escolhido um capacitor eletrolítico de 1000 μF / 25V.

O esquema elétrico das etapas da fonte de alimentação descritas até aqui podem ser observadas na figura 25.

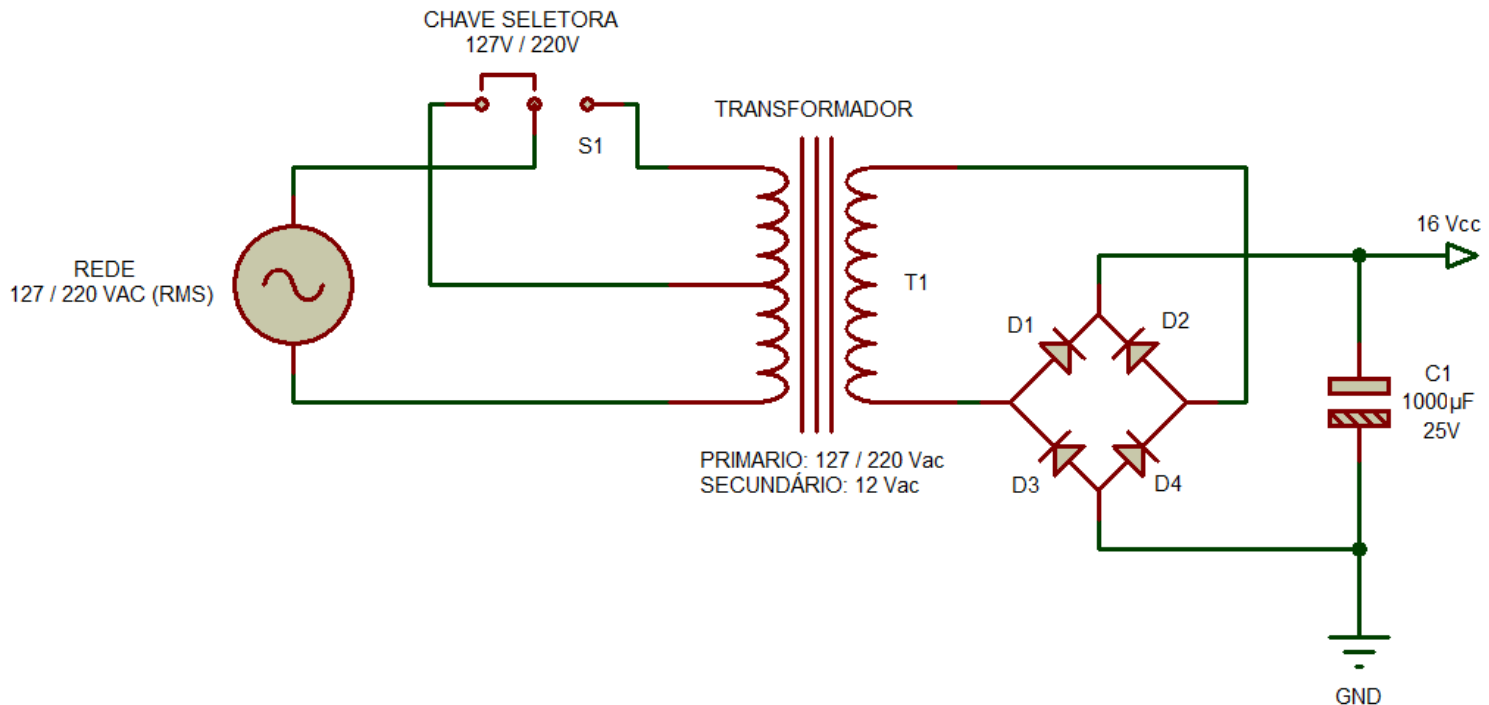


Figura 25 – Esquema elétrico da fonte de alimentação (fonte própria)

A “REDE” é o sinal senoidal da rede elétrica, que pode ser 127 Vac ou 220 Vac. A escolha da tensão da rede é feita através da chave “S1”, que trata-se de uma chave do tipo “HH”. Quando a chave está na posição 1-2 (conforme a figura 25), apenas metade do enrolamento primário do transformador é utilizado. Desta forma o circuito deverá ser ligado na rede 127 Vac. Quando a chave passa para a posição 2-3, o enrolamento primário completo do transformador é utilizado e o circuito deverá ser ligado na rede 220 Vca. Esta característica dos transformadores pode ser observada através da fórmula 2, a qual expressa a relação existente entre o número de espiras e as tensões do primário e secundário. Isolando V2 na fórmula temos:

$$V2 = \frac{N2 * V1}{N1}$$

Equação 6 – Tensão no secundário de um transformador

Pela equação 6 podemos observar que, mantendo N2 constante, se dobrarmos o valor de V1 teremos que dobrar também o valor de N1 para mantermos a tensão no secundário.

rio V2 constante. O transformador “T1” possui um enrolamento primário que pode ser ligado em 127 Vac ou 220 Vac e o enrolamento secundário será de 12 Vac. Este transformador suporta uma corrente máxima de 500 mA, conforme foi descrito anteriormente.

A ponte retificadora é composta pelos diodos “D1”, “D2”, “D3” e “D4”. O modelo de diodo utilizado é o 1N4007, muito utilizado comercialmente. A corrente direta máxima suportada por este diodo é 1 A, e a tensão reversa máxima é de 700 V. Como no projeto a corrente máxima será de 50 mA e a tensão de pico no secundário do transformador será de 16,97 V, este modelo de diodo atende perfeitamente as exigências do projeto.

O capacitor C1 será do tipo eletrolítico e seu valor será de 1000 μF / 25V, conforme especificado anteriormente.

O último estágio da fonte de alimentação é a regulação para a tensão de 5 Vcc. Ele é composto pelo regulador de tensão 7805 e por capacitores de filtro. Seu esquema elétrico pode ser observado na figura 26.

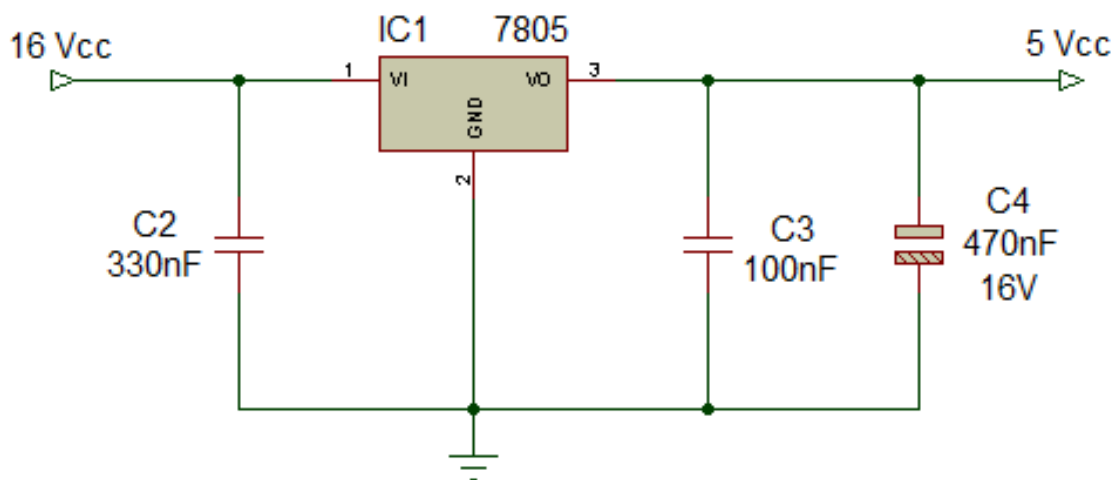


Figura 26 – Esquema elétrico do regulador de tensão (fonte própria)

Os capacitores “C2” e “C3” filtram, respectivamente, os sinais de entrada e saída contra ruídos. Seu uso e os valores escolhidos são recomendados pelo fabricante do regulador de tensão. O capacitor “C4” foi utilizado para suprir os picos de corrente do circuito.

4.2 Microcontrolador

O projeto foi desenvolvido utilizando-se o microcontrolador PIC 18F4520 da Microchip. Este modelo foi escolhido devido à literatura específica existente sobre ele e também

pelos recursos adicionais presentes na família 18F como, por exemplo, a adição de novas instruções, principalmente instruções de multiplicação e comparação, as quais facilitam a programação e reduzem o código fonte.

Entretanto outros modelos mais simples poderiam ter sido utilizados. Microcontroladores PIC da família 16F e com 28 terminais já atenderiam as necessidades do hardware. O importante é que o microcontrolador escolhido tenha um conversor analógico/digital de 10 bits para a leitura do sinal proveniente do sensor e que possua um módulo de comunicação USART. Os demais recursos de hardware utilizados poderiam ser implementados em qualquer modelo de microcontroladores PIC, desde que o mesmo tenha uma quantidade de terminais de I/O suficiente.

Outro fator que deve ser observado é a quantidade de memória de programa. Dependendo da complexidade do programa, este pode chegar a ocupar vários kbytes de memória. Existem PIC's com diversas capacidades de memória, sendo este um fator que causa grande influência no custo do microcontrolador. Portanto este também foi um dos fatores que levaram à escolha do PIC 18F4520 para o projeto. Este modelo possui 32 kbytes de memória de programa, uma quantidade elevada que suporta com folga mesmo os programas mais complexos.

A figura 27 apresenta o circuito do microcontrolador. Na figura 27 não foi mostrado os terminais de alimentação do PIC 18F4520, pois o esquema eletrônico foi desenvolvido e simulado utilizando-se o *software Proteus*, o qual omite os pinos de alimentação. Entretanto, neste modelo a alimentação é realizada pelos pinos 11 e 32 para o Vdd (+5 Vcc) e pelos pinos 12 e 31 para o Vss (GND).

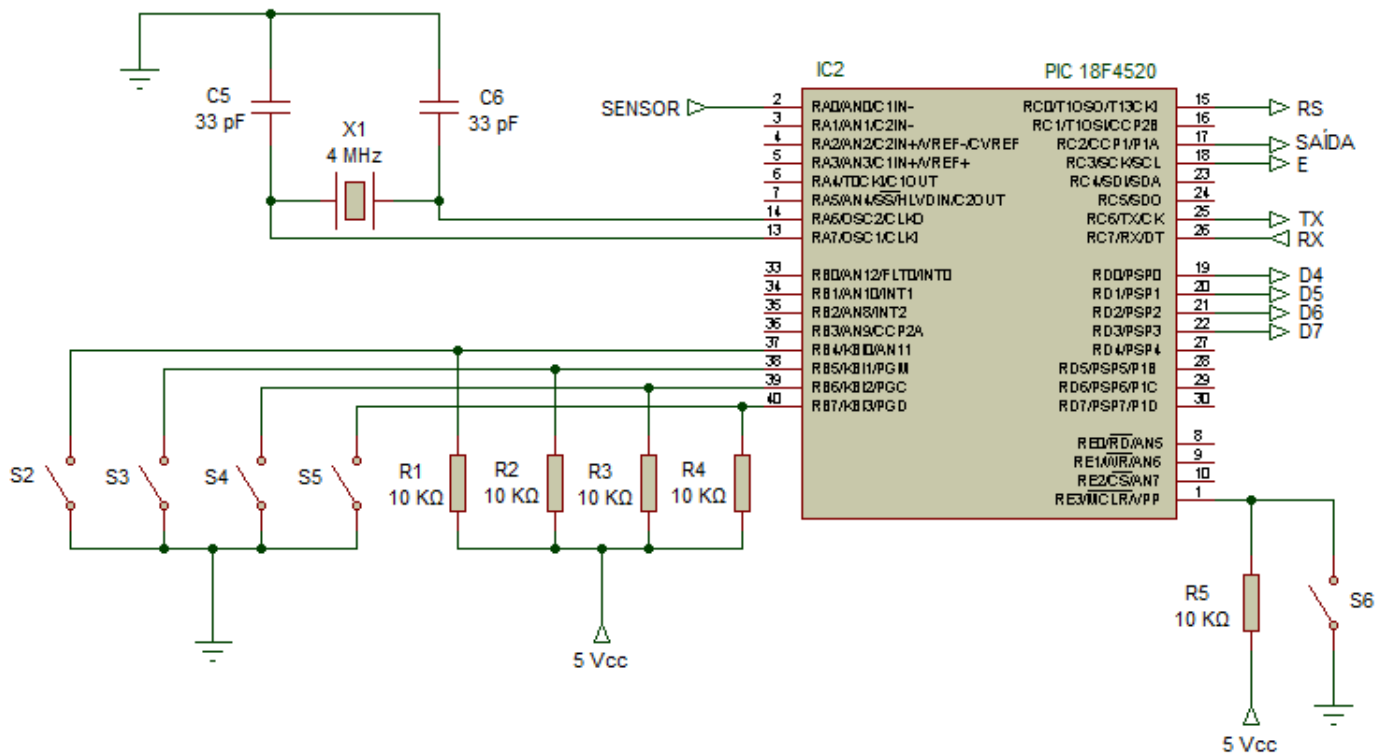


Figura 27 – Esquema elétrico do microcontrolador (fonte própria)

4.2.1 Geração de clock

Foi utilizado um cristal ressonador de 4 MHz para geração de clock para o microcontrolador. Os valores dos capacitores cerâmicos “C5” e “C6” são recomendados pelo fabricante e variam de acordo com a frequência do cristal utilizado. Com uma frequência de 4MHz temos um período de 250nS, uma vez que a frequência é o inverso do período. Entretanto, segundo Zanco (2005), os microcontroladores da família PIC dividem a frequência do oscilador principal por quatro, dando origem a um sinal de clock interno quatro vezes menor que o valor do clock principal. Este clock interno, ou ciclo de instrução, é utilizado como referência para a execução das instruções. Todas as instruções são executadas em um ciclo de instrução, com exceção das que provocam desvio no programa que são executadas em dois ciclos. Assim, utilizando-se um cristal de 4MHz, o clock interno será de 1MHz e seu período será de 1μS. Ou seja, as instruções que não provocam desvios são executadas em 1μS e as que provocam desvios são executadas em 2μS. Esta é uma informação importante quando se trabalha com rotinas de *delay*.

Existem vários métodos de geração de clock, incluindo a utilização do gerador interno de clock do PIC, dispensando o uso de componentes externos. Entretanto, quando se necessita de precisão no sinal do clock, o uso do oscilador é altamente recomendado. No projeto, esta necessidade de precisão no clock se dá devido o uso da comunicação serial.

O cristal ressonador externo é conectado aos pinos OSC1 e OSC2 do microcontrolador, ou seja, nos pinos 13 e 14. O uso de um cristal externo ocupa dois pinos de I/O.

4.2.2 Botões de configuração

As chaves “S2”, “S3”, “S4” e “S5” são chaves do tipo tátil e serão utilizadas para o operador interagir com o controle de temperatura, ajustando diversos parâmetros. A chave “S2” corresponde ao botão “+”, o qual incrementa o valor do parâmetro selecionado. A chave “S3” corresponde ao botão “-”, o qual decrementa o valor do parâmetro selecionado. A chave “S4” corresponde ao botão “inicial” e retorna para a tela inicial, caso o *display* esteja mostrando alguma outra tela. A chave “S5” corresponde ao botão “P” e avança para a próxima tela de configuração.



Figura 28 – Botões de configuração (fonte própria)

Os resistores “R1”, “R2”, “R3” e “R4” foram utilizados para manter o barramento das chaves em nível alto quando elas não estiverem acionadas. Ao acionar uma chave, a mesma irá aterrar a sua linha, provocando uma transição de nível alto para nível baixo, a qual provoca uma interrupção no microcontrolador. Quando a chave é liberada ocorre uma transição de nível baixo para nível alto na linha, a qual também provoca uma interrupção. Entretanto, estas interrupções apenas serão tratadas quando houver nível baixo de tensão na

linha, ou seja, quando a chave estiver pressionada. O valor dos resistores “R1”, “R2”, “R3” e “R4” são indiferentes, porém foram utilizados valores elevados de resistência para que não haja um consumo desnecessário de corrente elétrica.

A chave “S6” corresponde ao botão de reset. Quando acionada, provoca um reset no microcontrolador, ou seja, ele recomeça a execução do programa desde o princípio. Sua função é restabelecer o correto funcionamento do circuito, caso o mesmo apresente alguma anormalidade decorrente de travamentos na execução do programa por parte do microcontrolador. Neste caso, os registros especiais são carregados com seu valor padrão e o programa é desviado para o vetor de reset, ou seja, para o endereço 0x00. O resistor “R5” foi utilizado para manter nível alto de tensão na linha de reset quando a chave “S6” não estiver acionada, uma vez que o reset ocorre quando a tensão no pino 1 do microcontrolador é levada para nível baixo.

4.2.3 *Display* LCD

O projeto conta com um *display* de cristal líquido que facilita a interface com o usuário. Sua função é exibir as telas nas quais serão configurados os parâmetros do controlador de temperatura. Será apresentado no *display* LCD um total de quatro telas.

A primeira tela corresponde à tela inicial, na qual será mostrado o valor do *set point* selecionado e o valor medido da temperatura. Caso o *display* esteja mostrando alguma outra tela, ao se pressionar a tecla “inicial” (chave “S4”), o *display* retornará para a tela inicial.



Figura 29 – Tela inicial do *display* LCD (fonte própria)

A segunda tela corresponde ao ajuste do *set point*. Quando do *display* mostrar esta tela, o valor do *set point* poderá ser incrementado através do botão “+” (chave “S2”) ou decrementado através do botão “-” (chave “S3”).



Figura 30 – Tela de ajuste do *set point* (fonte própria)

A terceira tela corresponde ao ajuste da temperatura máxima. Este parâmetro determina o valor máximo que o *set point* poderá ser ajustado. Esta configuração é importante para qualquer sistema de controle de temperatura, pois não permite que a planta seja danificada por um ajuste incorreto de *set point* por parte do usuário. Em controladores de temperatura comerciais, este parâmetro é comumente protegido por senha.



Figura 31 – Tela de ajuste da temperatura máxima (fonte própria)

A quarta tela corresponde ao ajuste de temperatura mínima. Este parâmetro determina o valor mínimo que o *set point* poderá ser ajustado. Esta configuração também tem como objetivo evitar problemas decorrentes de ajuste incorreto do *set point*, pois alguns sistemas necessitam que a temperatura não caia abaixo de um determinado valor. Da mesma forma que o valor máximo, em controladores de temperatura comerciais este parâmetro é comumente protegido por senha.



Figura 32 – Tela de ajuste da temperatura mínima (fonte própria)

Segundo Barbacena e Fleury (1996), existem basicamente dois tipos de *display* LCD, os *displays* gráficos e de caractere. A diferença entre eles se dá pela quantidade de pixels disponíveis. Os *displays* de caractere possuem menos pixels para a formação da imagem,

podendo-se apenas formar os caracteres do código ASCII e alguns caracteres especiais simples. As características que devem ser levadas em consideração na escolha de um *display* LCD de caractere são a quantidade de linhas, a quantidade de colunas, a cor do *display*, a presença ou não do backlight e o tamanho da matriz que forma cada caractere. No projeto, foi utilizado um *display* LCD de caractere de 2 linhas, 16 colunas, com a tela azul (caracteres escritos em branco), com backlight e com matriz de 7x5. A pinagem e o circuito do *display* LCD podem ser observados na tabela 5 e na figura 33.

PINO	FUNÇÃO	DESCRIÇÃO
1	VSS	GND
2	VDD	+5 VCC
3	VEE	Tensão para ajuste do contraste
4	RS	1 (envio de dado) ou 0 (envio de instrução)
5	R/W	1 (leitura do LCD) ou 0 (escrita no LCD)
6	E	Habilita LCD
7	D0	Barramento de dados
8	D1	
9	D2	
10	D3	
11	D4	
12	D5	
13	D6	
14	D7	
15	A	Anodo do led de backlight
16	K	Catodo do led de backlight

Tabela 5 – Pinagem do *display* LCD [15]

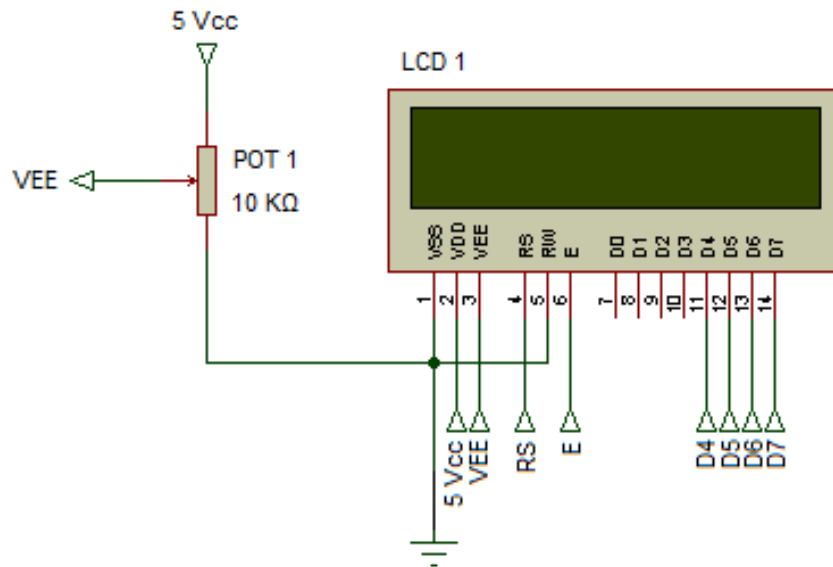


Figura 33 – Esquema elétrico do *display* LCD (fonte própria)

A alimentação do *display* LCD é realizada através dos pinos 1 (GND) e 2 (VCC). A alimentação nominal do *display* é 5 Vcc, podendo variar na faixa de 4.7 Vcc a 5.3 Vcc. O contraste pode ser controlado através de uma tensão aplicada ao pino 3 (VEE) do LCD. O contraste varia inversamente proporcional a esta tensão, ou seja, quanto menor a tensão aplicada maior será o contraste obtido. No projeto foi utilizada uma tensão de 0.9 Vcc para o contraste. Este valor pode ser ajustado através do potenciômetro “POT 1”.

O pino 4 (RS) é utilizado para informar o *display* LCD se a informação contida no barramento de dados é uma instrução ou um dado. Se for uma instrução este pino deverá ser colocado em nível baixo e se for um dado o pino deverá ser colocado em nível alto. Este pino é controlado pelo microcontrolador, o qual é responsável pelo envio dos dados e instruções.

O pino 5 (R/W) informa ao *display* LCD se será realizada uma operação de leitura ou de escrita. Se for uma operação de escrita este pino deverá ser colocado em nível baixo e se for uma operação de leitura o pino deverá ser colocado em nível alto. A operação de leitura comumente utilizada quando se trabalha com LCD é a leitura do busy flag, na qual o *display* informa que está pronto para receber um novo comando. No projeto não serão utilizadas operações de leitura, portanto o pino R/W foi aterrado. Para que seja assegurado que um comando não seja enviado para o *display* quando este não estiver pronto para recebê-lo, o microcontrolador obedecerá às temporizações recomendadas pelo fabricante.

Os pinos 7 a 14 correspondem ao barramento de dados de 8 bits do *display* LCD. A comunicação pode ser realizada utilizando-se todos os 8 bits do barramento ou utilizando-se apenas os últimos 4 bits (D4, D5, D6 e D7). Neste caso, envia-se o nibble mais significativo

primeiro e depois se envia o nibble menos significativo. Este é o tipo de comunicação que foi utilizada no projeto, pois necessita de menos fios, o que simplifica a elaboração do circuito.

Antes de se utilizar o *display* LCD este deve receber alguns comandos de configuração. A tabela 6 mostra o conjunto de instruções que o *display* está programado para receber e o tempo necessário para sua execução. Na tabela 6, B0 - B7 corresponde ao barramento de dados.

INSTRUÇÃO	R S	R/ W	B7	B6	B5	B4	B3	B2	B1	B0	DESCRIÇÃO e tempo de execução (uS)	t
Limpa Display	0	0	0	0	0	0	0	0	0	1	-Limpa todo o display e retorna o cursor para a primeira posição da primeira linha	1.6 mS
Home p/ Cursor	0	0	0	0	0	0	0	0	1	*	-Retorna o cursor para a 1. coluna da 1. Linha -Retorna a mensagem previamente deslocada a sua posição original	1.6 mS
Fixa o modo de funcionamento	0	0	0	0	0	0	0	1	X	S	-Estabelece o sentido de deslocamento do cursor (X=0 p/ esquerda, X=1 p/ direita) -Estabelece se a mensagem deve ou não ser deslocada com a entrada de um novo caracter (S=1 SIM, X=1 p/ direita) -Esta instrução tem efeito somente durante a leitura e escrita de dados.	40 uS
Controle do Display	0	0	0	0	0	0	1	D	C	B	-Liga (D=1) ou desliga display (D=0) -Liga(C=1) ou desliga cursor (C=0) -Cursor Piscante(B=1) se C=1	40 uS
Desloca cursor ou mensagem	0	0	0	0	0	1	C	R	*	*	-Desloca o cursor (C=0) ou a mensagem (C=1) para a Direita se (R=1) ou esquerda se (R=0) - Desloca sem alterar o conteúdo da DDRAM	40 uS
Fixa o modo de utilização do módulo LCD	0	0	0	0	1	Y	N	F	*	*	-Comunicação do módulo com 8 bits(Y=1) ou 4 bits(Y=0) -Número de linhas: 1 (N=0) e 2 ou mais (N=1) -Matriz do caracter: 5x7(F=0) ou 5x10(F=1) - Esta instrução deve ser ativada durante a inicialização	40 uS
Posiciona no endereço da CGRAM	0	0	0	1	Endereço da CGRAM						-Fixa o endereço na CGRAM para posteriormente enviar ou ler o dado (byte)	40 uS
Posiciona no endereço da DDRAM	0	0	1	Endereço da DDRAM							-Fixa o endereço na DDRAM para posteriormente enviar ou ler o dado (byte)	40 uS

Leitura do Flag Busy	0	1	B F	AC							-Lê o conteúdo do contador de endereços (AC) e o BF . O BF (bit 7) indica se a última operação foi concluída (BF =0 <i>concluída</i>) ou está em execução (BF =1).	0
Escreve dado na CGRAM / DDRAM	0	1	Dado a ser gravado no LCD								- Grava o byte presente nos pinos de dados no local apontado pelo contador de endereços (<i>posição do cursor</i>)	40 uS
Lê Dado na CGRAM / DDRAM	1	1	Dado lido do módulo								- Lê o byte no local apontado pelo contador de endereços (<i>posição do cursor</i>)	40 uS

Tabela 6 – Conjunto de instruções do *display* LCD [15]

A tabela 7 descreve os comandos que foram utilizados no projeto para inicialização do *display* LCD.

Valor do byte enviado	Descrição do comando
B'00110011'	Comando de inicialização do <i>display</i>
B'00110011'	Comando de inicialização do <i>display</i>
B'00101000'	Configura o <i>display</i> para comunicação 4 bits, 2 linhas e matriz de 5x7
B'00001000'	Desliga o <i>display</i> e desliga o cursor
B'00000001'	Limpa a tela do <i>display</i>
B'00000110'	Configura deslocamento automático da mensagem para a direita
B'00000011'	Retorna o cursor para a linha 1 e coluna 1
B'00001100'	Liga o <i>display</i> e desliga o cursor

Tabela 7 – Comandos utilizados para a inicialização do *display* LCD [15]

No envio de dados para o *display* LCD, antes de se enviar o dado propriamente dito, deve-se selecionar o endereço da posição na qual será mostrado o caractere. Segue na tabela 8 os endereços das posições de um *display* 16x2.

LCD 16x2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
linha 1	80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
linha 2	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF

Tabela 8 – Endereços de um *display* 16x2 [15]

4.3 Leitura de temperatura

A aquisição dos dados de temperatura é muito importante para o projeto, uma vez que este trata justamente do controle de temperatura. A escolha do sensor e o condicionamento do sinal proveniente dele são pontos preponderantes para se ter estabilidade e confiabilidade na aquisição dos dados medidos.

4.3.1 Sensor LM35

O sensor semicondutor LM35 foi o escolhido para ser utilizado no projeto. Sua escolha se deve principalmente ao fato da resposta linear que este sensor apresenta, tendo uma relação direta entre a temperatura medida em graus Celsius e sua tensão de saída. Esta característica do sensor simplifica a programação do microcontrolador, uma vez que não terá que ser realizada nenhuma espécie de linearização. Dependendo do sensor, o processo de linearização exige cálculos complexos demais para serem implementados utilizando-se linguagem *assembly*.

O sensor LM35 apresenta também uma precisão de 0.5°C , que atende as necessidades da aplicação proposta. O esquema de ligação do sensor LM35 utilizando-se o ambiente simulado do software *Proteus* pode ser observado na figura 34.

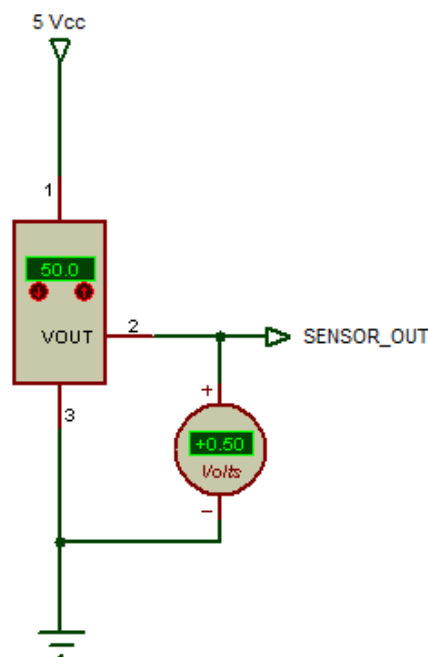


Figura 34 – Esquema de ligação do sensor LM35 (fonte própria)

Através da simulação no software *Proteus* podemos observar claramente a relação linear de $10\text{mV}/^{\circ}\text{C}$ que o sensor apresenta. Ou seja, para uma temperatura de 50°C selecionada no sensor, obtemos como resposta uma tensão de saída de 500mV .

4.3.2 Condicionamento de sinal do sensor

O conversor analógico/digital do PIC 18F4520 nos oferece duas possibilidades de seleção da tensão de referência que iremos utilizar. Podemos utilizar como referência a tensão presente entre os pinos 4 (RA2) e 5 (RA3) ou então podemos utilizar como referência a tensão de alimentação do PIC, ou seja, Vdd e Vss. Esta escolha é feita através dos bits de configuração VCFG0 e VCFG1, contidos no registrador ADCON1. Para simplificar o hardware, o projeto utilizou a tensão de alimentação como referência para o conversor analógico/digital. Desta forma, o mesmo estará apto a receber e converter tensões que variam dentro da faixa de 0 a 5 Vcc.

O controlador de temperatura foi desenvolvido para realizar medições e controle de temperatura na faixa de 0°C a 99,9°C. Com o sensor LM35 apresentando uma relação de 10mV/°C, a sua tensão de saída irá variar de 0 a 0,999 Volt. Desta forma, a tensão que o sensor de temperatura irá apresentar utilizaria apenas 20% da faixa de tensão que o conversor analógico/digital do PIC está configurado para reconhecer. Isto resultaria em perda de precisão na leitura, pois o conversor teria que trabalhar com variações muito pequenas de tensão. Além do fato que sinais de pequena amplitude são mais suscetíveis a interferências.

Para resolver este problema, o sinal de saída do sensor passa por um amplificador não inversor antes de chegar até o microcontrolador. O esquema elétrico do sistema de aquisição de dados do sensor pode ser observado na figura 35.

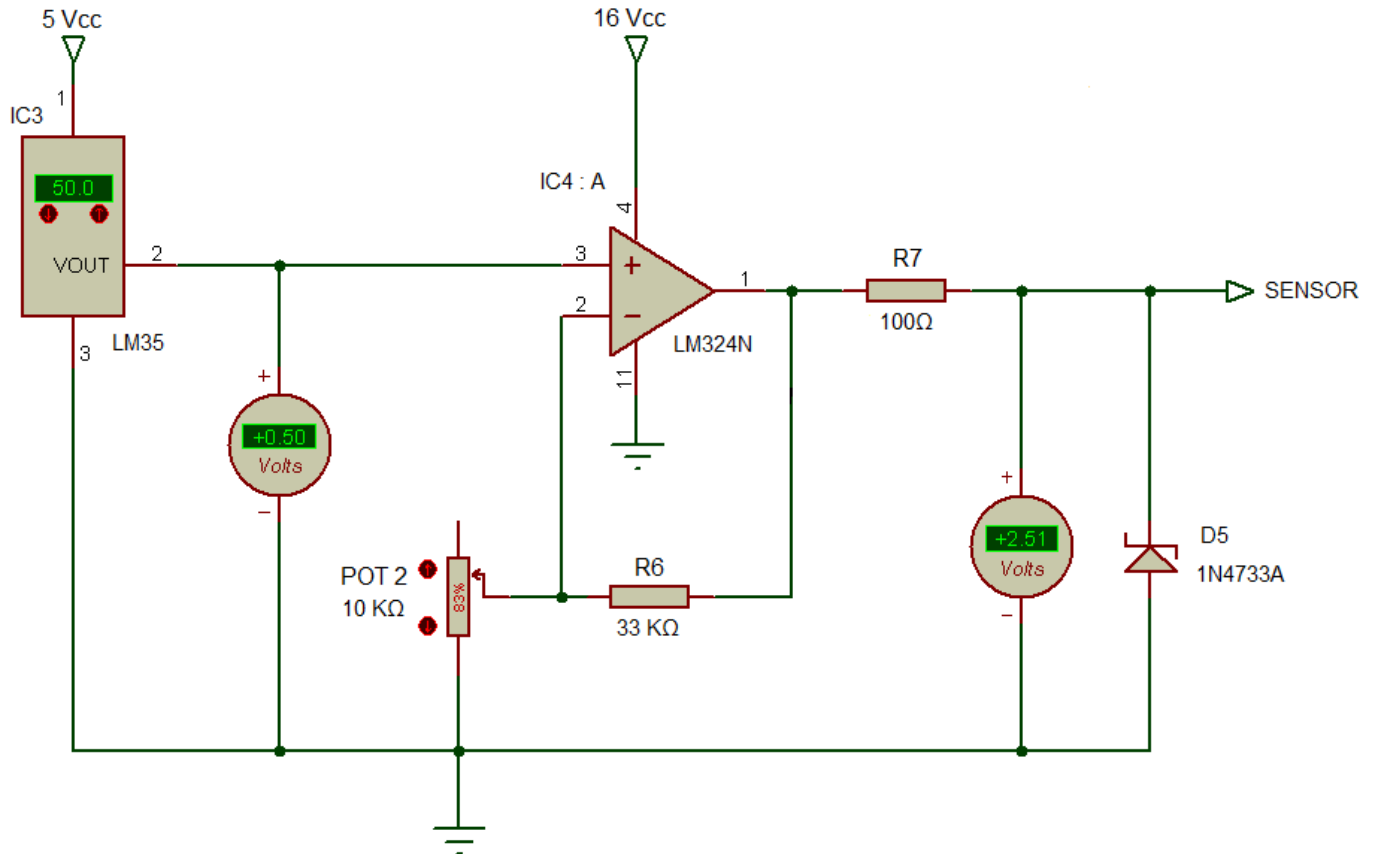


Figura 35 – Amplificador de sinal do sensor LM35 (fonte própria)

Para converter o sinal de 0 a 1 Volt de saída do sensor em um sinal de 0 a 5 Volts que é a faixa de trabalho do conversor analógico/digital do PIC, o sinal do sensor precisa passar por um amplificador com 5 de ganho. Esta é a função que o amplificador operacional realiza. Como podemos observar na figura 35, o amplificador recebe um sinal de 500mV que o sensor envia e o converte em um sinal de 2,5V, ou seja, amplifica o sinal em 5 vezes. O ganho é determinado pelo resistor R6 e pelo potenciômetro POT2, que se trata de um potenciômetro de precisão multivoltas. Para que a saída seja amplificada 5 vezes o potenciômetro de 10 K Ω precisa estar com 83% de seu valor, ou seja, 8,3 K Ω . Estes valores podem ser comprovados pela fórmula 9, que calcula o valor de saída de um amplificador não inversor, sendo R_f representado pelo resistor R6 e R₁ representado pelo potenciômetro POT2.

$$V_o = \left(1 + \frac{33K}{8,3K} \right) \cdot 0,5 \quad V_o = 2,5 \text{ V}$$

Equação 7 – Cálculo do valor de saída do amplificador

O amplificador operacional está sendo representando por IC4:A, pois o LM324N possui 4 amplificadores em um mesmo encapsulamento. O circuito integrado está sendo alimentado com a tensão de 16Vcc presente na saída do retificador da fonte de alimentação. O diodo D5 possui uma tensão zener de 5,1V, sua função é proteger a entrada analógica do microcontrolador caso a tensão de saída do amplificador exceda 5V, o que poderia causar a queima da entrada analógica. O resistor R7 limita a corrente do diodo zener em caso de sobretensão, além de garantir uma impedância mínima de entrada para o conversor analógico/digital. Segundo Sousa, Souza e Lavinia (2010), esta impedância de entrada deverá ser entre 50Ω e 2,5KΩ e está relacionada com o tempo de adequação do capacitor interno do conversor analógico/digital.

4.4 Eletrônica de potência

O controlador de temperatura desenvolvido neste projeto necessita de uma interface entre o circuito microcontrolado e a carga que irá acionar, uma vez que os terminais de I/O do PIC tem uma capacidade reduzida de fornecimento de corrente. Este interfaceamento será realizado através da eletrônica de potência.

4.4.1 Carga do controlador

A carga a qual se destina acionar é uma manta termoelétrica destinada a aquecimento localizado. É composta por um circuito resistivo que provoca aquecimento através do efeito Joule. Os resistores de aquecimento são constituídos por um fio de liga níquel/cromo bobinado em um cordão de Kevlar e revestido por PVC (policloreto de vinila). Os dados da manta termoelétrica podem ser observados na tabela 9.

MANTA TERMOELÉTRICA	
Tensão de alimentação:	220 Vac
Valor final da resistência:	1622 Ω
Corrente elétrica drenada:	136 mA
Potência dissipada:	30 W

Tabela 9 – Especificações da carga (fonte própria)

4.4.2 Circuito de acionamento da carga

O acionamento da carga é realizado pelo circuito mostrado na figura 36.

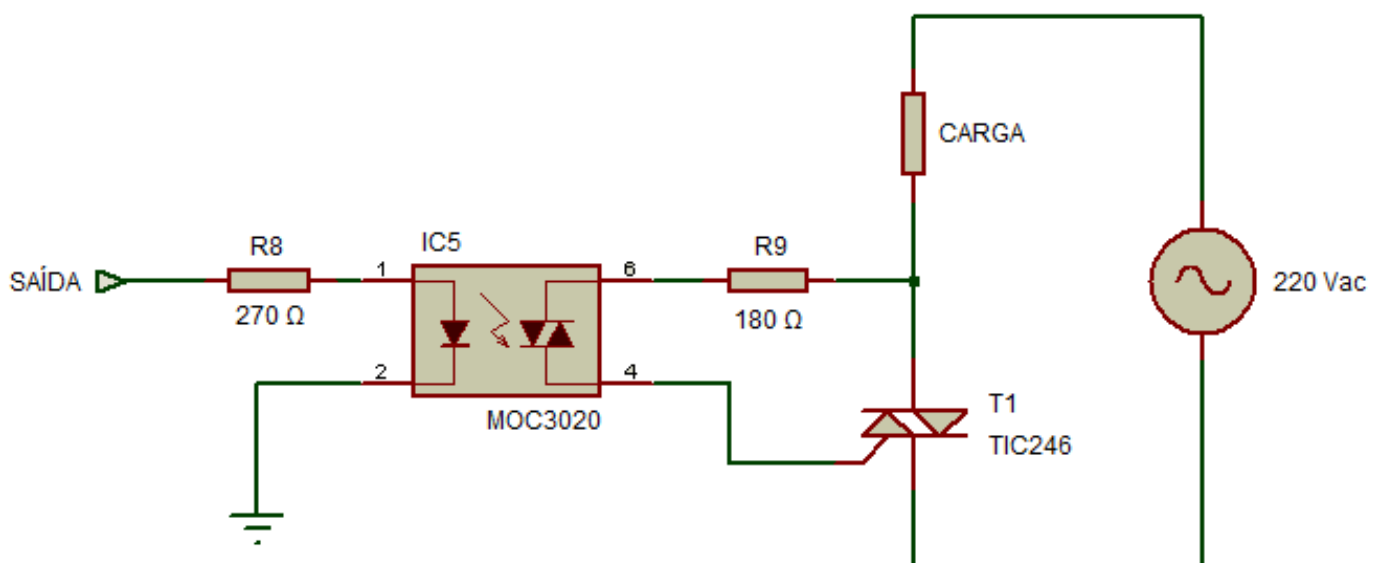


Figura 36 – Circuito de acionamento da carga (fonte própria)

O circuito de acionamento da carga funciona como uma chave analógica eletricamente isolada. No momento em que o microcontrolador aciona o seu terminal “SAÍDA”, o LED do optotriac MOC3020 envia um sinal de luz para o gate de seu triac interno, fazendo com que entre em estado de condução. Com o triac do MOC3020 conduzindo, uma amostra do sinal senoidal da rede é injetada no gate do triac de potência TIC246, fazendo com que este também entre em estado de condução, acionando assim a carga.

O circuito integrado MOC3020 tem a função de realizar o interfaceamento entre o circuito de 5 Vcc do microcontrolador e o circuito de 220 Vac da carga, além de realizar o isolamento elétrico entre os circuitos, protegendo assim o microcontrolador. Entretanto, como seu triac interno possui um limite máximo de corrente de 100 mA, é necessário um segundo estágio no circuito de interface para que a carga seja acionada. Este segundo estágio é realizado pelo triac de potência TIC246, cuja corrente máxima em estado de condução é de 16 Ampères (Datasheet TIC246, 2012).

Segundo Triacs [...] (2012), a tensão de disparo no gate do triac é relativa à tensão presente no terminal MT1. Caso a tensão de disparo seja retirada da tensão do terminal MT1 o triac não entrará em estado de condução. Por este motivo, embora o triac não possua polaridade para condução de corrente, deve-se observar a posição dos terminais MT1 e MT2 para que o disparo seja realizado. Segue na figura 37 a posição dos pinos no triac TIC246 para o encapsulamento TO-220.

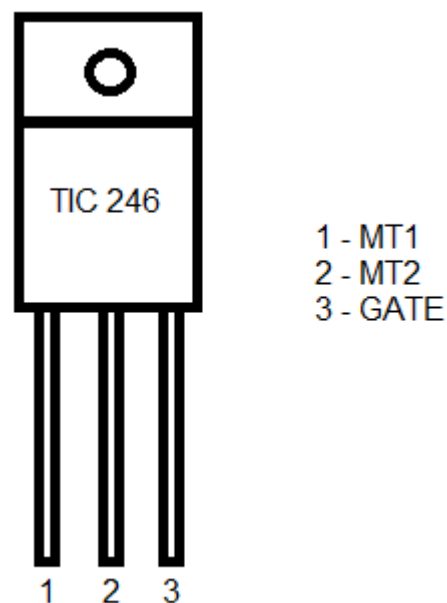


Figura 37 – Pinagem do triac TIC246 [23]

O resistor R8 foi calculado considerando a corrente típica do LED para disparo do optotriac de 15 mA e a tensão típica do LED de 1,2 Volts.

$$R8 = \frac{V_{cc} - V_d}{I_d} \qquad R8 = \frac{5 - 1,2}{0,015} = 253 \, \Omega$$

Equação 8 – Cálculo do resistor para o optoacoplador

Como 253 Ω não é um valor comercial foi adotado o valor de 270 Ω para o resistor R8.

4.5 Supervisório

O projeto conta com um *software* supervisor que simula virtualmente o controlador de temperatura, contando com as mesmas funcionalidades do equipamento físico. Ele permite o ajuste de todos os parâmetros configuráveis do controlador, além de indicar a temperatura medida em tempo real. A tela inicial do *software* supervisor pode ser vista na figura 38.

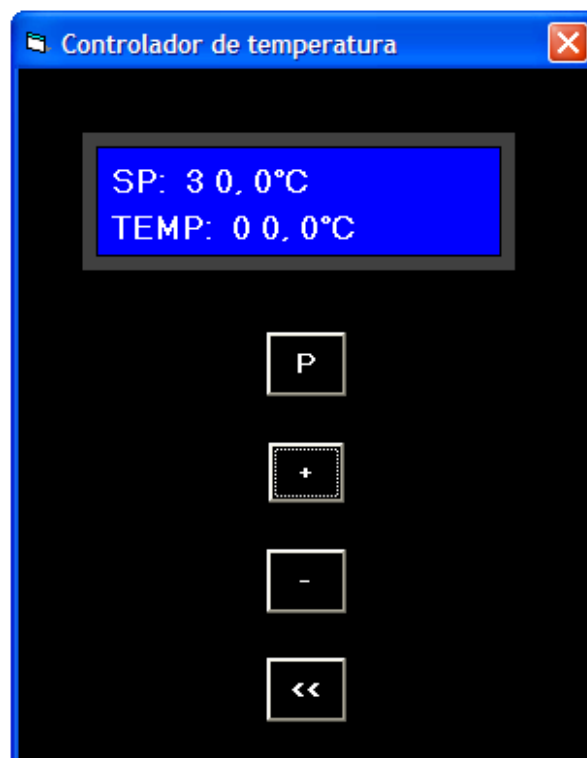


Figura 38 – Interface do software supervisor (fonte própria)

Os botões realizam as mesmas funções que o controlador físico. O botão “P” avança entre as telas de configuração. Os botões “+” e “-” incrementam ou decrementam os parâmetros. O botão “<<” retorna para a tela inicial.

Quando um botão é pressionado no *software* supervisorio, o computador envia um comando para o controlador de temperatura para que a função do botão seja executada nos dois ambientes, físico e virtual. Da mesma forma, quando um botão é pressionado no controlador de temperatura, este envia um comando para o computador para que o comando seja executado no *software*. Dessa forma, todos os parâmetros configuráveis do controlador de temperatura podem ser ajustados em ambos os ambientes, sendo os valores atualizados automaticamente. Segue nas figuras 39, 40 e 41 as demais telas de configuração de parâmetros.



Figura 39 – Ajuste do valor do *set point* (fonte própria)



Figura 40 – Ajuste da temperatura mínima (fonte própria)

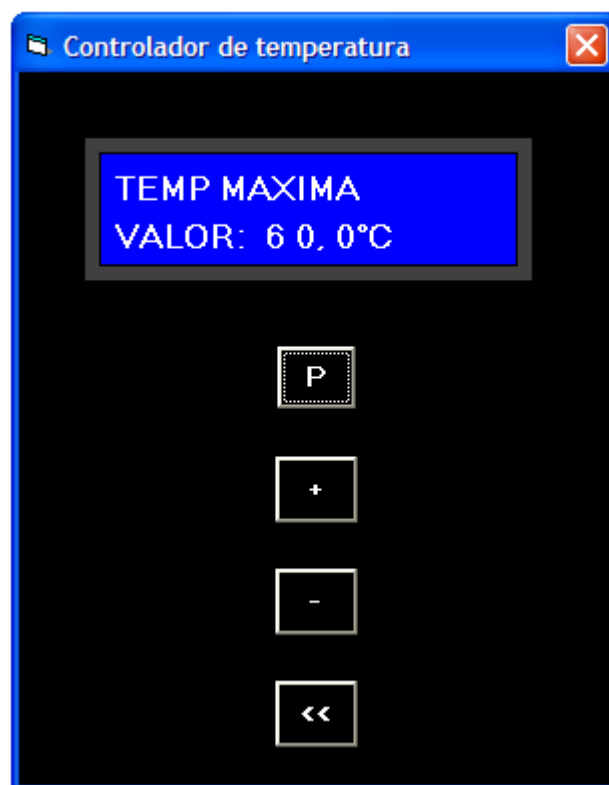


Figura 41 – Ajuste da temperatura máxima (fonte própria)

O *software* supervisor foi desenvolvido utilizando-se a linguagem de programação *Visual Basic*. Esta é uma linguagem de alto nível e orientada a objeto, permitindo a interação visual do usuário com o *software*.

4.5.1 Interface

A comunicação entre o computador e o controlador de temperatura é realizada de forma serial através do padrão RS-232. Para que a comunicação aconteça, tanto o computador quanto o controlador devem seguir o mesmo protocolo, ou seja, alguns parâmetros devem ser respeitados por ambos. Segue na tabela 10 os parâmetros de configuração da comunicação serial utilizada no projeto.

CONFIGURAÇÃO DA PORTA SERIAL	
Tipo de comunicação:	assíncrona
Baud rate:	9600 bps
Bits de dados:	8
Paridade:	sem paridade
Bit de parada:	1
Controle de fluxo:	sem controle de fluxo

Tabela 10 – Configuração da porta serial

Como o padrão RS-232 utiliza valores de tensão distintos do padrão TTL para indicar os níveis lógicos, é necessário realizar a conversão para que possa ser possível a comunicação entre o computador e o controlador de temperatura. Esta tarefa é realizada pelo circuito integrado MAX232. O circuito de conversão pode ser observado na figura 42.

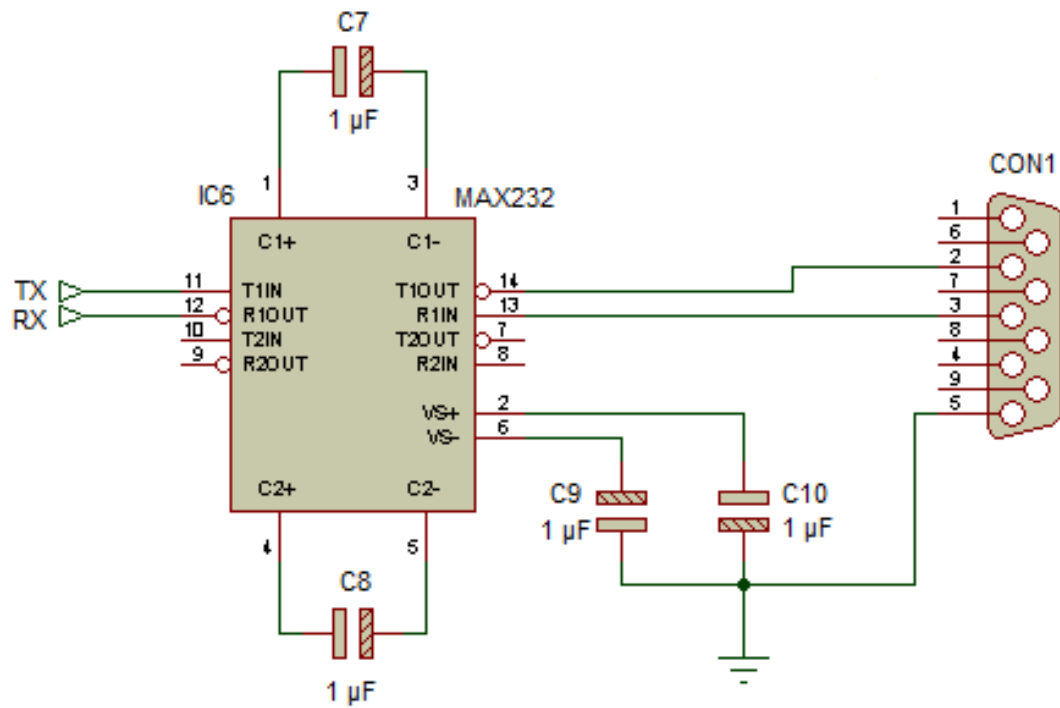


Figura 42 – Conversão RS-232 para TTL (fonte própria)

4.6 Placa de circuito impresso

A placa de circuito impresso do controlador de temperatura foi desenvolvida utilizando-se o software *Eagle* versão 6.3. A figura 43 apresenta o esquema eletrônico completo do projeto.

No esquema, “CON1” é o conector de comunicação com os botões, “CON2” é o conector de comunicação com o *display* LCD, “CON3” é o conector do sensor LM35 e “CON4” é o conector da interface serial. Os pontos “AC1” e “AC2” é para ligação do secundário do transformador e os pontos “L1” e “L2” são para ligação do circuito resistivo da manta termoe-létrica.

A partir do desenho do esquema eletrônico foi realizado o roteamento das trilhas da placa de circuito impresso. Segue na figura 44 o desenho das trilhas e a disposição dos componentes. As trilhas correspondem as linhas azuis. As linhas em vermelho são *jumpers*.

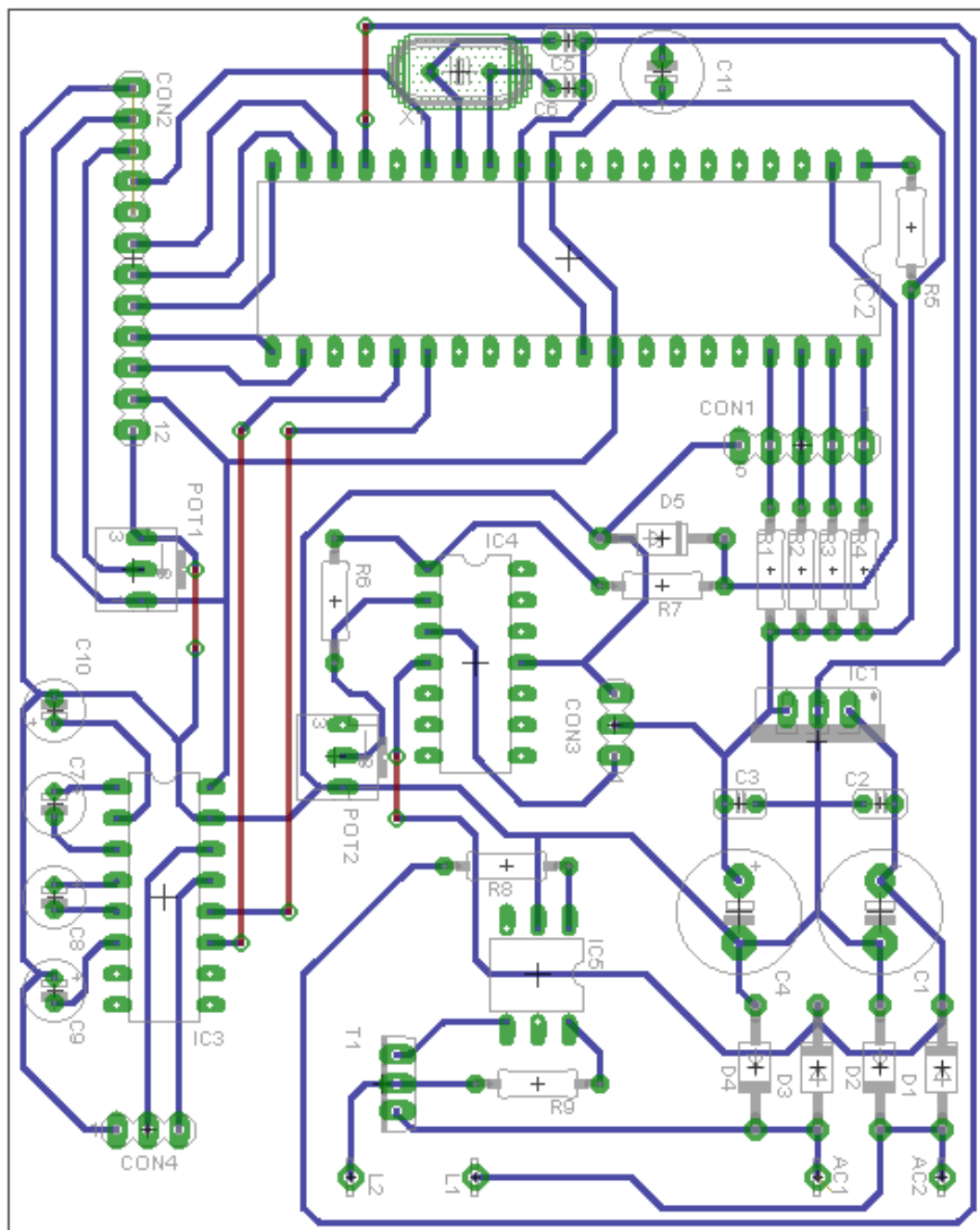


Figura 44 – Roteamento das trilhas de cobre (fonte própria)

5 RESULTADOS

Os testes do controlador de temperatura desenvolvido em conjunto com a manta termoelétrica tiveram bons resultados. Segue nas tabelas 11, 12 e 13 os resultados dos testes realizados para diversas situações diferentes.

CONDIÇÃO INICIAL (SP = 30°C)	
Valor do SP:	30°C
Temperatura máxima medida	31.5°C
Temperatura mínima medida:	29.7°C
Temperatura ambiente:	24.1°C
Tempo para se atingir a temperatura do SP:	2 minutos

Tabela 11 – Teste na condição inicial do controlador

ELEVAÇÃO DO SP (SP = 30°C → SP = 40.5°C)	
Valor do SP:	40.5°C
Temperatura máxima medida	40.9°C
Temperatura mínima medida:	40.2°C
Temperatura ambiente:	24.1°C
Tempo para se atingir a temperatura do SP:	10 minutos

Tabela 12 – Teste de elevação do SP

REDUÇÃO DO SP (SP = 40.5°C → SP = 35.5°C)	
Valor do SP:	35.5°C
Temperatura máxima medida	35.8°C
Temperatura mínima medida:	35.3°C
Temperatura ambiente:	24.1°C
Tempo para se atingir a temperatura do SP:	5 minutos

Tabela 13 – Teste de redução do SP

6 CONCLUSÃO

As propriedades do calor localizado já são amplamente utilizadas nas áreas médica e fisioterápica, sendo seu uso bastante intensivo em determinados tratamentos. Na área doméstica produtos para aquecimento corporal estão se difundindo cada dia mais no mercado brasileiro e já são itens comuns no mercado mundial. Embora haja uma grande variação no tipo de material empregado e na tecnologia utilizada para prover o aquecimento, todos estes produtos possuem como ponto em comum a presença de um controlador de temperatura para gerenciar seu funcionamento.

O desempenho do controlador de temperatura projetado mostrou-se satisfatório quando utilizado em conjunto com as mantas termoelétricas, aplicação esta para a qual foi desenvolvido. As opções de ajuste de seus parâmetros por parte do usuário foram inclusas, sendo imprescindíveis para o correto funcionamento do controlador. O *software* supervisor permite acesso total ao sistema de aquecimento sem a necessidade do operador estar presente no local onde o mesmo está instalado.

Entretanto, o valor da temperatura medida apresentou uma pequena oscilação em relação ao valor configurado no *set point*. Esta variação já era esperada e se deve ao fato de se ter utilizado uma lógica de controle simples, ou seja, o controle *on/off*. Nesta condição, a carga é desligada quando a temperatura ultrapassa o valor selecionado no *set point* e volta a ser acionada quando a temperatura se tornar menor que este valor. Como todo material possui uma inércia térmica, a temperatura não se altera subitamente, precisando de um determinado tempo para se ajustar às novas condições que foram impostas. Desta forma, quando o *set point* é atingido e a alimentação é desligada, a temperatura não para de aumentar instantaneamente, pois é necessário um tempo para que o calor acumulado comece a se dissipar e, assim, o material possa esfriar. O mesmo ocorre quando o valor de temperatura cai abaixo do valor do *set point*. É necessário um tempo para que o calor gerado possa aquecer o material que, neste momento, está perdendo calor para o ambiente. Assim, utilizando-se a lógica de controle *on/off*, não será possível estabilizar o valor medido de temperatura de forma que este não varie em torno do *set point*. Para resolver este problema deve-se utilizar uma lógica de controle que seja mais elaborada. Neste caso, o método de controle PID (proporcional integral derivativo) poderia ser aplicado com o objetivo de estabilizar o valor da temperatura no valor selecionado no *set point*.

A finalidade deste projeto foi desenvolver um controlador de temperatura com características básicas para uma aplicação específica. Evidentemente que diversas outras funcionalidades poderiam ter sido incorporadas ao projeto a fim de torná-lo mais robusto, mas esta não era a principal intenção deste trabalho. O objetivo primordial foi aplicar os conhecimen-

tos adquiridos durante a graduação para dar o primeiro passo no desenvolvimento de um equipamento que, no futuro, possa vir a se tornar um produto vendido comercialmente para um segmento específico do mercado. Neste contexto, o objetivo deste trabalho foi alcançado.

REFERÊNCIA BIBLIOGRÁFICA

- [1] SOUZA, Vitor Amadeu. **Microcontroladores Holtek**. Disponível em: <<http://www.cerneck.com.br/MicrocontroladoresHoltek.pdf>>. Acesso em: 07 jun. 2012.
- [2] SONG, Siang. **RISC – Reduced Instruction Set Computer**. Disponível em: <<http://www.ime.usp.br/~song/mac412/oc-risc.pdf>> Acesso em: 09 jun. 2012.
- [3] ZANCO, Wagner da Silva. **Microcontroladores PIC: Uma abordagem prática e objetiva**. 1. ed. São Paulo: Érica, 2005.
- [4] IDOETA, Ivan Valeije; CAPUANO, Francisco Gabriel. **Elementos de eletrônica digital**. 35. ed. São Paulo: Érica, 2003.
- [5] BORGES, Antônio; SILVA, Gabriel. **Organização de computadores**. Disponível em: <<http://equipe.nce.ufrj.br/gabriel/orgcomp2/capitulo3.pdf>>. Acesso em: 11 jun. 2012.
- [6] MORIBE, Sérgio. **Microcontroladores**. Disponível em: <http://pessoal.utfpr.edu.br/smoribe/download_microcontroladores/3_ARQUITETURA.pdf>. Acesso em: 15 jun. 2012.
- [7] SOUSA, Daniel Rodrigues de; SOUZA, David José de; LAVINIA, Nicolás César. **Desbravando o microcontrolador PIC18: Recursos avançados**. 1. ed. São Paulo: Érica, 2010.
- [8] VASCONCELOS, Laércio. **Fonte de alimentação linear**. Disponível em: <<http://www.laercio.com.br/artigos/hardware/hard-052/hard-052d.htm>>. Acesso em: 20 jun. 2012.
- [9] MARQUES, Angelo Eduardo; CRUZ, Eduardo Cesar Alves; JÚNIOR, Salomão Choueri. **Dispositivos semicondutores: Diodos e transistores**. 8. ed. São Paulo: Érica, 2002.
- [10] SOUZA, Ivair José. **Circuitos retificadores**. Disponível em: <http://ivairsouza.com/circuitos_retificadores.pdf>. Acesso em: 23 jun. 2012.
- [11] BOYLESTAD, Robert; NASHELSKY, Louis. **Dispositivos eletrônicos e teoria de circuitos**. 8. ed. São Paulo: Pearson, 2004.

[12] **Datasheet 7508**. Disponível em: <<http://www.alldatasheet.com/datasheet-pdf/pdf/69437/KEC/7805.html>>. Acesso em: 28 jun. 2012.

[13] Microchip Technology Inc. **Datasheet PIC 18F4520**. Disponível em: <<http://ww1.microchip.com/downloads/en/DeviceDoc/39631E.pdf>>. Acesso em: 07 jun. 2012.

[14] **Datasheet 1N4007**. Disponível em: <<http://www.alldatasheet.com/datasheet-pdf/pdf/14624/PANJIT/1N4007.html>>. Acesso em: 23 jun. 2012.

[15] BARBACENA, Ilton; FLEURY, Cláudio Afonso. **Display LCD**. Disponível em: <http://www2.eletronica.org/apostilas-e-ebooks/componentes/LCD_30324b.pdf>. Acesso em: 01 jul. 2012.

[16] **Serial port using visual basic .NET and windows**. Disponível em: <<http://www.me.umn.edu/courses/me2011/smartprodcourse/technotes/docs/serial-port-vb.pdf>>. Acesso em: 07 jul. 2012.

[17] **Datasheet LCD**. Disponível em: <http://www.winstar.com.tw/products_detail_ov.php?lang=pt&ProID=28>. Acesso em: 01 jul. 2012.

[18] BALBINOT, Alexandre; BRUSAMARELLO, Valner João. **Instrumentação e fundamentos de medidas**. 1. ed. São Paulo: LTC, v. 1, 2006.

[19] **Datasheet LM35**. Disponível em: <<http://www.alldatasheet.com/datasheet-pdf/pdf/8866/NSC/LM35.html>>. Acesso em: 10 jul. 2012.

[20] **Tiristor Triac**. Disponível em: <<http://www.electronica-pt.com/index.php/content/view/131/37/>>. Acesso em: 11 ago. 2012.

[21] **Datasheet MOC3020**. Disponível em: <<http://www.alldatasheet.com/datasheet-pdf/pdf/5038/MOTOROLA/MOC3020.html>>. Acesso em: 11 ago. 2012.

[22] **MOC3020**. Disponível em: <<http://www.projetostecnologicos.com/Componentes/Optos/Optoacopladores/MOC3020/MOC3020.html>>. Acesso em: 17 ago. 2012.

[23] **Datasheet TIC246**. Disponível em: < <http://www.alldatasheet.com/datasheet-pdf/pdf/20140/POINN/TIC246.html>>. Acesso em: 3 set. 2012.

[24] **TRIACs e SCRs**. Disponível em: <http://www.centelhas.com.br/biblioteca/triacs_e_scrs.pdf>. Acesso em: 23 ago. 2012.

[25] **Datasheet LM324**. Disponível em: < <http://www.alldatasheet.com/datasheet-pdf/pdf/17871/PHILIPS/LM324.html>>. Acesso em: 12 jul. 2012.

[26] ZAIDAN, Evelise. **Lesões, dores e câibras**: quando aplicar gelo ou calor? Disponível em: <<http://revistacontrarelogio.com.br/materia/lesoes-dores-e-caibras-quando-aplicar-gelo-ou-calor/>>. Acesso em: 01 out. 2012.

[27] SILVA, Ana Paula Gonçalves; SALVADOR, Marcelo. **O que são sistemas supervisórios?** Disponível em: < http://www.wectrus.com.br/artigos/sist_superv.pdf>. Acesso em: 13 out. 2012.

[28] **COMUNICAÇÃO serial RS232**. Disponível em: <http://www.cerne-tec.com.br/Artigo_07_ComunicacaoSerial.pdf>. Acesso em: 20 out. 2012.

[29] **RS-232**. Disponível em: <<http://ldev.wordpress.com/2009/04/18/rs-232/>>. Acesso em: 20 out. 2012.

[30] **Datasheet MAX232**. Disponível em: <<http://www.alldatasheet.com/datasheet-pdf/pdf/207380/TI/MAX232.html>>. Acesso em: 20 out. 2012.

ANEXOS

ANEXO A – Código fonte do microcontrolador

```

*****
;
;
;          TRABALHO DE CONCLUSÃO DE CURSO          *
;    PROJETO DE CONTROLADOR DE TEMPERATURA PARA AQUECIMENTO *
;          DE MANTA TERMOELÉTRICA                  *
;
;
;AUTOR: RAFAEL DE CASTRO ROCHA                      *
;DATA: 20/02/2012                                   *
;
*****

*****
;
;          CONTROLE DO DISPLAY          *
;
*****
;FUNC = 00000001 (TELA INICIAL)
;FUNC = 00000010 (ALTERAÇÃO DO SP)
;FUNC = 00000100 (ALTERAÇÃO TEMPERATURA MÁXIMA)
;FUNC = 00001000 (ALTERAÇÃO TEMPERATURA MÍNIMA)
;
*****

*****
;
;          ARQUIVO DE DEFINICOES          *
;
*****
#include <P18F4520.INC>
;
*****

*****
;
;    CONFIGURACAO DO MICROCONTROLADOR    *
;
*****

CONFIG OSC = XT, FCMEN = OFF, IESO = OFF, PWRT = ON, BOREN = ON, BORV = 0
CONFIG WDT = OFF, WDTPS = 128, MCLRE = ON, LPT1OSC = OFF, PBADEN = OFF
CONFIG CCP2MX = PORTC, STVREN = ON, LVP = OFF, DEBUG = OFF, XINST = OFF
CONFIG CP0 = OFF, CP1 = OFF, CP2 = OFF, CP3 = OFF, CPB = OFF, CPD = OFF
CONFIG WRT0 = OFF, WRT1 = OFF, WRT2 = OFF, WRT3 = OFF, WRTB = OFF
CONFIG WRTC = OFF, WRTD = OFF, EBTR0 = OFF, EBTR1 = OFF, EBTR2 = OFF
CONFIG EBTR3 = OFF, EBTRB = OFF

*****
;
;          VARIÁVEIS          *
;
*****

CBLOCK          0X00          ;ENDERECO INICIAL DA MEMORIA DO USUARIO (VARIÁVEIS)

```

```

SP_NUM_H           ;VALOR DA TEMPERATURA DO SP CONVERTIDO EM NÚMERO
SP_NUM_L           ;VALOR DA TEMPERATURA DO SP CONVERTIDO EM NÚMERO
TEMPO_MSEG         ;VARIÁVEL AUXILIAR DA ROTINA DE DELAY
TEMPO_SEG          ;VARIÁVEL AUXILIAR DA ROTINA DE DELAY
SEGUNDO            ;VARIÁVEL AUXILIAR DA ROTINA DE DELAY
W_TEMP             ;BACKUP DO REGISTRO W NA CHAMADA DE INTERRUPÇÃO
STATUS_TEMP        ;BACKUP DO REGISTRO STATUS NA CHAMADA DE INTERRUPÇÃO
BSR_TEMP           ;BACKUP DO REGISTRO BSR NA CHAMADA DE INTERRUPÇÃO
DADO_LCD           ;VARIÁVEL AUXILIAR DA ROTINA LCD_CHAR
SP_DEZENA          ;VALOR DA DEZENA DO SP
SP_UNIDADE         ;VALOR DA UNIDADE DO SP
SP_DECIMO          ;VALOR DO DECIMO DO SP
CONVERT_CHAR       ;VARIÁVEL AUXILIAR DA ROTINA CONVERTER
FUNC              ;INDICA EM QUAL TELA O DISPLAY ESTÁ
TMAX_DEZENA        ;VALOR DA DEZENA DA TEMPERATURA MÁXIMA
TMAX_UNIDADE       ;VALOR DA UNIDADE DA TEMPERATURA MÁXIMA
TMAX_DECIMO        ;VALOR DO DÉCIMO DA TEMPERATURA MÁXIMA
TMIN_DEZENA        ;VALOR DA DEZENA DA TEMPERATURA MÍNIMA
TMIN_UNIDADE       ;VALOR DA UNIDADE DA TEMPERATURA MÍNIMA
TMIN_DECIMO        ;VALOR DO DÉCIMO DA TEMPERATURA MÍNIMA
TMAX_INTEIRO       ;VALOR DA TEMPERATURA MÁXIMA (PARTE INTEIRA) DO SP CONVERTIDO EM
                   ;NÚMERO
TMIN_INTEIRO       ;VALOR DA TEMPERATURA MÍNIMA (PARTE INTEIRA) DO SP CONVERTIDO EM
                   ;NÚMERO

AUX1               ;VARIÁVEL AUXILIAR NA CONVERSÃO DO VALOR LIDO DE TEMPERATURA
AUX2               ;VARIÁVEL AUXILIAR NA CONVERSÃO DO VALOR LIDO DE TEMPERATURA
CONT               ;VARIÁVEL QUE RECEBE O RESULTADO DA DIVISÃO NA CONVERSÃO DO
                   ;VALOR LIDO ;DE TEMPERATURA

TEMP_DEZENA        ;VALOR DA DEZENA DA TEMPERATURA
TEMP_UNIDADE       ;VALOR DA UNIDADE DA TEMPERATURA
TEMP_DECIMO        ;VALOR DO DÉCIMO DA TEMPERATURA
CONTADOR           ;VARIÁVEL AUXILIAR DA ROTINA DE MÉDIA
MEDIA_L            ;VARIÁVEL AUXILIAR DA ROTINA DE MÉDIA
MEDIA_H            ;VARIÁVEL AUXILIAR DA ROTINA DE MÉDIA
MEDIA_AUX_L        ;VARIÁVEL AUXILIAR DA ROTINA DE MÉDIA
MEDIA_AUX_H        ;VARIÁVEL AUXILIAR DA ROTINA DE MÉDIA
VALOR_L            ;VARIÁVEL AUXILIAR DA ROTINA DE MÉDIA
VALOR_H            ;VARIÁVEL AUXILIAR DA ROTINA DE MÉDIA
CONTROLE           ;VARIÁVEL AUXILIAR DA ROTINA DE MÉDIA
DADO_RX            ;DADO RECEBIDO DO MÓDULO USART
DADO_TX            ;DADO TRANSMITIDO DO MÓDULO USART
DEZENA_CHAR        ;CÓDIGO ASCII DA DEZENA DO SP
UNIDADE_CHAR       ;CÓDIGO ASCII DA UNIDADE DO SP
DECIMO_CHAR        ;CÓDIGO ASCII DO DÉCIMO DO SP

ENDC

```

1

```

*****
1
*
1          ENTRADAS          *
*****

#DEFINE      BOTAO_MAIS      PORTB,4      ;BOTÃO '+'
#DEFINE      BOTAO_MENOS     PORTB,5      ;BOTÃO '-'
#DEFINE      BOTAO_VOLTA     PORTB,6      ;BOTÃO 'VOLTAR'
#DEFINE      BOTAO_P         PORTB,7      ;BOTÃO 'P'
*****
1

*****
1
*
1          SAIDAS            *
*****

#DEFINE      DADO            PORTD        ;COMUNICAÇÃO DE DADOS COM DISPLAY LCD
#DEFINE      RS              LATC,0       ;REGISTER SELECT DO DISPLAY LCD (DADO/COMANDO)
#DEFINE      SAIDA           LATC,2       ;SAÍDA PARA ACIONAMENTO DA CARGA
#DEFINE      ENABLE          LATC,3       ;CLOCK DO DISPLAY LCD
*****
1

*****
1
*
1          VETOR DE RESET    *
*****

ORG          0X00            ;ENDERECO INICIAL DE RESET
BRA          INICIO

*****
1

*****
1
*
1          INTERRUPCAO      *
*****

ORG          0X18            ;ENDERECO INICIAL DE INTERRUPCAO

MOVWF W_TEMP                ;SALVA W
MOVFF STATUS,STATUS_TEMP    ;SALVA STATUS
MOVFF BSR,BSR_TEMP           ;SALVA BSR

***** IDENTIFICA INTERRUPCAO *****
1

BTFSK INTCON,RBIF            ;TESTA INT DO PORTB
BRA INT_BOTAO                ;SE OCORREU, EXECUTA ROTINA

BTFSK PIR1,ADIF              ;TESTA INT DO CONVERSOR A/D
BRA INT_AD                   ;SE OCORREU, EXECUTA ROTINA

BTFSK PIR1,RCIF              ;TESTA INT DO MÓDULO USART
BRA INT_USART                ;SE OCORREU, EXECUTA ROTINA

```

```

        BRA     END_INT                ;SENAO SAI DA ROTINA DE INTERRUPTAO

;***** ROTINA DE INT EM PORTB *****
,

INT_BOTAO

        BCF     INTCON,RBIF            ;LIMPA FLAG DE INT NO PORTB

        MOVLW   .50                    ;DEBOUNCE
        MOVWF   TEMPO_MSEG             ;GERA DELAY DE 50 MS
        CALL    DELAY_MSEG

        BTFSS   BOTAO_MAI
        BRA     BOT_MAI

        BTFSS   BOTAO_MENOS           ;TESTA OS 4 BOTOES
        BRA     BOT_MENOS

        BTFSS   BOTAO_VOLTA
        BRA     BOT_VOLTA

        BTFSS   BOTAO_P
        BRA     BOT_P

        BRA     END_INT_DIRETO         ;SE NENHUM DELES ESTIVER ACIONADO, SAI DA ROTINA DE INTE
                                        ;RUPÇÃO SEM ATUALIZAR A TELA

;*****
,

BOT_MAI

        CALL    FUNC_MAI              ;CHAMA ROTINA DO BOTÃO

        MOVLW   'C'                    ;INFORMA AO SUPERVISÓRIO QUE O BOTÃO FOI ACIONADO
        BTFSS   TXSTA,TRMT
        BRA     $.2
        MOVWF   TXREG
        BRA     END_INT               ;VAI PARA O FIM DA ROTINA DE INT

BOT_MENOS

        CALL    FUNC_MENOS            ;CHAMA ROTINA DO BOTÃO

        MOVLW   'D'                    ;INFORMA AO SUPERVISÓRIO QUE O BOTÃO FOI ACIONADO
        BTFSS   TXSTA,TRMT
        BRA     $.2
        MOVWF   TXREG
        BRA     END_INT               ;VAI PARA O FIM DA ROTINA DE INT

```

BOT_VOLTA

```

CALL    FUNC_VOLTA    ;CHAMA ROTINA DO BOTÃO

MOVLW   'E'           ;INFORMA AO SUPERVISÓRIO QUE O BOTÃO FOI ACIONADO
BTFSS   TXSTA,TRMT
BRA     $.2
MOVWF   TXREG
BRA     END_INT       ;VAI PARA O FIM DA ROTINA DE INT

```

BOT_P

```

CALL    FUNC_P        ;CHAMA ROTINA DO BOTÃO

MOVLW   'B'           ;INFORMA AO SUPERVISÓRIO QUE O BOTÃO FOI ACIONADO
BTFSS   TXSTA,TRMT
BRA     $.2
MOVWF   TXREG
BRA     END_INT       ;VAI PARA O FIM DA ROTINA DE INT

```

```

;*****
;

```

```

;***** ROTINA DE INT DO CONVERSADOR AD *****
;

```

INT_AD

```

CALL    CALC_MEDIA    ;CHAMA ROTINA DE CONVERSÃO DO VALOR LIDO

BCF     PIR1,ADIF     ;LIMPA FLAG DE INT DO CONVERSADOR A/D

BRA     END_INT_DIRETO ;VAI PARA O FIM DA ROTINA DE INT SEM ATUALIZAR TELA

```

```

;*****
;

```

```

;***** ROTINA DE INT DO CONVERSADOR AD *****
;

```

INT_USART

```

MOVFF   RCREG,DADO_RX ;SALVA DADO RECEBIDO DA USART

BCF     PIR1,RCIF     ;LIMPA FLAG DE INT DO CONVERSADOR A/D

MOVLW   'B'           ;QUANDO RECEBER O CARACTER "B"
BCF     STATUS,Z      ;FAZ A FUNÇÃO DO BOTÃO "P"
XORWF   DADO_RX,W
BTFSC   STATUS,Z
CALL    FUNC_P

```

```

    MOVLW 'C'                ;QUANDO RECEBER O CARACTER "C"
    BCF    STATUS,Z          ;FAZ A FUNÇÃO DO BOTÃO "MAIS"
    XORWF  DADO_RX,W
    BTFSC  STATUS,Z
    CALL   FUNC_MAIS

    MOVLW 'D'                ;QUANDO RECEBER O CARACTER "D"
    BCF    STATUS,Z          ;FAZ A FUNÇÃO DO BOTÃO "MENOS"
    XORWF  DADO_RX,W
    BTFSC  STATUS,Z
    CALL   FUNC_MENOS

    MOVLW 'E'                ;QUANDO RECEBER O CARACTER "E"
    BCF    STATUS,Z          ;FAZ A FUNÇÃO DO BOTÃO "VOLTA"
    XORWF  DADO_RX,W
    BTFSC  STATUS,Z
    CALL   FUNC_VOLTA

    BRA    END_INT           ;VAI PARA O FIM DA ROTINA DE INT

;*****
,

;***** FIM DO TRATAMENTO DE INTERRUPÇÃO *****

END_INT

;***** ATUALIZAÇÃO DE TELA *****

    BTFSC  FUNC,0            ;ATUALIZA TELAS NO DISPLAY
    CALL   TELA_INICIAL
    BTFSC  FUNC,1
    CALL   TELA_SP
    BTFSC  FUNC,2
    CALL   TELA_TEMP_MAX
    BTFSC  FUNC,3
    CALL   TELA_TEMP_MIN

    MOVWF  .250
    CALL   DELAY_MSEG

    BTFSS  BOTAO_MAIS        ;TESTA PARA VER SE O BOTÃO ESTÁ AINDA PRESSIONADO
    BRA    BOT_MAIS

    BTFSS  BOTAO_MENOS
    BRA    BOT_MENOS

END_INT_DIRETO

```



```
***** RESTAURA REGISTROS *****
```

```
MOVFF BSR_TEMP,BSR      ;RECUPERA REGISTRO BSR
MOVFF W_TEMP,W          ;RECUPERA REGISTRO W
MOVFF STATUS_TEMP,STATUS ;RECUPERA REGISTRO STATUS
```

```
*****
```

```
RETIE      ;RETORNA DA CHAMADA DE INTERRUPTÃO
```

```
*****
```

```
*****
```

```
*          INICIO          *
```

```
*****
```

INICIO

```
*****
```

```
*          INICIALIZACAO DOS PORTS          *
```

```
*****
```

```
CLRF PORTA      ; GARANTE TODAS AS SAÍDAS EM ZERO
CLRF PORTB
CLRF PORTC
CLRF PORTD
CLRF PORTE
CLRF LATA
CLRF LATB
CLRF LATC
CLRF LATD
CLRF LATE
```

```
MOVLW B'11111111'
MOVWF TRISA      ; CONFIGURA I/O DO PORTA
```

```
MOVLW B'11111111'
MOVWF TRISB      ; CONFIGURA I/O DO PORTB
```

```
MOVLW B'11110010'
MOVWF TRISC      ; CONFIGURA I/O DO PORTC
```

```
MOVLW B'11110000'
MOVWF TRISD      ; CONFIGURA I/O DO PORTD
```

```
MOVLW B'00000111'
MOVWF TRISE      ; CONFIGURA I/O DO PORTE
```

```
*****
```

```

*****
1
;
; INICIALIZACAO DAS VARIÁVEIS *
*****
1

    CLRF    DELAY_MSEG
    CLRF    DELAY_SEG
    CLRF    SEGUNDO
    CLRF    W_TEMP           ;ZERA VARIÁVEIS
    CLRF    STATUS_TEMP
    CLRF    BSR_TEMP
    CLRF    DADO_LCD
    CLRF    CONVERT_CHAR
    CLRF    CONTROLE

    CLRF    TEMP_DEZENA
    CLRF    TEMP_UNIDADE     ;INICIA VARIÁVEIS DE TEMPERATURA MEDIDA PELO SENSOR
    CLRF    TEMP_DECIMO

    MOVLW   .3               ;INICIA SP COM 30,0°
    MOVWF   SP_DEZENA
    MOVLW   .0
    MOVWF   SP_UNIDADE
    MOVLW   .0
    MOVWF   SP_DECIMO

    MOVLW   .6               ;INICIA TEMPERATURA MÁXIMA COM 60,0°
    MOVWF   TMAX_DEZENA
    MOVLW   .0
    MOVWF   TMAX_UNIDADE
    MOVLW   .0
    MOVWF   TMAX_DECIMO

    MOVLW   .1               ;INICIA TEMPERATURA MÍNIMA COM 10,0°
    MOVWF   TMIN_DEZENA
    MOVLW   .0
    MOVWF   TMIN_UNIDADE
    MOVLW   .0
    MOVWF   TMIN_DECIMO

    CALL    CONVERTE_SP      ;CONVERTE VARIÁVEIS EM NÚMERO
    CALL    CONVERTE_TMAX
    CALL    CONVERTE_TMIN

    MOVLW   .1               ;FUNC INDICA TELA INICIAL
    MOVWF   FUNC

*****
1

```

```

;*****
;
; CONFIGURACAO DOS REGISTRADORES DE PROPOSITOS ESPECIAIS *
;*****
;

```

```

MOV LW B'11000010' ;CONFIGURA TIMER0 (TIMER0 COM 8 BITS E PRESCALER DE 1:8)
MOV WF T0CON

MOV LW B'11001000' ;HABILITA INTERRUPTOES GERAIS,HABILITA INTERRUPTAO DOS PERIFERICOS
MOV WF INTCON ;E HABILITA INTERRUPTAO POR MUDANCA DE NIVEL NO PORTB

MOV LW B'10000000' ;DESABILITA RESISTORES DE PULL-UP NO PORTB E SET ALTA PRIORIDADE
MOV WF INTCON2 ;PARA INTERRUPTAO POR MUDANCA DE NIVEL NO PORTB

MOV LW B'01100000' ;HABILITA INTERRUPTAO DO CONVERSOR A/D E DO MODULO DE RECEPCAO
MOV WF PIE1 ;DA USART

MOV LW B'10101101' ;RESULTADO JUSTIFICADO A DIREITA, TEMPO DE CONVERSÃO DE 12 TAD E
MOV WF ADCON2 ; FREQUENCIA DE FOSC/16 (TAD = 4US)

MOV LW B'00001110' ;HABILITA ENTRADA ANALÓGICA PARA AN0 (DEMAIS I/O) E REFERÊNCIA DE
MOV WF ADCON1 ;TENSÃO VSS E VDD

MOV LW B'00000000' ;DESLIGA O CONVERSOR A/D E SELECIONA O CANAL 0
MOV WF ADCON0

MOV LW B'00100100' ;TRANSMISSÃO ASSÍNCRONA DE 8 BITS COM ALTA VELOCIDADE
MOV WF TXSTA

MOV LW .25 ;BAUND RATE DE 9600 BPS
MOV WF SPBRG

MOV LW B'10010000' ;HABILITA PORTA SERIAL E RECEPCAO ASSÍNCRONA DE 8 BITS
MOV WF RCSTA

MOV LW B'00000000' ;DESABILITA AUTOBAUD, WAKE-UP E GERADOR DE BAUD RATE DE 8 BITS
MOV WF BAUDCON

```

```

;*****
;

```

```

;*****
;
; INICIALIZAR DISPLAY LCD *
;*****
;

```

```

MOV LW .50
CALL DELAY_MSEG

MOV LW B'00110011' ;INICIALIZAÇÃO DO DISPLAY
CALL LCD_COM

MOV LW B'00110010' ;INICIALIZAÇÃO DO DISPLAY

```

```

CALL    LCD_COM

MOVLW   B'00101000'    ;CONFIGURA LCD PARA COMUNICAÇÃO 4 BITS, 2 LINHAS E 5X7 PIXELS POR
CALL    LCD_COM        ; CARACTERE

MOVLW   B'00001000'    ;DESLIGA DISPLAY, DELIGA CURSOR
CALL    LCD_COM

MOVLW   B'00000001'    ;LIMPA O DISPLAY
CALL    LCD_COM

MOVLW   B'00000110'    ;CONFIGURA DESLOCAMENTO AUTOMÁTICO DA MENSAGEM PARA A DIREITA
CALL    LCD_COM

MOVLW   B'00000011'    ;RETORNA CURSOR PARA A LINHA 1 COLUNA 1
CALL    LCD_COM

MOVLW   B'00001100'    ;LIGA DISPLAY, DELIGA CURSOR
CALL    LCD_COM

;*****
;

;*****
;
;          ROTINA PRINCIPAL          *
;*****
;

CALL    CHAR_GRAUS    ;CRIA CARACTERE 'º' NA CGRAM
CALL    TELA_INICIAL  ;INICIA NA TELA INICIAL

BSF     ADCON0,ADON    ;LIGA CONVERSOR A/D

LOOP

;LEITURA DO VALOR DE TEMPERATURA

BSF     ADCON0,ADON    ;LIGA O CONVERSOR A/D
BSF     ADCON0,GO

TESTE2

BTFSC   ADCON0,GO      ;AGUARDA LEITURA TERMINAR
BRA     TESTE2

;COMPARA O VALOR DA TEMPERATURA LIDA COM O VALOR DO SP

MOVF    SP_DEZENA,W
CPFSGT  TEMP_DEZENA
BRA     TESTE_D_M
BRA     DESLIGA

```

TESTE_D_M

```
    MOVF    SP_DEZENA,W
    CPFSLT  TEMP_DEZENA
    BRA     TESTE_U
    BRA     LIGA
```

TESTE_U

```
    MOVF    SP_UNIDADE,W
    CPFSGT  TEMP_UNIDADE
    BRA     TESTE_U_M
    BRA     DESLIGA
```

TESTE_U_M

```
    MOVF    SP_UNIDADE,W
    CPFSLT  TEMP_UNIDADE
    BRA     TESTE_DIG
    BRA     LIGA
```

TESTE_DIG

```
    MOVF    SP_DECIMO,W
    CPFSGT  TEMP_DECIMO
    BRA     LIGA
    BRA     DESLIGA
```

;SE TEMP LIDA < TEMP SP

;LIGA SAIDA

LIGA

```
    BSF     SAIDA
    BRA     PULA
```

;SE TEMP LIDA > TEMP SP

;DESLIGA SAIDA

DESLIGA

```
    BCF     SAIDA
```

PULA

```
    BCF     ADCON0,ADON          DESLIGA CONVERSOR A/D
```

;LOOP DA LEITURA DE TEMP

```
    BRA     LOOP
```

```
.******
;
```

```

;*****
;
; SUB ROTINAS *
;*****
;*****

;INCLUI ARQUIVO CONTENDO AS SUB ROTINAS DE ACESSO AO DISPLAY LCD
#INCLUDE <LCD.INC>

;INCLUI ARQUIVO CONTENDO AS SUB ROTINAS DE TRATAMENTO DE VALORES ANALÓGICOS
#INCLUDE <ADC.INC>

;INCLUI ARQUIVO CONTENDO AS SUB ROTINAS DE TRATAMENTO DOS BOTÕES
#INCLUDE <BOTOES.INC>

;INCLUI ARQUIVO CONTENDO AS SUB ROTINAS DE DELAY
#INCLUDE <DELAY.INC>

;*****
;*****

;*****
;
; FIM DO PROGRAMA *
;*****
;*****

END

;*****
;*****

;*****
;
; * *
; * ROTINAS DE CONVERSÃO *
; * *
; * *
;*****
;*****

;***** VALOR ANALOGICO MEDIO *****
;*****

;ROTINA CALCULA O VALOR MEDIO DAS ULTIMAS 60 ENTRADAS ANALOGICAS

CALC_MEDIA

MOV LW .60 ;INICIA VARIÁVEIS
MOV WF CONTADOR

MOV LW .0
MOV WF MEDIA_L
MOV WF MEDIA_H
MOV WF MEDIA_AUX_L
MOV WF MEDIA_AUX_H

```

LOOP_MEDIA

```
BSF    ADCON0,ADON    ;LIGA O CONVERSADOR A/D
```

```
BSF    ADCON0,GO      ;INICIA CONVERSÃO
```

TESTE

```
BTFSC  ADCON0,GO      ;AGUARDA A LEITURA TERMINAR
BRA     TESTE
```

```
BCF    ADCON0,ADON    ;DESLIGA O CONVERSADOR A/D
```

```
MOVFF  ADRESL,VALOR_L  ;SALVA VALORES LIDOS
MOVFF  ADRESH,VALOR_H
```

```
MOVF   VALOR_L,W
BCF    STATUS,C
ADDWF  MEDIA_AUX_L,F
BTFSC  STATUS,C
INCF   MEDIA_AUX_H,F   ;SOMA 60 LEITURAS
MOVF   VALOR_H,W
ADDWF  MEDIA_AUX_H,F
DECFSZ CONTADOR,F
BRA    LOOP_MEDIA
```

```
MOVLW  .0
BCF    STATUS,Z
XORWF  MEDIA_AUX_H,W   ;VERIFICA SE VALOR DA SOMA É MENOR QUE 256
BTFSS  STATUS,Z
BRA    DIV
```

VOLTA1

```
MOVLW  .60
BCF    STATUS,C
SUBWF  MEDIA_AUX_L,F   ;SE SIM, DIVIDE APENAS A VARIÁVEL MEDIA_AUX_L POR 60
BTFSS  STATUS,C
BRA    FIM_MEDIA
BCF    STATUS,C
INCF   MEDIA_L,F
BRA    VOLTA1
```

DIV ;SE NÃO, DIVIDE AS VARIÁVEIS MEDIA_AUX_L E MEDIA_AUX_H POR 60

```
MOVLW  .60
BCF    STATUS,C
SUBWF  MEDIA_AUX_L,F
BTFSS  STATUS,C
DECFSZ MEDIA_AUX_H,F
BRA    DIV3
```

DIV2

```

BCF     STATUS,Z
INCF    MEDIA_L,F
BTFSC   STATUS,Z
INCF    MEDIA_H,F
MOVLW   .60
BCF     STATUS,C
SUBWF   MEDIA_AUX_L,F
BTFSS   STATUS,C
BRA     FIM_MEDIA
BRA     DIV2

```

DIV3

```

BCF     STATUS,Z
INCF    MEDIA_L,F
BTFSC   STATUS,Z
INCF    MEDIA_H,F
BRA     DIV

```

;VARIÁVEIS MEDIA_H E MEDIA_L CONTEM O VALOR DA MÉDIA

FIM_MEDIA

```

CALL    CONVERTE_SP    ;FAZ SP COMO UM NÚMERO ÚNICO

CALL    CONVERT        ;CONVERTE O VALOR DA MÉDIA EM DÍGITOS PARA O LCD

RETURN

```

,

***** CONVERSÃO DO VALOR DA TEMPERATURA *****
,

CONVERT

```

CLRF    TEMP_DEZENA            ;INICIA VARIÁVEIS
CLRF    TEMP_UNIDADE
CLRF    TEMP_DECIMO

MOVFF   MEDIA_L, MEDIA_AUX_L
MOVFF   MEDIA_H, MEDIA_AUX_H

MOVLW   .0
BCF     STATUS,Z
XORWF   MEDIA_AUX_H,W          ;VERIFICA SE VALOR DA SOMA É MENOR QUE 256
BTFSS   STATUS,Z
BRA     DIV_DEZENA1

```



```

MOVLW .100
BCF    STATUS,Z
XORWF  MEDIA_AUX_L,W      ;VERIFICA SE VALOR DA SOMA É MENOR QUE 256
BTFSC  STATUS,Z
BRA    CONV_UNIDADE       ;SE SIM, DEZENA = 0

```

VOLTA2

```

MOVLW .100
BCF    STATUS,C
SUBWF  MEDIA_AUX_L,F      ;SE NÃO, DIVIDE VARIÁVEL MEDIA_AUX_L POR 100
BTFSS  STATUS,C
BRA    CONV_UNIDADE
INCF   TEMP_DEZENA,F
BRA    VOLTA2

```

DIV_DEZENA1 ;SE NÃO , DIVIDE AS VARIÁVEIS MEDIA_AUX_L E MEDIA_AUX_H POR 60

```

MOVLW .100
BCF    STATUS,C
SUBWF  MEDIA_AUX_L,F
BTFSS  STATUS,C
DECFSZ MEDIA_AUX_H,F
BRA    DIV_DEZENA3

```

DIV_DEZENA2

```

INCF   TEMP_DEZENA,F
MOVLW .100
BCF    STATUS,C
SUBWF  MEDIA_AUX_L,F
BTFSS  STATUS,C
BRA    CONV_UNIDADE
BRA    DIV_DEZENA2

```

DIV_DEZENA3

```

INCF   TEMP_DEZENA,F
BRA    DIV_DEZENA1

```

CONV_UNIDADE

```

MOVFF  MEDIA_L, MEDIA_AUX_L
MOVFF  MEDIA_H, MEDIA_AUX_H

MOVLW .100
MULWF  TEMP_DEZENA

MOVF   PRODL,W
SUBWF  MEDIA_AUX_L,F

MOVF   PRODH,W

```

```
SUBWF MEDIA_AUX_H,F
```

```
MOVLW .10
```

```
BCF STATUS,Z
```

```
XORWF MEDIA_AUX_L,W ;VERIFICA SE VALOR DA SOMA É MENOR QUE 10
```

```
BTFSC STATUS,Z
```

```
BRA CONV_DECIMO ;SE SIM, UNIDADE = 0
```

VOLTA3

```
MOVLW .10
```

```
BCF STATUS,C
```

```
SUBWF MEDIA_AUX_L,F ;SE NÃO, DIVIDE A VARIÁVEL MEDIA_AUX_L POR 10
```

```
BTFSS STATUS,C
```

```
BRA CONV_DECIMO
```

```
INCF TEMP_UNIDADE,F
```

```
BRA VOLTA3
```

CONV_DECIMO

```
MOVFF MEDIA_L, MEDIA_AUX_L
```

```
MOVFF MEDIA_H, MEDIA_AUX_H
```

```
MOVLW .100
```

```
MULWF TEMP_DEZENA
```

```
MOVF PRODL,W
```

```
SUBWF MEDIA_AUX_L,F
```

```
MOVF PRODH,W
```

```
SUBWF MEDIA_AUX_H,F
```

```
MOVLW .10
```

```
MULWF TEMP_UNIDADE
```

```
MOVF PRODL,W
```

```
SUBWF MEDIA_AUX_L,F
```

```
MOVF PRODH,W
```

```
SUBWF MEDIA_AUX_H,F
```

```
MOVFF MEDIA_AUX_L,TEMP_DECIMO
```

;O RESULTADO DA LEITURA DE TEMPERATURA É ENVIADO PARA O *DISPLAY* APENAS SE ESTIVERMOS NA TELA INICIAL

```
BTFSS FUNC,0
```

```
;TESTA PARA VER SE ESTÁ NA TELA INICIAL
```

```
RETURN
```

```
;SE NÃO, SAI DA ROTINA SEM IMPRIMIR NO DISPLAY
```

```
MOVLW 0XC6
```

```
;SE SIM, ENVIA OS 3 DÍGITOS DE TEMPERATURA PARA O DISPLAY
```

```
CALL LCD_COM
```

```
;VAI PARA POSIÇÃO CORRETA NO DISPLAY
```

```

MOVF  TEMP_DEZENA,W
CALL  CONVERTER          ;CONVERTE VALOR DA VARIÁVEL PARA CÓDIGO ASCII
MOVWF  DEZENA_CHAR
CALL  LCD_CHAR           ;ENVIA CÓDIGO ASCII PARA O DISPLAY

MOVLW  0XC7
CALL  LCD_COM            ;VAI PARA POSIÇÃO CORRETA NO DISPLAY
MOVF  TEMP_UNIDADE,W
CALL  CONVERTER          ;CONVERTE VALOR DA VARIÁVEL PARA CÓDIGO ASCII
MOVWF  UNIDADE_CHAR
CALL  LCD_CHAR           ;ENVIA CÓDIGO ASCII PARA O DISPLAY

MOVLW  0XC9
CALL  LCD_COM            ;VAI PARA POSIÇÃO CORRETA NO DISPLAY
MOVF  TEMP_DECIMO,W
CALL  CONVERTER          ;CONVERTE VALOR DA VARIÁVEL PARA CÓDIGO ASCII
MOVWF  DECIMO_CHAR
CALL  LCD_CHAR           ;ENVIA CÓDIGO ASCII PARA O DISPLAY

```

;ATUALIZAÇÃO DO SUPERVISÓRIO

```

MOVLW  'G'              ;INFORMA AO SUPERVISÓRIO QUE IRÁ
BTFSS  TXSTA,TRMT       ;ENVIAR A DEZENA DA TEMPERATURA
BRA    $.2
MOVWF  TXREG

MOVLW  .1
CALL  DELAY_MSEG

BTFSS  TXSTA,TRMT       ;ENVIAR A DEZENA DA TEMPERATURA
BRA    $.2
MOVFF  DEZENA_CHAR,TXREG

MOVLW  .1
CALL  DELAY_MSEG

MOVLW  'H'              ;INFORMA AO SUPERVISÓRIO QUE IRÁ
BTFSS  TXSTA,TRMT       ;ENVIAR A UNIDADE DA TEMPERATURA
BRA    $.2
MOVWF  TXREG

MOVLW  .1
CALL  DELAY_MSEG

BTFSS  TXSTA,TRMT       ;ENVIAR A UNIDADE DA TEMPERATURA
BRA    $.2
MOVFF  UNIDADE_CHAR,TXREG

MOVLW  .1
CALL  DELAY_MSEG

```

```

MOVLW 'I'                ;INFORMA AO SUPERVISÓRIO QUE IRÁ
BTFSS TXSTA,TRMT         ;ENVIAR O DÉCIMO DA TEMPERATURA
BRA    $.2
MOVWF TXREG

```

```

MOVLW .1
CALL   DELAY_MSEG

```

```

BTFSS TXSTA,TRMT         ;ENVIA O DÉCIMO DA TEMPERATURA
BRA    $.2
MOVFF DECIMO_CHAR, TXREG

```

```

RETURN

```

```

;*****
;

```

```

;**** CONVERTE VALOR DAS VARIÁVEIS DE UNIDADE E DEZENA DO SP ****
;

```

```

CONVERTE_SP                ;CONVERTE PARTE INTEIRA DO SP EM UM ÚNICO NÚMERO

```

```

MOVF   SP_DEZENA,W        ;MULTIPLICA DEZENA POR 100
MULLW  .100
MOVFF  PRODH,SP_NUM_H
MOVFF  PRODL,SP_NUM_L

```

```

MOVF   SP_UNIDADE,W       ;MULTIPLICA UNIDADE POR 10 E SOMA
MULLW  .10
MOVF   PRODL,W
BCF    STATUS,C
ADDWF  SP_NUM_L,F
BTFSC  STATUS,C
INCF   SP_NUM_H

```

```

MOVF   SP_DECIMO,W        ;SOMA DÉCIMO
BCF    STATUS,C
ADDWF  SP_NUM_L,F
BTFSC  STATUS,C
INCF   SP_NUM_H

```

```

RETURN

```

```

;*****
;

```

```

;*** CONVERTE VALOR DAS VARIÁVEIS DE UNIDADE E DEZENA DE TMAX ***

```

```

CONVERTE_TMAX          ;CONVERTE PARTE INTEIRA DE TEMPERATURA MÁXIMA EM UM ÚNICO NÚMERO

```

```

    MOVF    TMAX_DEZENA,W      ;MULTIPLICA DEZENA POR 10
    MULLW   .10
    MOVF    PRODL,W
    ADDWF   TMAX_UNIDADE,W     ;SOMA COM A UNIDADE
    MOVWF   TMAX_INTEIRO       ;SALVA NA VARIÁVEL CORRESPONDENTE

    RETURN

```

```

;*****
1

```

```

;*** CONVERTE VALOR DAS VARIÁVEIS DE UNIDADE E DEZENA DE TMIN ***

```

```

CONVERTE_TMIN          ;CONVERTE PARTE INTEIRA DE TEMPERATURA MÍNIMA EM UM ÚNICO NÚMERO

```

```

    MOVF    TMIN_DEZENA,W      ;MULTIPLICA DEZENA POR 10
    MULLW   .10
    MOVF    PRODL,W
    ADDWF   TMIN_UNIDADE,W     ;SOMA COM A UNIDADE
    MOVWF   TMIN_INTEIRO       ;SALVA NA VARIÁVEL CORRESPONDENTE

    RETURN

```

```

;*****
1

```

```

;*****
1
;*                                     *
1
;*  ROTINAS PARA TRATAMENTO DOS BOTÕES  *
1
;*                                     *
3
;*****
1

```

```

;***** AÇÃO DO BOTÃO MAIS *****
1

```

```

FUNC_MAIS

```

```

    BTFSC   FUNC,0             ;NA TELA INICIAL O BOTÃO "MAIS" NÃO TEM FUNÇÃO
    RETURN

```

```

    BTFSC   FUNC,1
    CALL    BOT_INC_SP         ;CHAMA ROTINA DE INCREMENTO NO VALOR DO SP

```

```

    BTFSC   FUNC,2
    CALL    BOT_INC_MAX        ;CHAMA ROTINA DE INCREMENTO NO VALOR DA TEMPERATURA MÁXIMA

```

```

BTFSC  FUNC,3
CALL   BOT_INC_MIN    ;CHAMA ROTINA DE INCREMENTO NO VALOR DA TEMPERATURA MÍNIMA

BSF     CONTROLE,0

RETURN                ;SAI DA ROTINA

```

,***** AÇÃO DO BOTÃO MENOS *****

FUNC_MENOS

```

BTFSC  FUNC,0          ;NA TELA INICIAL O BOTÃO "MENOS" NÃO TEM FUNÇÃO
RETURN

BTFSC  FUNC,1
CALL   BOT_DEC_SP      ;CHAMA ROTINA DE INCREMENTO NO VALOR DO SP

BTFSC  FUNC,2
CALL   BOT_DEC_MAX     ;CHAMA ROTINA DE DECREMENTO NO VALOR DA TEMPERATURA MÁXIMA

BTFSC  FUNC,3
CALL   BOT_DEC_MIN     ;CHAMA ROTINA DE DECREMENTO NO VALOR DA TEMPERATURA MÍNIMA

BSF     CONTROLE,0

RETURN                ;SAI DA ROTINA

```

,***** AÇÃO DO BOTÃO VOLTA *****

FUNC_VOLTA

```

MOVLW  .1              ;VAI PARA A TELA INICIAL
MOVWF  FUNC

RETURN                ;SAI DA ROTINA

```

,***** AÇÃO DO BOTÃO VOLTA *****

FUNC_P

```

BTFSC  FUNC,3
GOTO   FUNC_INI

RLNCF  FUNC             ;ROTACIONA VARIÁVEL PARA A ESQUERDA (ACIONA PRÓXIMA TELA)
RETURN                ;SAI DA ROTINA

```

FUNC_INI

```
MOVLW .1
MOVWF FUNC
RETURN
```

,***** ALTERA VALOR DO SP PARA MAIS *****

BOT_INC_SP

```
MOVF SP_DEZENA,W
CPFSEQ TMAX_DEZENA ;VERIFICA SE DEZENA DE TMAX = DEZENA DE SP
GOTO SP_MAX_OK ;SE FOR DIFERENTE EXECUTA ROTINA DE INCREMENTO DE SP NOR
;MALMENTE

MOVF SP_UNIDADE,W
CPFSEQ TMAX_UNIDADE ;VERIFICA SE UNIDADE DE TMAX = UNIDADE DE SP
GOTO SP_MAX_OK ;SE FOR DIFERENTE EXECUTA ROTINA DE INCREMENTO DE SP NOR
;MALMENTE

MOVF SP_DECIMO,W
CPFSEQ TMAX_DECIMO ;VERIFICA SE DÉCIMO DE TMAX = DÉCIMO DE SP
GOTO SP_MAX_OK ;SE FOR DIFERENTE EXECUTA ROTINA DE INCREMENTO DE SP NOR
;MALMENTE

RETURN ;SE SP = TMAX, ENTÃO SAI DA ROTINA
```

SP_MAX_OK

```
BCF STATUS,Z

MOVLW .9
XORWF SP_DECIMO,W ;TESTA SE O VALOR DA VARIÁVEL É 9
BTFSC STATUS,Z
GOTO INCREMENTA_UNIDADE_SP ;SE SIM, ALTERA UNIDADE

INCF SP_DECIMO ;SE NÃO, INCREMENTA DÉCIMO

RETURN ;SAI DA ROTINA
```

INCREMENTA_UNIDADE_SP

```
CLRF SP_DECIMO ;ZERA DECIMO, POIS MUDOU DE UNIDADE

MOVLW .9
XORWF SP_UNIDADE,W ;TESTA SE O VALOR DA VARIÁVEL É 9
BTFSC STATUS,Z
GOTO INCREMENTA_DEZENA_SP ;SE SIM, ALTERA DEZENA
```

```

    INCF    SP_UNIDADE          ;SE NÃO, INCREMENTA UNIDADE

    RETURN                                ;SAI DA ROTINA

INCREMENTA_DEZENA_SP

    CLRF    SP_UNIDADE          ;ZERA UNIDADE, POIS MUDOU DE UNIDADE

    MOVLW   .9
    XORWF   SP_DEZENA,W        ;TESTA SE O VALOR DA VARIÁVEL É 9
    BTFSC   STATUS,Z
    GOTO     END_INT           ;SE SIM, SAI DA ROTINA DE INTERRUPÇÃO, POIS ESTÁ NO VALOR
                                ;MÁXIMO (99,9°)

    INCF    SP_DEZENA          ;SE NÃO, INCREMENTA DEZENA

    RETURN                                ;SAI DA ROTINA

,*****

,***** ALTERA VALOR DA TEMPERATURA MAX PARA MAIS *****

BOT_INC_MAX

    MOVLW   .9
    XORWF   TMAX_DECIMO,W      ;TESTA SE O VALOR DA VARIÁVEL É 9
    BTFSC   STATUS,Z
    GOTO     INC_UNIDADE_TMAX   ;SE SIM, ALTERA UNIDADE

    INCF    TMAX_DECIMO        ;SE NÃO, INCREMENTA DÉCIMO

    RETURN                                ;SAI DA ROTINA

INC_UNIDADE_TMAX

    CLRF    TMAX_DECIMO        ;ZERA DECIMO, POIS MUDOU DE UNIDADE

    MOVLW   .9
    XORWF   TMAX_UNIDADE,W     ;TESTA SE O VALOR DA VARIÁVEL É 9
    BTFSC   STATUS,Z
    GOTO     INC_DEZENA_TMAX    ;SE SIM, ALTERA DEZENA

    INCF    TMAX_UNIDADE        ;SE NÃO, INCREMENTA UNIDADE

    RETURN                                ;SAI DA ROTINA

INC_DEZENA_TMAX

    CLRF    TMAX_UNIDADE        ;ZERA UNIDADE, POIS MUDOU DE UNIDADE

```



```

MOVLW .9
XORWF TMAX_DEZENA,W      ;TESTA SE O VALOR DA VARIÁVEL É 9
BTFSC STATUS,Z
GOTO END_INT              ;SE SIM, SAI DA ROTINA DE INTERRUPÇÃO, POIS ESTÁ NO VALOR
                           ;MÁXIMO (99,9°)

INCF TMAX_DEZENA           ;SE NÃO, INCREMENTA DEZENA

RETURN                    ;SAI DA ROTINA

```

```

;*****
,

```

```

;***** ALTERA VALOR DA TEMPERATURA MIN PARA MAIS *****
,

```

BOT_INC_MIN

```

MOVF SP_DEZENA,W
CPFSEQ TMIN_DEZENA        ;VERIFICA SE DEZENA DE TMIN = DEZENA DE SP
GOTO INC_MIN_OK           ;SE FOR DIFERENTE EXECUTA ROTINA DE INCREMENTO DE TMIN
                           ;NORMALMENTE

MOVF SP_UNIDADE,W
CPFSEQ TMIN_UNIDADE       ;VERIFICA SE UNIDADE DE TMIN = UNIDADE DE SP
GOTO INC_MIN_OK           ;SE FOR DIFERENTE EXECUTA ROTINA DE INCREMENTO DE TMIN
                           ;NORMALMENTE

MOVF SP_DECIMO,W
CPFSEQ TMIN_DECIMO        ;VERIFICA SE DÉCIMO DE TMIN = DÉCIMO DE SP
GOTO INC_MIN_OK           ;SE FOR DIFERENTE EXECUTA ROTINA DE INCREMENTO DE TMIN
                           ;NORMALMENTE

CALL BOT_INC_SP           ;SE SP = TMIN, INCREMENTA SP PARA QUE ESTE NÃO FIQUE COM
                           ;VALOR MENOR QUE TMIN

```

INC_MIN_OK

```

MOVLW .9
XORWF TMIN_DECIMO,W      ;TESTA SE O VALOR DA VARIÁVEL É 9
BTFSC STATUS,Z
GOTO INC_UNIDADE_TMIN    ;SE SIM, ALTERA UNIDADE

INCF TMIN_DECIMO          ;SE NÃO, INCREMENTA DÉCIMO

RETURN                    ;SAI DA ROTINA

```

INC_UNIDADE_TMIN

```

CLRF TMIN_DECIMO          ;ZERA DECIMO, POIS MUDOU DE UNIDADE

MOVLW .9

```

```

XORWF TMIN_UNIDADE,W      ;TESTA SE O VALOR DA VARIÁVEL É 9
BTFSC STATUS,Z
GOTO INC_DEZENA_TMIN      ;SE SIM, ALTERA DEZENA

INCF TMIN_UNIDADE         ;SE NÃO, INCREMENTA UNIDADE

RETURN                    ;SAI DA ROTINA

```

INC_DEZENA_TMIN

```

CLRf TMIN_UNIDADE         ;ZERA UNIDADE, POIS MUDOU DE UNIDADE

MOVLW .9
XORWF TMIN_DEZENA,W      ;TESTA SE O VALOR DA VARIÁVEL É 9
BTFSC STATUS,Z
GOTO END_INT             ;SE SIM, SAI DA ROTINA DE INTERRUÇÃO, POIS ESTÁ NO VALOR
                           ;MÁXIMO (99,9°)

INCF TMIN_DEZENA         ;SE NÃO, INCREMENTA DEZENA

RETURN                    ;SAI DA ROTINA

```

***** ALTERA VALOR DO SP PARA MENOS *****

BOT_DEC_SP

```

MOVf SP_DEZENA,W
CPFSEQ TMIN_DEZENA       ;VERIFICA SE DEZENA DE TMIN = DEZENA DE SP
GOTO SP_MIN_OK           ;SE FOR DIFERENTE EXECUTA ROTINA DE DECREMENTO DE SP
                           ;NORMALMENTE

MOVf SP_UNIDADE,W
CPFSEQ TMIN_UNIDADE      ;VERIFICA SE UNIDADE DE TMIN = UNIDADE DE SP
GOTO SP_MIN_OK           ;SE FOR DIFERENTE EXECUTA ROTINA DE DECREMENTO DE SP
                           ;NORMALMENTE

MOVf SP_DECIMO,W
CPFSEQ TMIN_DECIMO       ;VERIFICA SE DECIMO DE TMIN = DECIMO DE SP
GOTO SP_MIN_OK           ;SE FOR DIFERENTE EXECUTA ROTINA DE DECREMENTO DE SP
                           ;NORMALMENTE

RETURN                    ;SE SP = TMIN, ENTÃO SAI DA ROTINA

```

SP_MIN_OK

```

TSTFSZ SP_DECIMO
GOTO DEC_SP_1
GOTO DECREMENTA_UNIDADE_SP ;SE SIM, ALTERA UNIDADE

```

DEC_SP_1

DECF SP_DECIMO ;SE NÃO, DECREMENTA DÉCIMO

RETURN ;SAI DA ROTINA

DECREMENTA_UNIDADE_SP

MOVLW .9 ;DÉCIMO VAI PARA 9

MOVWF SP_DECIMO

TSTFSZ SP_UNIDADE

GOTO DEC_SP_2

GOTO DECREMENTA_DEZENA_SP ;SE SIM, ALTERA DEZENA

DEC_SP_2

DECF SP_UNIDADE ;SE NÃO, DECREMENTA UNIDADE

RETURN ;SAI DA ROTINA

DECREMENTA_DEZENA_SP

MOVLW .9 ;DÉCIMO VAI PARA 9

MOVWF SP_UNIDADE

TSTFSZ SP_DEZENA

GOTO DEC_SP_3

GOTO END_INT

;SE SIM, SAI DA ROTINA DE INTERRUPÇÃO, POIS

ESTÁ NO VALOR MÍNIMO (00,0°)

DEC_SP_3

DECF SP_DEZENA ;SE NÃO, DECREMENTA DEZENA

RETURN ;SAI DA ROTINA

***** ALTERA VALOR DA TEMPERATURA MÁXIMA PARA MENOS *****

BOT_DEC_MAX

MOVF SP_DEZENA,W

CPFSEQ TMAX_DEZENA

;VERIFICA SE DEZENA DE TMAX = DEZENA DE SP

GOTO DEC_MAX_OK

;SE FOR DIFERENTE EXECUTA ROTINA DE DECRE-

MENTO DE TMAX NORMALMENTE

```

MOVF    SP_UNIDADE,W
CPFSEQ  TMAX_UNIDADE      ;VERIFICA SE UNIDADE DE TMAX = UNIDADE DE SP
GOTO    DEC_MAX_OK        ;SE FOR DIFERENTE EXECUTA ROTINA DE DECREMENTO DE TMAX
                                ;NORMALMENTE

MOVF    SP_DECIMO,W
CPFSEQ  TMAX_DECIMO      ;VERIFICA SE DÉCIMO DE TMAX = DÉCIMO DE SP
GOTO    DEC_MAX_OK        ;SE FOR DIFERENTE EXECUTA ROTINA DE DECREMENTO DE TMAX
                                ;NORMALMENTE

CALL    BOT_DEC_SP        ;SE SP = TMIN, DECREMENTA SP PARA QUE ESTE NÃO FIQUE COM
                                ;VALOR MAIOR QUE TMAX

DEC_MAX_OK

TSTFSZ  TMAX_DECIMO
GOTO    DEC_MAX_1
GOTO    DEC_UNIDADE_TMAX  ;SE SIM, ALTERA UNIDADE

DEC_MAX_1

DEC     TMAX_DECIMO        ;SE NÃO, DECREMENTA DÉCIMO

RETURN                                ;SAI DA ROTINA

DEC_UNIDADE_TMAX

MOVLW   .9                ;DÉCIMO VAI PARA 9
MOVWF   TMAX_DECIMO

TSTFSZ  TMAX_UNIDADE
GOTO    DEC_MAX_2
GOTO    DEC_DEZENA_TMAX   ;SE SIM, ALTERA DEZENA

DEC_MAX_2

DEC     TMAX_UNIDADE      ;SE NÃO, DECREMENTA UNIDADE

RETURN                                ;SAI DA ROTINA

DEC_DEZENA_TMAX

MOVLW   .9                ;DÉCIMO VAI PARA 9
MOVWF   TMAX_UNIDADE

TSTFSZ  TMAX_DEZENA
GOTO    DEC_MAX_3
GOTO    END_INT           ;SE SIM, SAI DA ROTINA DE INTERRUPÇÃO, POIS ESTÁ NO VALOR
                                ;MÍNIMO (00,0°)

```

DEC_MAX_3

DECF TMAX_DEZENA ;SE NÃO, DECREMENTA DEZENA

RETURN ;SAI DA ROTINA

***** ALTERA VALOR DA TEMPERATURA MÍNIMA PARA MENOS *****

BOT_DEC_MIN

TSTFSZ TMIN_DECIMO

GOTO DEC_MIN_1

GOTO DEC_UNIDADE_TMIN ;SE SIM, ALTERA UNIDADE

DEC_MIN_1

DECF TMIN_DECIMO ;SE NÃO, DECREMENTA DÉCIMO

RETURN ;SAI DA ROTINA

DEC_UNIDADE_TMIN

MOVLW .9 ;DÉCIMO VAI PARA 9

MOVWF TMIN_DECIMO

TSTFSZ TMIN_UNIDADE

GOTO DEC_MIN_2

GOTO DEC_DEZENA_TMIN ;SE SIM, ALTERA DEZENA

DEC_MIN_2

DECF TMIN_UNIDADE ;SE NÃO, DECREMENTA UNIDADE

RETURN ;SAI DA ROTINA

DEC_DEZENA_TMIN

MOVLW .9 ;DÉCIMO VAI PARA 9

MOVWF TMIN_UNIDADE

TSTFSZ TMIN_DEZENA

GOTO DEC_MIN_3

GOTO END_INT ;SE SIM, SAI DA ROTINA DE INTERRUPÇÃO, POIS ESTÁ NO VALOR
;MÍNIMO (00,0°)

DEC_MIN_3

DECF TMIN_DEZENA ;SE NÃO, DECREMENTA DEZENA

RETURN ;SAI DA ROTINA

1

1
*
1 *
* ROTINAS DE DELAY *
1 *
*
1 *

1

***** DELAY DE 1MS *****
1

DELAY_MSEG ;ESTA ROTINA CONTARA X MS, ONDE X E O VALOR DO REGISTRO W NO MOMENTO DA
;CHAMADA

MOVWF TEMPO_MSEG

BCF INTCON,TMR0IF ;LIMPA FLAG DO TIMER0

DELAY

MOVLW .132 ;INICIA CONT COM O VALOR DE 132 DECIMAL. ISTO SIGNIFICA QUE O TIMER
;IRA SER INCREMENTADO 124 VEZES (256-132)
MOVWF TMR0L ;COM UM PRESCALER DE 1/8 O TIMER LEVA 1 MS PRA ESTOURAR

DELAY_VOLTA

BTFSS INTCON,TMR0IF
BRA DELAY_VOLTA

BCF INTCON,TMR0IF
DECFSZ TEMPO_MSEG
BRA DELAY

RETURN

1

***** DELAY DE 1S *****
1

DELAY_SEG ;ESTA ROTINA CONTARA X SEGUNDOS, ONDE X E O VALOR DO REGISTRO W NO MO
;MENTO DA CHAMADA

SEG1

MOVWF TEMPO_SEG

```

MOVLW .4
MOVWF SEGUNDO

```

SEG2

```

MOVLW .250
CALL  DELAY_MSEG
DECFSZ SEGUNDO
BRA   SEG2

```

```

DECFSZ TEMPO_SEG
BRA   SEG1

```

```

RETURN

```

```

;*****
;

```

```

;*****
;

```

```

;*                                     *
;                                     *
;*   ROTINAS PARA O DISPLAY LCD      *
;                                     *
;*                                     *
;                                     *
;*****
;

```

```

;***** ENVIO DE CARACTERE PARA O LCD *****
;

```

LCD_CHAR

```

MOVWF DADO_LCD
SWAPF DADO_LCD,W      ;TROCA NIBBLES

```

```

BSF    RS              ;ENVIO DE CARACTERE (RS=1)
MOVWF  DADO            ;COLOCA DADOS NO BARRAMENTO DE COMUNICAÇÃO
BSF    ENABLE          ;HABILITA ENABLE
MOVLW  .5
CALL   DELAY_MSEG
BCF    ENABLE          ;DESABILITA ENABLE
MOVLW  .5
CALL   DELAY_MSEG

```

```

MOVFF  DADO_LCD,DADO   ;COLOCA DADOS NO BARRAMENTO DE COMUNICAÇÃO
BSF    ENABLE          ;HABILITA ENABLE
MOVLW  .5
CALL   DELAY_MSEG
BCF    ENABLE          ;DESABILITA ENABLE
MOVLW  .5
CALL   DELAY_MSEG

```

```

RETURN

```

```

;*****
;

```

```
,***** ENVIO DE COMANDO PARA O LCD *****
```

```
LCD_COM
```

```

MOVWF DADO_LCD
SWAPF DADO_LCD,W      ;TROCA NIBBLES

BCF    RS              ;ENVIO DE INSTRUÇÃO (RS=0)
MOVWF DADO              ;COLOCA DADOS NO BARRAMENTO DE COMUNICAÇÃO
BSF    ENABLE          ;HABILITA ENABLE
MOVLW .5
CALL   DELAY_MSEG

BCF    ENABLE          ;DESABILITA ENABLE
MOVLW .5
CALL   DELAY_MSEG

MOVFF DADO_LCD,DADO    ;COLOCA DADOS NO BARRAMENTO DE COMUNICAÇÃO
BSF    ENABLE          ;HABILITA ENABLE
MOVLW .5
CALL   DELAY_MSEG
BCF    ENABLE          ;DESABILITA ENABLE
MOVLW .5
CALL   DELAY_MSEG

RETURN
```

```
,*****
```

```
,***** CRIA TELA INICIAL DO DISPLAY *****
```

```
TELA_INICIAL
```

```

MOVLW B'00000001'      ;LIMPA O DISPLAY
CALL   LCD_COM

MOVLW B'00000011'      ;RETORNA CURSOR PARA A LINHA 1 COLUNA 1
CALL   LCD_COM

MOVLW 'S'
CALL   LCD_CHAR
MOVLW 'P'
CALL   LCD_CHAR
MOVLW ':'
CALL   LCD_CHAR
MOVLW ' '
CALL   LCD_CHAR
```



```

MOVLW 0X86
CALL  LCD_COM
MOVLW ','
CALL  LCD_CHAR

```

```

MOVLW 0X88
CALL  LCD_COM
MOVLW 0X00
CALL  LCD_CHAR
MOVLW 'C'
CALL  LCD_CHAR

```

```

MOVLW 0XC0
CALL  LCD_COM

```

```

MOVLW 'T'
CALL  LCD_CHAR
MOVLW 'E'
CALL  LCD_CHAR
MOVLW 'M'
CALL  LCD_CHAR
MOVLW 'P'
CALL  LCD_CHAR
MOVLW ':'
CALL  LCD_CHAR
MOVLW ' '
CALL  LCD_CHAR

```

```

MOVLW 0XC8
CALL  LCD_COM
MOVLW ','
CALL  LCD_CHAR

```

```

MOVLW 0XCA
CALL  LCD_COM
MOVLW 0X00
CALL  LCD_CHAR
MOVLW 'C'
CALL  LCD_CHAR

```

;COLOCAR VALORES DE TEMPERATURA DO SP

```

MOVLW 0X84          ;VAI PARA POSIÇÃO NO DISPLAY
CALL  LCD_COM
MOVF  SP_DEZENA,W
CALL  CONVERTER      ;CONVERTE VALOR DA VARIÁVEL PARA CÓDIGO ASCII
CALL  LCD_CHAR        ;ENVIA CÓDIGO ASCII PARA O DISPLAY

```

```

MOVLW 0X85          ;VAI PARA POSIÇÃO NO DISPLAY
CALL  LCD_COM
MOVF  SP_UNIDADE,W

```

```
CALL    CONVERTER    ;CONVERTE VALOR DA VARIÁVEL PARA CÓDIGO ASCII
CALL    LCD_CHAR     ;ENVIA CÓDIGO ASCII PARA O DISPLAY
```

```
MOVLW  0X87          ;VAI PARA POSIÇÃO NO DISPLAY
CALL    LCD_COM
MOVF    SP_DECIMO,W
CALL    CONVERTER    ;CONVERTE VALOR DA VARIÁVEL PARA CÓDIGO ASCII
CALL    LCD_CHAR     ;ENVIA CÓDIGO ASCII PARA O DISPLAY
```

```
RETURN
```

```
,*****
```

```
,***** CRIA TELA DE ALTERAÇÃO DO SP *****
```

```
TELA_SP
```

```
BTFSC  CONTROLE,0
BRA     ALTERA_SP
```

```
MOVLW  B'00000001'    ;LIMPA O DISPLAY
CALL    LCD_COM
```

```
MOVLW  B'00000011'    ;RETORNA CURSOR PARA A LINHA 1 COLUNA 1
CALL    LCD_COM
```

```
MOVLW  'A'
CALL    LCD_CHAR
MOVLW  'J'
CALL    LCD_CHAR
MOVLW  'U'
CALL    LCD_CHAR
MOVLW  'S'
CALL    LCD_CHAR
MOVLW  'T'
CALL    LCD_CHAR
MOVLW  'E'
CALL    LCD_CHAR
MOVLW  ' '
CALL    LCD_CHAR
MOVLW  'D'
CALL    LCD_CHAR
MOVLW  'O'
CALL    LCD_CHAR
MOVLW  ' '
CALL    LCD_CHAR
MOVLW  'S'
CALL    LCD_CHAR
MOVLW  'P'
CALL    LCD_CHAR
```

```

MOVLW 0XC0          ;MUDA DE LINHA
CALL  LCD_COM

```

```

MOVLW 'V'
CALL  LCD_CHAR
MOVLW 'A'
CALL  LCD_CHAR
MOVLW 'L'
CALL  LCD_CHAR
MOVLW 'O'
CALL  LCD_CHAR
MOVLW 'R'
CALL  LCD_CHAR
MOVLW ':'
CALL  LCD_CHAR
MOVLW ''
CALL  LCD_CHAR

```

```

MOVLW 0XC9
CALL  LCD_COM
MOVLW ','
CALL  LCD_CHAR

```

```

MOVLW 0XCB
CALL  LCD_COM
MOVLW 0X00
CALL  LCD_CHAR
MOVLW 'C'
CALL  LCD_CHAR

```

;COLOCAR VALORES DE TEMPERATURA DO SP

ALTERA_SP

```

BCF    CONTROLE,0

```

```

MOVLW 0XC7          ;VAI PARA POSIÇÃO NO DISPLAY
CALL  LCD_COM
MOVF  SP_DEZENA,W
CALL  CONVERTER      ;CONVERTE VALOR DA VARIÁVEL PARA CÓDIGO ASCII
CALL  LCD_CHAR        ;ENVIA CÓDIGO ASCII PARA O DISPLAY

```

```

MOVLW 0XC8          ;VAI PARA POSIÇÃO NO DISPLAY
CALL  LCD_COM
MOVF  SP_UNIDADE,W
CALL  CONVERTER      ;CONVERTE VALOR DA VARIÁVEL PARA CÓDIGO ASCII
CALL  LCD_CHAR        ;ENVIA CÓDIGO ASCII PARA O DISPLAY

```

```

MOVLW 0XCA          ;VAI PARA POSIÇÃO NO DISPLAY
CALL  LCD_COM

```

```

MOVF  SP_DECIMO,W
CALL  CONVERTER      ;CONVERTE VALOR DA VARIÁVEL PARA CÓDIGO ASCII
CALL  LCD_CHAR       ;ENVIA CÓDIGO ASCII PARA O DISPLAY

```

```

RETURN

```

```

,*****
,

```

```

,***** CRIA TELA DE ALTERAÇÃO DA TEMP MAX *****
,

```

```

TELA_TEMP_MAX

```

```

BTFSC  CONTROLE,0
BRA     ALTERA_TMAX

```

```

MOVLW  B'00000001'      ;LIMPA O DISPLAY
CALL   LCD_COM

```

```

MOVLW  B'00000011'      ;RETORNA CURSOR PARA A LINHA 1 COLUNA 1
CALL   LCD_COM

```

```

MOVLW  'T'
CALL   LCD_CHAR
MOVLW  'E'
CALL   LCD_CHAR
MOVLW  'M'
CALL   LCD_CHAR
MOVLW  'P'
CALL   LCD_CHAR
MOVLW  ' '
CALL   LCD_CHAR
MOVLW  'M'
CALL   LCD_CHAR
MOVLW  'A'
CALL   LCD_CHAR
MOVLW  'X'
CALL   LCD_CHAR
MOVLW  'I'
CALL   LCD_CHAR
MOVLW  'M'
CALL   LCD_CHAR
MOVLW  'A'
CALL   LCD_CHAR

```

```

MOVLW  0XC0              ;MUDA DE LINHA
CALL   LCD_COM

```

```

MOVLW  'V'
CALL   LCD_CHAR

```

```

MOVLW 'A'
CALL  LCD_CHAR
MOVLW 'L'
CALL  LCD_CHAR
MOVLW 'O'
CALL  LCD_CHAR
MOVLW 'R'
CALL  LCD_CHAR
MOVLW ':'
CALL  LCD_CHAR
MOVLW ''
CALL  LCD_CHAR

```

```

MOVLW 0XC9
CALL  LCD_COM
MOVLW ','
CALL  LCD_CHAR

```

```

MOVLW 0XCB
CALL  LCD_COM
MOVLW 0X00
CALL  LCD_CHAR
MOVLW 'C'
CALL  LCD_CHAR

```

;COLOCAR VALORES DE TEMPERATURA MÁXIMA

ALTERA_TMAX

```

BCF    CONTROLE,0

```

```

MOVLW 0XC7                ;VAI PARA POSIÇÃO NO DISPLAY
CALL  LCD_COM
MOVF  TMAX_DEZENA,W
CALL  CONVERTER            ;CONVERTE VALOR DA VARIÁVEL PARA CÓDIGO ASCII
CALL  LCD_CHAR             ;ENVIA CÓDIGO ASCII PARA O DISPLAY

```

```

MOVLW 0XC8                ;VAI PARA POSIÇÃO NO DISPLAY
CALL  LCD_COM
MOVF  TMAX_UNIDADE,W
CALL  CONVERTER            ;CONVERTE VALOR DA VARIÁVEL PARA CÓDIGO ASCII
CALL  LCD_CHAR             ;ENVIA CÓDIGO ASCII PARA O DISPLAY

```

```

MOVLW 0XCA                ;VAI PARA POSIÇÃO NO DISPLAY
CALL  LCD_COM
MOVF  TMAX_DECIMO,W
CALL  CONVERTER            ;CONVERTE VALOR DA VARIÁVEL PARA CÓDIGO ASCII
CALL  LCD_CHAR             ;ENVIA CÓDIGO ASCII PARA O DISPLAY

```

RETURN

,

***** CRIA TELA DE ALTERAÇÃO DA TEMP MIN *****

TELA_TEMP_MIN

BTFSC CONTROLE,0
BRA ALTERA_TMIN

MOVLW B'00000001' ;LIMPA O *DISPLAY*
CALL LCD_COM

MOVLW B'00000011' ;RETORNA CURSOR PARA A LINHA 1 COLUNA 1
CALL LCD_COM

MOVLW 'T'
CALL LCD_CHAR
MOVLW 'E'
CALL LCD_CHAR
MOVLW 'M'
CALL LCD_CHAR
MOVLW 'P'
CALL LCD_CHAR
MOVLW ' '
CALL LCD_CHAR
MOVLW 'M'
CALL LCD_CHAR
MOVLW 'I'
CALL LCD_CHAR
MOVLW 'N'
CALL LCD_CHAR
MOVLW 'I'
CALL LCD_CHAR
MOVLW 'M'
CALL LCD_CHAR
MOVLW 'A'
CALL LCD_CHAR

MOVLW 0XC0 ;MUDA DE LINHA
CALL LCD_COM

MOVLW 'V'
CALL LCD_CHAR
MOVLW 'A'
CALL LCD_CHAR
MOVLW 'L'
CALL LCD_CHAR

```

MOVLW 'O'
CALL  LCD_CHAR
MOVLW 'R'
CALL  LCD_CHAR
MOVLW ':'
CALL  LCD_CHAR
MOVLW ' '
CALL  LCD_CHAR

```

```

MOVLW 0XC9
CALL  LCD_COM
MOVLW ','
CALL  LCD_CHAR

```

```

MOVLW 0XCB
CALL  LCD_COM
MOVLW 0X00
CALL  LCD_CHAR
MOVLW 'C'
CALL  LCD_CHAR

```

;COLOCAR VALORES DE TEMPERATURA MÍNIMA

ALTERA_TMIN

```

BCF    CONTROLE,0

```

```

MOVLW 0XC7                ;VAI PARA POSIÇÃO NO DISPLAY
CALL  LCD_COM
MOVF  TMIN_DEZENA,W
CALL  CONVERTER            ;CONVERTE VALOR DA VARIÁVEL PARA CÓDIGO ASCII
CALL  LCD_CHAR            ;ENVIA CÓDIGO ASCII PARA O DISPLAY

```

```

MOVLW 0XC8                ;VAI PARA POSIÇÃO NO DISPLAY
CALL  LCD_COM
MOVF  TMIN_UNIDADE,W
CALL  CONVERTER            ;CONVERTE VALOR DA VARIÁVEL PARA CÓDIGO ASCII
CALL  LCD_CHAR            ;ENVIA CÓDIGO ASCII PARA O DISPLAY

```

```

MOVLW 0XCA                ;VAI PARA POSIÇÃO NO DISPLAY
CALL  LCD_COM
MOVF  TMIN_DECIMO,W
CALL  CONVERTER            ;CONVERTE VALOR DA VARIÁVEL PARA CÓDIGO ASCII
CALL  LCD_CHAR            ;ENVIA CÓDIGO ASCII PARA O DISPLAY

```

```

RETURN

```

```

-*****
1

```

```

;***** CRIA CARACTERE DE GRAUS *****
;

```

```

CHAR_GRAUS

```

```

    MOVLW 0X40          ;CRIA CARACTERE 'º' NO ENDEREÇO 0X40 DA CGRAM
    CALL  LCD_COM

```

```

    MOVLW B'00001000'
    CALL  LCD_CHAR
    MOVLW B'00010100'
    CALL  LCD_CHAR
    MOVLW B'00010100'
    CALL  LCD_CHAR
    MOVLW B'00001000'
    CALL  LCD_CHAR
    MOVLW B'00000000'
    CALL  LCD_CHAR
    MOVLW B'00000000'
    CALL  LCD_CHAR
    MOVLW B'00000000'
    CALL  LCD_CHAR
    MOVLW B'00000000'
    CALL  LCD_CHAR
    MOVLW B'00000000'
    CALL  LCD_CHAR
    MOVLW B'00000000'

```

```

    RETURN

```

```

;*****
;

```

```

;***** CONVERTE NÚMERO EM CÓDIGO ASCII *****
;

```

```

CONVERTER          ;ROTINA CONVERTE O VALOR CONTIDO NO W NO MOMENTO DA CHAMADA EM
                   ;SEU CÓDIGO ASCII EQUIVALENTE

```

```

    MOVWF CONVERT_CHAR          ;SALVA CONTEÚDO DE W

```

```

    MOVLW .0
    XORWF CONVERT_CHAR,W        ;TESTA SE O VALOR DA VARIÁVEL É 0
    BTFSC STATUS,Z              ;SE SIM, RETORNA COM O CÓDIGO ASCII EQUIVALENTE
    RETLW '0'

```

```

    MOVLW .1
    XORWF CONVERT_CHAR,W        ;TESTA SE O VALOR DA VARIÁVEL É 1
    BTFSC STATUS,Z              ;SE SIM, RETORNA COM O CÓDIGO ASCII EQUIVALENTE
    RETLW '1'

```

```

    MOVLW .2
    XORWF CONVERT_CHAR,W        ;TESTA SE O VALOR DA VARIÁVEL É 2
    BTFSC STATUS,Z              ;SE SIM, RETORNA COM O CÓDIGO ASCII EQUIVALENTE

```



```

RETLW  '2'

MOVLW  .3
XORWF  CONVERT_CHAR,W           ;TESTA SE O VALOR DA VARIÁVEL É 3
BTFSC  STATUS,Z                 ;SE SIM, RETORNA COM O CÓDIGO ASCII EQUIVALENTE
RETLW  '3'

MOVLW  .4
XORWF  CONVERT_CHAR,W           ;TESTA SE O VALOR DA VARIÁVEL É 4
BTFSC  STATUS,Z                 ;SE SIM, RETORNA COM O CÓDIGO ASCII EQUIVALENTE
RETLW  '4'

MOVLW  .5
XORWF  CONVERT_CHAR,W           ;TESTA SE O VALOR DA VARIÁVEL É 5
BTFSC  STATUS,Z                 ;SE SIM, RETORNA COM O CÓDIGO ASCII EQUIVALENTE
RETLW  '5'

MOVLW  .6
XORWF  CONVERT_CHAR,W           ;TESTA SE O VALOR DA VARIÁVEL É 6
BTFSC  STATUS,Z                 ;SE SIM, RETORNA COM O CÓDIGO ASCII EQUIVALENTE
RETLW  '6'

MOVLW  .7
XORWF  CONVERT_CHAR,W           ;TESTA SE O VALOR DA VARIÁVEL É 7
BTFSC  STATUS,Z                 ;SE SIM, RETORNA COM O CÓDIGO ASCII EQUIVALENTE
RETLW  '7'

MOVLW  .8
XORWF  CONVERT_CHAR,W           ;TESTA SE O VALOR DA VARIÁVEL É 8
BTFSC  STATUS,Z                 ;SE SIM, RETORNA COM O CÓDIGO ASCII EQUIVALENTE
RETLW  '8'

MOVLW  .9
XORWF  CONVERT_CHAR,W           ;TESTA SE O VALOR DA VARIÁVEL É 9
BTFSC  STATUS,Z                 ;SE SIM, RETORNA COM O CÓDIGO ASCII EQUIVALENTE
RETLW  '9'

RETLW  '0'                       ;SE O VALOR FOR MAIOR QUE 9, RETORNA 0

RETURN

```

```

*****
1

```

ANEXO B – Código fonte do supervisor

```
Dim dado As String
Dim tempo As Integer
Dim tela As Integer
Dim temp_total As Integer
Dim sp_total As Integer
Dim max_total As Integer
Dim min_total As Integer

Dim sp_dezena As Integer
Dim sp_unidade As Integer
Dim sp_decimo As Integer

Dim temp_dezena As Integer
Dim temp_unidade As Integer
Dim temp_decimo As Integer

Dim max_dezena As Integer
Dim max_unidade As Integer
Dim max_decimo As Integer

Dim min_dezena As Integer
Dim min_unidade As Integer
Dim min_decimo As Integer

Private Sub bt_inicial_Click()

com.Output = "E" 'envia o caractere "E"

bot_volta      'executa rotina do botão

End Sub

Private Sub bt_mais_Click()

com.Output = "C" 'envia o caractere "C"

bot_mais

End Sub

Private Sub bt_menos_Click()

com.Output = "D" 'envia o caractere "D"

bot_menos

End Sub
```

```
Private Sub bt_p_Click()

com.Output = "B" 'envia o caractere "B"

bot_p

End Sub

Private Sub Form_Load()

'inicia comunicação serial

com.PortOpen = True

'inicia variáveis

tela = 0

sp_dezena = 3
sp_unidade = 0
sp_decimo = 0

max_dezena = 6
max_unidade = 0
max_decimo = 0

min_dezena = 1
min_unidade = 0
min_decimo = 0

atualiza_tela

End Sub

Private Sub timer2_Timer()

'recebe dado da porta serial

dado = ""
dado = com.Input

Select Case dado

    Case "B" 'se recebeu char "B"

        bot_p 'executa função do botão P

    Case "C" 'se recebeu char "C"

        bot_mais 'executa função do botão mais
```

Case "D" 'se recebeu char "D"

bot_menos 'executa função do botão menos

Case "E" 'se recebeu char "E"

bot_volta 'executa função do botão volta

Case "G" 'se recebeu char "G"

le_temp_dezena 'chama função

Case "H" 'se recebeu char "H"

le_temp_unidade 'chama função

Case "I" 'se recebeu char "I"

le_temp_decimo 'chama função

End Select

End Sub

Public Function atualiza_tela()

'atualiza *display* de acordo com a tela que está

If tela = 0 Then

lbl_1.Caption = "SP: " & Str(sp_dezena) & Str(sp_unidade) & "," & Str(sp_decimo) & "°C"

lbl_2.Caption = "TEMP: " & Str(temp_dezena) & Str(temp_unidade) & "," & Str(temp_decimo) & "°C"

End If

If tela = 1 Then

lbl_1.Caption = "AJUSTE DO SP"

lbl_2.Caption = "VALOR: " & Str(sp_dezena) & Str(sp_unidade) & "," & Str(sp_decimo) & "°C"

End If

If tela = 2 Then

lbl_1.Caption = "TEMP MAXIMA"

lbl_2.Caption = "VALOR: " & Str(max_dezena) & Str(max_unidade) & "," & Str(max_decimo) & "°C"

End If

If tela = 3 Then

lbl_1.Caption = "TEMP MINIMA"

lbl_2.Caption = "VALOR: " & Str(min_dezena) & Str(min_unidade) & "," & Str(min_decimo) & "°C"

End If

End Function

Public Function bot_mais()

'incrementa SP, temperatura máxima e temperatura mínima

sp_total = sp_dezena & sp_unidade & sp_decimo

max_total = max_dezena & max_unidade & max_decimo

min_total = min_dezena & min_unidade & min_decimo

If tela = 0 Then

GoTo fim_bt_mais

End If

If tela = 1 Then

If sp_total = max_total Then

GoTo fim_bt_mais

Else

If sp_decimo <> 9 Then

sp_decimo = sp_decimo + 1

Else

sp_decimo = 0

If sp_unidade <> 9 Then

sp_unidade = sp_unidade + 1

Else

sp_unidade = 0

sp_dezena = sp_dezena + 1

End If

End If

End If

End If

If tela = 2 Then

If max_total = 999 Then

GoTo fim_bt_mais

Else

If max_decimo <> 9 Then

max_decimo = max_decimo + 1

Else

max_decimo = 0

If max_unidade <> 9 Then

max_unidade = max_unidade + 1

Else

max_unidade = 0

max_dezena = max_dezena + 1

End If

End If

End If

End If

If tela = 3 Then

If min_total = 900 Then

GoTo fim_bt_mais

```

Else
    If min_decimo <> 9 Then
        min_decimo = min_decimo + 1
    Else
        min_decimo = 0
        If min_unidade <> 9 Then
            min_unidade = min_unidade + 1
        Else
            min_unidade = 0
            min_dezena = min_dezena + 1
        End If
    End If
End If
End If
End If

fim_bt_mais:

atualiza_tela

End Function

Public Function bot_menos()

'decrementa SP, temperatura máxima e temperatura mínima

sp_total = sp_dezena & sp_unidade & sp_decimo
max_total = max_dezena & max_unidade & max_decimo
min_total = min_dezena & min_unidade & min_decimo

If tela = 0 Then
    GoTo fim_bt_menos
End If

If tela = 1 Then
    If sp_total = min_total Then
        GoTo fim_bt_menos
    Else
        If sp_decimo <> 0 Then
            sp_decimo = sp_decimo - 1
        Else
            sp_decimo = 9
            If sp_unidade <> 0 Then
                sp_unidade = sp_unidade - 1
            Else
                sp_unidade = 9
                sp_dezena = sp_dezena - 1
            End If
        End If
    End If
End If
End If
End If

```

```

If tela = 2 Then
  If max_total = 100 Then
    GoTo fim_bt_menos
  Else
    If max_decimo <> 0 Then
      max_decimo = max_decimo - 1
    Else
      max_decimo = 9
    If max_unidade <> 0 Then
      max_unidade = max_unidade - 1
    Else
      max_unidade = 9
      max_dezena = max_dezena - 1
    End If
  End If
End If
End If

```

```

If tela = 3 Then
  If min_total = 0 Then
    GoTo fim_bt_menos
  Else
    If min_decimo <> 0 Then
      min_decimo = min_decimo - 1
    Else
      min_decimo = 9
    If min_unidade <> 0 Then
      min_unidade = min_unidade - 1
    Else
      min_unidade = 9
      min_dezena = min_dezena - 1
    End If
  End If
End If
End If

```

fim_bt_menos:

atualiza_tela

End Function

Public Function bot_volta()

'vai para a tela inicial

tela = 0

atualiza_tela

End Function

```
Public Function bot_p()
```

```
'avança de tela
```

```
If tela < 3 Then
```

```
    tela = tela + 1
```

```
Else
```

```
    tela = 0
```

```
End If
```

```
atualiza_tela
```

```
End Function
```

```
Public Function le_temp_dezena()
```

```
'aguarda o envio da dezena da temperatura
```

```
Do
```

```
dado = com.Input
```

```
Loop While dado = "G"
```

```
temp_dezena = Val(dado)
```

```
atualiza_tela
```

```
End Function
```

```
Public Function le_temp_unidade()
```

```
'aguarda o envio da unidade da temperatura
```

```
Do
```

```
dado = com.Input
```

```
Loop While dado = "H"
```

```
temp_unidade = Val(dado)
```

```
atualiza_tela
```

```
End Function
```

```
Public Function le_temp_decimo()
```

```
'aguarda o envio do décimo da temperatura
```


Do

dado = com.Input

Loop While dado = "I"

temp_decimo = Val(dado)

atualiza_tela

End Function