

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
COORDENAÇÃO DE AUTOMAÇÃO INDUSTRIAL
CURSO SUPERIOR DE TECNOLOGIA EM AUTOMAÇÃO INDUSTRIAL

AVELINO RAIMUNDO JUNIOR

GUSTAVO IMAI ROSSI

TERMO REGISTRADOR DIGITAL MICROCONTROLADO

TRABALHO DE CONCLUSÃO DE CURSO

PATO BRANCO

2013

AVELINO RAIMUNDO JUNIOR
GUSTAVO IMAI ROSSI

TERMO REGISTRADOR DIGITAL MICROCONTROLADO

Trabalho de Conclusão de Curso de graduação apresentado à disciplina de Trabalho de Diplomação, do Curso de Tecnologia em Automação Industrial da Universidade Tecnológica Federal do Paraná – UTFPR, Câmpus Pato Branco, como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Prof. Dr. Fábio Luiz Bertotti

PATO BRANCO

2013

TERMO DE APROVAÇÃO

O trabalho de diplomação intitulado “**TERMO REGISTRADOR DIGITAL MICROCONTROLADO**” foi considerado **APROVADO** de acordo com a ata da banca examinadora N° 148 de 2013.

Fizeram parte da banca os professores:

FÁBIO LUIZ BERTOTTI (Orientador)

ANDRÉ MACARIO BARROS

CEZAR RAFAEL CLAURE TORRICO

AGRADECIMENTOS

Primeiramente nossos agradecimentos vão aos professores que com seu empenho conseguiram nos fornecer o conhecimento necessário para finalizarmos esta etapa da nossa formação. Damos ênfase ao agradecer o professor Dr. Fábio Luiz Bertotti pelo empenho na orientação do nosso trabalho de conclusão de curso.

Posteriormente agradecemos nossas famílias, que foram o alicerce para a construção do nosso caminho até a finalização do nosso curso superior.

EPÍGRAFE

I call my field “knowledge management” but you can’t manage knowledge. Nobody can. What you do - what a company does - is manage the environment that optimizes knowledge. (PRUSAK, Laurence, 1997).

Eu denomino meu campo de Gestão do Conhecimento, mas você não pode gerenciar conhecimento. Ninguém pode. O que você pode fazer - o que a empresa pode fazer – é gerenciar o ambiente que otimize o conhecimento. (PRUSAK, Laurence, 1997).

RESUMO

RAIMUNDO JUNIOR, Avelino; ROSSI, Gustavo Imai. Termo registrador Digital Microcontrolado. 2013. 14529 f. Trabalho de Conclusão de Curso (Graduação em Tecnologia em Automação Industrial) – Programa de Graduação em Tecnologia, Universidade Tecnológica Federal do Paraná. Pato Branco, 2013.

Este trabalho apresenta o desenvolvimento do protótipo de um Termo Registrador Digital Microcontrolado que tem como função obter dados de temperatura em uma máquina de pasteurização de alimentos através de um transdutor tipo PT100 instalado na mesma. Um sistema microcontrolado é responsável pela aquisição do sinal do sensor de temperatura, pela apresentação do valor da temperatura obtida em um display de LCD e pelo armazenamento deste valor junto com a informação temporal do instante da aquisição em uma memória externa não volátil. Os dados armazenados nesta memória podem ser transferidos a um computador através de uma interface de comunicação serial com padrão RS232. Os resultados obtidos são apresentados e discutidos no final deste trabalho.

Palavras-chave: Termo Registrador, Microcontrolador, transdutor PT100, Memória Flash, comunicação serial.

ABSTRACT

RAIMUNDO JUNIOR, Avelino; ROSSI, Gustavo Imai. Microcontrolled Temperature Data Logger. 2013. 52 f. Graduation Conclusion Work (Graduation in Industrial Automation Technology) – Technology Graduation Program, Federal University of Technology – Paraná. Pato Branco, 2013.

This work presents the development of a Microcontrolled Temperature Data Logger prototype whose function is to obtain and record temperature data on a milk pasteurization machine by means of a PT100 temperature type transducer. The microcontrolled system carries out data acquisition, processes the obtained temperature value, shows it on an LCD display and stores this value together with acquisition time information in a non-volatile external memory. Data stored in this memory can be transferred to a computer through a RS-323 serial communication interface. The results obtained are presented and discussed at the end of this work.

Keywords: Data Logger, Microcontroller, PT100 temperature transducer, Flash memory, serial communication.

LISTA DE FIGURAS

Figura 1 - Processo de transmissão e reconhecimentos dos bits em comunicação serial síncrona.....	21
Figura 2 - Processo de transmissão serial síncrona.....	21
Figura 3 - Momentos de START e STOP da comunicação serial.	23
Figura 4 - Processo de quantização de um sinal analógico amostrado $V(k,w)$, onde k é o número da amostra e w a palavra digital resultante deste processo.	26
Figura 5 – Diagrama em blocos do Termo Registrador Digital Microcontrolado.	29
Figura 6 - kit de desenvolvimento Lauchpad para MSP430	30
Figura 7 - Diagrama de blocos do microcontrolador MSP430F2234.....	33
Figura 8 - Especificação dos pinos no MSP430F2234.....	33
Figura 9 - Portas lógicas do MAX3222.....	37
Figura 10 - Conexões externas do MAX3222.....	37
Figura 11 - Soquete Adaptador para MSP430F2234, desenho gerado no software EAGLE 5.6.0.....	39
Figura 12 - Circuito de interfaceamento com o display.	40
Figura 13 - Circuito de condicionamento de sinal do transdutor PT100	41
Figura 14 – Esquema elétrico do circuito RTC interligado ao microcontrolador.....	42
Figura 15 - Ligação dos botões ao microcontrolador.	42
Figura 16 Circuito de conexão da da Memória.....	43
Figura 17 - Circuito do MAX3222	44
Figura 18 - Circuito de interfaceamento da memória e MAX3222 com o microcontrolador.....	45
Figura 19 - Termo registrador digital montado em uma protoboard, onde o circulo indicado por A é o soquete adaptador para o MSP430F2234, B é o cabo para programação do microcontrolador, C é o display, D é o circuito do RTC, E são os push-bottons, F é a memória flash e G é o circuito do MAX3222.....	46
Figura 20 - Circuito de instrumentação e condicionamento de sinal.	47

LISTA DE QUADROS

Quadro 1 - Função de inicialização do conversor A/D	52
Quadro 2 - Inicialização da comunicação I2C	53
Quadro 3 - Inicialização da comunicação SPI	54

LISTA DE TABELAS

Tabela 1: Configurações do módulo USCI dos microcontroladores MSP430 para os modos de funcionamento da SPI.	24
Tabela 2 - Especificação dos pinos da memória AT25DF641	35
Tabela 3 - Especificação dos pinos do RTC8563.....	36
Tabela 4 - Módulos LCD a caracter disponíveis no mercado	38

LISTA DE ABREVIATURAS

A/D	Analógico / Digital
LEDs(<i>Light Emitting Diode</i>)	Diodo emissor de luz
PWM (<i>Pulse Width Modulation</i>)	Largura de pulso modulado
RISC (<i>Reduced Instruction Set Computer</i>)	Reduzido computador conjunto de instruções
CPU (Central Processor Unit)	Unidade Central de Processamento
RAM (<i>Random Access Memory</i>)	Memória de acesso aleatório
MIPS(<i>Microprocessor without interlocked pipeline stages</i>)	Microprocessador sem estágios de pipeline intertravados
MODEM (modulador/ demodulador)	Modulador e demodulador
I ² C (<i>Inter Integrated Communication</i>) Comunicação entre Integrados)	É um barramento serial (Comunicação entre Integrados)
SCL	Sinal de <i>clock</i>
SDA	Sinal de dados
ACK – <i>acknowledgement</i>	Sinal de reconhecimento
SPI(<i>Serial Peripheral Interface Bus</i>)	Barramento serial de interface periférica
SIMO (<i>Slave In Master Out</i>)	Escravo entra mestre sai.
SOMI (<i>Slave Out Master In</i>)	Escravo sai mestre entra.
CKPL	Polaridade do sinal de <i>clock</i>
RTS (ready to send)	Pronto para enviar
CTS (clear to send)	Limpar para enviar
LSB (<i>Least Significant Bit</i>)	Bit menos significativo
RTC (real time clock)	Relógio de Tempo Real
EPROM (<i>erasable programmable read-only memory</i>)	Memória programável apagável somente de leitura
EEPROM (<i>Electrically-Erasable Programmable Read-Only Memory</i>)	Memória eletricamente programável apagável somente de leitura
USB (<i>Universal Serial Bus</i>)	Barramento serial universal
DIP (<i>Dual In-line Package</i>)	(Duplo pacote em linha) Tipo de encapsulamento
CCStudio	Code Composer Studio [®]
TI	Texas Instruments
IRDA(<i>Infrared Data Association</i>)	Transmissores infravermelhos
SOIC (<i>Svenska Ostindiska Companiet</i>)	Tipo de encapsulamento
GND (Ground)	Referência elétrica a 0 Volt
LCD (<i>liquid-crystal display</i>)	Display de cristal líquido
SMD (<i>Surface Mounted Components</i>)	Tecnologia de montagem superficial
TSSOP	Tipo de encapsulamento
PCI	Placa de Circuito Impresso

SUMÁRIO

1. INTRODUÇÃO.....	14
1.2 OBJETIVOS	15
1.2.1 Objetivo Geral	15
1.2.2 Objetivos Específicos	15
2. FUNDAMENTAÇÃO TEÓRICA.....	16
2.1 CONCEITOS SOBRE PASTEURIZAÇÃO	16
2.2 SENSORES E AQUISIÇÃO DE SINAIS.....	16
2.3 TERMORESISTOR PT-100	17
2.4 MICROCONTROLADORES.....	18
2.4.1 Microcontroladores da família MSP430 da Texas Instruments	19
2.5 COMUNICAÇÃO SERIAL	20
2.5.1 Comunicação Serial I ² C	22
2.5.2 Comunicação serial SPI	23
2.6 CONVERSOR A/D	25
2.7 RELÓGIO DE TEMPO REAL.....	27
2.8 MEMÓRIA FLASH.....	27
3. MATERIAIS E MÉTODOS.....	29
3.1 MATERIAIS UTILIZADOS.....	30
3.1.1 Placa de Desenvolvimento MSP-EXP430G2 <i>LauchpadExperimenter Board</i> ...30	30
3.1.2 Ferramenta de desenvolvimento de código.....	31
3.1.3 O Microcontrolador MSP430F2234	31
3.1.4 Memória Flash AT25DF641	34
3.1.5 RTC PCF8563.....	35
3.1.6 Dispositivo de interface de linha padrão RS-232.....	36
3.1.7 Display de LCD	38
3.2 METODOLOGIA.....	39

3.2.1 Desenvolvimento de soquete adaptador para o MSP430F2234	39
3.2.2 Conexão do circuito de emulação do kit Launchpad ao microcontrolador MSP430F2234	40
3.2.3 Interfacemanto com o Display.	40
3.2.4 Desenvolvimento de circuito de condicionamento de sinal para o transdutor PT100.....	41
3.2.5 Desenvolvimento do circuito do RTC e interfaceamento com o microcontrolador	41
3.2.6 Interfaceamento com teclado	42
3.2.7 Interfaceamento com memória externa.....	43
3.2.8 Desenvolvimento de circuito de interface de linha padrão RS232 e interfacemanto com o microcontrolador	43
4. RESULTADOS	46
5. CONCLUSÕES	49
REFERÊNCIAS	50
APÊNDICE A.....	52
APÊNDICE B.....	53
APÊNDICE C	54

1. INTRODUÇÃO

Segundo o item 2.1.1.1 da Instrução Normativa 51, que trata do leite pasteurizado, publicada pelo Ministério da Agricultura em 18 de setembro de 2002:

“O leite quando destinado ao consumo humano direto, deve ser submetido a tratamento térmico na faixa de temperatura de 72 a 75 °C (setenta e dois a setenta e cinco graus Celsius) durante 15 a 20s (Quinze a vinte segundos), em equipamento de pasteurização a placas, dotado de painel de controle com termo registrador e termo regulador automáticos” (MINISTÉRIO DA AGRICULTURA, 2002).

Em virtude dos registradores utilizados atualmente serem analógicos, com registros em discos de papel, vislumbrou-se a possibilidade de desenvolvimento de um termo registrador digital microcontrolado. Um termo registrador digital microcontrolado confere maior confiabilidade, agilidade na verificação, consistência de dados e autonomia no processo de registros das informações geradas pelo equipamento, quando comparado com os registradores analógicos.

O Termo registrador digital microcontrolado tem como objetivo fazer aquisição de temperaturas obtidas no processo de pasteurização de leite e seus derivados, e registrar os resultados obtidos em um documento digital que possa ser consultado e ou analisado pelo representante da agência nacional de vigilância sanitária que irá inspecionar o estabelecimento onde ocorre o processo de beneficiamento.

Através da aquisição de dados extraídos do processo produtivo, o termo registrador conseguirá registrar em arquivo digital, de forma segura e inviolável, as informações sobre variações de temperatura no processo de pasteurização de leite, permitindo acesso para análise e *download* dos dados registrados a qualquer momento.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Desenvolver um termo registrador digital microcontrolado aplicado ao processo de pasteurização de leite.

1.2.2 Objetivos Específicos

Os objetivos específicos deste trabalho são:

- Realizar estudos das normas vigentes para o processo de pasteurização de produtos lácteos, onde o equipamento será utilizado.
- Conduzir estudos sobre:
 - O micro controlador MSP430F2234T;
 - Os periféricos que serão utilizados no microcontrolador;
 - Sensores e aquisição de sinais;
 - Protocolos de comunicação serial;
 - Memória externa para armazenamento de dados.
- Projeto e implementação do *hardware*;
- Desenvolvimento de *firmware* para o micro controlador;
- Realização de testes de funcionamento do protótipo.

2. FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresentará, a partir de uma revisão bibliográfica, os conceitos teóricos sobre os principais assuntos envolvidos neste trabalho. Serão apresentados conceitos sobre pasteurização de leite, medição de temperatura utilizando sensor, o microcontrolador e suas funções, comunicação serial, utilização de memória e armazenamento de dados, conversão de sinais e geração de informação de tempo real.

2.1 CONCEITOS SOBRE PASTEURIZAÇÃO

A pasteurização é um processo empregado para garantir que o leite seja isento de microrganismos causadores de doenças no homem e consiste de um tratamento térmico em que a temperatura do leite é elevada para valores entre 71°C e 75°C, menor que o ponto de ebulição, por um determinado tempo, e este é resfriado instantaneamente. Este processo é suficiente para eliminar os microrganismos patogênicos do leite e também a quase totalidade da flora bacteriana (MINISTÉRIO DA AGRICULTURA, 2002).

2.2 SENSORES E AQUISIÇÃO DE SINAIS

Um sensor é um dispositivo que detecta fenômenos físicos e gera algum tipo de sinal para dispositivos de medição ou controle, diferente de um transdutor que recebe energia de uma fonte e a transforma em outro tipo de energia. Tipicamente, as informações oriundas dos sensores são a base para o monitoramento e controle na maioria dos processos industriais (ORDOZES, PENTEADO e SILVA, 2005).

Em muitos sistemas de controle e monitoramento é necessária a medição de alguma grandeza física como, por exemplo, temperatura, pressão, velocidade, entre outras. A maioria destas grandezas é inerente a fenômenos físicos e, em geral,

sua natureza é analógica. Trata-se de sinais contínuos no tempo que são formados por infinitos valores, diferentemente de sinais digitais que assumem valores bem definidos, por exemplo, (ORDOZES, PENTEADO e SILVA, 2005).

Geralmente, quando os sinais analógicos dos sensores são processados por sistemas digitais, há a necessidade do condicionamento do sinal para que estes sinais sejam adequados à faixa de tensão de entrada do conversor A/D (Analógico para Digital). Alguns dispositivos digitais, como os microcontroladores, costumam disponibilizar um conversor A/D interno, o que elimina a necessidade do uso de um dispositivo externo. Assim, com o emprego de microcontroladores no controle de processos, o valor analógico convertido para digital pode ser processado pelo *software* de controle de acordo com decisões lógicas baseadas em comparações. Podem-se executar, também, operações matemáticas que definem as funções de controle (ORDOZES, PENTEADO e SILVA, 2005).

2.2.1 Termorresistor PT-100

As termorresistências são transdutores de temperatura muito utilizados na indústria, pois além de ótima precisão, possuem uma larga faixa de trabalho e ainda permitem ligações a longa distância. O princípio de funcionamento é baseado na variação de resistência de um filamento bastante delgado de um metal, como platina ou níquel, com a temperatura (THOMAZINI, 2005).

Thomazini (2005) diz que a termorresistência de platina é a mais usada industrialmente devido a sua estabilidade e precisão. Esta termorresistência tem sua curva padronizada conforme a norma DIN – IEC 751 – 1985 e tem como característica uma resistência de 100Ω a 0°C , dando origem a denominação PT100. A faixa de temperatura de trabalho vai de -200°C a 650°C , porém a ITS – 90 padronizou o seu uso até 962°C , aproximadamente. A resistência de isolamento a temperatura ambiente entre cada terminal e a bainha deve ser pelo menos $100\text{M}\Omega$ (THOMAZINI, 2005).

O auto-aquecimento do transdutor PT100 é causado pela corrente que passa pela resistência, oriunda do instrumento de leitura, e pode levar a erros que

comprometam a medição. Para evitar este efeito, deve-se limitar a corrente de excitação do sensor a valores da ordem de 1 a 2 mA. A elevação da temperatura equivalente ao aumento da dissipação de calor na termorresistência não deve exceder 0,3°C (THOMAZINI, 2005).

A variação da resistência com a temperatura é chamada de coeficiente de temperatura e é especificada com a média da variação entre 0 e 100°C. Este coeficiente é de 0,385Ω por °C, conforme a norma IEC 60751 (NOVUS PRODUTOS ELETRÔNICOS, 2012).

2.3 MICROCONTROLADORES

Matic e Andric (2000) relatam que a história dos microcontroladores iniciou em 1969 com a empresa INTEL, através da idéia de uma empresa japonesa chamada BUSICOM, que enviou engenheiros aos Estados Unidos com o intuito de encomendar circuitos integrados para serem utilizados em calculadoras. A INTEL ficou responsável pelo desenvolvimento do projeto, e propôs uma idéia diferente da construção sugerida pelos japoneses, ou seja, um circuito integrado comandado por um programa armazenado no próprio circuito. A empresa japonesa optou por uma solução mais simples, mas a INTEL utilizou-se da concepção e criou o primeiro microprocessador, sendo assim, em 1971 a INTEL comprou o direito de venda do circuito integrado da empresa BUSICOM. Passado o tempo, somando-se a grande concorrência e o conseqüente desenvolvimento dos circuitos integrados, surgiram no mercado circuitos que integravam simultaneamente processador e periféricos, resultando no primeiro *chip* contendo um “microcomputador”, denominado posteriormente de microcontrolador (MATIC e ANDRIC, 2000).

Atualmente, os microcontroladores vêm ocupando cada vez mais espaço nas vidas das pessoas, sendo empregados nos mais variados equipamentos. Por aliarem baixo custo, fácil manutenção, sofisticação e personalização, muitos fabricantes optam por utilizar microcontroladores nos seus equipamentos ao invés da eletrônica tradicional (PEREIRA, 2004).

Souza (2010) diz que os microcontroladores podem ser definidos resumidamente como um “pequeno” componente eletrônico que pode ser programado para executar tarefas no controle de processos lógicos, que se baseiam nas ações lógicas que devem ser executadas dependendo do estado dos periféricos de entrada e/ou saída, periféricos estes que podem ser LEDs (*Light Emitting Diode*), displays de segmentos, sensores diversos, relês e muitos outros.

O microcontrolador é um componente eletrônico programável, e sua lógica de operação é estruturada na forma de um programa e gravada dentro do componente, e depois de gravada, cada vez que o microcontrolador for alimentado, o programa que está inserido internamente nele será executado. Os microcontroladores possuem em um único e pequeno encapsulamento todos os componentes necessários para controlar um processo, ou seja, é provido, internamente, de memória de programa, memória de dados, portas de entrada e saída paralela, temporizadores, contadores, interface de comunicação serial, geradores de sinais PWM (*Pulse Width Modulation*), conversores A/D, etc. As características citadas são o que diferem o microcontrolador de um microprocessador, pois o microprocessador não possui todos estes recursos agrupados em um único componente (SOUZA, 2010).

2.3.1 Microcontroladores da família MSP430 da Texas Instruments

Para PEREIRA (2005), os microcontroladores da família MSP430, fabricados pela empresa Texas Instruments, possuem um *design* simples, porém poderoso. Desenvolvido em arquitetura RISC (*Reduced Instruction Set Computer*), estes microcontroladores combinam um conjunto reduzido de instruções à arquitetura de barramento clássica de Von Neumann, o que permite a CPU (*Central Processing Unit*) possuir um único espaço de endereçamento de memória.

Os microcontroladores da família MSP430 possuem em sua CPU três barramentos distintos, sendo um de endereços, outro de dados e o terceiro de controle. O tamanho dos barramentos de dados e endereço é de 16 *bits*. Isto significa que é possível acessar pelo barramento de endereços até 65.536 posições de memória e pelo barramento de dados a CPU pode processar informações em

lotes de 16 *bits* (PEREIRA, 2005). As principais características dos microcontroladores da família MSP430 são (TEXAS INSTRUMENTS, 2008):

- Baixo consumo de energia:
 - Operam com tensões de 1,8V até 3,6V
 - 0,1 μ A retendo dados na RAM (*Random Access Memory*);
 - 0,8 μ A modo relógio de tempo real;
 - 250 μ A com MIPS (*Microprocessor without interlocked pipeline stages*) ativo.
- Alto desempenho, ideal para medição analógica de precisão;
 - Temporizador de comparação para medir elementos resistivos.
- CPU RISC de 16 bits que permite aplicações complexas com códigos pequenos;
 - Arquivo de registro que elimina gargalo de trabalho;
 - *Design* compacto, o que reduz o consumo de energia e custo;
 - Otimizado para programação moderna de alto nível;
 - 27 instruções básicas e sete modos de endereçamento;
 - Extensa capacidade de interrupção vetorizada.
- No sistema de *flash* programável permite alterações de código, melhorias flexíveis de campo e dados de registro.

2.4 COMUNICAÇÃO SERIAL

Nicolosi (2002) comenta que uma comunicação serial permite a troca de dados entre máquinas bidirecionalmente com apenas dois fios, ou também através de um MODEM (modulador/ demodulador) que recebe e transfere dados por meio do sistema telefônico tradicional.

O sistema de comunicação SERIAL transmite um *byte* ou *word*, bit-a-bit, numa sequência pré-estipulada, para que o receptor receba a informação e transforme-a em um *byte* ou *word* igual ao transmitido inicialmente. Basicamente, existem dois modos de comunicação SERIAL, sendo Serial Síncrona e Serial Assíncrona. A comunicação Serial Síncrona faz com que uma das saídas envie ou

receba os dados, e outra transmite um sinal de sincronismo, fazendo com que o receptor entenda qual *bit* é “0” e qual *bit* é “1” na comunicação. Também precisa-se enviar um *byte* para iniciar a geração do sincronismo, e outro *byte* para indicar o fim da transmissão, conforme mostrado nas Figuras 1 e 2.

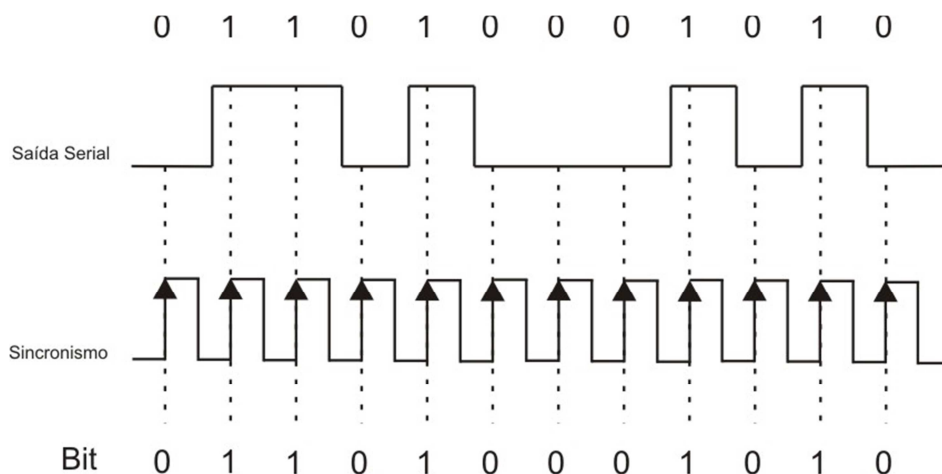


Figura 1 - Transmissão e reconhecimentos dos *bits* em comunicação serial síncrona.

Fonte: Adaptado de Nicolosi (2002).

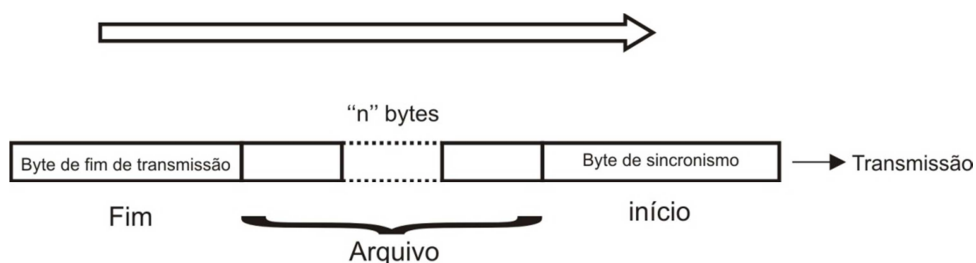


Figura 2 - Processo de transmissão serial síncrona.

Fonte: Adaptado de Nicolosi (2002).

A comunicação Serial Assíncrona não precisa de sincronismo, ela gera um padrão de transmissão para cada *byte* transmitido. Quando o *byte* será transmitido, inicia-se por um *bit* de início, ou *bit* de “start” e finaliza-se por um *bit* de fim, ou *bit* de “stop”. O sistema de recepção sempre espera o *bit* de start (transição de 1 para 0), quando o recebe começa a recepção dos outros *bit* do *byte*, até receber o *bit* de stop (transição de 0 para 1), momento este que finaliza a comunicação, até reiniciar o processo recebendo outro *bit* de start. Além disso, para

que ocorra comunicação entre o transmissor e o receptor é necessário que as frequências de ambos sejam as mesmas (NICOLOSI, 2002).

2.4.1 Comunicação Serial I²C

Para facilitar o desenvolvimento de sistemas modulares para televisores e outros aparelhos eletrônicos, a Philips criou o protocolo síncrono *half duplex* a dois fios, do tipo mestre-escravo, denominado I²C (*Inter Integrated Communication* ou Comunicação entre Integrados). O protocolo tem sua comunicação iniciada e comandada por um dispositivo chamado de mestre e dispositivo que responde as suas solicitações é o escravo, que se encontra do outro lado da linha, portanto a denominação de protocolo síncrono mestre-escravo.

Um dos fios utilizados pelo barramento I²C é o sinal de *clock* (chamado de SCL) e o segundo é o sinal de dados (chamado de SDA). Para permitir a ligação de diversos dispositivos na mesma linha, o protocolo especifica as saídas em coletor aberto. Para manter o barramento de comunicação em nível lógico alto ou “1”, utilizam-se dois resistores de *pull-up*. O I²C também suporta a presença de diversos mestres ligados simultaneamente no mesmo barramento, cujo modo é denominado *multimastering*, porém apenas um deles pode estar ativo, caso contrário ocorrerá colisão de dados no barramento (PEREIRA, 2005). As regras básicas de funcionamento do protocolo I²C são:

1. A informação presente na linha de dados (SDA) somente é lida durante a fase alta da linha de *clock* (SCL);
2. Somente é permitido alterar o nível da linha de dados (SDA) durante a fase baixa da linha de *clock* (SCL);
3. Quando o barramento não está em uso, ambas as linhas permanecem desligadas, portanto forçadas nível alto pelos resistores de *pull-up*.

Para indicar o início de uma transmissão (*START*), a linha SDA é forçada da condição “1” para “0”, durante a fase alta do *clock*. Para indicar o fim de uma transmissão (*STOP*), a linha SDA passa de “0” para “1” durante a fase alta da linha

SCL. Observa-se na Figura 3 que somente é permitida a emissão da condição de parada ao fim da comunicação e não ao término de um caractere (PEREIRA, 2005).

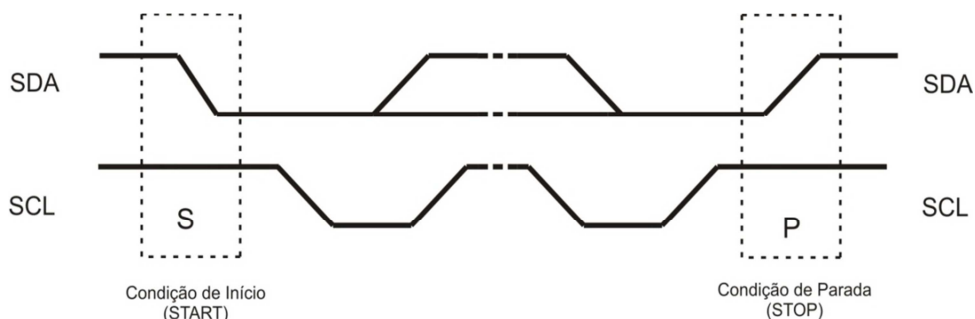


Figura 3 - Momentos de *START* e *STOP* da comunicação serial.

Fonte: Pereira (2005).

Após dado início a comunicação através do *bit* de *START*, são transmitidos oito *bits* de dados, começando pelo *bit* mais significativo. Após o último *bit* (menos significativo), o receptor deve gerar uma condição de reconhecimento (*ACK* – *acknowledgement*), o que é feito forçando a linha SDA em nível “0” antes do pulso de *clock* da linha SCL (PEREIRA, 2005).

2.4.2 Comunicação serial SPI

No protocolo de comunicação serial padrão SPI (*Serial Peripheral Interface Bus*), comunicação é feita através de três fios, sendo uma linha para troca de dados denominada SIMO (*Slave In Master Out*) que envia informações no modo mestre ou recebe no modo escravo, outra linha para troca de dados denominada SOMI (*Slave Out Master In*), que envia informações no modo escravo e recebe no modo mestre, e a terceira linha é denominada UCLK, que envia sinal de *clock* no modo mestre, e recebe sinal de *clock* no modo escravo. Um quarto sinal pode ser utilizado, que serve para seleção do dispositivo (*STE*, *CS* ou *SS*), permitindo o dispositivo mestre selecionar um entre vários dispositivos escravos. Este sinal também indica o início e o término da comunicação ao dispositivo escravo (PEREIRA, 2005).

O protocolo SPI possibilita quatro tipos de funcionamento, diferenciando-se um do outro através da fase do sinal do *clock* que o dado é lido. Nos modos 0 e

1, o sinal de *clock* é ativo em nível alto, e nos modos 2 e 3 o *clock* é ativo em nível baixo, selecionados pelos *bits* UxTCTL:CKPH (fase do sinal de *clock*) e UxTCTL:CKPL (polaridade do sinal de *clock*) do periférico USCI dos microcontroladores representados na Tabela 1a 1. O dispositivo com o qual deseja-se a comunicação é que define a escolha do modo de funcionamento (PEREIRA, 2005).

Tabela 1: Configurações do módulo USCI para funcionamento da SPI.

Modo	KPL	KPH	Descrição
0	0	0	O dado é armazenado na borda de subida do <i>clock</i>
1	0	1	O dado é armazenado na borda de descida do <i>clock</i>
2	1	0	O dado é armazenado na borda de descida do <i>clock</i>
3	1	1	O dado é armazenado na borda de subida do <i>clock</i>

Fonte: Pereira (2005).

2.4.3 Comunicação serial RS232.

O padrão RS232 é um padrão de comunicação serial onde os *bit* são enviados um a um em sequência, e normalmente o *bit* menos significativo primeiro. O protocolo é assíncrono, isto é, não tem uma linha de *clock*, sendo assim é responsabilidade do transmissor e do receptor efetuarem controles de tempo para saber quando cada *bit* inicia e finaliza (PEREIRA, 2005).

O RS-232 utiliza dois sinais de controle, o RTS (*ready to send*) e o CTS (*clear to send*) para efetuar o controle de fluxo via *hardware*. Basicamente, quando o transmissor deseja começar um envio ele sinaliza através do pino RTS. O receptor ao perceber que o transmissor deseja enviar algum dado, prepara-se para recebê-lo e seta o pino CTS. Apenas depois de receber o sinal CTS o transmissor pode começar a transmissão (PEREIRA, 2005).

2.5 CONVERSOR A/D

Para que um sinal analógico possa ser interpretado por um microcontrolador, microprocessador ou sistema digital, este precisa ser convertido em um sinal digital. Para que isto aconteça, primeiramente deve-se condicionar o sinal analógico, através de um circuito condicionador de sinais, para que, posteriormente, o sinal condicionado possa ser digitalizado, cujo processo é realizado por um circuito conhecido como conversor A/D (PAZOS, 2002).

Pazos (2002) comenta que o teorema de Nyquist ou teorema da amostragem deve ser obedecido no processo de conversão de um sinal analógico para um digital correspondente. Este teorema estabelece que para um sinal analógico, que apresenta uma frequência máxima f_{max} , ser amostrado, processado e recuperado (convertido em analógico novamente), a frequência de amostragem mínima (inversa do período de amostragem) deve ser maior que duas vezes f_{max} .

Um sinal analógico é contínuo no tempo, ou seja, o sinal é formado por infinitos valores ao longo de um intervalo de tempo. Converter infinitos valores analógicos de tensão em palavras digitais produz uma quantidade infinita de dados, tornando inviável sua implementação, armazenamento e processamento destes dados. Por isso, amostras de tensão do sinal analógico são coletadas em um instante de tempo tendendo a zero, como um ponto em um gráfico de tensão por tempo, e este procedimento se repete em um intervalo de tempo fixo e determinado, conhecido como tempo de amostragem. A cada amostra do sinal analógico coletado é associada a uma palavra digital dentro de uma faixa de valores que varia de 0 a $2^n - 1$, onde n é o tamanho da palavra em *bits*. Desta forma, a faixa de variação da tensão do sinal analógico é dividida em 2^n degraus de tensão e cada um destes degraus equivale a uma palavra digital. Neste processo, conhecido como quantização, a tensão de uma amostra do sinal analógico é retida por um tempo suficiente para ser enquadrada em um dos degraus de tensão e, conseqüentemente, a amostra é associada a uma palavra digital, conforme pode ser observada na Figura 4 onde k é o número da amostra e w a palavra digital resultante deste processo. Portanto, estes procedimentos constituem o processo de conversão A/D.

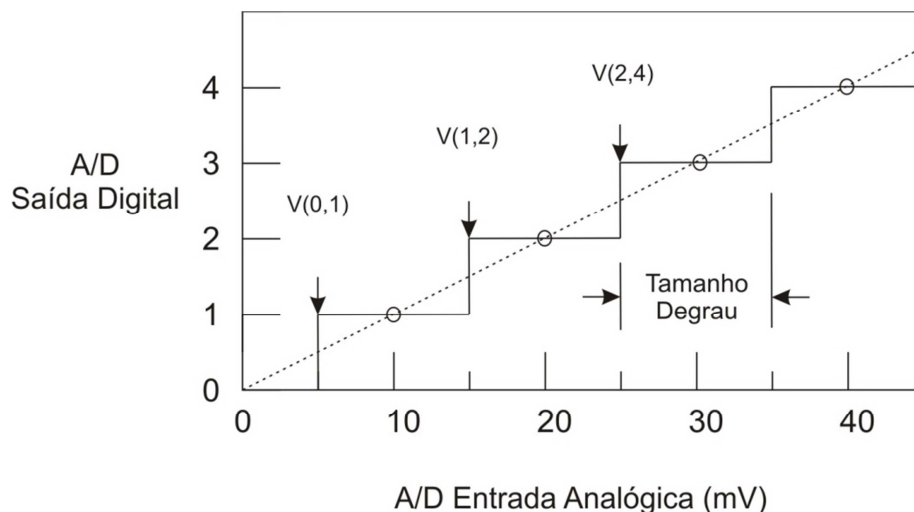


Figura 4 - Processo de quantização de um sinal analógico amostrado $V(k,w)$.
Fonte: Pazos (2002).

O processo de conversão A/D, intrinsecamente, produz erros. Um deles é o erro de quantização, que é a diferença de tensão entre a tensão da amostra do sinal analógico e a tensão média do degrau (ver Figura 4). Quanto maior a resolução do conversor A/D, ou seja, o tamanho da palavra digital resultante da conversão, menor será a amplitude de tensão do degrau e, conseqüentemente, menor será o erro de quantização. Outro problema existente é a não linearidade, devido ao fato de não conseguir-se implementar degraus com mesmo valor de tensão. Na prática, o tempo necessário para que uma amostra de tensão do sinal analógico seja retido (processo de retenção – *holding*) não é idealmente zero, contribuindo para o erro total do processo de conversão A/D. Além disso, o tempo necessário para que uma conversão A/D ocorra limita a máxima frequência de amostragem e, conseqüentemente, restringe a frequência máxima que o sinal a ser convertido pode ter. Também vale observar a sensibilidade do conversor A/D em relação às variações da fonte que o alimenta e o erro de *zero-offset* que é a diferença entre o primeiro nível de transição e 0,5 LSB (*Least Significant Bit*) bit menos significativo (PAZOS, 2002).

2.6 RELÓGIO DE TEMPO REAL

O Relógio de Tempo Real (RTC) é um circuito integrado que fornece informações de tempo para um circuito mestre que necessite destes dados como, por exemplo, um microcontrolador. A base de tempo é fornecida por um sinal de um oscilador, cuja frequência é determinada por um cristal de quartzo. Mesmo após o desligamento da alimentação deste dispositivo, o RTC se mantém em funcionamento através de uma bateria externa. As principais características dos RTCs são (NXP, 2012):

- Contagem de minutos, segundos, horas, dia do mês, mês, dia da semana e ano com compensação de ano bissexto;
- Memória RAM interna não volátil para armazenamento de dados;
- Detecção de falhas;
- Interface serial;
- Calendário;
- Alarme de tarefas.

2.7 MEMÓRIA FLASH

As memórias *flash* são do tipo não-volátil, pois as informações armazenadas não são perdidas quando há interrupção do fornecimento de energia. A combinação entre as memórias EPROM (*erasable programmable read-only memory*) e EEPROM (*Electrically-Erasable Programmable Read-Only Memory*) resultou na memória *flash*, também conhecidas como EEPROM *flash*. Este tipo de memória une a capacidade de ser eletricamente apagável e programável, como as memórias EEPROM, e o fato de sua célula requerer somente um transistor, semelhante às memórias EPROM. Além disso, possui alta densidade e baixo custo (PEDRONI, 2010).

O tempo de apagamento e escrita das memórias *flash* são curtos. Existem memórias *flash* que usam uma operação de apagamento no qual todas as células do

chip são apagadas ao mesmo tempo. Tipicamente, este processo necessita centenas de milissegundos para ser executado. Memórias *flash* mais modernas podem ser apagadas por setor, ou seja, partes específicas da matriz de memória podem ser apagadas, evitando ter de apagar e reprogramar todas as células quando há necessidade de atualizar apenas uma parte da memória. A velocidade de escrita de uma memória *flash* típica é de 10 μ s por *byte* (TOCCI, WIDMER e MOSS, 2007).

Tocci, Widmer e Moss (2007) explicam que os dispositivos de memória flash atuais não diferem muito dos de primeira geração, pois a arquitetura do núcleo da memória e o conjunto básico de códigos de comando são muito semelhantes, porém, as memórias mais modernas possuem novos recursos e novos códigos de comando para controlar esses recursos, por exemplo, dispositivos atuais possuem muito mais capacidade, consomem menos energia, são menores em tamanho físico e o custo por *bit* é menor que os antecessores. Além disso, outro diferencial é a capacidade de ler e escrever dados no mesmo momento que outro bloco de memória está sendo apagado, também permite que se utilize uma única fonte de alimentação pois a tensão de programação é gerada internamente (TOCCI, WIDMER e MOSS, 2007).

3. MATERIAIS E MÉTODOS

Neste Capítulo, o projeto do termo registrador microcontrolado proposto será descrito, seu funcionamento e as partes que o compõe serão apresentadas, assim como os materiais utilizados e a metodologia empregada no desenvolvimento. A Figura 5 apresenta um diagrama em blocos do termo-registrador.

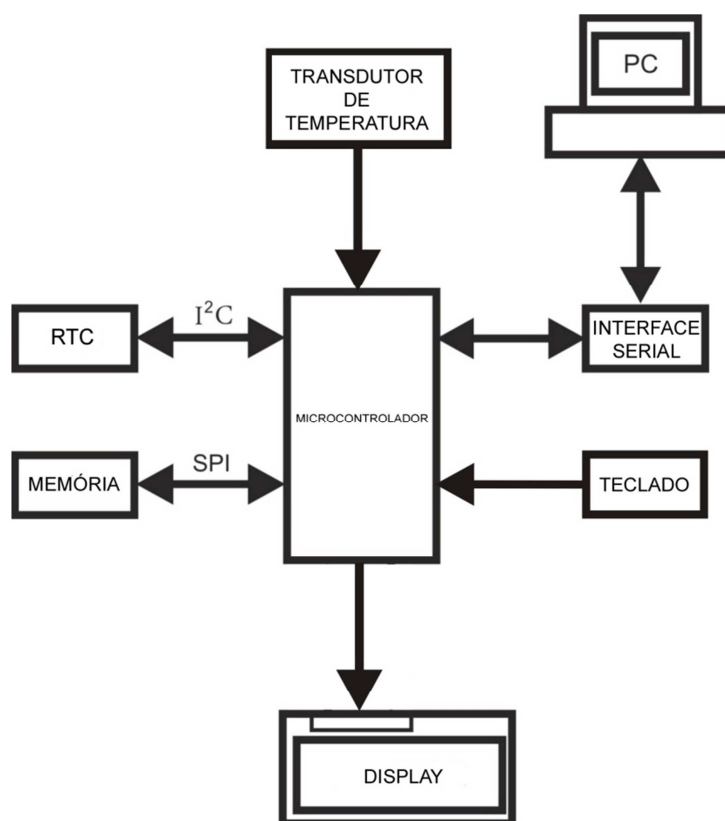


Figura 5–Diagrama em blocos do Termo registrador Digital Microcontrolado.

Fonte: Autoria própria.

De acordo com a Figura 5, o elemento central do termo registrador é um microcontrolador, cuja função é efetuar a leitura de temperatura do leite, através de um transdutor de temperatura, armazenar esta informação em uma memória *flash* juntamente com as informações fornecidas pelo RTC, e apresentar em um *display*. O sistema de *download* das informações contidas na memória *flash* será feito através de uma comunicação serial e salvas em forma de arquivo de texto no computador.

3.1 MATERIAIS UTILIZADOS

3.1.1 Placa de Desenvolvimento MSP-EXP430G2 *Launchpad Experimental Board*

O MSP-EXP430G2 *LaunchPad Experimenter Board* que é mostrado na Figura 6 é um kit de desenvolvimento de sistemas microcontrolados de baixo custo utilizando microcontroladores da família MSP430 da Texas Instruments. Este kit possui um emulador USB (*Universal Serial Bus*) integrado e oferece recursos de *hardware* necessários para desenvolver aplicações baseadas nos dispositivos da série MSP430G2xx e MSP430F20xx de microcontroladores (TEXAS INSTRUMENTS, 2012).

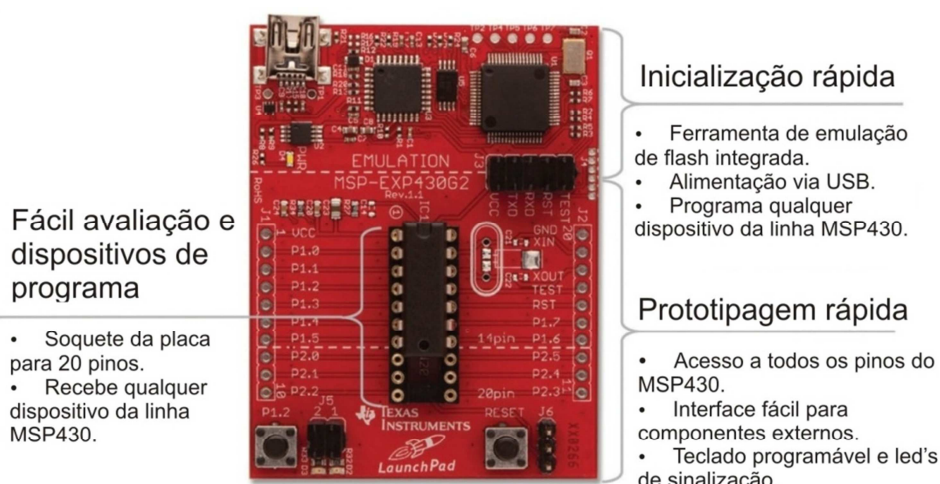


Figura 6 - kit de desenvolvimento *Launchpad* para MSP430

Fonte: Adaptado de Texas Instruments (2012).

O kit *LaunchPad* possui um conector DIP (*Dual In-line Package*) em sua placa que suporta microcontroladores da família MSP430G2xx com até 20 pinos conectores para conexão dos pinos do microcontrolador em circuitos externos, dois leds, 2 teclas, e oferece uma ferramenta integrada de emulação que permite a gravação do código desenvolvido no microcontrolador (programação), a depuração deste código que está gravado no dispositivo e a emulação de uma interface serial através da conexão USB com o computador (TEXAS INSTRUMENTS, 2012).

3.1.2 Ferramenta de desenvolvimento de código.

O Code Composer Studio® (CCStudio) é um ambiente de desenvolvimento integrado para a Texas Instruments (TI) e incorpora várias famílias de processadores. O CCStudio incorpora diversas ferramentas usadas para desenvolver e depurar sistemas embarcados. Ele possui compiladores para todas as famílias de microcontroladores da TI, editor de código fonte, ambiente de projeto de construção, depurador, perfilador, simuladores do sistema operacional em tempo real e diversos outros recursos. O ambiente de desenvolvimento oferece uma interface de usuário que possibilita percorrer através de cada etapa. O programa possui estrutura baseada no *software* de código aberto Eclipse que foi originalmente desenvolvido como uma estrutura aberta para a criação de ferramentas de desenvolvimento (TEXAS INSTRUMENTS, 2013).

A ferramenta de desenvolvimento CCStudio possui compiladores C/C++ adaptados para arquiteturas embarcadas. Estes processam sinal digital e procuram tirar o máximo proveito de desempenho dessas arquiteturas (TEXAS INSTRUMENTS, 2013).

3.1.3 O Microcontrolador MSP430F2234

O microcontrolador utilizado no projeto é o dispositivo MSP430F2234, fabricado pela empresa Texas Instruments, que foi escolhido por ser o dispositivo utilizado nas matérias durante o curso, pelo seu baixo custo, facilidade de acesso as ferramentas de desenvolvimento, baixo consumo de energia, conversor analógico digital integrado, disponibilidade no mercado e por possuir as características de desempenho necessárias para o funcionamento do projeto. A Figura 7 apresenta o diagrama em blocos interno deste microcontrolador, enquanto a Figura 8 mostra a função dos seus pinos. As principais características deste dispositivo são (TEXAS INSTRUMENTS, 2011):

- Baixa tensão de alimentação: 1,8V a 3,6V;
- Baixo consumo de energia: modo ativo 270 μ A com frequência de 1MHz em 2.2V;

- Sai do modo *Stand By* em menos de 1 μ s;
- Arquitetura RISC de 16 Bits;
- Tempo de ciclos de instrução de 62,5 ns;
- Sistema de *clock* com as seguintes características:
 - Frequência interna de até 16 Mhz com quatro frequências de calibração com tolerância de +- 1%;
 - Oscilador interno de baixa frequência e baixo consumo de energia;
 - Oscilador de baixa frequência baseado em cristal de 32.678Hz ou ressonador;
 - Aceita fonte de *clock* externa.
- *Timer A* de 16 *Bits* com três canais;
- *Timer B* de 16 *Bits* com três canais;
- Interface de comunicação serial universal que atende os seguintes protocolos:
 - UART com detecção automática de *Baudrate*;
 - IrDA com codificador e decodificador;
 - SPI;
 - I²C.
- Conversor A/D de 10 *Bits* com referência interna, *autoscan* e controle de transferência de dados;
- Dois amplificadores operacionais configuráveis;
- Programação via interface serial específica;
- Não necessita de tensão de programação externa e de proteção de código por fusível de segurança;
- Módulo de emulação *On-Chip*;
- Memória *Flash* de 8KB (programa) + 256B (dados);
- Memória RAM de 512B.

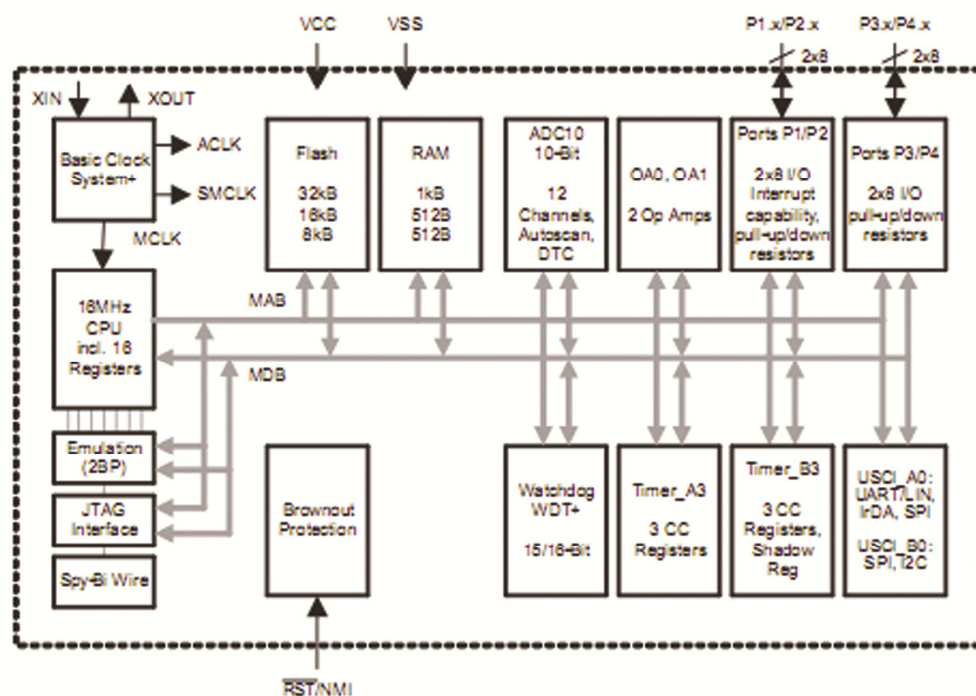


Figura 7 - Diagrama de blocos do microcontrolador MSP430F2234.

Fonte: Texas Instruments (2011).

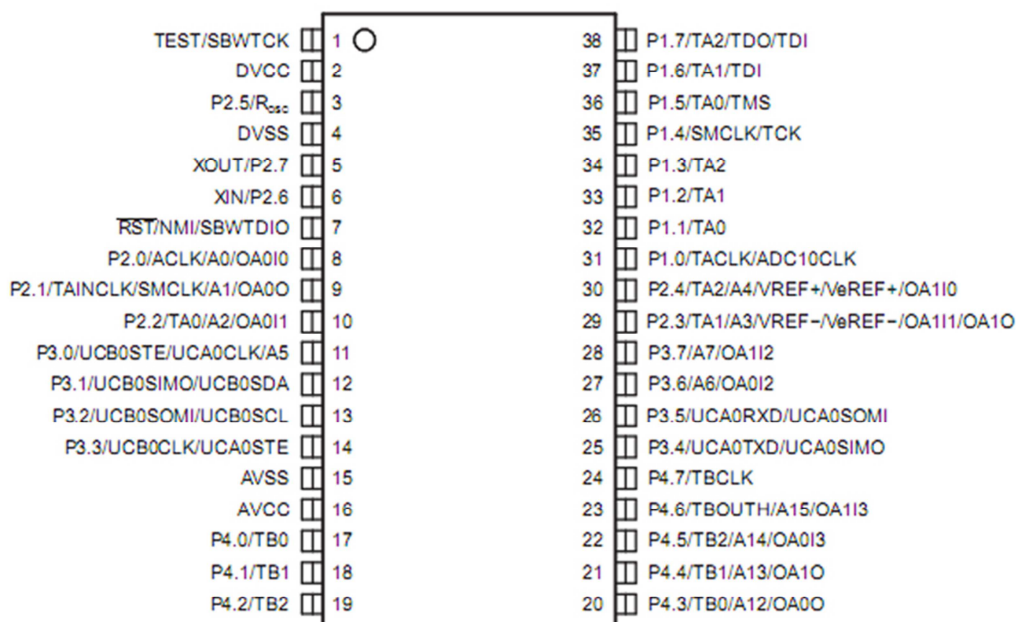


Figura 8 - Especificação dos pinos no MSP430F2234

Fonte: Texas Instruments (2011).

3.1.4 Memória *Flash* AT25DF641

A Memória *Flash* modelo AT25DF641, fabricada pela ATMEL, é um dispositivo criado para ser aplicado onde há necessidade de grande capacidade de armazenamento de dados. Este foi um dos motivos para escolha deste dispositivo, incluindo seu baixo custo e por possuir um pino que habilita e desabilita a comunicação com a memória. Esta memória proporciona grande flexibilidade, pois suas matrizes de memória podem ser apagadas em quatro níveis de granularidade, ou seja, apagando em blocos de 4 *kbytes* blocos de 32 *kbytes*, blocos de 64 *kbytes* e também um nível de apagamento completo. Suas matrizes são divididas em setores físicos de tamanho uniforme, contendo 128 setores de 64 *kbytes* (total de $128 * 64k = 8MB$) cada, sendo que cada setor pode ser protegido individualmente contra operações do programa e apagamento indevido. O tamanho dos setores físicos são otimizados para permitir que código e aplicações de armazenamento de dados residam em suas próprias regiões isoladas (ATMEL, 2010). A Tabela 2 apresenta as especificações dos pinos da memória. As principais características desta memória são (ATMEL, 2010):

- Interface serial SPI com operação em alta frequência (até 75 MHz);
- Arquitetura flexível de apagamento para o código e aplicações de armazenamento de dados;
- Setor de proteção individual ou função Proteger/Desproteger Global;
- Proteção de setores via *hardware* controlado pelo pino WP/;
- Setor de bloqueio;
- 128 bytes programáveis no registrador de segurança OTP;
- Checagem automática de falhas de programação e apagamento;
- Reset controlado via *Software*;
- Baixa potência de dissipação: 5 mA em atividade de leitura a 20MHz;
- Durabilidade: 100.000 ciclos de programação/apagamentos;
- Retém os dados gravados por 20 anos;
- Suporta faixas de temperaturas com parâmetros industriais;
- Soquete SOIC (*Svenska Ostindiska Companiet*) de 16 pinos.

Tabela 2 - Especificação dos pinos da memória AT25DF641

CS/	Seleciona o dispositivo para iniciar ou terminar uma operação.	
SCK	Este pino é utilizado para fornecer <i>clock</i> para o dispositivo e utilizado para controlar o fluxo de dados.	
SI (SIO)	Entrada serial, usado para transferir dados para o dispositivo.	
SO (SOI)	Saída serial, usado para transferir dados para fora do dispositivo.	
WP/	Controla o bloqueio de <i>hardware</i> do dispositivo.	
HOLD/	Usado temporariamente para pausa de comunicação serial.	
V _{CC}	Alimentação do dispositivo.	
GND	Referência de terra para a fonte de alimentação.	

Fonte: Adaptado de Atmel (2010).

3.1.5 RTC PCF8563

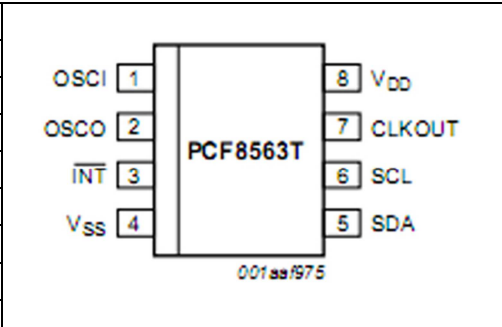
O PCF8563 é um RTC (*Real Time Clock*) com calendário e otimizado para baixo consumo de energia. Possui saída de relógio programável, saída de interrupção e detector de baixa tensão. Os endereços e os dados são transferidos através de uma interface de comunicação serial no padrão I²C, cuja velocidade máxima de transferência é de 400 *Kbit/s*. O endereço de registro é incrementado automaticamente após cada *byte* de dados escrito ou lido (NXP, 2012). A Tabela 3 mostra as especificações dos pinos do RTC8563. As principais características do dispositivo PCF8563 são (NXP, 2012):

- Fornece informação de ano, mês, dia, dia da semana, horas, minutos e segundos com base na oscilação de frequência de um cristal de quartzo de 32768 KHz.
- Tensão de operação do *clock*: de 1V a 5,5V à temperatura ambiente.

- Corrente de *Backup*¹ baixa: 0,25 μ A com alimentação de 3V e temperatura ambiente de 25°C.
- Comunicação serial I²C a dois fios, com frequência de 400 KHz;
- Saída de *clock* programável para dispositivos periféricos (32.678 Hz, 1024 Hz, 32 Hz e 1 Hz);
- Funções de alarme e temporizador;
- Capacitor oscilador integrado;
- Circuito de *reset* interno;
- Endereço do escravo I²C para leitura de A3h e de escrita A2h.

Tabela 3 - Especificação dos pinos do RTC8563

PINO	FUNÇÃO
OSCI	Entrada do oscilador.
OSCO	Saída do oscilador.
INT/	Saída de interrupção.
V _{SS}	Referência de terra.
SDA	Entrada e saída de dados serial.
SCL	Entrada de <i>clock</i> para comunicação serial.
CLKOUT	Saída de <i>clock</i> .
V _{DD}	Alimentação do dispositivo.



Fonte: Adaptado de NXP (2012).

3.1.6 Dispositivo de interface de linha padrão RS-232

O dispositivo MAX3222 consiste em dois *driver* e dois receptores de linha, conforme pode ser observado na Figura 9, possuindo dois circuitos de bomba de carga (*charge pump*) usando capacitores para prover as tensões de linha, conforme mostra a Figura 10, com proteção contra descargas eletrostáticas pino a pino para até +- 15 kV. Estes dispositivos operam com taxas de envio de dados de até 250 *kbit/s*, com *slew rate* (taxa de variação) de 30 V/ μ s no *driver* de saída. O MAX3222 pode ser colocado no modo *Power-Down* (desligado) definindo pelo pino PWRDOWN em nível baixo, que consome apenas 1 μ A da fonte de alimentação. Quando o dispositivo está ligado, os receptores permanecem ativos enquanto os

¹Corrente de *Backup*: corrente necessária para manter o tempo quando a alimentação principal falhar.

drivers são colocados em estado de alta impedância. Além disso, quando desligado, a bomba de carga existente é desativada, $V+$ é rebaixado para V_{cc} e $V-$ é elevado para nível GND (*Ground*). A saída do receptor também pode ser colocada no estado de alta impedância, definindo $\overline{EN/}$ para nível alto (TEXAS INSTRUMENTS, 2004).

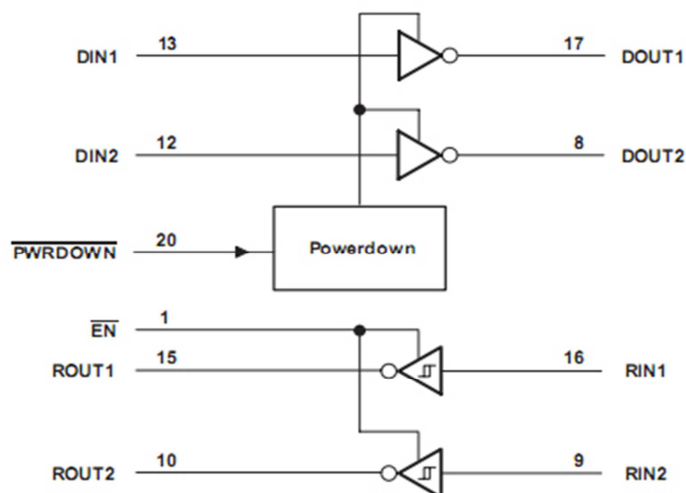


Figura 9 - Portas lógicas do MAX3222.

Fonte: Texas Instruments (2004).

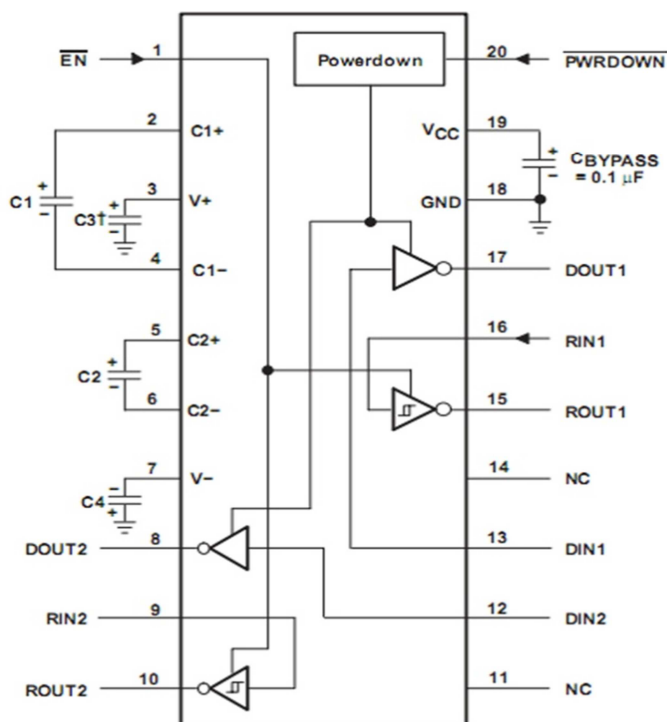


Figura 10 - Conexões externas do MAX3222

Fonte: Texas Instruments (2004).

3.1.7 Display de LCD

O *display* de LCD (*liquid-crystal display*) é uma interface de saída que pode ser gráfica ou a caractere. Os módulos gráficos podem ser encontrados com resoluções de 122x32, 128x64, 240x64 e 240x127 *dots pixel*, e geralmente possuem 20 pinos para interfaceamento (INTECH, 1996). Os *displays* LCD do tipo caractere são especificados pelo seu número de linhas e colunas, conforme a Tabela 4.

Tabela 4 - Módulos LCD a caractere disponíveis no mercado

Número de colunas	Número de Linhas	Quantidade de Pinos
8	2	14
12	2	14/15
16	1	14/16
16	2	14/16
16	4	14/16
20	1	14/16
20	2	14/16
20	4	14/16
24	2	14/16
24	4	14/16
40	2	16
40	4	16

Fonte: Adaptado de Intech (1996).

O *display* LCD possui um controlador próprio que permite sua interligação através de pinos a outra placa, estes pinos permitem a alimentação e a interligação dos barramentos de dados e controle com o circuito do usuário. Para que aconteça a comunicação do circuito do usuário com o *display* também é necessário um protocolo de comunicação que envia *bytes* de instruções e *bytes* de dados.

Para facilitar a leitura em ambientes escuros, alguns módulos LCD possuem *LED backlight*, que é uma iluminação de fundo. A corrente consumida pelo LED ligado varia de 100 a 200 mA, dependendo do modelo (INTECH, 1996). Para o

projeto foi escolhido um *display* LCD de dezesseis colunas e duas linhas 16X2, com *LED backlight* e 16 pinos para interfaceamento.

3.2 METODOLOGIA

3.2.1 Desenvolvimento de soquete adaptador para o MSP430F2234

Utilizando o *software* EAGLE versão 5.6.0 foi desenvolvido um soquete para adaptar o Microcontrolador MSP430F2234, conforme mostra a Figura 11, pois este possui encapsulamento SMD do tipo TSSOP de 38 pinos e não permite encaixe em *protoboard*.

O *software* EAGLE é um programa utilizado para confeccionar desenho de placa de circuito impresso (PCI ou PCB), utilizando as bibliotecas de componentes disponível no programa se constrói um esquema elétrico que serve de base para o projeto do circuito impresso.

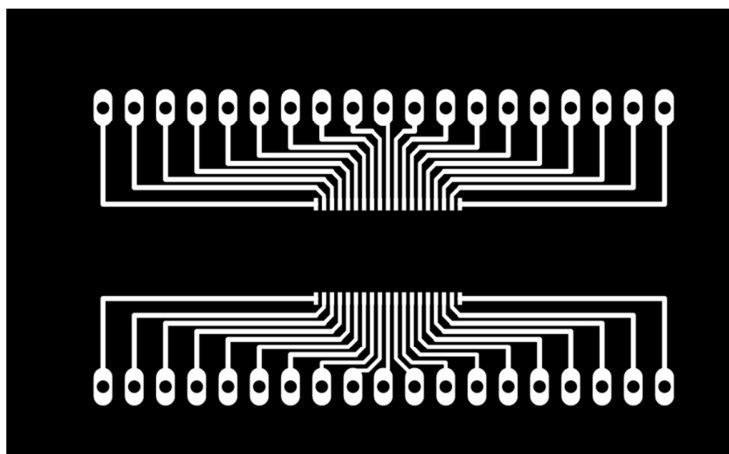


Figura 11 - Soquete Adaptador para MSP430F2234.

Fonte: Autoria própria.

3.2.2 Conexão do circuito de emulação do kit *Lauchpad* ao microcontrolador MSP430F2234

Para possibilitar a gravação e a depuração dos programas ao microcontrolador MSP430F2234 utilizando o kit *Lauchpad Experiment Board*, foi adaptado um cabo de 6 vias para interligação das linhas de comunicação e níveis de tensão na *proto board* onde o microcontrolador foi alocado.

3.2.3 Interfacemanto com o *Display*.

O *display* LCD 16x2 possui um microcontrolador em sua placa que, através de seus pinos de entrada e saída, permite a configuração e a entrada e saída de dados para que o mesmo comunique com o microcontrolador MSP430 e apresente os dados desejados em sua tela. Para fazer a escrita no *display* utiliza-se os códigos de caracteres da tabela ASCII. O *trimpot* é utilizado como divisor de tensão para regular a intensidade da luz de *backlight*. O diagrama esquemático de interligação do *display* com microcontrolador é apresentado na Figura 12.

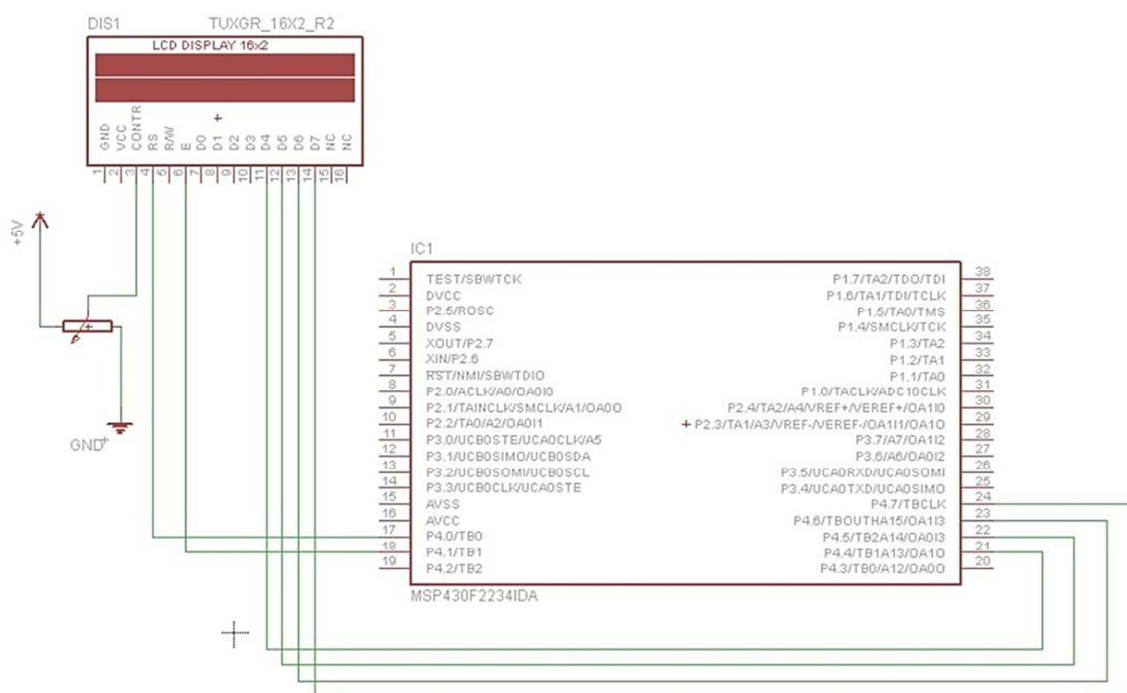


Figura 12 - Circuito de interfaceamento com o *display*.

Fonte: Autoria própria.

3.2.4 Desenvolvimento de circuito de condicionamento de sinal para o transdutor PT100

Para fazer a aquisição do sinal resistivo do PT100, utilizou-se uma ponte de Wheatstone que fornece um valor de tensão diretamente proporcional a variação da resistência, mas devido a pequena variação de tensão fornecida pelo circuito um Amplificador Operacional de instrumentação modelo AD623 foi utilizado com um ganho igual a 2 que dobrar o valor da tensão e tornando possível a leitura dessa tensão pelo conversor A/D. O circuito foi montado conforme Figura 13.

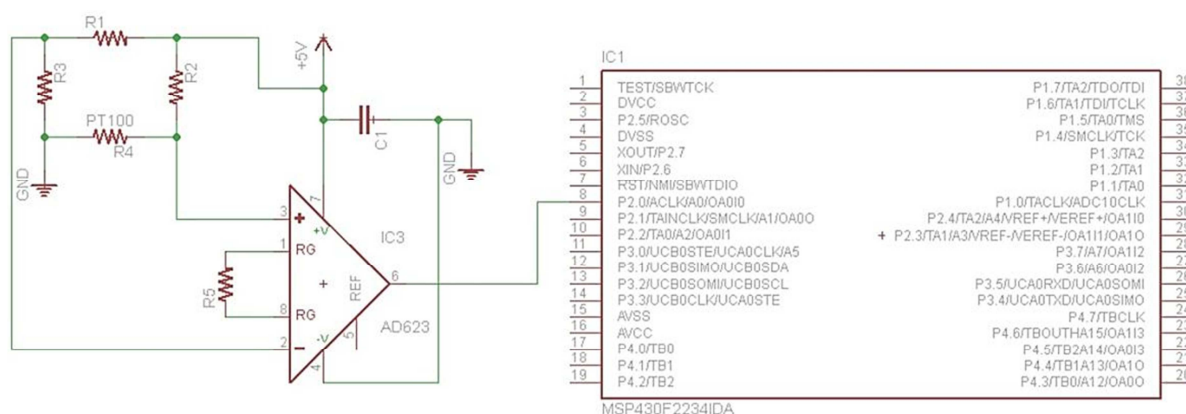


Figura 13 - Circuito de condicionamento de sinal do transdutor PT100

Fonte: Autoria própria.

3.2.5 Desenvolvimento do circuito do RTC e interfaceamento com o microcontrolador

O circuito do RTC foi desenvolvido sobre uma Placa de Circuito Impresso (PCI) com matriz de furos. Utilizou-se uma bateria de lítio de 3V para manter o funcionamento do RTC na ausência de alimentação e um cristal de quartzo de 32,768 KHz para fornecer o *clock* necessário para o RTC. Como o RTC8563 utiliza comunicação serial I²C a dois fios para fornecer os dados gerados ao microcontrolador, foi necessário inserir dois resistores de *pull-up*, um na linha SDA e outro na linha SCL, conforme mostra o esquema da Figura 14.

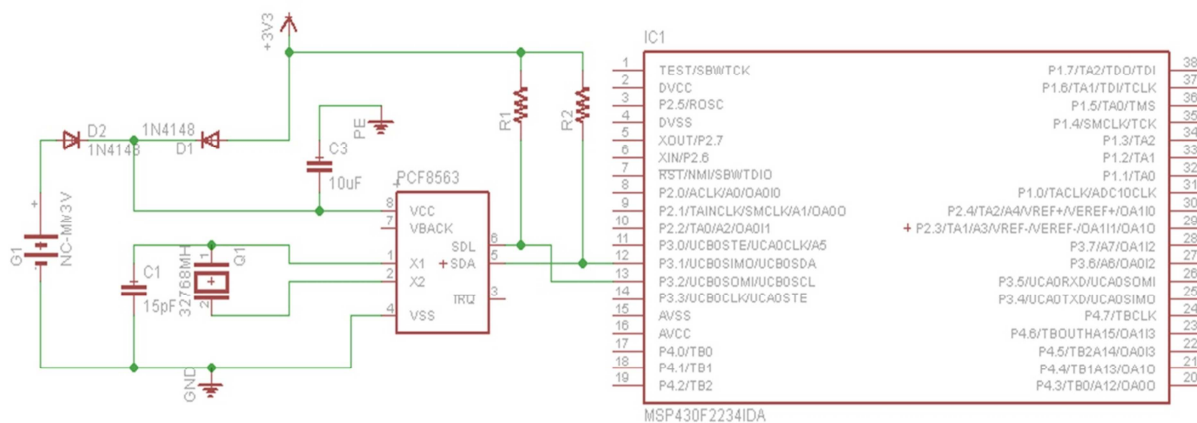


Figura 14–Esquema elétrico do circuito RTC interligado ao microcontrolador

Fonte: Autoria própria.

3.2.6 Interfaceamento com teclado

Para o ajuste das configurações iniciais do RTC como data e hora, foram implementados três chaves (*push-button*) interligados no microcontrolador, conforme o circuito da Figura 15. Assim quando acionar S1 terá como função selecionar uma função, S2 incrementar e S3 decrementar um valor.

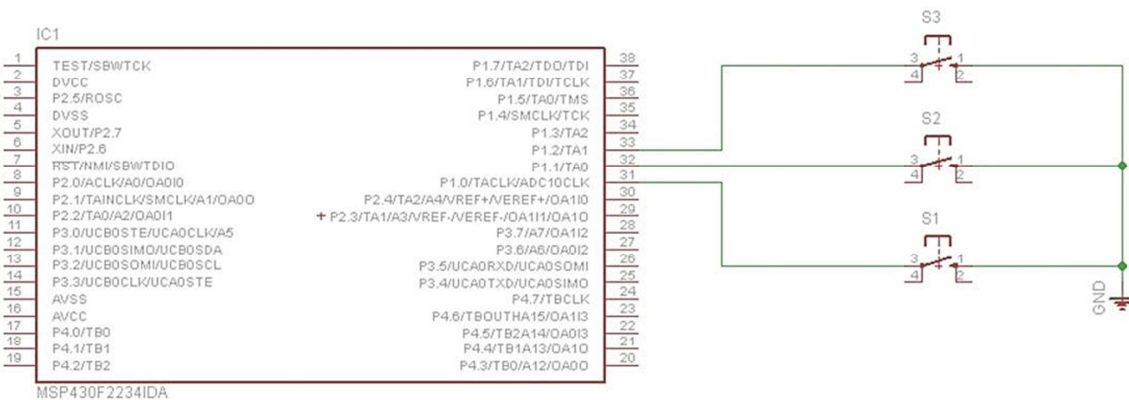


Figura 15 - Ligação dos botões ao microcontrolador.

Fonte: Autoria própria.

3.2.7 Interfaceamento com memória externa

Para a memória AT25DF641, que possui um encapsulamento SMD do tipo SOIC de 16 pinos, foi confeccionado um adaptador, utilizando matriz de furos, para possibilitar o acoplamento da mesma na *protoboard*. O circuito de interligação da memória ao microcontrolador foi montado conforme a Figura 16.

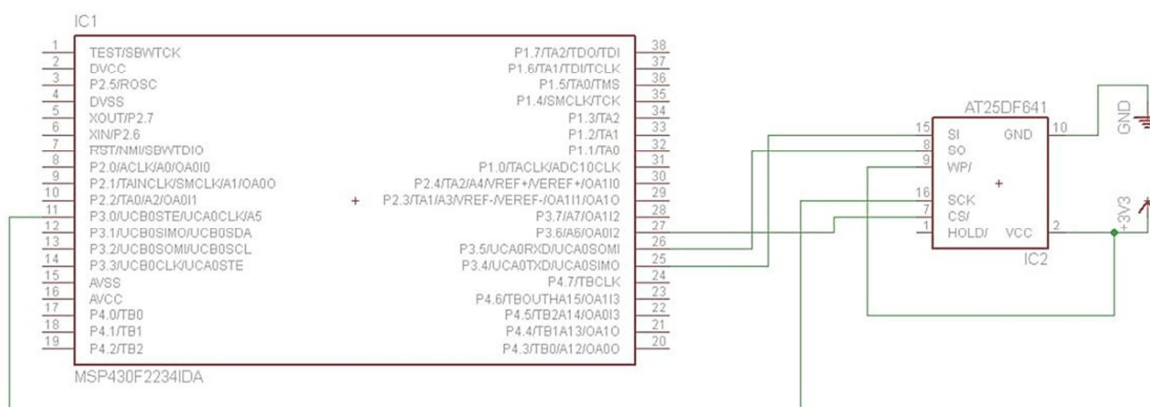


Figura 16 Circuito de conexão da Memória

Fonte: Autoria própria.

3.2.8 Desenvolvimento de circuito de interface de linha padrão RS232 e interfacemaneto com o microcontrolador

Utilizando o *software* EAGLE 5.6.0 foi confeccionado um circuito para o funcionamento do *DRIVER/RECEIVER* MAX3222, utilizando-se quatro capacitores, conforme apresentado na Figura 17.

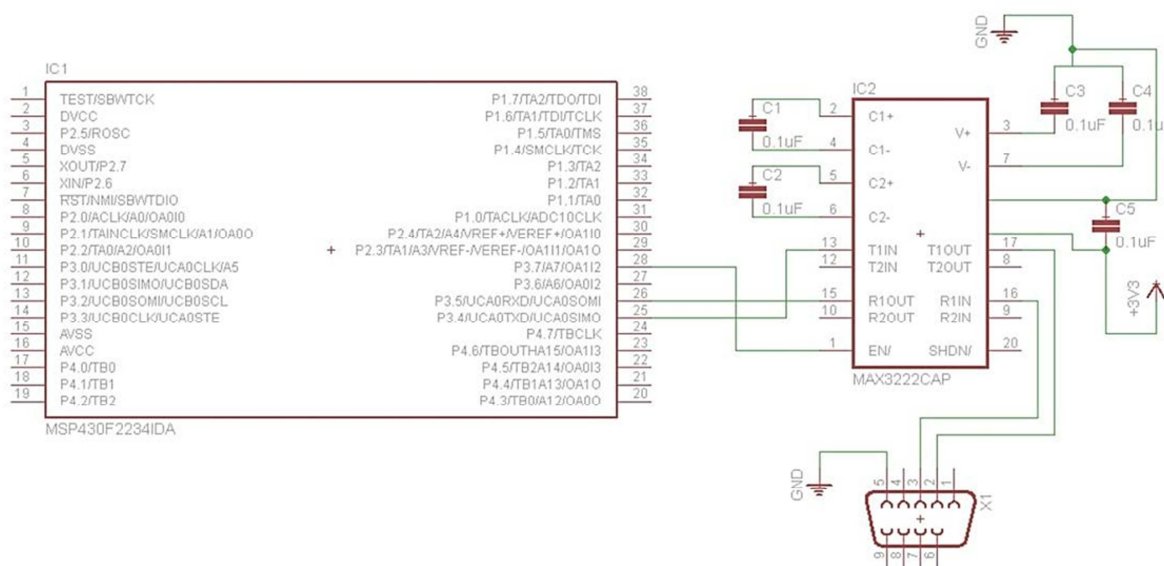


Figura 17 - Circuito do MAX3222

Fonte: Autoria própria.

Para estabelecer a compatibilidade com o padrão RS-232 utilizou-se o MAX3222, que além de ser um conversor de nível, oferece uma opção que permite ativar e desativar o circuito, quando desativado os *drivers* são colocados em estado de alta impedância, permitindo que a porta de comunicação serial do micro controlador seja utilizada por outro dispositivo. Esta opção foi a forma encontrada para conectar duas interfaces seriais com protocolos diferentes no mesmo periférico do microcontrolador. A interface de comunicação serial UART é configurada para o modo RS232 para envio de dados para o computador e reconfigurada para o modo SPI para permitir a escrita/leitura de dados na memória *flash*. Como a memória e o MAX3222 possuem pino de habilitação de dispositivo (*enable*), é possível desativar um dispositivo enquanto o outro está usando o barramento serial, evitando que conflitos ocorram. O diagrama do circuito de interfaceamento da memória e do MAX3222 com o microcontrolador são apresentados na Figura 18.

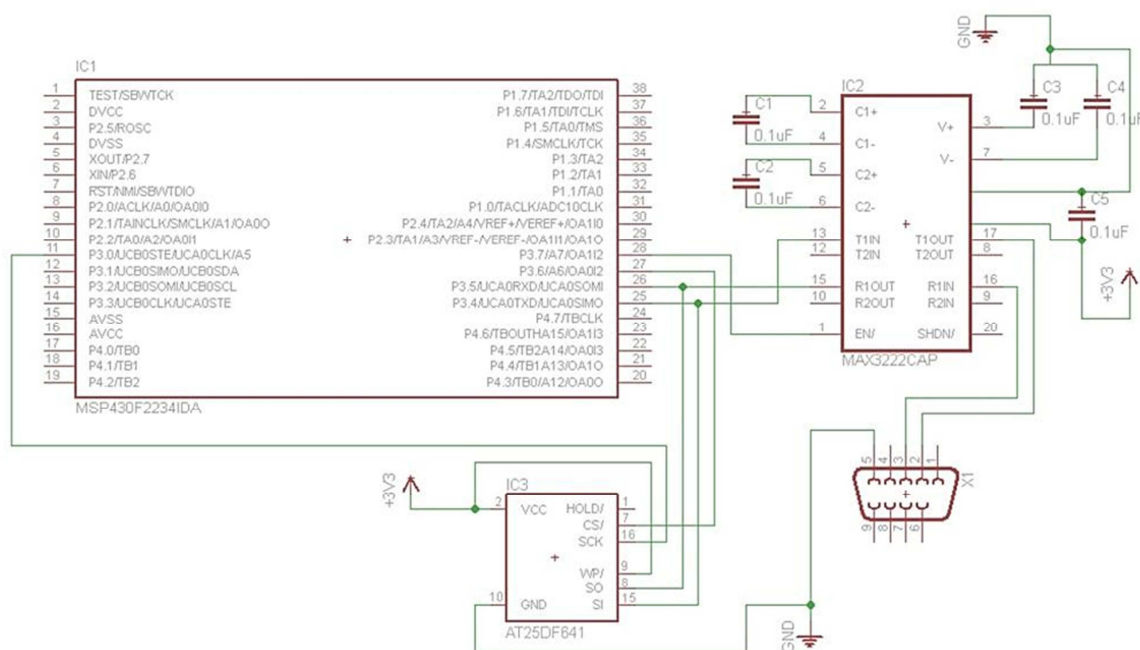


Figura 18 - Circuito de interfaceamento da memória e MAX3222 com o microcontrolador.

Fonte: Autoria própria.

4. RESULTADOS

Após adquiridos todos os componentes necessários, iniciou-se a confecção do soquete adaptador para o MSP430F2234 conforme item A da Figura 19, pois este possui encapsulamento TSSOP-38. Esta confecção possibilitou e facilitou a conexão do microcontrolador com a *protoboard*. Para comunicação e alimentação do microcontrolador com o kit *Launchpad* foi confeccionado um cabo de 6 vias conforme item B da Figura 19.

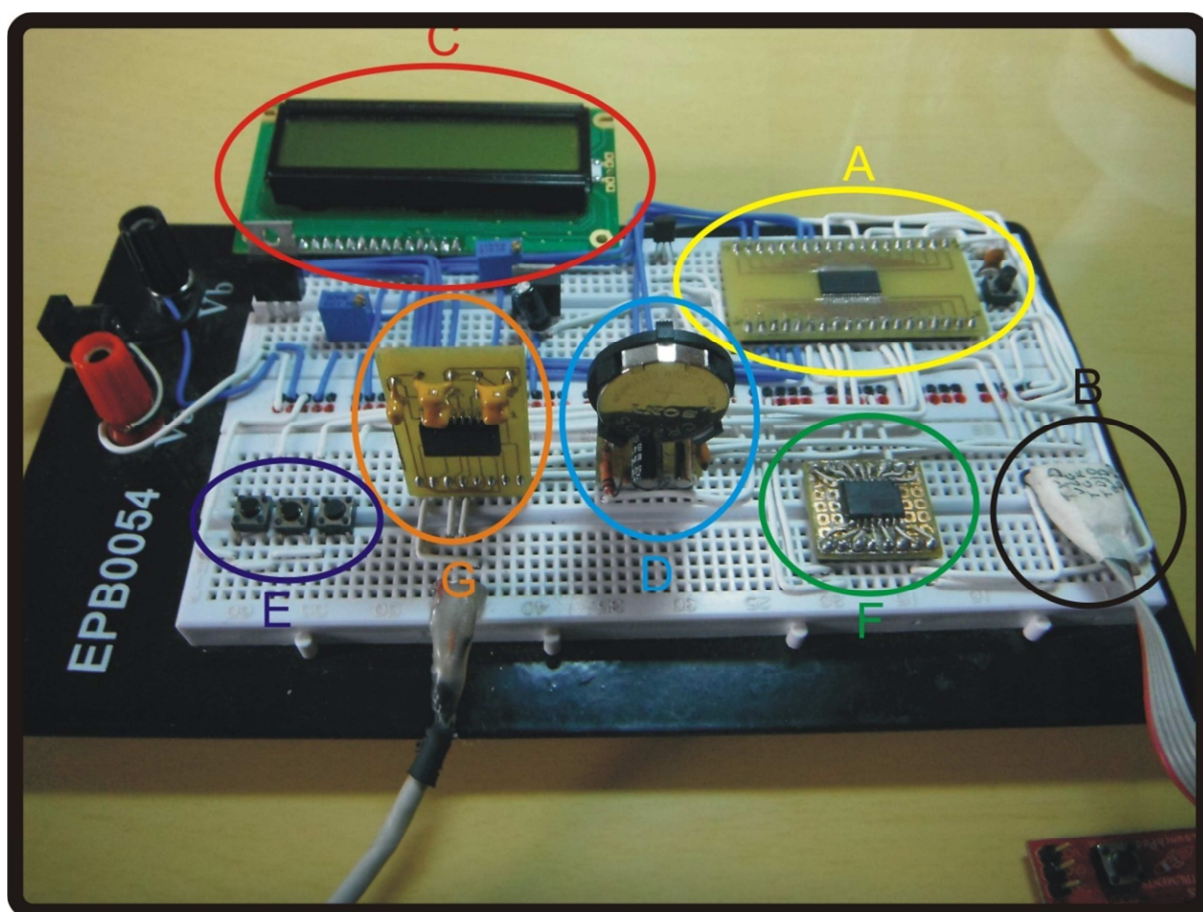


Figura 19 - Termo registrador digital montado em uma *protoboard*, onde o círculo indicado por A é o soquete adaptador para o MSP430F2234, B é o cabo para programação do microcontrolador, C é o *display*, D é o circuito do RTC, E são os *push-buttons*, F é a memória *flash* e G é o circuito do MAX3222

Fonte: Autoria própria.

Com a conexão entre *Launchpad* e o microcontrolador funcionando corretamente, possibilitou-se a programação e o interfaceamento dos periféricos ao microcontrolador. Sendo assim, o próximo passo dado foi interligar o *display* LCD ao

circuito conforme item C da Figura 19 e programá-lo com o protocolo de comunicação adequado, tendo como resultado a saída de dados em forma de caracteres.

Confeccionou-se, também, um circuito de instrumentação e condicionamento de sinal. Este circuito faz com que o sinal resistivo gerado pelo sensor seja convertido em sinal de tensão, permitindo a utilização desta tensão no circuito A/D do microcontrolador. O circuito pode ser visualizado na Figura 20.

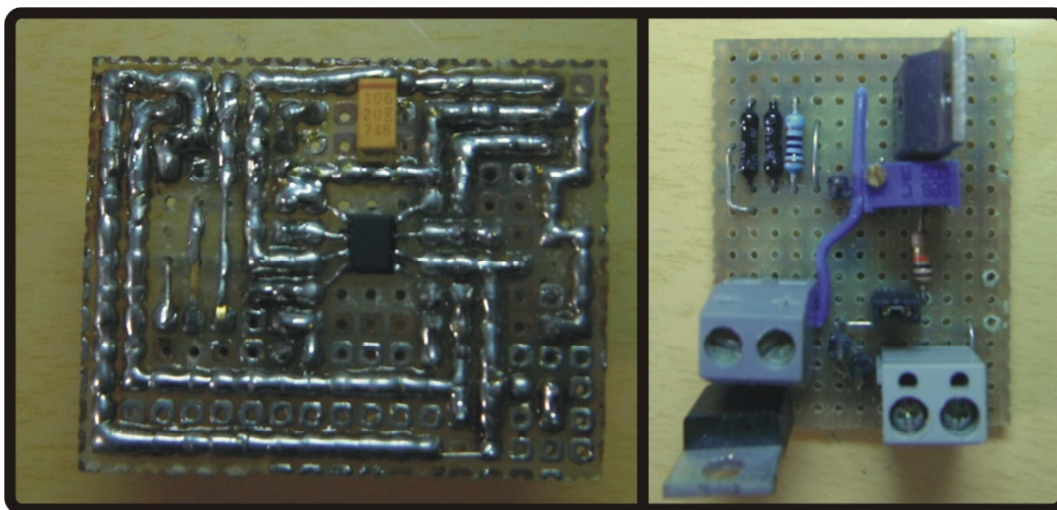


Figura 20 - Circuito de instrumentação e condicionamento de sinal.

Fonte: Autoria própria.

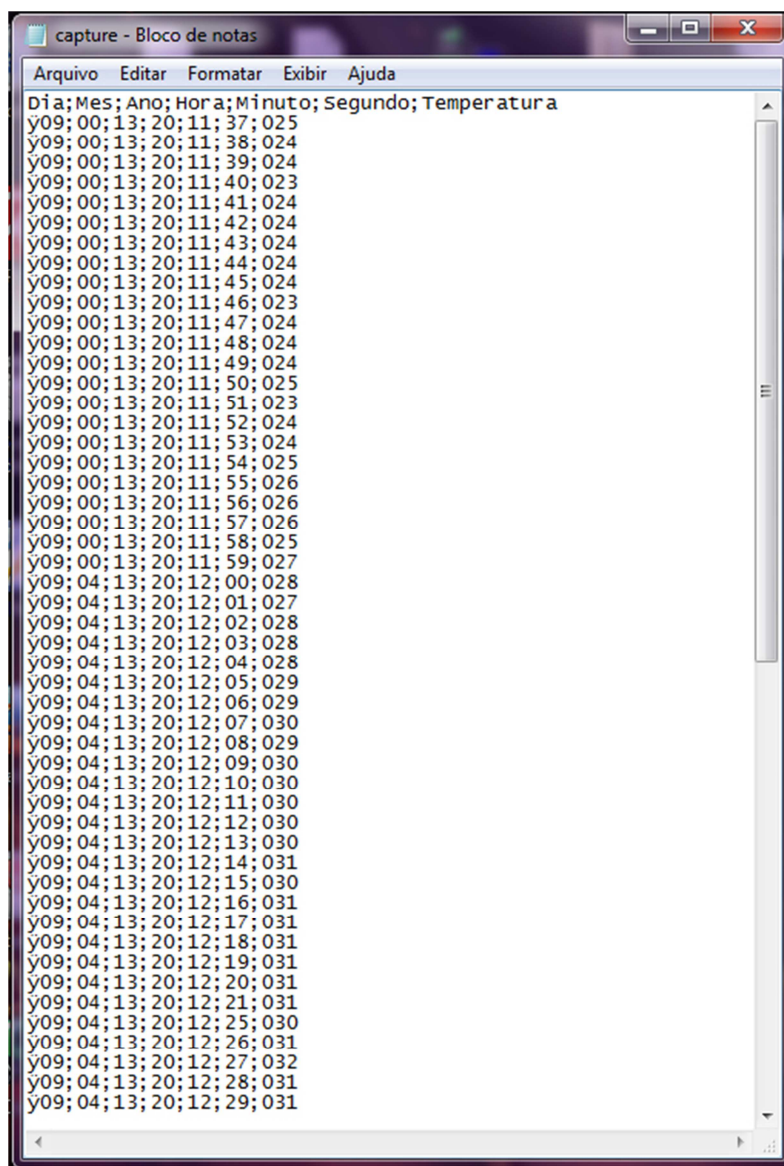
O RTC foi montado sobre uma placa de matriz com uma barra de pinos para facilitar a sua conexão ao *protoboard* conforme item D da Figura 19. Os pinos possibilitaram a comunicação de dados e a alimentação. Depois de integrado o RTC e programado, obteve-se os dados de dia, mês, ano, hora, minuto e segundo, necessários para identificar o período em que a coleta de dados de temperatura foi registrado.

Para que o RTC possa ter suas informações iniciais configuradas, incluiu-se ao circuito um teclado de 3 teclas, que foram montadas na *protoboard* conforme item E da Figura 19.

A memória *flash* externa foi adaptada sobre uma matriz de furos para facilitar sua conexão com o circuito conforme item F da Figura 19. Com o programa de depuração foi possível armazenar e ler os dados em endereços pré-determinados da memória *flash*.

Para o circuito de interface de linha padrão RS232 foi confeccionada uma placa de circuito impresso com o circuito integrado MAX3222 conforme item G da Figura 19. Este circuito possibilitou utilizar dois protocolos de comunicação, UART (descarga de dados em um computador) e SPI (comunicação da memória), em um mesmo módulo de comunicação serial do microcontrolador MSP430F2234.

O arquivo de texto gerado com os dados registrados em um período de teste pode ser visto na Figura 21.



```
capture - Bloco de notas
Arquivo Editar Formatar Exibir Ajuda
Dia;Mes;Ano;Hora;Minuto;Segundo;Temperatura
y09;00;13;20;11;37;025
y09;00;13;20;11;38;024
y09;00;13;20;11;39;024
y09;00;13;20;11;40;023
y09;00;13;20;11;41;024
y09;00;13;20;11;42;024
y09;00;13;20;11;43;024
y09;00;13;20;11;44;024
y09;00;13;20;11;45;024
y09;00;13;20;11;46;023
y09;00;13;20;11;47;024
y09;00;13;20;11;48;024
y09;00;13;20;11;49;024
y09;00;13;20;11;50;025
y09;00;13;20;11;51;023
y09;00;13;20;11;52;024
y09;00;13;20;11;53;024
y09;00;13;20;11;54;025
y09;00;13;20;11;55;026
y09;00;13;20;11;56;026
y09;00;13;20;11;57;026
y09;00;13;20;11;58;025
y09;00;13;20;11;59;027
y09;04;13;20;12;00;028
y09;04;13;20;12;01;027
y09;04;13;20;12;02;028
y09;04;13;20;12;03;028
y09;04;13;20;12;04;028
y09;04;13;20;12;05;029
y09;04;13;20;12;06;029
y09;04;13;20;12;07;030
y09;04;13;20;12;08;029
y09;04;13;20;12;09;030
y09;04;13;20;12;10;030
y09;04;13;20;12;11;030
y09;04;13;20;12;12;030
y09;04;13;20;12;13;030
y09;04;13;20;12;14;031
y09;04;13;20;12;15;030
y09;04;13;20;12;16;031
y09;04;13;20;12;17;031
y09;04;13;20;12;18;031
y09;04;13;20;12;19;031
y09;04;13;20;12;20;031
y09;04;13;20;12;21;031
y09;04;13;20;12;25;030
y09;04;13;20;12;26;031
y09;04;13;20;12;27;032
y09;04;13;20;12;28;031
y09;04;13;20;12;29;031
```

Figura 21- Arquivo com os dados registrados em um período de teste.

Fonte: Autoria própria.

5. CONCLUSÕES

A execução do projeto exigiu conhecimentos de diversas disciplinas estudadas durante o curso, tais como eletricidade, programação aplicada, circuitos elétricos, dispositivos eletrônicos, métodos e técnicas de pesquisa, sistemas de comunicação de dados, sistemas digitais, eletrônica aplicada, sistemas microprocessados, aquisição de sinais, microcontroladores, controle de processos, e também incluindo os conhecimentos adquiridos em outras disciplinas não citadas.

A instrução normativa 51 do Ministério da Agricultura foi de difícil acesso e compreensão, sendo necessário a consulta de um especialista para fundamentação dos procedimentos que foram adotados.

Encontrou-se algumas dificuldades na programação do microcontrolador, principalmente nos protocolos de comunicação serial. Tais dificuldades foram resolvidas com estudos mais aprofundados no assunto.

Os objetivos foram alcançados, podendo ainda o projeto ser melhorado com a inclusão de circuitos de proteção para que quando utilizado em ambiente industrial não ocorram alterações nos resultados desejados devido interferências geradas por equipamentos externos.

REFERÊNCIAS

- ATMEL. Datasheet AT25DF641. Disponível em: <<http://www.atmel.com/Images/doc3680.pdf>>. Acesso em: abril 2012.
- DALLAS SEMICONDUCTOR. Datasheet RTC DS1307. Disponível em: <<http://www.sparkfun.com/datasheets/Components/DS1307>>. Acesso em: agosto 2012.
- INTECH. **Manual display LCD**. 1996.[s.n.], 1996.
- MATIC, Nebojsa; ANDRIC, Dragan. **The PIC Microcontroller, tradução de Alberto Jerônimo**. [S.l.]: V. Book 1, 2000.
- MINISTÉRIO DA AGRICULTURA. Instrução Normativa 51. Disponível em: <<http://extranet.agricultura.gov.br/sislegis-consulta/consultarLegislacao.do?operacao=visualizar&id=8932>>. Acesso em: março 2012.
- NICOLOSI, Denis E. C. **Microcontrolador 8051 Detalhado**. 2. ed. São Paulo: Editora Érica Ltda, 2002.
- NOVUS PRODUTOS ELETRÔNICOS. Catálogo Técnico PT100. Disponível em: <http://www.novus.com.br/downloads/Arquivos/folheto_pt100.pdf>. Acesso em: Março 2012.
- NXP. **Datasheet PCF8563: Real-time clock/calendar**. 2012. Rev. 10 - [s.n.], 2012.
- NXP SEMICONDUCTORS. Datasheet PCF8563. Disponível em: <<http://www.nxp.com/search?rows=10&type=keyword&q=pcf8563&page=1&tab=Documents>>. Acesso em: abril 2012.
- ORDOZES, Edward D. M.; PENTEADO, Cesar G.; SILVA, Alexandre C. R. **Microcontroladores e FPGA's Aplicações em Automação**. São Paulo: Novatec, 2005.
- PAZOS, Fernando. **Automação de Sitemas e Robótica**. Rio de Janeiro: Axcel, 2002.
- PEDRONI, Volnei A. **Eletrônica Digital Moderna e VHDL**. Rio de Janeiro: Elsevier Editora Ltda, 2010.
- PEREIRA, FÁBIO. **Microcontroladores HC908Q: Teoria e Prática**. 1. ed. São Paulo: Editora Érica Ltda, 2004.
- PEREIRA, Fábio. **Microcontroladores MSP430: Teoria e Prática**. 1. ed. São Paulo: Editora Érica Ltda, 2005.

SOUZA, David J. D. **Desbravando o PIC**. 12. ed. São Paulo: Editora Érica Ltda, 2010.

TEXAS INSTRUMENTS. **Datasheet MAX3222 - 3-V to 5-V multichannel RS-232 line Driver/Receiver**. 2004.[s.n.], 2004.

TEXAS INSTRUMENTS. **Guia do usuário família MSP430**. Disponível em: <<http://www.ti.com/sitesearch/docs/universalsearch.tsp?searchTerm=user%20guide%20msp430&linkId=3>>. Acesso em: 27 Janeiro 2013.

TEXAS INSTRUMENTS. Datasheet MSP430F2234. Disponível em: <<http://www.ti.com/product/msp430f2234>>. Acesso em: março 2012.

TEXAS INSTRUMENTS. Launchpad para MSP430. Disponível em: <<http://www.ti.com.br/launchpad>>. Acesso em: abril 2012.

THOMAZINI, Daniel. **Sensores Industriais: Fundamentos e Aplicações**. 1. ed. São Paulo: Editora Érica Ltda, 2005.

TOCCI, Ronald J.; WIDMER, Neal S.; MOSS, Gregory L. **Sistemas Digitais: princípios e aplicações**. 10. ed. São Paulo: Pearson, 2007.

APÊNDICE A

FUNÇÃO DE INICIALIZAÇÃO DO CONVERSOR A/D

```
//////////////////////////////////// ini AD //////////////////////////////////////  
void ini_ADC10(void){  
  
    ADC10CTL0 |= ADC10ON + ADC10IE;  
    ADC10AE0 |= 0x01;  
    ADC10CTL0 |= ENC + REFON + REF2_5V; // Habilita ADC10  
  
}
```

Quadro 1 - Função de inicialização do conversor A/D

Fonte: Autoria própria.

APÊNDICE B

INICIALIZAÇÃO DA COMUNICAÇÃO I2C

```

voidi2c_iniciar(void){
    UCB0CTL1 |= UCSWRST; // Desativa o controlador do i2c
    UCB0CTL1 |= UCSSEL_2; // Define o SMCLOCK como fonte para o SCL
    PORT_I2C_SEL |= BIT2 | BIT1; // Define as portas do i2c
    UCB0CTL0 |= UCMST + UCSYNC + UCMODE_3; // Ativa modo I2Cmaster e sincrono
    UCB0BR0 = 160; // 16Mhz/160 ~ 100kHz
    UCB0BR1 = 0;
    UCB0I2CSA = 0x50; // Endereco do RTC Slave
    UCB0CTL1 &= ~UCSWRST; // Inicia o I2C
    UCB0CTL1 |= UCTR + UCTXSTT; // Envia um START no i2c
    int timeout=1000;
    while (!(IRQ_REG & TX_IFG)&& timeout) timeout--; // //Espera a transmissao
    if(timeout == 0) { //Se finalizou o while por timeout entre em modo Alerta
        seq_fase=7; //Modo Inativo
    }
    TX_BUFFER = 0xC0; // Endereco do Registrador para iniciar o RTC
    timeout=1000;
    while (!(IRQ_REG & TX_IFG)&& timeout) timeout--; // Espera a transmissao
    if(timeout == 0) {
        seq_fase=7;
    }
    IRQ_REG &= ~TX_IFG; //Limpa flag de TX
    UCB0CTL1 &= ~UCTR; //
    UCB0CTL1 |= UCTXSTP; // Finaliza com um STOP
    timeout=1000;
    while ((UCB0CTL1 & UCTXSTP)&& timeout) timeout--; // Aguarda a confirmacao
    if(timeout == 0) {
        seq_fase=7;
    }
}

```

Quadro 2 - Inicialização da comunicação I2C

Fonte: Autoria própria.

APÊNDICE C

INICIALIZAÇÃO DA COMUNICAÇÃO SPI

```

void ini_SPI(void){
    ini_P1();
    UCA0CTL0 |= UCMSB + UCMST + UCSYNC + UCCKPH; // MODO 3;
    primeiro dado transmitido MSB
                // Modomestre, modosincrono
    UCA0CTL1 |= UCSSEL_2; // fontedeclockSMCLK
    UCA0BR0 |= 0x02; // baud rate, divide clock por 2
    UCA0BR1 |= 0x00; //
    UCA0MCTL = 0; // Desliga o modulador de geração
do baudrate

    UCA0CTL1 &= ~UCSWRST; // **Initialize USCI state
machine**

    //IE2 |= UCA0RXIE; // Enable USCI0 RX interrupt
}

```

Quadro 3 - Inicialização da comunicação SPI

Fonte: Autoria própria.