

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

GUILHERME DE CEZARO STANGA

**SISTEMA DE MONITORAMENTO DE TEMPERATURA E UMIDADE PARA UM
SILO**

TRABALHO DE CONCLUSÃO DE CURSO 2

PATO BRANCO

2017

GUILHERME DE CEZARO STANGA

**SISTEMA DE MONITORAMENTO DE TEMPERATURA E UMIDADE PARA UM
SILO**

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Bacharel em Engenharia de Computação, do Departamento Acadêmico de Informática da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Robison Cris Brito
Coorientador: Prof. Dr. Fábio Favarim

PATO BRANCO

2017



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Pato Branco
Departamento Acadêmico de Informática
Curso de Engenharia de Computação



TERMO DE APROVAÇÃO

Às 9 horas e 30 minutos do dia 27 de junho de 2017, na sala V104, da Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco, reuniu-se a banca examinadora composta pelos professores Robison Cris Brito (orientador), Fábio Favarim (coorientador), Gustavo Weber Denardin e Marcelo Teixeira para avaliar o trabalho de conclusão de curso com o título **Sistema de Monitoramento de Temperatura e Umidade para um Silo**, do aluno **Guilherme de Cezaro Stanga**, matrícula 01210165, do curso de Engenharia de Computação. Após a apresentação o candidato foi arguido pela banca examinadora. Em seguida foi realizada a deliberação pela banca examinadora que considerou o trabalho aprovado.

Prof. Robison Cris Brito
Orientador (UTFPR)

Prof. Fábio Favarim
Coorientador (UTFPR)

Prof. Gustavo Weber Denardin
(UTFPR)

Prof. Marcelo Teixeira
(UTFPR)

Profa. Beatriz Terezinha Borsoi
Coordenador de TCC

Prof. Pablo Gauterio Cavalcanti
Coordenador do Curso de
Engenharia de Computação

A Folha de Aprovação assinada encontra-se na Coordenação do Curso.

AGRADECIMENTOS

Agradeço a minha família pelo apoio e consideração por serem a minha base moral.

Agradeço os professores que tive a honra de conhecer, que me passaram seu conhecimento para construir meu futuro profissional. Em especial meus orientadores e minha banca de TCC.

Agradeço a todos os demais funcionários da universidade, que de uma forma ou outra contribuíram em minha formação, em especial o DIMASU, onde trabalhei com ótimos colegas e líderes.

Agradeço aos colegas por sempre estarmos unidos no desenvolvimento dos trabalhos da faculdade.

RESUMO

DE CEZARO STANGA, Guilherme. Sistema de monitoramento de temperatura e umidade para um silo. 2017. 53pg. Trabalho de Conclusão de Curso de bacharelado em Engenharia de Computação - Universidade Tecnológica Federal do Paraná. Pato Branco, 2017.

O monitoramento de temperatura em um silo se faz necessário, pois a temperatura é o principal fator envolvido na deterioração do grão armazenado. Controlando a temperatura é possível reduzir perdas devido à proliferação de fungos e evitar a germinação dos grãos. Este trabalho propõe um sistema para monitoramento de temperatura interna de um silo e da temperatura e umidade do ar de entrada no silo. O monitoramento utiliza o software supervisor SCADABR para armazenamento e visualização dos dados de temperatura e umidade. Esse *software* supervisor se comunica através de um barramento RS-485 utilizando protocolo Modbus com os dispositivos sensores. O *hardware* do sistema utiliza sensores e microcontroladores para a coleta de dados para o software supervisor. O sistema fornece alarmes e envio de e-mail para usuários cadastrados quando atingidas temperaturas pré-determinadas, assim permitindo ao operador do silo a tomada de decisão.

Palavras-chave: Silo. Monitoramento de temperatura. Monitoramento de umidade. Sistema supervisor. Modbus. RS-485.

ABSTRACT

DE CEZARO STANGA, Guilherme. Temperature and humidity monitoring system for a silo. 2017. 53 f. Conclusion Work for a Bachelor in Computer Engineering - Universidade Tecnológica Federal do Paraná. Pato Branco, 2017.

Temperature monitoring in a silo is necessary because temperature is the main factor involved in the deterioration of stored grain. By controlling the temperature it is possible to reduce losses due to the proliferation of fungi and to avoid the germination of the grains. This work proposes a system for monitoring the internal temperature of a silo and the temperature and humidity of the incoming air in the silo. Monitoring uses SCADABR supervisory software for storing and viewing temperature and humidity data. This supervisory software communicates via an RS-485 bus using Modbus protocol with the sensing devices. The system hardware uses sensors and microcontrollers to collect data for supervisory software. The system provides alarms and sends e-mail to registered users when reached predetermined temperatures, thus allowing the operator of the silo the decision making.

Keywords: Silo. Temperature monitoring. Humidity monitoring. Supervisory system. Modbus. RS-485.

LISTA DE FIGURAS

Figura 1 – Processo de medição por um instrumento.....	14
Figura 2 – Representação utilizada para um RTD	16
Figura 3 – Curvas de variação da resistência em função da temperatura.....	17
Figura 4 – Representação de dois eletrodos intergigitados	18
Figura 5 – Topologias de redes	20
Figura 5 – Relação entre velocidade e comprimento	21
Figura 6 – Métodos de terminações.....	21
Figura 7 – Diagrama de ciclo de mensagens Modbus.....	23
Figura 8 – Representação de uma PDU Modbus	24
Figura 9 – Representação de um quadro no modo de transmissão ASCII	26
Figura 10 – Representação de um quadro no modo de transmissão RTU.....	26
Figura 11 – Exemplo de requisição para leitura de registradores	27
Figura 12 – Diagrama de estados para Função 03.....	28
Figura 13 – Diagrama de componentes e conexões do sistema.....	29
Figura 14 – Diagrama de componentes e conexões de um dispositivo sensor.....	34
Figura 15 – Circuito de instrumentação	35
Figura 16 – Configuração de clock e portas do microcontrolador.....	36
Figura 17 – Funções para o A/D	36
Figura 18 – Funções para o sensor DHT11	37
Figura 19 – Função main.	38
Figura 20 – Alteração na função eMBRegHoldingCB.....	39
Figura 21 – Esquemático de um dispositivo sensor.	40
Figura 22 – Desenho para a PCI de um dispositivo sensor.....	40
Figura 23 – MAX-485 (a) e rede RS-485 com resistores de terminação (b).....	41
Figura 24 – Modulo FT232RL.....	42
Figura 25 – Esquemático do conversor USB – RS-485.....	43
Figura 26 – Desenho para a PCI do conversor USB - RS-485.....	43
Figura 27 – Configurações do Data Base.	44
Figura 28 – Configurações do Data Point.	45
Figura 29 – Configurações de e-mail.	45
Figura 30 – Configurações de e-mail para usuários.	46
Figura 31 – Configurações de alarme.....	46
Figura 32 – Vista superior de um escravo.....	47
Figura 33 – Sinais do barramento RS-485.	48
Figura 34 – Conversor RS-485 - USB.....	48
Figura 35 – Interface para visualização dos valores dos sensores.....	49
Figura 36 – E-mail do ScadaBR recebido.	49
Figura 37 – Imagem geral do protótipo.	50

LISTA DE QUADROS

Quadro 1 – Relação entre Tipos de RTD mais usados	18
Quadro 2 – Relação entre Cód. de função e descrição.....	25
Quadro 3 – Faixas de endereço do protocolo Modbus.....	25
Quadro 4 – Principais funções do protocolo Modbus.....	27
Quadro 5 – Relação entre nome dos materiais e sua descrição	29

LISTA DE SIGLAS

A/D	Analógico/Digital
ASCII	<i>American Standard Code for Information Interchange</i>
C	Celsius
CRC	<i>Cyclic Redundancy check</i>
DCO	<i>Digitally Controlled Oscillator</i>
IHM	Interface Homem-Maquina
K	Kelvin
NTC	<i>Negative Temperature Coefficient</i>
OSI	<i>Open Systems Interconnection</i>
PDU	<i>Protocol Data Unit</i>
PIB	Produto Interno Bruto
PTC	<i>Positive Temperature Coefficient</i>
RTD	<i>Resistance Temperature Detector</i>
RTU	<i>Remote Terminal Unit</i>
SQL	<i>Structured Query Language</i>
TTL	<i>Transistor-transistor Logic</i>
UART	<i>Universal Asynchronous Receiver/Transmitter</i>
UR	Umidade Relativa
USB	<i>Universal Serial Bus</i>

SUMÁRIO

1	INTRODUÇÃO	10
1.1	CONSIDERAÇÕES INICIAIS.....	10
1.2	OBJETIVOS	11
1.2.1	Objetivo Geral.....	11
1.2.2	Objetivos Específicos	11
1.3	JUSTIFICATIVA.....	11
1.4	ESTRUTURA DO TRABALHO.....	13
2	REFERENCIAL TEÓRICO.....	14
2.1	INSTRUMENTAÇÃO ELETRÔNICA	14
2.2	SENSORES.....	15
2.2.1	Sensores de temperatura	15
2.2.1.1	Sensor RTD.....	16
2.2.2	Sensor de umidade.....	18
2.3	PADRÃO RS-485	19
2.4	PROTOCOLO MODBUS	22
3	MATERIAIS E MÉTODO.....	29
3.1	MATERIAIS	29
3.1.1	Sensor de temperatura	30
3.1.2	Microcontrolador.....	30
3.1.3	Conversão TTL/RS-485.....	30
3.1.4	Sensor digital de umidade e temperatura.....	31
3.1.5	Conversão USB/TTL.....	31
3.1.6	Supervisório	31
3.1.7	Amplificador de instrumentação	32
3.2	MÉTODO	33
3.2.1	Dispositivo sensor	33
3.2.2	Rede Mestre/Escravo	41
3.2.3	Sistema supervisório.....	43
4	RESULTADOS E DISCUSSÕES	47
5	CONSIDERAÇÕES FINAIS	51
	REFERÊNCIAS.....	53

1 INTRODUÇÃO

Este capítulo está dividido em três seções. Na Seção 1.1 é apresentada uma visão contextual do assunto referente à proposta, na Seção 1.2 são apresentados os objetivos gerais e específicos e na Seção 1.3 a justificativa para a realização desta proposta.

1.1 CONSIDERAÇÕES INICIAIS

O Brasil no cenário mundial é o terceiro maior exportador de grãos. Na safra 2016/2017 o Brasil produziu 216,54 milhões de toneladas de grãos em uma área plantada de 60,0 milhões de hectares, correspondendo a uma produtividade de 3.609Kg/ha. Destaca-se deste total, 46% a produção de soja e 40% a produção de milho (CONAB, 2017). A produção agrícola é responsável por 20% do Produto Interno Bruto (PIB) total da economia Brasileira. Nos últimos anos o país tem aumentado às taxas de colheita enquanto a área plantada não obteve significativo aumento, o que significa maior índice de produtividade. Esta produção é em parte resultado do emprego de melhores tecnologias no campo (CONAB, 2017).

Um fator preocupante no cenário agrícola Brasileiro é a quantidade insuficiente e a qualidade dos locais para armazenagem de grãos. Durante o armazenamento deve-se adequar os índices de temperatura e umidade para valores específicos para o grão armazenado com o propósito de evitar perdas do produto.

Em países como França, Argentina e Estados Unidos, a armazenagem dos grãos nas próprias fazendas dos produtores representa de 30 a 60% da safra, enquanto no Brasil, esta prática não é muito popular entre os produtores de grãos, apresentando um percentual de cerca de 5%. Geralmente os produtores armazenam sua produção em cooperativas. O principal fator deste baixo percentual é o econômico (D'ARCE, 2011).

Considerando a importância de que o ambiente de armazenamento de grãos seja correto em termos de temperatura e umidade e uniforme no silo, um sistema de monitoramento pode ser utilizado para o mapeamento dos índices de temperatura em pontos dispersos do silo. Este sistema pode contribuir com a eficiência do processo de secagem e armazenagem, ao fornecer informações para a tomada de decisões para corrigir os problemas relacionados a perdas no armazenamento de grãos.

O sistema de monitoramento conta com um conjunto de dispositivos capazes de medir temperatura e por meio de uma rede enviar os dados para um dispositivo final, sendo que existirá um software que receberá os dados e mostrará em uma Interface Homem-Maquina (IHM).

1.2 OBJETIVOS

A seguir estão, o objetivo geral e os objetivos específicos deste trabalho de conclusão de curso.

1.2.1 Objetivo Geral

Desenvolver um sistema de monitoramento de temperatura e umidade para um silo para auxiliar na tomada de decisões visando redução nas perdas na armazenagem.

1.2.2 Objetivos Específicos

- Projetar e desenvolver dispositivos sensores, para medição e transmissão de amostras de temperatura e umidade;
- Realizar a instrumentação de um sensor de temperatura;
- Projetar e implementar uma rede de comunicação RS-485 e aplicar o protocolo modbus para transportar dados dos dispositivos sensores até um dispositivo final;
- Aplicar um software supervisor para visualização e armazenamento dos dados coletados;

1.3 JUSTIFICATIVA

O presente trabalho propõe o desenvolvimento de um sistema para monitorar os índices de temperatura interna de um silo (estrutura armazenadora e secadora de grãos) e

da temperatura e umidade do ar de entrada do silo. Este sistema serve para tomada de decisões e registro de informações. O sistema possibilita observar os dados de temperatura em pontos dispersos no silo. Isso se faz necessário, pois a massa de grãos tem comportamento de isolante térmico, o que significa que diferentes regiões do silo podem estar com diferentes temperaturas (D'ARCE, 2011).

Com os dados das temperaturas do silo torna possível uma pessoa tomar decisões tais como acionar um sistema de atuadores para injeção de ar no silo, pois o sistema proposto não proporciona a atuação. Desta forma, adequando o ambiente do silo para as corretas características de secagem e armazenamento dos grãos armazenados, reduzindo, assim, as perdas.

Em um silo, o ar presente entre os grãos não é estático, está em movimento contínuo causado por correntes de convecção resultantes da diferença de densidade entre o ar quente e o ar frio. Quando ocorre movimentação do ar das regiões mais quentes para as mais frias, o ar quente apresenta maiores índices de umidade causados pelo arrefecimento do ar quente na região fria. Quando ocorrem estes acúmulos de umidade, o ambiente torna-se favorável a proliferação de pragas que degradam o grão, provocando perdas na produção (SILVA, 2005).

Para a correta secagem dos grãos é necessário que o ar injetado possua menor índice de umidade que a umidade no interior do silo (SILVA, 2005). Portanto, com os dados de umidade relativa do ar e da temperatura do ambiente externo é possível uma pessoa tomar decisões sobre o ar de entrada, por exemplo: aquecer ou resfriar o ar de entrada, com o intuito de diminuir sua umidade. Estes dados serão coletados por um sensor posicionado na parte externa do silo.

Um sensor de temperatura torna possível a avaliação das condições de armazenamento pelo fato do aumento da temperatura estar relacionada com focos de pragas ou movimentação de umidade. Como o índice de temperatura está relacionado com a umidade, não se necessita medir a umidade no interior do silo. Além disso, os sensores existentes para medição de umidade no mercado medem a umidade relativa do ar e não do grão, a medição de umidade no grão é realizada em laboratório com equipamentos específicos e de custo elevado (D'ARCE, 2011).

O motivo da realização deste trabalho é fornecer uma ferramenta que possibilite maior precisão no controle de perdas de grãos armazenados. O trabalho também vem a contribuir com os estudos já realizados em meio acadêmico.

1.4 ESTRUTURA DO TRABALHO

Este trabalho é estruturado nos seguintes capítulos. O Capítulo 2 é apresentado conceitos e fundamentos que formam a base deste trabalho. O Capítulo 3 explana os materiais e métodos utilizados para realização do trabalho, no Capítulo 4 são apresentados os resultados obtidos e o capítulo 5 as considerações finais.

2 REFERENCIAL TEÓRICO

Neste capítulo são explanados os principais conceitos que fundamentam o trabalho proposto.

2.1 INSTRUMENTAÇÃO ELETRÔNICA

Um instrumento é um dispositivo capaz de determinar valores de grandezas ou variáveis. Para que uma grandeza física (por exemplo, temperatura ou umidade) possa ser medida corretamente, deve-se passar por um processo de instrumentação. Na Figura 1 são mostrados os passos para obtenção do valor de uma varável medida.



Figura 1 – Processo de medição por um instrumento
 Fonte: Adaptado de Mercedes (2010, p. 4).

Neste processo, o sensor converte a grandeza física para uma grandeza elétrica (por exemplo, tensão ou resistência), o condicionamento é formado por um circuito eletrônico capaz de alterar as características do sinal medido para que possa ser possível uma resposta linear. Esta resposta linear é convertida de analógica para digital, geralmente por um circuito chamado Analógico/Digital (A/D). O dado digital pode ser processado por um dispositivo computacional e mostrado por meio de um display (MERCEDDES, 2010).

A seguir, algumas definições a respeito de medições (HELFRICK, 1994):

- Erro: Diferença entre o valor medido e o valor verdadeiro;
- Resolução: Menor variação na grandeza medida que pode ser detectada pelo instrumento;

- Sensibilidade: É a razão entre o sinal de saída do instrumento pelo sinal de entrada do instrumento ou a variável a ser medida;
- Exatidão: O quão próximo o valor medido está do valor verdadeiro;
- Precisão: Indica quanto um conjunto de valores medidos da mesma grandeza estão próximos umas das outras;

2.2 SENSORES

Sensores são dispositivos que alteram suas características em função de alguma forma de energia em um ambiente físico, respondendo a estes estímulos com um sinal proporcional ao estímulo medido.

Quando um sensor realiza uma medição de forma direta é chamado de transdutor. O transdutor é um dispositivo que transforma uma grandeza física (temperatura, umidade, etc.) em uma grandeza elétrica (tensão, corrente, etc.) possibilitando a leitura do meio físico por um dispositivo digital (THOMAZINI, 2005).

2.2.1 Sensores de temperatura

A temperatura é uma grandeza física e mede a quantidade de energia térmica de um corpo, representa aquecimento ou resfriamento em relação a um meio. Geralmente é medida em graus nas escalas Celsius (C) ou Kelvin (K) (DUNN, 2013).

Um sensor ou transdutor de temperatura é um dispositivo que produz como saída um sinal elétrico em função da temperatura que é medida. Atualmente existem 5 grupos principais de sensores de temperatura. Bimetais, termistores, termopares, transdutores integrados e detectores de temperatura resistivos, em inglês *Resistance Temperature Detector* (RTD) (MERCEDDES, 2010).

A seguir um detalhamento de cada tipo de sensor de temperatura.

- O sensor do tipo bimetálico é composto por uma barra formada por duas chapas de metal unidas com coeficientes de dilatação térmica distintas, a medição de temperatura é realizada pelo raio de deformação na barra;

- O sensor do tipo termistor é baseado na variação da resistência de semicondutores com a temperatura, são encontrados da forma *Positive Temperature Coefficient* (PTC) e *Negative Temperature Coefficient* (NTC);
- Os termopares são sensores formados por dois metais distintos e conectados em uma extremidade capazes de gerar tensão devido ao efeito seebeck;
- Os transdutores integrados são formados por circuitos eletrônicos baseados em transistores e diodos;
- Os sensores do tipo RTD são baseados na variação da resistência de um metal com a temperatura;

2.2.1.1 Sensor RTD

O funcionamento de um sensor RTD é baseado na variação de resistência elétrica de um metal (condutor) em função da temperatura, sua resistência aumenta conforme a temperatura aumenta e diminui conforme a temperatura diminui.

Quando uma corrente elétrica passa por uma resistência, os terminais da resistência produzem uma diferença de tensão, desta forma, medindo a diferença de tensão entre os terminais de um RTD é possível determinar a resistência e associá-la a temperatura. É um sensor do tipo ativo, isto significa que necessitam de uma alimentação auxiliar para operar.

Sensores RTD são formados por um filamento resistivo de metal, os metais mais utilizados são de níquel, platina e cobre. A nomenclatura de sensores RTD segue uma norma, depende do material constituído e de sua resistência a 0°C, por exemplo, PT-100 é formado de platina e possui resistência de 100Ω a 0°C (HELFRİK, 1994). O símbolo utilizado para representar este dispositivo é mostrado na Figura 2.



Figura 2 – Representação utilizada para um RTD

A relação de sua resistência com a temperatura é expressa pela Equação (1).

$$R_T = R_o(1 + \alpha \Delta T + \alpha_2 \Delta T^2 + \dots + \alpha_n \Delta T^n), em \text{ } ^\circ C \quad (1)$$

Sendo:

- R_T = Resistência da temperatura medida;
- R_o = Resistência da temperatura de referencia;
- α = Constantes do material;
- ΔT = Diferença entre temperatura medida e temperatura de referência;

O número de constantes necessárias depende de diversos fatores, como a precisão requerida, o material e a faixa de temperatura a ser medida. Geralmente é utilizada apenas uma constante de temperatura, isto resulta uma melhor linearidade em uma faixa de temperatura limitada (MERCEDDES, 2010). Desta forma, aproxima-se a Equação. (1) para Equação (2):

$$R_T = R_o(1 + \alpha \Delta T) \quad (2)$$

A Figura 3 mostra as curvas características de sensores RTD de diferentes materiais.

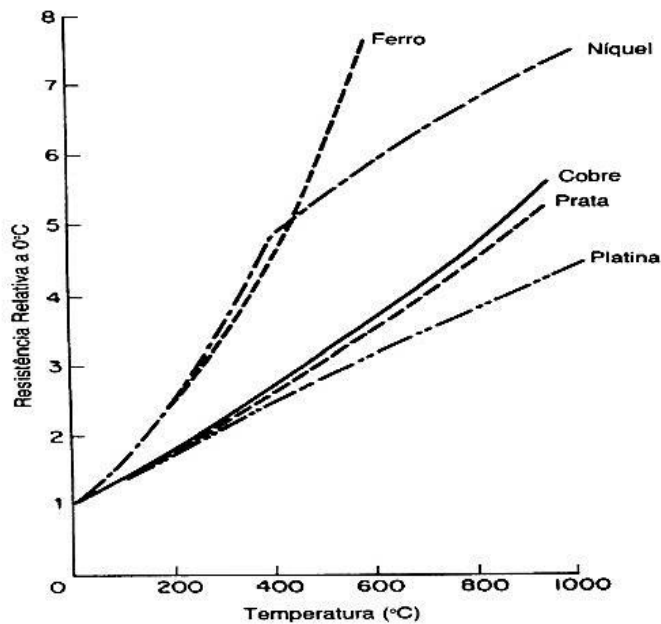


Figura 3 – Curvas de variação da resistência em função da temperatura
Fonte: Helfrick (1994, p. 264).

Pelo gráfico, pode-se observar que o sensor de níquel tem resposta não linear e maior sensibilidade por responder um aumento significativo de resistência em uma curta faixa de temperatura, enquanto que o de platina oferece a resposta mais linear e tem menos sensibilidade.

O Quadro 1 mostra uma relação entre três materiais mais usados.

Tipo	Faixa de temperatura	Exatidão	Notas
Platina	-170°C a +850°C	$\pm 2^{\circ}\text{C}$	Alta estabilidade, larga faixa de operação e tempo de resposta lento (15s).
Cobre	-180°C a +260°C	$\pm 1^{\circ}\text{C}$	Alta linearidade, alta precisão nas redondezas da temperatura ambiente e faixa de temperatura limitada.
Níquel	0°C a +60°C	$\pm 1^{\circ}\text{C}$	Alta sensibilidade, alto coeficiente de temperatura, não linear e faixa de temperatura limitada.

Quadro 1 – Relação entre Tipos de RTD mais usados
Fonte: Helfrick (1994, p. 264).

2.2.2 Sensor de umidade

Segundo Dunn (2013), umidade é uma medida da quantidade relativa de água em estado gasoso presente no ar ou em um gás. A Umidade Relativa (UR) é a razão entre a quantidade parcial de vapor de água presente no ar em certo volume pela quantidade total, ou de saturação, que o mesmo volume de ar pode suportar, dado as mesmas condições de temperatura e pressão. O índice UR é expresso em porcentagem e indica quão próximo o ar está da saturação (DUNN, 2013).

Um higrômetro é um dispositivo capaz de medir umidade de modo indireto pela mudança nas propriedades físicas ou elétricas de materiais em função da umidade. Entre os tipos de higrômetros, os mais comuns encontrados em sensores são do tipo resistivo e capacitivo.

O higrômetro resistivo, também chamado de sensor de umidade resistivo, é formado por dois eletrodos interdigitados contidos em uma substância isolante, como mostra a Figura 4 (DUNN, 2013).

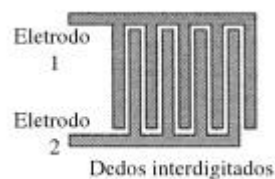


Figura 4 – Representação de dois eletrodos interdigitados
Fonte: Dunn (2013, p. 151).

Os eletrodos são revestidos com um material que absorve água, este material fornece um meio condutivo entre os eletrodos quando em contato com umidade, desta forma o valor de resistência do sensor diminui proporcionalmente com o aumento de umidade (DUNN, 2013).

O higrômetro capacitivo é formado, assim como um capacitor, de duas placas condutoras com um material dielétrico entre elas. O material dielétrico muda de forma linear em contato com a umidade do ar. Desta forma, o valor de capacitância do sensor aumenta conforme o aumento da umidade (DUNN, 2013).

2.3 PADRÃO RS-485

Uma interface de comunicação serial é realizada pela transmissão de bits um após o outro em um barramento. Pode ser síncrona quando um sinal de clock controla o fluxo de dados e assíncrona quando o controle do fluxo é realizado pelo tempo de propagação de cada bit.

Também conhecido por TIA/EIA-485, o padrão RS-485 define uma interface de comunicação serial entre dispositivos. Esta comunicação é realizada por dois cabos, A+ e B-, com transmissão diferencial de dados e resistores de terminação. Uma transmissão diferencial significa que os bits são representados pela diferença de tensão entre os dois cabos, da forma $A - B$, se esta diferença for superior que 200mV tem-se o nível lógico 1, se for inferior a -200mV o nível lógico é 0. O intervalo entre 200mV e -200mV é considerado como indefinido e pode servir para detectar se algum cabo está solto. Este padrão ainda permite um terceiro cabo para sinal de referência, também chamado terra. A transmissão diferencial traz vantagem por ser robusta em relação à interferência eletromagnética, pois quando são usados dois cabos em forma de par trançado e os resistores de terminação para balanceamento, quando um ruído é inserido no par trançado a diferença entre A+ e B- não chega a ser significativa (KRON, 2015).

Para que um circuito utilizando tensões de nível *Transistor-transistor Logic* (TTL) possa se comunicar pela interface RS-485, são utilizados módulos conversores, existem vários disponibilizados no mercado, entre eles estão o MAX-485 e LTC2850 (SILVA; PEREIRA; BUCHMANN, 2013).

Em relação à topologia, para distâncias pequenas e velocidades baixas a topologia da rede não tem muita influência, porém quando surgirem problemas na linha de transmissão o padrão recomenda a topologia daisy-chain. Nesta topologia, os componentes

da rede ficam em série conectados diretamente aos cabos de comunicação. A Figura 5 mostra as topologias existentes. A topologia de barramento com terminações é tolerável quando o comprimento dos cabos que interligam o dispositivo a linha de comunicação sejam o menor possível, ou seja, muito menor em relação ao comprimento do barramento da linha de comunicação (KRON, 2015).

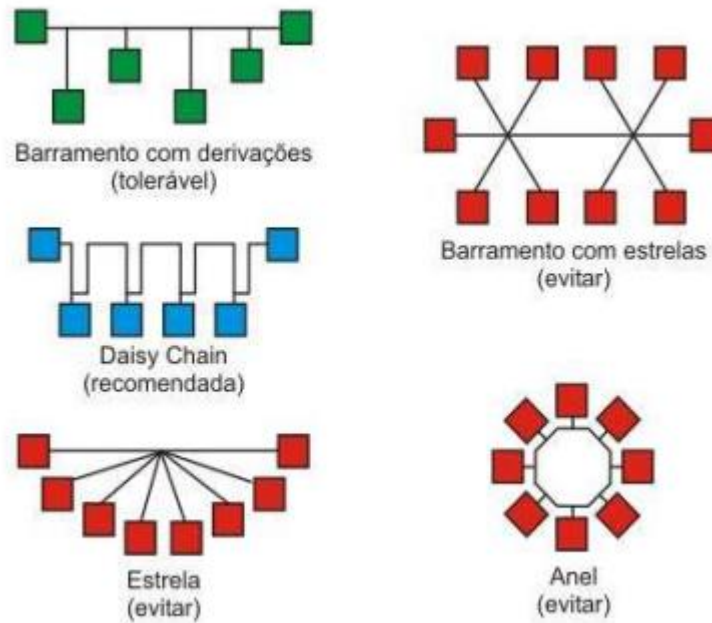


Figura 5 – Topologias de redes
Fonte: Kron (2015, p. 3).

O padrão RS-485 é dito half-duplex, a interface de comunicação é utilizada para transmissão e recepção de dados, mas não simultaneamente, quando um dispositivo está transmitindo, os demais devem manter-se em estado de recepção. A interface de comunicação pode-se tornar full-duplex com a utilização de mais dois cabos para possibilitar a transmissão e recepção ao mesmo tempo (TEXAS, 2008). A arquitetura utilizada na maioria das vezes é a Mestre/Escravo, sendo um dispositivo Mestre responsável por decidir qual Escravo irá transmitir em um intervalo de tempo.

O padrão permite até 32 dispositivos conectados no mesmo barramento, dependendo do valor de resistência de cada dispositivo, utilizando cargas menores que 12K, denominado como “carga unitária”, é possível ampliar para 256 dispositivos no barramento. O comprimento máximo dos cabos de comunicação é definido em 1200 metros, sendo que existe uma relação entre a velocidade de comunicação e o comprimento

do cabo, quanto maior o cabo menor deve ser a velocidade de comunicação (KRON, 2015). Esta relação é mostrada na Figura5.

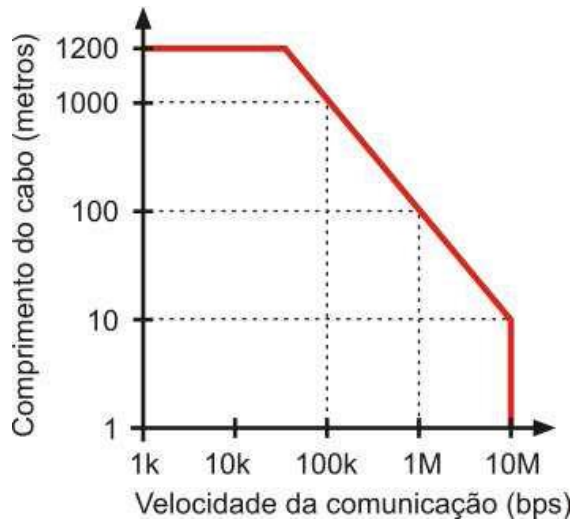


Figura 5 – Relação entre velocidade e comprimento
Fonte: Kron (2015, pg. 4).

Para que a comunicação pelo barramento seja confiável, as reflexões no barramento devem ser tratadas, para isso são usados resistores de terminação com valores de 120Ω, pois na maioria das vezes este é o valor de impedância do barramento, valores inferiores a este não são recomendados para a utilização (TEXAS, 2008).

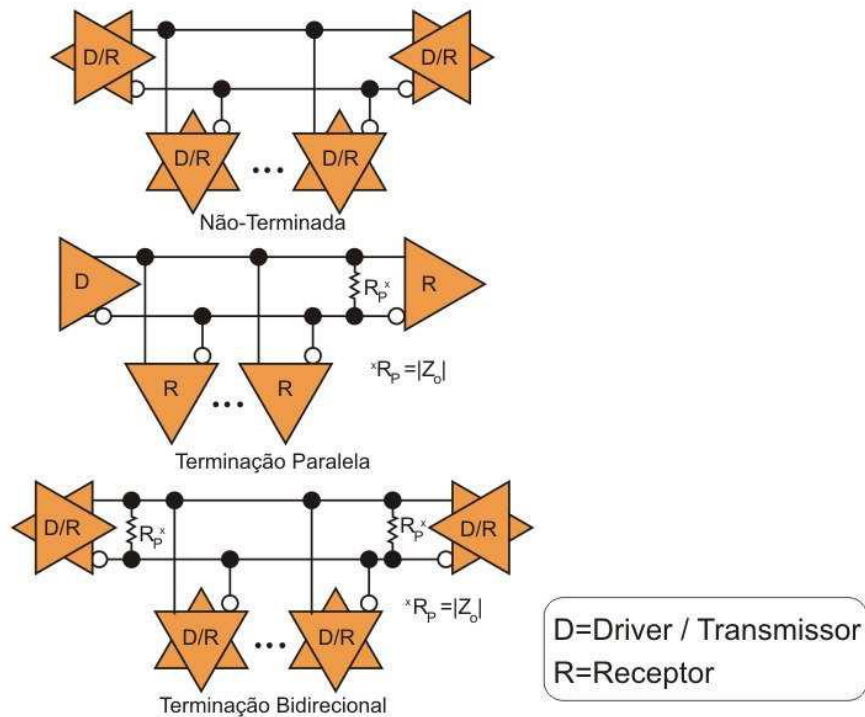


Figura 6 – Métodos de terminações
Fonte: Kron (2015, p. 2).

As redes não terminadas são utilizadas quando o comprimento dos cabos for inferior a 100 metros e velocidades inferiores a 19,2K bps, desta forma os resistores de terminação são desnecessários. Suas vantagens são o menor consumo, facilidade de implementação e são mais baratas. A terminação paralela é limitada a um único dispositivo para transmissão posicionado em uma das extremidades enquanto o resistor fica posicionado na extremidade oposta. Os demais dispositivos apenas recebem, sua vantagem é oferecer taxas de transmissão elevadas. A terminação Bidirecional é o método mais confiável de terminação, esta técnica possibilita os transmissores estarem posicionados em qualquer ponto do barramento, porém sua desvantagem é o aumento do consumo na rede (KRON, 2015).

O padrão RS-485 pertence à camada física do modelo OSI, do inglês *Open Systems Interconnection*, e não especifica um protocolo de comunicação, é possível ser implementado qualquer protocolo, sendo que o mais utilizado é o Modbus (KRON, 2015).

2.4 PROTOCOLO MODBUS

Um protocolo é um conjunto de regras que define a ordem e o formato de mensagens trocadas por dispositivos em uma rede e suas ações a serem realizadas no recebimento e envio de mensagens (KUROSE; ROSS, 2010).

O protocolo Modbus foi desenvolvido pela Modicon em 1979, é uma estrutura de mensagem aberta e seu princípio de funcionamento é do tipo requisição/resposta. Este protocolo pertence à camada 7 do modelo OSI provendo comunicação cliente/servidor para diversos barramentos e redes, sendo que também pode ser implementado na camada física em comunicação serial do tipo mestre/escravo. Sua principal característica é oferecer serviços definidos por funções. É utilizado em barramentos seriais assíncronos como RS-232 ou RS-485 ou também em redes TCP/IP (MODBUS, 2006).

A comunicação entre dispositivos utilizando Modbus serial em um barramento RS-485 é do tipo mestre/escravo e ocorre da seguinte forma: um dispositivo da rede sendo o mestre é responsável por coordenar a rede, ele define a vez de qual escravo deve transmitir e os escravos não se comunicam entre si. O Mestre pode mandar mensagens *unicast* quando quer se comunicar com um único escravo e também pode mandar mensagens *broadcast* quando quer enviar mensagens para todos os escravos, porém, os escravos não devem responder a mensagens *broadcast*, este tipo de mensagem serve para escrita (MODBUS, 2006).

O princípio de funcionamento ocorre da seguinte forma: o Mestre envia para o barramento uma consulta, nela contém o endereço do dispositivo escravo que irá responder e o código de função a ser executado pelo escravo, um campo de dados que dependendo da função não é necessário haver informações, e um campo de checagem de erros. Quando o escravo responde a consulta, ele coloca seu próprio endereço neste campo, indicando qual escravo respondeu, e preenche o campo de dados de acordo com o código de função. O campo checagem de erros é o resultado de um calculo para verificar se a mensagem enviada contém erros. A Figura 7 mostra o processo de troca de mensagens (MODBUS, 2006).

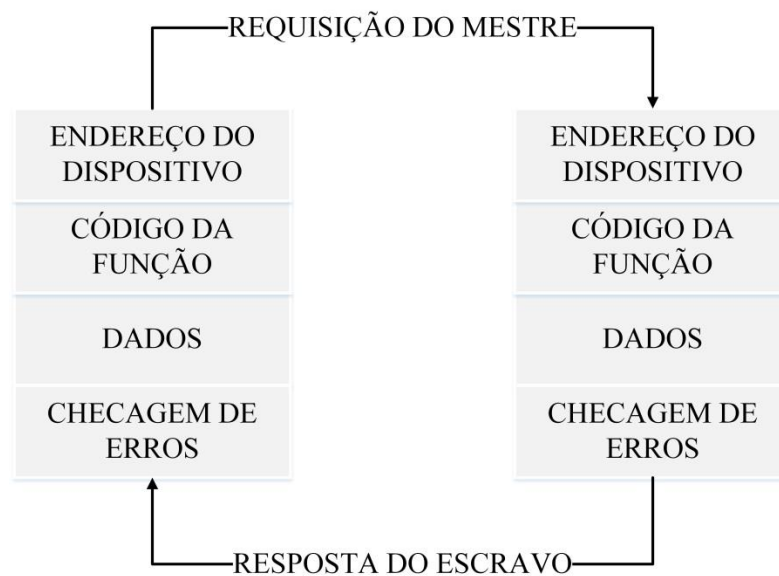


Figura 7 – Diagrama de ciclo de mensagens Modbus
Fonte: Composto a partir de Modbus (2006).

O número de endereços especificado pelo Modbus serial é 256. O endereço 0 é definido como endereço de broadcast, os endereços de 1 até 247 são para endereçamento de escravos e os endereços de 248 até 255 são reservados. Sendo que o dispositivo Mestre não tem endereço (MODBUS, 2006).

O protocolo Modbus especifica uma unidade de dados de protocolo, do inglês, *Protocol Data Unit* (PDU) independente de outras camadas do modelo OSI. Uma PDU é um conjunto de dados transmitido entre mesmas camadas de rede. Quando uma camada superior recebe uma PDU, esta adiciona seus campos para formar uma nova PDU, este processo se chama encapsulamento. Uma PDU Modbus contém um campo de código de função e um campo de dados, mostrada na Figura 8. O tamanho de um PDU é limitado a 254 bytes (MODBUS, 2002).

PDU MODBUS	
Código de função	Dados

Figura 8 – Representação de uma PDU Modbus

Fonte: Adaptado de: Modbus (2002, p. 8).

O campo de dados contém as informações trocadas entre o mestre e o escravo. O código da função indica para o escravo qual função ele deve realizar (escrita, leitura,...) e indica para o mestre se houve erros na função solicitada. Quando uma função é executada com sucesso, o escravo responde com o mesmo valor no código de função. Já se o escravo não for capaz de atender a função solicitada, ele retorna uma função do tipo *exception* informando o motivo que ocasionou o erro (MODBUS, 2002).

Quando o mestre envia uma requisição para um escravo, quatro situações podem ocorrer (MODBUS, 2006):

- O escravo receber a requisição do mestre sem erros processá-la e responde-la.
- O escravo pode não receber a requisição por problemas no barramento, desta forma, não responde a requisição e o mestre verifica um *time out*.
- O escravo recebe a requisição, porém detectou problemas de comunicação (ex.: erro de CRC ou paridade), desta forma não irá responder ao mestre e este verifica seu *time out*.
- O escravo recebe a requisição do mestre sem erros de comunicação, porém não consegue atender a requisição desejada (ex.: leitura de registrador inexistente). Então ele retorna uma mensagem de exceção informando a natureza do erro.

O Quadro 2 mostra os principais códigos de exceção e sua descrição.

Código de Exceção	Descrição
1	O código de função recebido não é suportado. Pode não estar implementado na aplicação.
2	A requisição tentou acessar um endereço inválido. Isso pode acontecer somente se o endereço inicial e a quantidade de valores requisitada ultrapassar 2^{16} .
3	A requisição tem dados incorretos. O mestre solicitou acessar mais dados do

	que o permitido pelo escravo ou pelo protocolo.
4	Ocorreu um erro irreversível durante uma tentativa de processar a requisição. Esse é um código de exceção geral, que indica que a requisição era válida, mas o escravo não conseguiu executá-la.

Quadro 2 – Relação entre Cód. de função e descrição.

Fonte: Modbus (2006, p. 49).

O protocolo define como os dados são armazenados e acessados. O Quadro 3 apresenta as faixas de endereços do protocolo Modbus. As faixas de endereço possibilitam restringir ou permitir acesso a diferentes tipos de dados. O esquema de endereçamento é definido pelo dispositivo escravo.

Bloco	Tipo de dado	Permissão ao mestre	Prefixo
Discrete Input	1 bit	Leitura	1
Coils	1 bit	Leitura/Escrita	0
Input Registers	16 bit	Leitura	3
Holding Registers	16 bit	Leitura/Escrita	4

Quadro 3 – Faixas de endereço do protocolo Modbus.

Fonte: Adaptado de: Modbus (2006, p. 6).

A especificação do protocolo define que cada faixa de dados contém um espaço de endereçamento de no máximo 2^{16} (65.536) elementos. O protocolo utiliza prefixos para referir-se a determinado tipo de dado. Por exemplo, para referir-se a um holding register 14 no endereço 13, o manual de um dispositivo pode referir-se a um item de dados no endereço 4.014.

O protocolo especifica dois modos de transmissão serial assíncrona, ASCII, do inglês *American Standard Code for Information Interchange* e RTU, do inglês *Remote Terminal Unit*. Estes modos definem o modo como os bits são transmitidos pelo barramento, como os dados são empacotados nos campos de mensagens e como são desempacotados. O modo de transmissão deve ser o mesmo para todos os dispositivos no barramento. Os dispositivos do barramento devem ser configurados para qual modo usar, sendo que o modo padrão previamente implementado em diversos dispositivos comerciais é o RTU (MODBUS, 2002).

A Figura 9 mostra um quadro ASCII.

Modelo ASCII					
Começo	Endereço	Cód. Função	Dados	Checagem	Fim
1 caractere	2 caracteres	2 caracteres	N caracteres	2 caracteres	2 caracteres

Figura 9 – Representação de um quadro no modo de transmissão ASCII

Fonte: Adaptado de: Modbus (2002, p. 17).

Neste modo cada byte da mensagem é transmitido como dois caracteres ASCII (0-9-A-F). Este modo traz vantagem da mensagem ser legível pela tabela ASCII, porém pode consumir mais recursos na rede. Outra vantagem é permitir que os intervalos de tempo sejam de um segundo para percorrer os caracteres de uma mensagem sem ocasionar erros. Os dispositivos verificam constantemente o barramento para verificar o início de uma mensagem definido pelo caractere ‘:’ (dois pontos) em ASCII 3A. O campo para verificação de erros é baseado no método LRC. O campo de fim é definido como os caracteres CR+LF. Cada símbolo contém 10 bits: 1 bit para informar início, 7 bits de dados da forma LSB (bit menos significativo primeiro), 1 bit de paridade e um bit de parada (MODBUS, 2002).

No modo de transmissão RTU, cada símbolo é formado por dois dígitos hexadecimais. Cada símbolo contém 11 bits: 1 bit de início, 8 bits de dados da forma LSB, 1 bit de paridade e um bit de parada. A vantagem deste modo em relação ao modo ASCII é a maior densidade de dados enviados na mesma velocidade de comunicação. A Figura 10 mostra um quadro no modo RTU (MODBUS, 2002).

Modelo RTU			
Endereço	Cód. Função	Dados	Checagem
1 byte	1 byte	N bytes	2 bytes

Figura 10 – Representação de um quadro no modo de transmissão RTU

Fonte: Adaptado de: Modbus (2002, p. 13).

Neste modo não existem caracteres que indicam o início e o fim de um quadro, a separação de mensagens é realizada pela ausência de dados na transmissão do barramento em um tempo de 3,5 vezes o tempo de um caractere. Desta forma, quando uma mensagem é enviada após este tempo mínimo, os dispositivos da rede assumirão que o primeiro bit representa o início de uma nova mensagem. Os dispositivos entenderão que a mensagem chegou ao fim quando o tempo decorrer novamente. Caso ocorra durante a transmissão de

uma mensagem o tempo entre os caracteres for maior que 1,5 vezes o tempo de um caractere, a mensagem é considerada inválida e o Mestre irá descartar estes dados. Desta forma, cada mensagem deverá ser transmitida em um fluxo contínuo de caracteres. O método de checagem de erro neste modo é o CRC, do inglês *Cyclic Redundancy Check* (MODBUS, 2002).

Em relação ao campo Código de Função, o protocolo especifica a faixa entre 1 a 255, sendo que apenas a faixa entre 1 a 127 é utilizada, pois o bit mais significativo é utilizado para respostas de exceção. As funções mais utilizadas em dispositivos que utilizam o protocolo Modbus são mostradas no Quadro 4 (MODBUS, 2006).

Código de Função	Descrição
1 – Read Coils	Faz a leitura de um bloco de bits do tipo coil.
2 – Read Discrete Inputs	Faz a leitura de bits do tipo entradas discretas.
3 – Read Multiple Registers	Faz a leitura de um bloco de registradores do tipo holding.
5 – Write Single Coil	Faz a escrita em um único bit do tipo coil.
6 – Write Single Register	Faz a escrita em um único registrador do tipo coil.
16 – Write Multiple Registers	Faz a escrita em um bloco de registradores do tipo holding.

Quadro 4 – Principais funções do protocolo Modbus.

Fonte: Modbus (2006, p. 11).

Para este trabalho destaca-se o funcionamento da função 3. A PDU de requisição informa no campo de dados o endereço do registro inicial e o numero de registros. Na PDU os registros são endereçados a partir de zero. Desta forma registros numerados de 1-16 são endereçados como 0-15. Logo, no campo de dados para esta requisição do Mestre, são utilizados dois bytes para informar o endereço de início e dois bytes para informar o numero de registradores (MODBUS, 2006).

A Figura 11 mostra como seria uma requisição de código de função 3 para ler registradores na faixa entre 108-110.

Request		Response	
Field Name	(Hex)	Field Name	(Hex)
Function	03	Function	03
Starting Address Hi	00	Byte Count	06
Starting Address Lo	6B	Register value Hi (108)	02
No. of Registers Hi	00	Register value Lo (108)	2B
No. of Registers Lo	03	Register value Hi (109)	00
		Register value Lo (109)	00
		Register value Hi (110)	00
		Register value Lo (110)	64

Figura 11 – Exemplo de requisição para leitura de registradores

Fonte: Modbus (2006, p. 15).

A requisição informa que o endereço começa em 6Bh (107) e tem 3h (3) registradores, o escravo então responde com os respectivos valores no campo de dados da PDU de resposta: para o registrador 108 o valor 02 2Bh (555), para o registrador 109 o valor 00 00h (0) e para o registrador 110 o valor 00 64h (100) (MODBUS, 2006).

A Figura 12 mostra o diagrama de estados para o tratamento de uma requisição modbus por um escravo. O escravo ao receber uma requisição, realiza a validação do campo de código de função para verificar se a função está implementada. A seguir, é verificado se os registradores são suportados e é verificado se o endereço requisitado está na faixa de endereçamento correta. Após o processamento da requisição, é realizada a leitura dos registradores e enviado uma resposta. Caso alguma condição não for atendida, uma resposta de exceção correspondente é enviada (MODBUS, 2006).

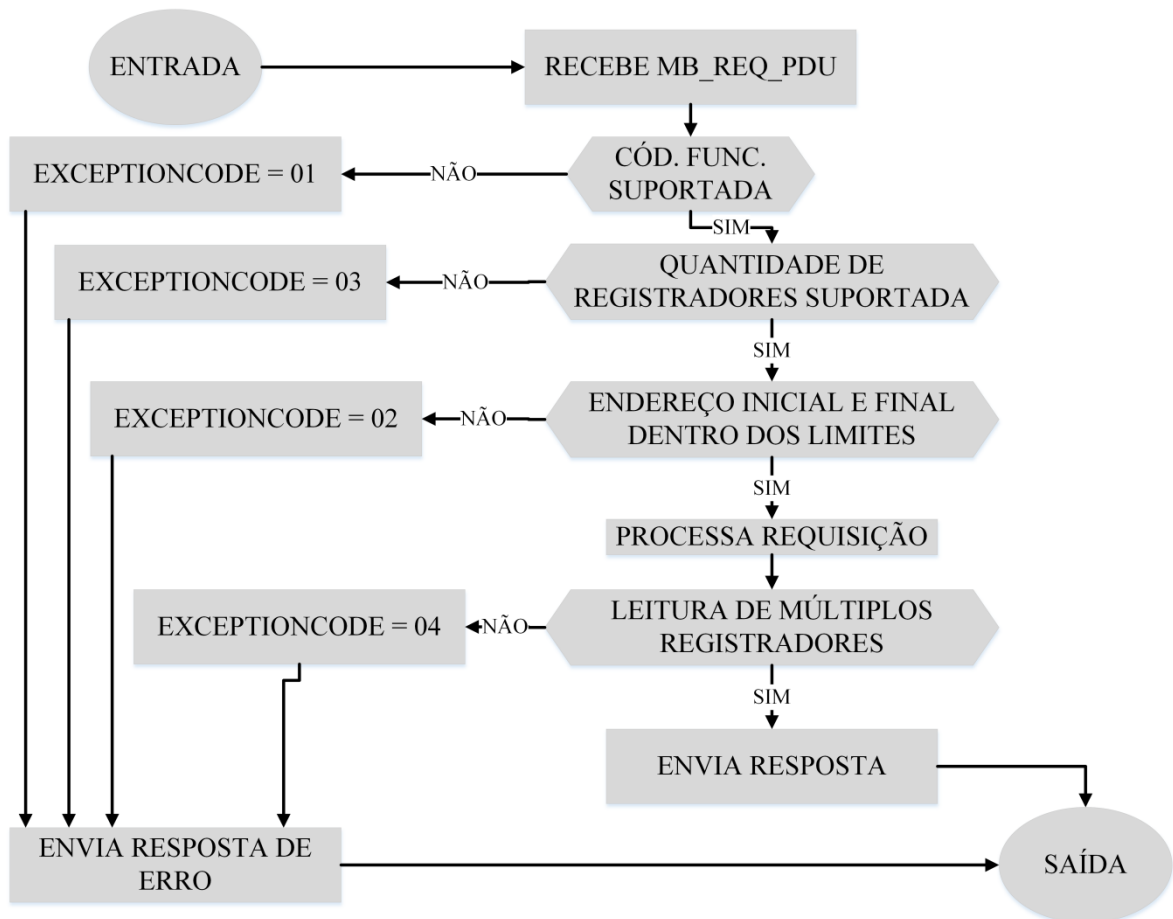


Figura 12 – Diagrama de estados para Função 03.
 Fonte: Adaptado de Modbus (2006, p. 16).

3 MATERIAIS E MÉTODO

O sistema proposto utiliza uma arquitetura mestre/escravo com protocolo Modbus em um barramento RS-485, no qual os escravos são dispositivos sensores microcontrolados responsáveis por coletar dados das variáveis do silo e, utilizando um meio físico, transportá-los para o mestre, sendo este um software supervisorio executando em computador de propósito geral que tem por função coletar os dados dos escravos, armazenar e mostrar os valores por meio de uma interface visual.

A Figura 13 mostra os componentes do processo. O componente TTL/RS-485 serve para converter os níveis de tensões do padrão TTL para o nível de tensão do padrão RS-485.

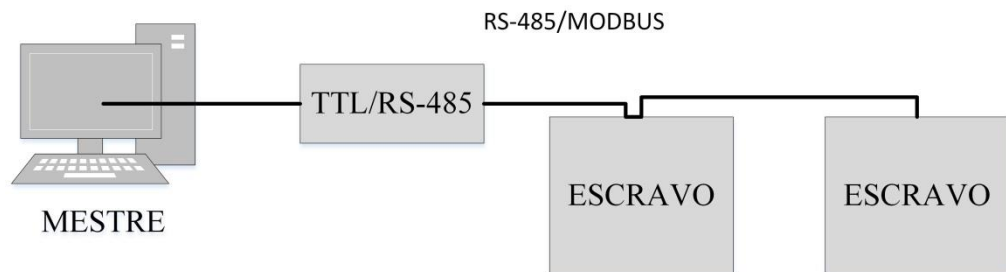


Figura 13 – Diagrama de componentes e conexões do sistema

3.1 MATERIAIS

O Quadro 5 resume os principais materiais utilizados no projeto e suas funções.

Material	Descrição
PT-100	Sensor de temperatura resistivo
MSP430G2553	Microcontrolador
MAX485	Conversor entre TTL e RS-485
DHT11	Sensor digital temperatura e umidade relativa do ar
PLACA FTDI FT232RL	Conversor entre USB e TTL
SCADABR	<i>Software</i> supervisorio
INA826	Amplificador de instrumentação

Quadro 5 – Relação entre nome dos materiais e sua descrição

Nas seções a seguir é feito um detalhamento de cada material a ser utilizado no projeto.

3.1.1 SENSOR DE TEMPERATURA

Para mensurar a temperatura dos grãos no silo foi escolhido o sensor PT-100 na versão em inox (V4A) disponível na UTFPR – Câmpus Pato Branco. Este sensor é o mais usado na indústria devido sua precisão e boa linearidade. Estes fatores influenciaram na escolha para este trabalho.

Este sensor é formado pelo metal platina e possui resistência de 100Ω a 0°C . Pode medir temperaturas na faixa de -40°C a 500°C , o que torna suficiente para mensurar as temperaturas no interior do silo que normalmente são entre 10°C a 120°C . Possui erro de $0,1^{\circ}\text{C}$ na faixa de 0°C a 100°C , a corrente de excitação deve ser menor que $0,3\text{mA}$, seus coeficientes são $a_1 = 3,9 \cdot 10^{-3} \text{ } 1/^{\circ}\text{C}$ e $a_2 = -5,8 \cdot 10^{-7} \text{ } 1/^{\circ}\text{C}^2$ para uma faixa de temperatura de 0°C até 850°C .

3.1.2 Microcontrolador

O microcontrolador MSP430G2553 é fabricado pela Texas Instruments, possui tensão de alimentação de $3,6 \text{ V}$, 2 timers de 16 bits, 1 conversor Analógico-Digital (AD) de 10 bits, 3 modos de comunicação (UART, SPI e I²C), memória flash de 16Kb, memória SRAM de 512bytes e frequência de operação de 16MHz. Possui um Kit de desenvolvimento chamado Launchpad e o fabricante disponibiliza o software Code Composer para o desenvolvimento de aplicações (TEXAS, 2013).

Este microcontrolador foi escolhido devido ao seu custo relativamente baixo e suas especificações serem suficientes para este trabalho, além do autor já possuir alguma experiência no mesmo.

3.1.3 Conversão TTL/RS-485

O Max485 é um circuito integrado capaz de converter tensões de nível TTL para RS-485. Possui tensão de alimentação de 5V e opera em temperaturas entre 0 a 70°C . Suporta até 32 dispositivos no mesmo barramento (MAXIM, 2003).

3.1.4 Sensor digital de umidade e temperatura

O DHT11 é um sensor digital composto de dois sensores. Um sensor de temperatura e um sensor de umidade relativa do ar. O sensor de temperatura é um termistor do tipo NTC e o sensor de umidade é baseado no efeito resistivo.

Possui faixa de medição de umidade entre 20 a 90% com precisão de $\pm 5\%$ e mede temperaturas na faixa de 0 a 50°C com precisão de $\pm 2^\circ\text{C}$. Alimentação de 3 a 5V e tempo de resposta de 2 segundos.

3.1.5 Conversão USB/TTL

A placa conversora é formada pelo integrado FT232RL do fabricante FTDI. Vem implementado no integrado o protocolo USB 2.0. A placa contém pinos para indicar atividade de transmissão e recebimento de dados. Possui drivers para Windows, Linux e MAC OSX. Pode ser alimentado com tensões de 3,3 a 5,25V.

3.1.6 Supervisório

O *software* ScadaBR foi escolhido por ser “*open-source*” e possuir licença gratuita. O ScadaBR é um software do tipo SCADA (*Supervisory Control and Data Acquisition*) que significa controle supervisão e aquisição de dados.

Um sistema SCADA tem por finalidade servir de interface entre um operador (pessoa) e processos automatizados. Além de monitorar o sistema também pode, por meio de uma programação lógica, controlar o processo. O sistema deve comunicar-se com os equipamentos do processo, possuir um sistema para registro de dados e uma interface gráfica para o operador, denominada interface homem-máquina (IHM) a qual disponibiliza elementos gráficos para representar o processo de forma a ser o mais próximo ao processo real. As principais características de um sistema SCADA são (ROSÁRIO, 2005):

- Proporcionar facilidade na visualização gráfica e operação do sistema;
- Geração automática de relatórios;

- Acompanhamento das variáveis controladas;
- Acesso automático a um banco de dados;
- Acionamento de alarmes;

O ScadaBR é uma aplicação multiplataforma com base na tecnologia Java. Ser multiplataforma significa que a aplicação pode ser executada em diversos sistemas operacionais a partir de um servidor de aplicações. Ao executar o aplicativo, que pode ser acessado de um navegador de Internet, a interface principal do ScadaBR fornece visualização das variáveis, gráficos, configuração dos protocolos, alarmes e construção de telas do tipo IHM com várias opções de configuração. Após as configurações de protocolos e entradas e saídas é possível montar interfaces de operador Web utilizando o navegador além de possibilitar a criação de aplicativos personalizados em qualquer linguagem de programação a partir do código-fonte disponibilizado ou de sua API “web-services” (SCADABR, 2010).

3.1.7 Amplificador de instrumentação

O INA826 é um amplificador de instrumentação fabricado pela Texas Instruments. Possui as seguintes características (TEXAS, 2016):

- Rejeição de modo comum de 104 dB, Min (G = 10) e 100 dB, Min em 5kKz(G = 10)
- Baixa tensão de offset: 150 μ V, Max.
- Ruído: $18\text{nV}/\sqrt{\text{Hz}}$, $G \geq 100$.
- Consumo de corrente: 200 μ A.
- Tensão de operação: Single Supply entre 3 a 36V e Dual supply $\pm 1,5$ a ± 18 V.

O ganho do amplificador é definido pela Equação (3).

$$G = 1 + \left(\frac{49,4K\Omega}{Rg} \right) \quad (3)$$

Outros componentes que também foram utilizados são: Resistores, capacitores, LEDs, conectores borne, botões, cabos, *jumpers*, e placas de cobre.

Além dos materiais destacados, para o desenvolvimento foi utilizado os seguintes recursos de software:

- Code Composer Studio e linguagem C para programação dos microcontroladores.

- TINA-TI para simulação de circuitos.
- EAGLE para confecção de placa de circuito impresso.
- Biblioteca para comunicação com sensor DHT11.
- Biblioteca FreeModbus.

3.2 MÉTODO

O desenvolvimento do sistema pode ser dividido em 3 partes principais. O desenvolvimento: dos dispositivos sensores (escravos), da rede mestre/escravo e do sistema supervisor (mestre).

3.2.1 Dispositivo sensor

Cada dispositivo sensor é composto de sensores, circuitos de instrumentação, uma unidade de alimentação, um MSP430G2553 e um MAX485. A Figura 14 mostra o diagrama de componentes e conexões de um dispositivo sensor. Apenas um dos dispositivos sensores conterá o sensor DHT11 para mensurar a temperatura e umidade relativa do ar. Como este sensor tem interface digital, ele se comunica diretamente com o MSP430G2553.

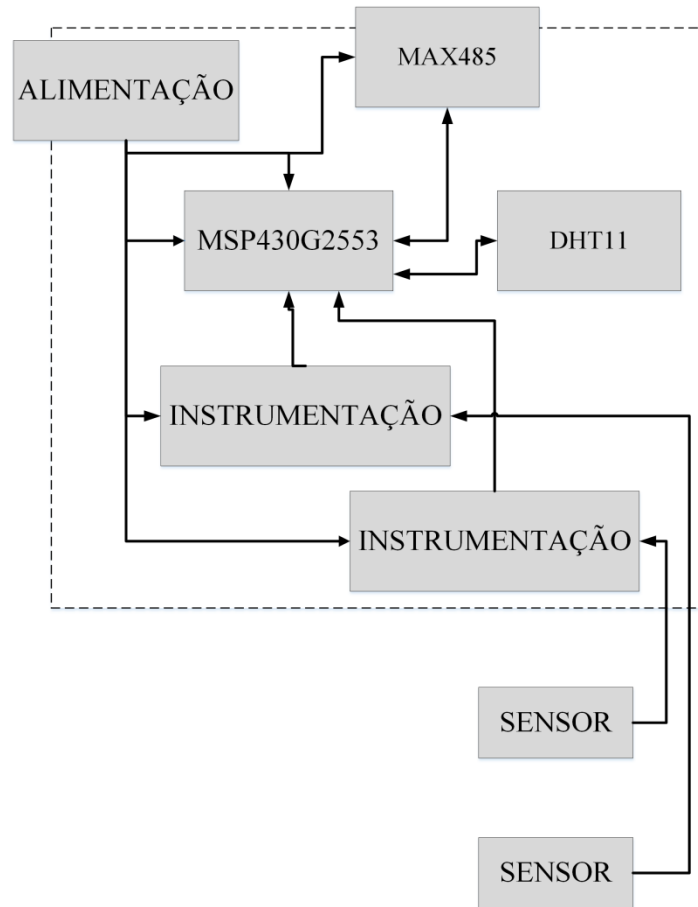


Figura 14 – Diagrama de componentes e conexões de um dispositivo sensor

Os dispositivos sensores são alimentados por fontes de bancada, fornecendo tensões de 5V, 3,6V, -3,6V e GND e ficarão posicionados fora da massa de grãos para que a umidade dos grãos não os danifique. Desta forma, somente os sensores PT-100 estarão imersos na massa de grãos através de cabos que podem chegar a até 20 metros. Como esta distância pode interferir na leitura devido à resistência elétrica dos cabos, este problema poderá ser tratado no bloco de instrumentação.

No bloco de instrumentação foi utilizado um circuito elétrico que tem como função adequar o sinal dos sensores PT-100 para a leitura do A/D no MSP430G2553. O circuito utilizado é uma ponte de Wheatstone com ligação de três fios, que foi primeiramente simulado pelo software TINA-TI, a Figura 15 mostra o circuito. O fabricante TEXAS INSTRUMENTS fornece uma planilha para o cálculo dos resistores utilizados na ponte, sendo necessário informar as variáveis desejadas. Desta forma foi definido para uma faixa de temperatura entre 0°C a 50°C a saída (V_o) do INA826 seja entre 0 a 3,6V, compatível com as tensões do microcontrolador utilizado. Os resistores R26, R27 e R28 são para simular a distância dos cabos. Foram utilizados trimpots de 500Ω para os resistores R2 e R4 com a finalidade de realizar ajustes, pois os resistores não possuem valores perfeitos.

Como a fonte para a ponte é de 5V, os resistores R3 e R9 são de cerca de 5k Ω para que a corrente que circula pelo PT-100 fique em torno de 1mA, o recomendado para este sensor (TEXAS, 2011).

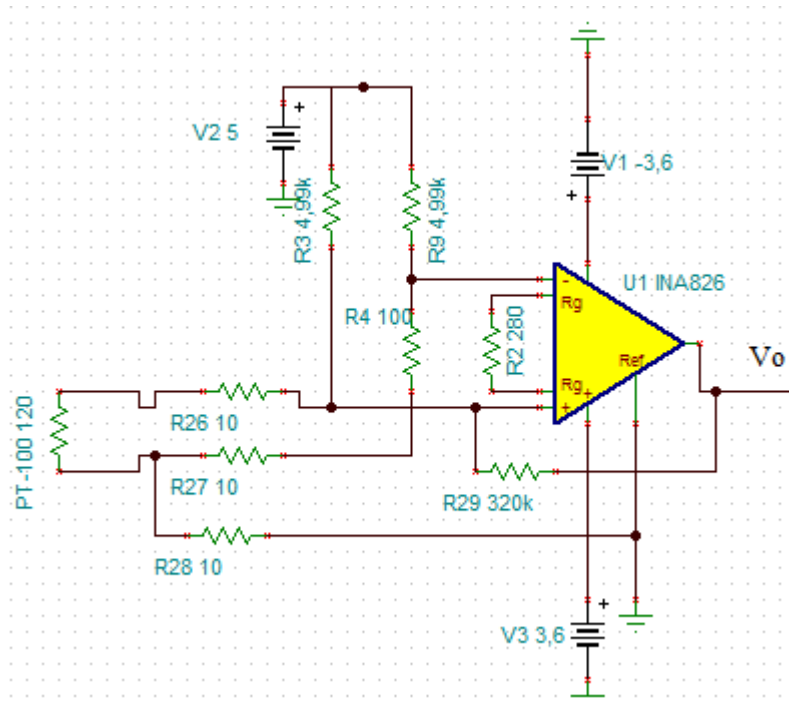


Figura 15 – Circuito de instrumentação

Após desenvolvido o circuito de instrumentação, o próximo passo foi programar o microcontrolador para realizar as seguintes atividades:

- Converter o sinal analógico da instrumentação para digital;
- Comunicar-se com o sensor DHT11;
- Comunicar-se com a rede mestre/escravo;

Para a programação dos microcontroladores dos dois escravos foi utilizado linguagem C e o software Code Composer. Primeiramente foi escolhida a fonte de clock a ser utilizada, sendo escolhido o oscilador interno DCO (*Digitally Controlled Oscillator*) utilizado para o clock principal (MCLK) e secundário (SMCLK). A seguir, foram configuradas as portas do microcontrolador. As funções para estas configurações são mostradas na Figura 16.

```

//Configuracao do clock
void config_clock(void)
{
    WDTCTL = WDTPW + WDTHOLD; // Para o WDT
    DCOCTL = CALDCO_1MHZ; // DCO com freq. calibrada de 1 MHz
    BCSCTL1 = CALBC1_1MHZ;
}
//Configuracao das portas
void config_portas(void)
{
    P1DIR = BIT0 + BIT1 + BIT2 +BIT3 + BIT4 + BIT5 + BIT6 + BIT7;
    P1SEL = BIT1 + BIT2; // Conecta pinos 3 e 4 nos sinais UCA0TXD/UCA0RXD
    P1SEL2 = BIT1 + BIT2;
    P1IFG = 0; // limpa flag de interrupcao da P1.
    P2DIR |= BIT0+BIT1+BIT2+BIT3+BIT4+BIT5+BIT6+BIT7; //toda a porta P2 como saida
    P2OUT = 0;
}

```

Figura 16 – Configuração de clock e portas do microcontrolador

A seguir, foram implementadas três funções para o conversor A/D do microcontrolador. Para inicializa-lo, para tratamento de interrupções e para leitura dos canais. A função de inicialização configura o A/D para múltiplos canais, sendo cada canal conectado a saída de cada circuito de instrumentação. O disparo de amostragem é por software, assim, os dados somente serão amostrados quando o escravo for requisitado. A Figura 17 detalha as funções para o A/D.

```

//Configuracao do ad
void ini_ADC10(void){
/* clock: MCLK
 * realiza amostragem a partir de canal 7
 * sequencia de canais sem repeticao
 * interrupcao por software
 */
    ADC10CTL1 = INCH_7 + CONSEQ_1 + ADC10SSEL_2 + SHS_0;
    ADC10CTL0 = ADC10SHT_2 + MSC+ ADC10ON + ADC10IE;
    ADC10DTC1 = 8; //tamanho do bloco a ser transferido
    ADC10AE0 = BIT6 + BIT7; // Habilita os canais para ad
    ADC10CTL0 |= ENC +ADC10SC ; // Habilita conversao (gera borda de subida)
}
//Leitura do ad
void ler_adc(void)
{
    ADC10CTL0 &= ~ENC; //desabilita conversao
    while (ADC10CTL1 & BUSY); // espera se o ad esta ocupado
    ADC10SA = (unsigned int)adc; //define um vetor para copiar os dados da conversao
    ADC10CTL0 |= ENC + ADC10SC;// habilita amostragem e conversao
}
//Tratamento de interrupcoes
#pragma vector=ADC10_VECTOR
__interrupt void ADC10_ISR(void)
{
    ADC10CTL0 &= ~ADC10IFG;//limpa flag de interrupcao
    ADC10SA = (short)&adc[0];//vetor para enviar a conversao
    __bic_SR_register_on_exit(CPUOFF);// Clear CPUOFF bit from 0(SR)
}

```

Figura 17 – Funções para o A/D

O próximo passo foi a comunicação com o sensor DHT11, para isso foi utilizado uma biblioteca desenvolvida por Bryce Feigum disponível em <https://github.com/bafeigum/DHT11-Library-for-MSP430>, qual necessita de um bit de uma porta e um timer. Desta forma foi desenvolvido duas funções, uma para inicializar o timerA1 e outra para tratamento de interrupções, mostradas na Figura 18.

```

//Tratamento de interrupcoes
#pragma vector = TIMER1_A0_VECTOR
__interrupt void CCR0_ISR(void){
    TOUT=1;
    CLR (TA1CCTL0, CCIFG);
}
//Configuracao do timer A1
void config_timer_A1(void)
{
    /*fonte de clock: SMCLK
    * fator divisão: 4
    * modo up
    */
    TA1CCR0 = 50000; // inicializa timer para contar em 5Hz
    TA1CTL = TASSEL_2 + ID_2 + MC_1 + TACLK;
}

```

Figura 18 – Funções para o sensor DHT11

Para comunicar-se com a rede mestre/escravo foi utilizado foi utilizado o FreeModbus, uma implementação de código aberto para o escravo do protocolo Modbus, disponível em <http://www.freemodbus.org/>. Esta implementação possui porte para o MSP430 e utiliza um timer e a UART. Algumas funções também vem implementadas pela biblioteca, como a função *Read Multiple Registers* de código 3, já explanada no Quadro 4.

A Figura 19 mostra a função main executada no microcontrolador, primeiramente é chamada a função eMBInit para inicializar a configuração do porte serial, sendo passado os parâmetros para a comunicação, sendo elas: Modo RTU, a identificação do escravo (neste caso 1) e demais parâmetros que serão utilizados por esta função para configurar o modo UART do microcontrolador e são mostrados na Figura 19. Depois de configurado a inicialização e habilitado o protocolo pela função eMBEnable, o escravo fica na espera de alguma requisição pelo mestre, executando a função eMBPoll em um laço repetitivo.

```

void main(void)
{
    volatile eMBErrorCode    eStatus;
    WDTCTL = WDTPW | WDTHOLD; // Stop watchdog timer
    config_clock();
    config_portas();
    ini_ADC10();
    #if S_TH //se este escravo nao possuir o DHT11, as funcoes do mesmo nao sao compiladas
    config_timer_A1();
    #endif
    _enable_interrupt(); //Habilita interrupcoes
    //Configuracoes da UART: 9600 bps / 8 bits / sem paridade/1 stop bit
    //esta configurado direto no port
    if( ( eStatus = eMBInit( MB_RTU, 1, 0, 9600, MB_PAR_NONE ) ) != MB_ENOERR )
    {
    }
    /* Habilita o protocolo Modbus RTU */
    else if( ( eStatus = eMBCEnable( ) ) != MB_ENOERR )
    {
    }
    else
    {
        for(;;)
        {
            ( void )eMBPoll( );
            habilita=1; //variavel para controle de execucao
        }
    }
}

```

Figura 19 – Função main.

No recebimento de uma requisição, a função eMBPoll verifica se o escravo solicitado é o mesmo e qual função a executar. Para este trabalho a única função usada é a 03. Deste modo, a função eMBRegHoldingCB é chamada, esta função foi implementada para que seja feita a leitura dos sensores quando o escravo for solicitado. Como a função pode ser chamada dependendo do numero de registradores, uma variável de controle foi criada para controlar o numero de execuções. A Figura 20 mostra a alteração na função de leitura de registradores.

```

case MB_REG_READ: //se for para leitura de registradores
while( usNRegs > 0 )
{
  if(habilita){//para executar somente uma vez,
  P1OUT &= ~(BIT0); //para indicar inicio da leitura de sensores
  #if S_TH
  TA1CCTL0 = CCIE;
  read_Packet(Packet);
  RH_byte1 = Packet[0];
  RH_byte2 = Packet[1];
  T_byte1 = Packet[2];
  T_byte2 = Packet[3];
  checksum = Packet[4];
  if (check_Checksum(Packet))
    SET (P1OUT, 0x40);
  SET (TA1CTL, TACLR);
  SET (TA1CTL, 0x10);
  TA1CCR0 = 50000;
  CLR(TA1CCTL0,CCIE);//desabilitar as interrupcoes
  usRegHoldingBuf[2]=RH_byte1;
  usRegHoldingBuf[3]=T_byte1;
  #endif
  ler_adc();
  usRegHoldingBuf[0]=adc[0]*0.0489;
  usRegHoldingBuf[1]=adc[1]*0.0489;
  P1OUT |= (BIT0); //para indicar termino do tratamento de sensores
  //pisca led

  habilita=0;
}
*pucRegBuffer++ = ( unsigned char )( usRegHoldingBuf[iRegIndex] >> 8 );
*pucRegBuffer++ = ( unsigned char )( usRegHoldingBuf[iRegIndex] & 0xFF );
iRegIndex++;
usNRegs--;
}

```

Figura 20 – Alteração na função eMBRegHoldingCB

Para transformar o valor do AD em valor de temperatura (°C), foi definido uma equação de reta. Como o AD do MSP430 utilizado possui 10 bits, a leitura pode variar de 0 a 1023, sendo que a temperatura varia de 0 a 50 °C. Logo pela equação 4 o coeficiente da reta resulta em 0.0489, valor este que é multiplicado pela leitura de cada canal do AD na Figura 15.

$$m = \frac{50 - 0}{1023 - 0} \approx 0.0489 \quad (4)$$

O próximo passo foi montar o circuito para o dispositivo sensor. Primeiramente foram realizados testes em uma protoboard com cada parte separada, o microcontrolador comunicando com o DHT11, o circuito de instrumentação com o microcontrolador e o microcontrolador comunicando com o mestre. Após a fase de testes, deu-se inicio ao desenvolvimento do circuito no software Eagle. A Figura 21 mostra o esquemático do circuito gerado. Os pinos 15 e 14 do MSP430 são conectados na saída (Vo) do INA826, já mostrado na Figura 15, possui um jumper para ativar o resistor de terminação caso este for o último escravo da rede e também para o pino 11 caso este escravo possua o sensor

3.2.2 Rede Mestre/Escravo

Para a comunicação entre os escravos e o mestre, a rede mestre/escravo terá como meio físico o padrão RS-485. O padrão RS-485 é vantajoso para esta aplicação, pois podem ser conectados até 32 escravos e com uma velocidade de comunicação de 9600bps o cabo da rede pode alcançar até 1,2Km, permitindo ao sistema interligar vários silos. Outra vantagem é o padrão por ser de comunicação diferencial, é robusto em relação a interferências.

Como este padrão trabalha com níveis de tensão diferentes das tensões usadas no computador de propósito geral e nos microcontroladores, será usado o componente MAX485 responsável por converter tensões TTL para tensões diferenciais. A Figura 23 mostra em (a) o circuito integrado MAX485 e sua pinagem, e em (b) mostra a aplicação do MAX485 em uma rede RS-485 comunicando-se com diversos dispositivos.

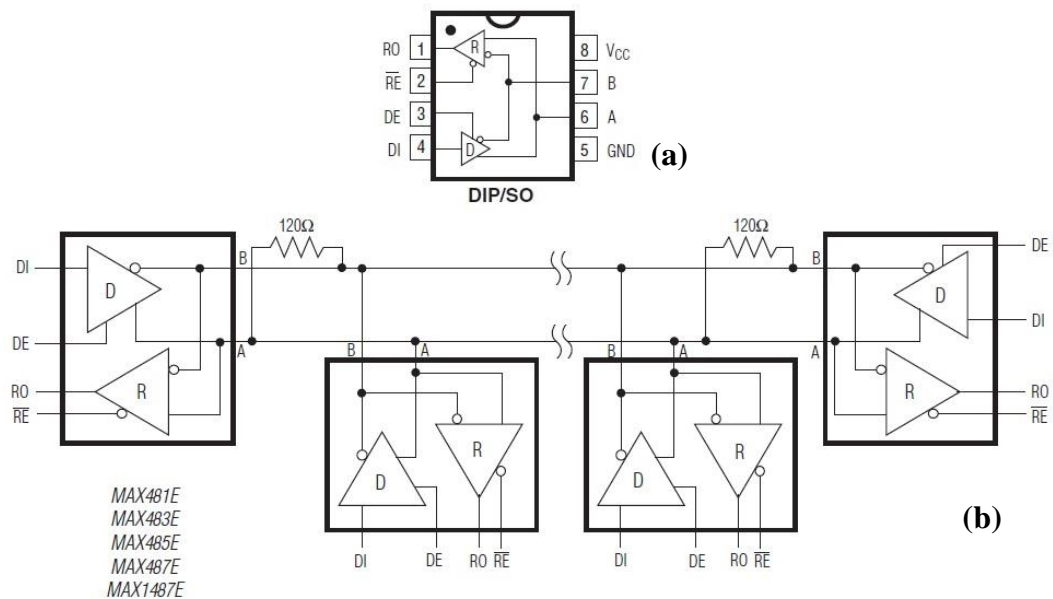


Figura 23 – MAX-485 (a) e rede RS-485 com resistores de terminação (b)
Fonte: MAXIM.

O pino 1 transmissão, o pino 4 recepção, os pinos 2 e 3 são para controle de recepção e transmissão, podem ser conectados servindo como um único pino de controle, quando em nível lógico 0, habilita recepção e desabilita transmissão, quando em nível lógico 1, desabilita recepção e habilita transmissão. Os pinos A e B são as saídas diferenciais.

O software supervisor utiliza uma saída USB, do inglês *Universal Serial Bus*, do computador, desta forma foi necessário desenvolver um dispositivo para realizar a conversão do padrão USB para RS-485 sendo utilizado o integrado FT232RL junto com um MAX485, sendo que o primeiro componente converte de USB para TTL e o segundo de TTL para RS-485.

O FT232RL é um integrado programável, porém já vem pré-programado de fábrica com algumas funções que são descritas em seu *datasheet*. Em sua pré-programação, o pino CBUS2 está programado para receber o sinal TXDEN, este sinal fica em nível alto quando o dispositivo envia dados e em nível baixo o restante do tempo. Logo foi conectado este pino junto com os dois pinos de controle do MAX485. A Figura 24 mostra o módulo utilizado que contém o FT232RL e o pino utilizado para o sinal TXDEN.



Figura 24 – Módulo FT232RL

Após montado o circuito e testado em protoboard, foi desenvolvido no software Eagle um desenho para montagem de sua placa, a Figura 25 mostra o esquemático. Onde foi utilizado *jumpers* (JP1, JP2, JP3) para escolha caso for necessário a utilização de resistor de terminação (R2) e os resistores de *pull-up* (R1) e *pull-down* (R3). O dispositivo é alimentado pela própria saída USB.

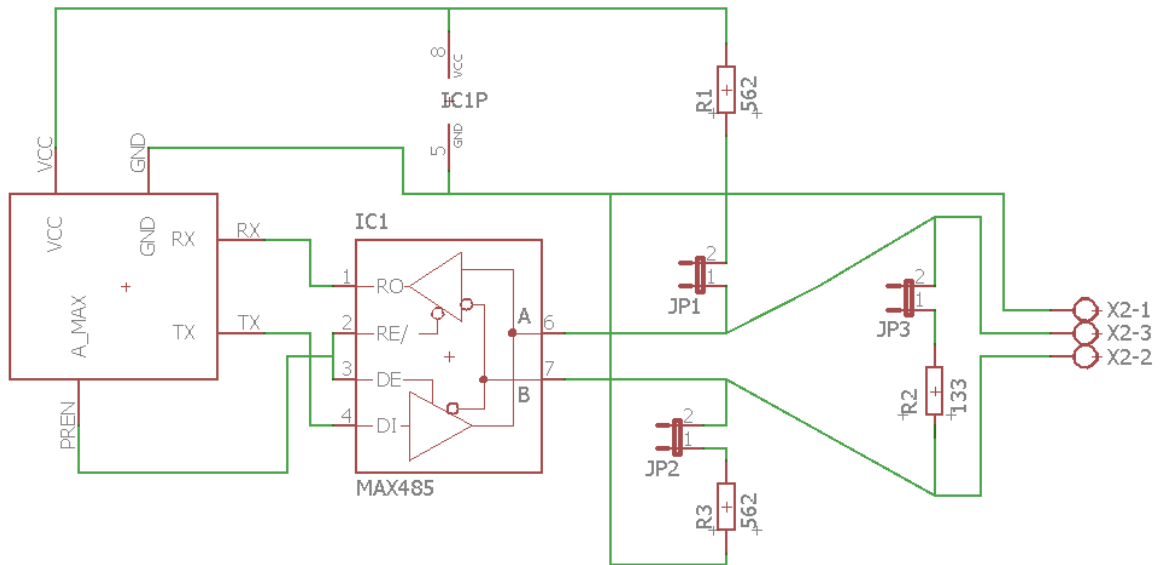


Figura 25 – Esquemático do conversor USB – RS-485

A Figura 26 mostra o desenho gerado a partir do esquemático com a organização dos componentes e as trilhas.

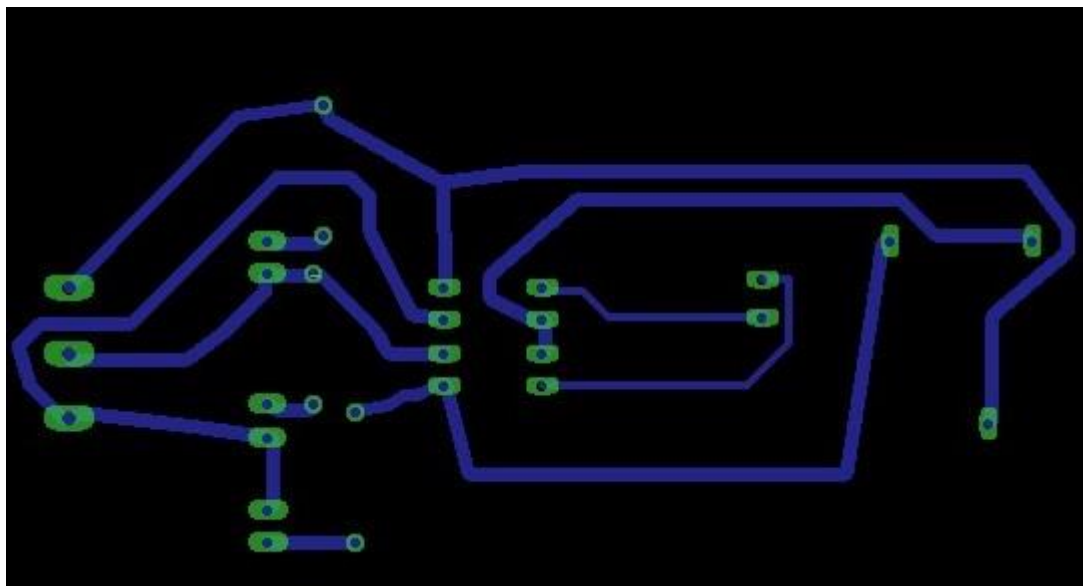


Figura 26 – Desenho para a PCI do conversor USB - RS-485.

3.2.3 Sistema supervisor

O software supervisor ScadaBR foi instalado em um computador de propósito geral. Inicialmente foi definida a base de dados do sistema, quais as variáveis envolvidas e como serão armazenadas em um banco de dados. Foi utilizado o banco de dados Derby que é disponibilizado na instalação do ScadaBR.

Entre as configurações do ScadaBR, primeiramente é definido um *Data Source*, do qual provém a origem dos dados. Para este trabalho foi utilizado um *Data Source* do tipo Modbus Serial, pois este tipo vem com a implementação do mestre do protocolo Modbus. A Figura 27 mostra a tela onde as propriedades do Data Source são configuradas. Foi configurado para o mestre enviar requisições a cada 3 segundos. O timeout é um tempo máximo que o mestre espera pela resposta de um escravo, se exceder este tempo, um alarme é acionado. Um detalhe importante é que os campos *Baud Rate*, *Data Bits*, *Stop Bits*, *Parity* e Codificação devem ter os mesmos valores que os configurados nos escravos.

Propriedades do modbus serial

Nome: Modb ser msp

Export ID (XID): DS_123521

Período de atualização: 3 segundo(s)

Quantificação:

Timeout (ms): 1000

Tentativas: 1

Apenas quantidades contínuas:

Criar pontos de monitor de escravo:

Máxima contagem de leitura de bits: 2000

Máxima contagem de leitura de registradores: 125

Máxima contagem de escrita de registradores: 120

Porta: COM4

Baud rate: 9600

Controle de fluxo de entrada: Nenhum

Controle de fluxo de saída: Nenhum

Data bits: 8

Stop bits: 1

Parity: Nenhuma

Codificação: RTU

Echo: Desligado

Simultaneidade: Função

Níveis de alarme de eventos

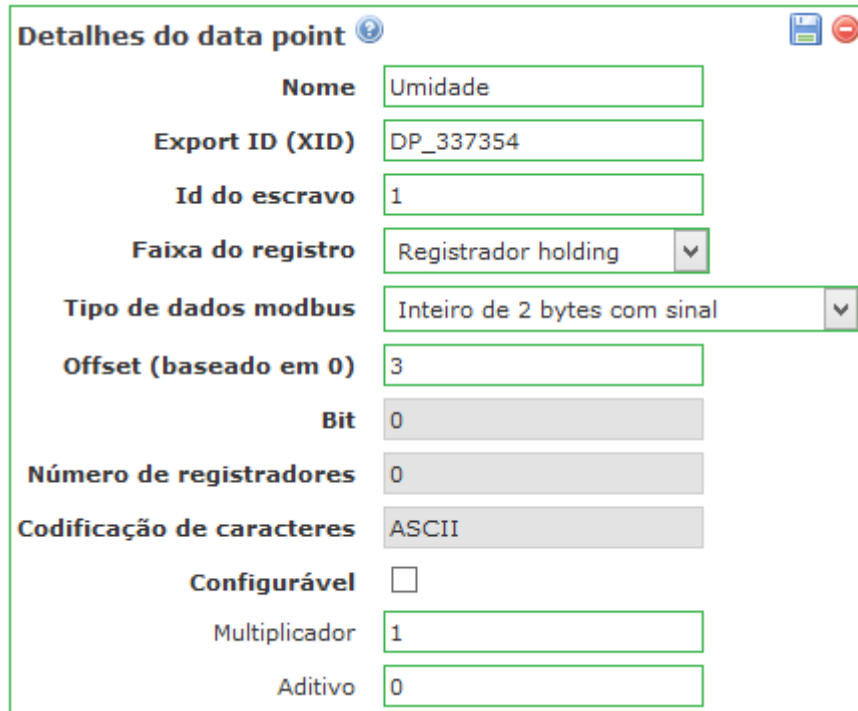
Exceção de data source: Urgente

Exceção de leitura de data point: Urgente

Exceção de escrita em data point: Urgente

Figura 27 – Configurações do Data Base.

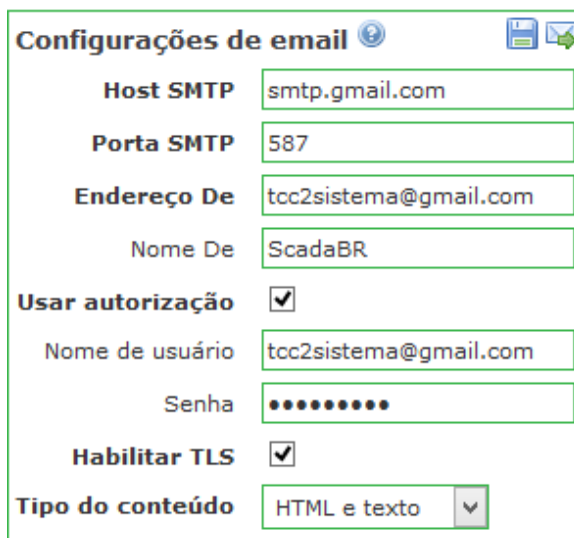
O próximo passo é adicionar *Data Points*, que são os dados a serem requisitados. Deve ser informado o número de identificação do escravo, o offset, que refere-se a posição do registrador e na Faixa do registro marca-se Registrador holding. A Figura 28 mostra a configuração de um *Data Point*.



Nome	Umidade
Export ID (XID)	DP_337354
Id do escravo	1
Faixa do registro	Registrador holding
Tipo de dados modbus	Inteiro de 2 bytes com sinal
Offset (baseado em 0)	3
Bit	0
Número de registradores	0
Codificação de caracteres	ASCII
Configurável	<input type="checkbox"/>
Multiplicador	1
Aditivo	0

Figura 28 – Configurações do Data Point.

Para o envio de e-mail, é necessário configurar uma conta de e-mail que o ScabaBR possa utilizar para o envio. Esta configuração está na aba Configurações do sistema. A Figura 29 mostra os campos para configuração, em que o Host e a Porta são referentes ao domínio do serviço de e-mail, neste caso, foi criada uma conta no Gmail.



Host SMTP	smtp.gmail.com
Porta SMTP	587
Endereço De	tcc2sistema@gmail.com
Nome De	ScadaBR
Usar autorização	<input checked="" type="checkbox"/>
Nome de usuário	tcc2sistema@gmail.com
Senha	••••••••
Habilitar TLS	<input checked="" type="checkbox"/>
Tipo do conteúdo	HTML e texto

Figura 29 – Configurações de e-mail.

Após a conta estar configurada, é necessário configurar na aba Usuários o e-mail do usuário, também pode ser escolhido para qual tipo de alarme o e-mail é enviado, neste caso foi escolhido o Crítico. A Figura 30 mostra essa configuração.

The screenshot shows a web interface for user management. On the left, a sidebar titled 'Usuários' contains two entries: 'admin' and 'cliente'. The main area is titled 'Detalhes de usuário' and contains the following configuration options:

- Nome de usuário:** admin
- Nova senha:** (empty field)
- Email:** guistanga@gmail.com
- Telefone:** (empty field)
- Administrador:**
- Desabilitado:**
- Enviar emails de alarme:** Crítico (dropdown menu)
- Recuperar eventos de auditoria próprios:**

Figura 30 – Configurações de e-mail para usuários.

Para o fornecimento de alarmes, cada Data Point deve ser configurado no campo Detectores de eventos. Para este trabalho foi configurado o acionamento de alarme Crítico para os sensores que ficam imersos nos grãos quando detectarem temperaturas acima de 30°C por mais de 10 segundos. O campo Alias é a mensagem que ficará gravada ao alarme correspondente. A Figura 31 mostra a tela de configuração de um *Data Point*.

The screenshot shows a web interface for event detectors configuration. The window is titled 'Detectores de eventos' and contains the following configuration options:

- Tipo:** Limite superior (dropdown menu)
- Tipo:** Detector de limite superior (text label)
- Export ID (XID):** PED_863339
- Alias:** Sensor A2 detectou temperatura acima do i (text field)
- Nível de alarme:** Crítico (dropdown menu)
- Limite superior:** 30
- Duração:** 10 (text field) and segundo(s) (dropdown menu)

Figura 31 – Configurações de alarme.

Também foram feitas configurações de geração de relatório diário que são enviados por e-mail, e configurado uma interface para que possam ser acompanhados os valores das temperaturas e umidade.

4 RESULTADOS E DISCUSSÕES

Após as etapas descritas até o momento, foram realizados testes em bancada com o protótipo montado. Para simular um silo os sensores foram colocados imersos em um recipiente contendo grãos de milho.

Foi feito ajustes com o trimpot de cada circuito de instrumentação para ajustar corretamente os valores de temperatura. O circuito de instrumentação comportou-se de maneira semelhante às simulações. Para verificar se o valor medido estaria correto, foi utilizado um multímetro para comparação, pois era o único instrumento para mensuração de temperatura disponível na sala de apoio da universidade. Os valores de temperatura provenientes dos sensores PT-100 mostraram-se semelhantes com os valores do multímetro, com erros próximos de 1°C devido à calibração com o trimpot não ficar perfeita. A Figura 32 mostra a parte superior da placa de um escravo montado junto com os cabos com os sensores PT-100, as fontes, os cabos para a rede RS-485 e contendo o sensor DHT11.

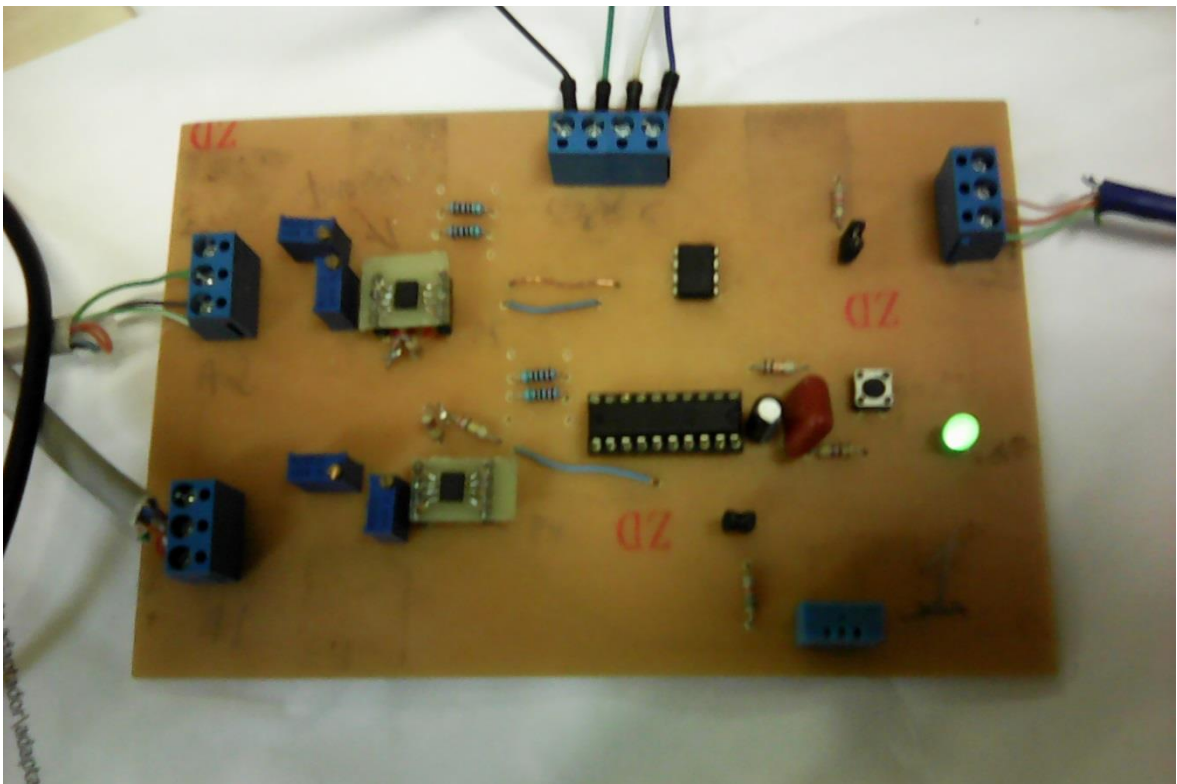


Figura 32 – Vista superior de um escravo.

O próximo passo foi testar a comunicação entre o mestre e os escravos. Foram realizados testes primeiramente sem os resistores de terminação, nos quais eventualmente

ocorreu alguma falha. Posteriormente, foram realizados testes com os resistores de terminação e os resistores de *pull-up* e *pull-down*, nos quais não houve ocorrência de falhas. A Figura 33 mostra uma captura dos sinais A (azul) e B (amarelo) do barramento RS-485 por um osciloscópio, o sinal em vermelho é a operação matemática $A - B$, resultando no sinal em padrão TTL.

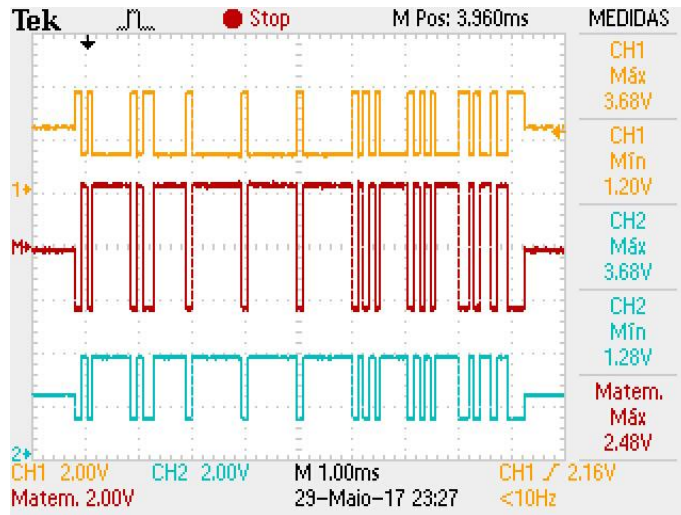


Figura 33 – Sinais do barramento RS-485.

A Figura 34 mostra a placa resultante do dispositivo conversor USB - RS-485 e as conexões do cabo USB e os cabos para a rede RS-485.

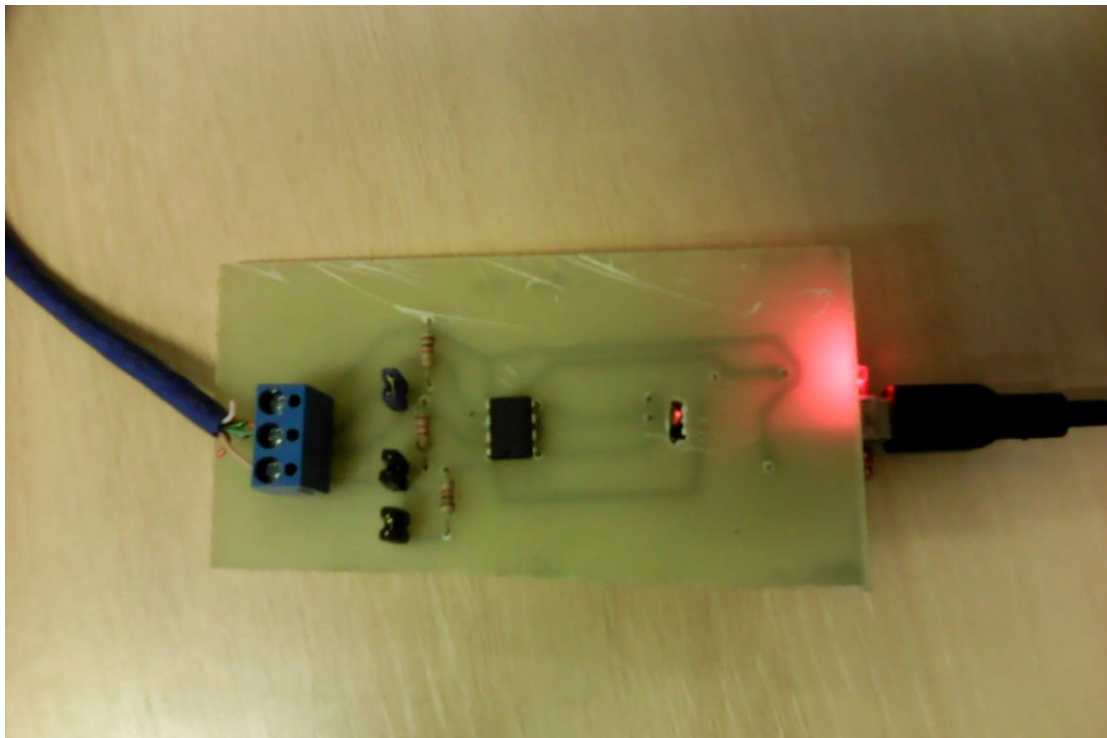


Figura 34 – Conversor RS-485 - USB.

A Figura 35 mostra a tela da IHM do ScabaBR com os valores dos sensores e a posição de cada sensor em um silo.

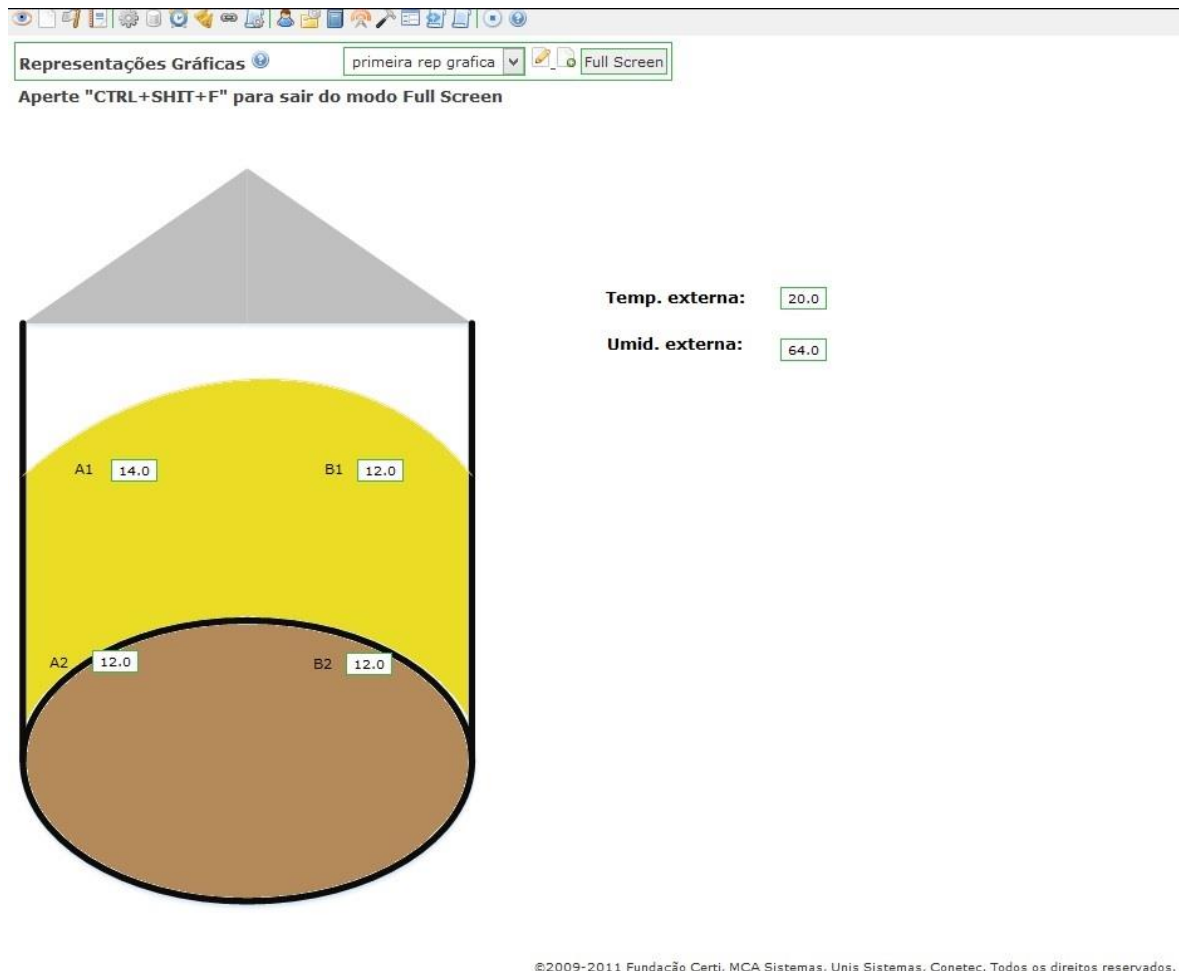


Figura 35 – Interface para visualização dos valores dos sensores.

Quando a temperatura de um dos pontos do silo passou de 30°C por mais de 10 segundos, um alarme sonoro e visual foi acionado no ScadaBR e um e-mail foi enviado para os usuários cadastrados. Além dos alarmes, o ScadaBR também enviou o relatório conforme o programado. A Figura 36 mostra o e-mail recebido por um usuário cadastrado.

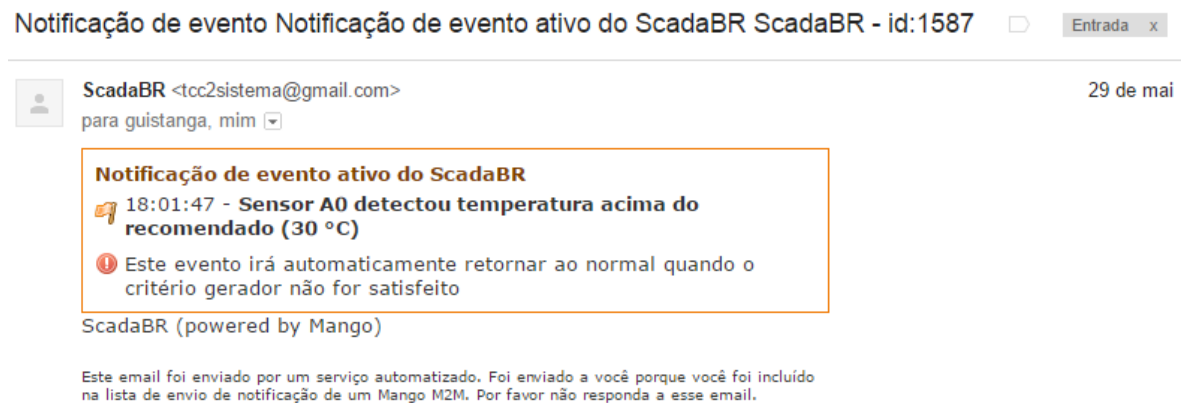


Figura 36 – E-mail do ScadaBR recebido.

E por fim, na Figura 37 é mostrado o sistema completo, com o ScadaBR executando em um computador, a rede RS-485 com o adaptador RS-485 – USB, os dispositivos sensores e um recipiente contendo grãos para os testes com os sensores pt-100.



Figura 37 – Imagem geral do protótipo.

5 CONSIDERAÇÕES FINAIS

De forma geral, todos os objetivos da proposta foram cumpridos. Foram projetados e desenvolvidos dispositivos sensores para medição e transmissão de amostras de temperatura e umidade. Foi realizada a instrumentação de sensores de temperatura. Foi desenvolvido uma rede RS-485 e aplicado o protocolo Modbus para comunicação entre o mestre e os escravos, e foi aplicado um software supervisor para visualização de dados e também envio de e-mail.

Desta forma, o sistema foi capaz de realizar a mensuração dos valores de temperatura e soar alarmes e enviar e-mail caso alguma temperatura no interior do silo passe de 30°C. Possibilitando ao operador de silo saber quando a temperatura passou do limite, para que ele possa tomar uma decisão.

O sensor PT-100 foi eficiente em sua função, porém, mesmo sendo um sensor com a vantagem de precisão e características lineares, traz desvantagens quando ao custo. O PT-100 custa em média R\$50,00 e para esta aplicação são necessários vários sensores, o que deixaria um projeto custoso. Além de seu circuito de instrumentação, que se não for corretamente calibrado o protótipo deixa de usufruir de suas vantagens.

Para esta aplicação não é necessário uma precisão da ordem de mais de uma casa decimal. Em conversa com técnicos que operam silos, pode ocorrer de em um dia ensolarado, mesmo o sistema de monitoramento da cerealista indicar temperatura elevada, os aeradores só são acionados na parte da noite, com a temperatura e umidade do ar mais baixa.

Uma melhoria que pode ser feita é adicionar uma lógica de controle para acionamento dos aeradores de forma automática. Outras melhorias que podem ser feitas são no circuito do dispositivo sensor, como adicionar filtros na saída da instrumentação e um circuito para impedir tensões negativas nas portas do AD do microcontrolador.

Em resumo, os seguintes tópicos destacam as possíveis melhorias no projeto:

- Estudar a substituição do sensor PT-100, levando em consideração sua instrumentação.
- Melhorar a alimentação do dispositivo sensor.
- Adicionar sistema de controle.
- Melhorar o circuito dos dispositivos escravos.
- Melhorar o desenho das placas de circuito impresso.

Após a entrega do trabalho foi realizado uma melhoria na alimentação do dispositivo sensor. Sendo criada uma fonte linear com reguladores de tensão para o dispositivo sensor não necessitar de fontes de bancada, para obtenção da alimentação diretamente da rede elétrica convencional, de 110V ou 220V.

REFERÊNCIAS

CONAB. **Acompanhamento da safra Brasileira - Grãos.** 2017. Disponível em: <http://www.conab.gov.br/OlalaCMS/uploads/arquivos/17_03_14_15_28_33_boletim_graos_marco_2017bx.pdf>. Acesso em 01 jul 2017

D'ARCE, Marisa A. B. Regitano. **Pós Colheita E Armazenamento De Grãos.** 2011. Disponível em <<http://www.esalq.usp.br/departamentos/lan/pdf/Armazenamentodegraos.pdf>> Acesso em: 07 nov. 2016.

DUNN, William C. **Fundamento de instrumentação e controle de processos:** Porto Alegre: Bookman 2013.

HELFRICK, Albert D. **Instrumentação eletrônica moderna e técnicas de medição.** 1 ed. Rio de Janeiro: Editora Prentice Hall do Brasil LTDA, 1994.

KRON **Conceitos Básicos de RS-485 e RS-422.** 2015 Disponível em: <www.kronweb.com.br/download2.php?id=353>. Acesso em: 07 nov. 2016.

KUROSE, James F; ROSS, Keith W. **Redes de computadores e a Internet: uma abordagem top-down.** 5 ed. São Paulo: Editora Pearson, 2010.

MAXIM. **Max-485E.** 2003. Disponível em: <<https://datasheets.maximintegrated.com/en/ds/MAX1487E-MAX491E.pdf>>. Acesso em: 08 nov 2015.

MERCEDES, Miguel G. **Instrumentación electrónica: transductores y acondicionadores de señal.** 1 ed. Santander: Editora Gráficas Apel, 2010.

MODBUS. **MODBUS APPLICATION PROTOCOL SPECIFICATION.** 2006. Disponível em: <http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf>. Acesso em: 07 nov. 2016.

MODBUS. **MODBUS OVER SERIAL LINE.** 2002. Disponível em: <http://www.modbus.org/docs/Modbus_over_serial_line_V1.pdf>. Acesso em: 07 nov. 2016.

ROSÁRIO, João Maurício. **Princípios de mecatrônica.** 1 ed. São Paulo: Editora Prentice Hall do Brasil LTDA, 2005.

SCADABR. **Manual de software.** 2010. Disponível em: <<http://www.scadabr.com.br/?q=downloads>>. Acesso em 01 nov 2015.

SILVA, Bruno S.; PEREIRA, Maurício de C.; BUCHMANN, Rafael M. **Padrões de comunicação serial clássicos: RS-232, RS-422 e RS-485.** Universidade Federal do Rio de Janeiro. Escola Politecnica, 2013.

SILVA, Luíz Cesar. **Secagem de Grãos.** 2005. Disponível em: <http://agais.com/manuscript/ag0405_secagem.pdf>. Acesso em: 21 set 2015

TEXAS. **Analog linearization of resistance temperature detectors.** 2011. Disponível em: <<http://www.ti.com/lit/an/slyt442/slyt442.pdf>>. Acesso em: 20 nov 2016.

TEXAS. **INA826 Precision, 200- μ A Supply Current, 3-V to 36-V Supply Instrumentation Amplifier with Rail-to-Rail Output.** 2016. Disponível em: <<http://www.ti.com/lit/ds/symlink/ina826.pdf>>. Acesso em: 08 nov 2016.

TEXAS. **Interface Circuits for TIA/EIA-485 (RS-485).** 2008. Disponível em: <<http://www.ti.com/lit/an/slla036d/slla036d.pdf>>. Acesso em: 10 out 2015.

TEXAS. **MSP430G2x53 Mixed signal microcontroller.** 2013. Disponível em: <<http://www.ti.com/lit/ds/symlink/msp430g2553.pdf>>. Acesso em 08 nov 2015.

THOMAZINI, Daniel. **Sensores industriais: Fundamentos e aplicações.** 1 ed. São Paulo: Editora: Érica, 2005.