

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ELÉTRICA
CURSO DE ENGENHARIA ELÉTRICA

PABLO HENRIQUE SEIBERT JAHNO

**IMPLEMENTAÇÃO DO PROTOCOLO MODBUS UTILIZANDO REDES SEM
FIO BASEADO NO PROTOCOLO IEEE 802.15.4**

TRABALHO DE CONCLUSÃO DE CURSO

PATO BRANCO

2016

PABLO HENRIQUE SEIBERT JAHNO

**IMPLEMENTAÇÃO DO PROTOCOLO MODBUS UTILIZANDO REDES
SEM FIO BASEADO NO PROTOCOLO IEEE 802.15.4**

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Conclusão de Curso, do Curso de Engenharia Elétrica do Departamento Acadêmico de Elétrica – DAELE – da Universidade Tecnológica Federal do Paraná – UTFPR, Câmpus Pato Branco, como requisito parcial para obtenção do título de Engenheiro Eletricista.

Orientador: Prof. Dr. Gustavo Weber Denardin,

PATO BRANCO

2016

TERMO DE APROVAÇÃO

O trabalho de Conclusão de Curso intitulado **IMPLEMENTAÇÃO DO PROTOCOLO MODBUS UTILIZANDO REDES SEM FIO BASEADO NO PROTOCOLO IEEE 802.15.4**, do aluno **PABLO HENRIQUE SEIBERT JAHNO** foi considerado **APROVADO** de acordo com a ata da banca examinadora N° **108** de 2016.

Fizeram parte da banca os professores:

Prof. Dr. Gustavo Weber Denardin

Prof. Dr. César Rafael Claire Torrico

Prof. Dr. Giovanni Alfredo Guarneri

A Ata de Defesa assinada encontra-se na Coordenação do Curso de Engenharia Elétrica

DEDICATÓRIA

Aos meus pais Gilmar Jahno e Sueli Terezinha Seibert.

AGRADECIMENTOS

Aos meus pais Gilmar e Sueli, e minha irmã Pricila, pelo incentivo e apoio nos momentos difíceis durante essa jornada.

Um agradecimento especial ao meu professor orientador Dr. Gustavo Weber Denardin, pela amizade, paciência e por estar presente em grande parte da minha graduação transmitindo conhecimentos e lições que não se encontram nos livros.

Aos meus colegas Otavio Augusto Gomes e Darlan Antônio Rigo pela amizade e apoio durante toda a trajetória do curso.

Aos amigos que fiz durante a graduação pelo companheirismo, apoio e experiências compartilhadas.

A UTFPR, por disponibilizar sua infraestrutura e os recursos necessários para minha graduação.

Aos professores do curso de Engenharia Elétrica pelo esforço, empenho, dedicação e todos os conhecimentos transmitidos durante o curso.

E por fim, agradeço a banca examinadora composta pelo Prof. Dr. César Rafael Claure Torrico e pelo Prof. Dr. Giovanni Alfredo Guarneri pelas considerações e esclarecimentos.

EPÍGRAFE

"Para se ter sucesso, é necessário amar de verdade o que se faz. Caso contrário, levando em conta apenas o lado racional, você simplesmente desiste. É o que acontece com a maioria das pessoas." (Steve Jobs)

RESUMO

JAHNO, Pablo H. S. Implementação do protocolo MODBUS utilizando redes sem fio baseado no protocolo IEEE 802.15.4. 2016, 53 f. Trabalho de Conclusão de Curso – Curso de Engenharia Elétrica, Universidade Tecnológica Federal do Paraná. Pato Branco, 2016.

Este trabalho apresenta as atividades desenvolvidas na busca do desenvolvimento de um protótipo de um módulo *gateway* MODBUS utilizando redes sem fio baseada no padrão IEEE 802.15.4. Inicialmente é apresentada a importância dos protocolos de comunicação na indústria e a utilização de redes sem fio. Na sequência é feita uma breve revisão teórica dos protocolos utilizados para desenvolvimento do protótipo. A partir dos conceitos apresentados os objetivos deste trabalho são introduzidos e são descritas as metodologias utilizadas para desenvolvimento do projeto. Após a implementação dos protótipos *gateway* e escravos, foram realizados diversos testes variando o tamanho do pacote transmitido e a distância entre os dispositivos. Dessa forma foi possível identificar a taxa de sucesso para cada cenário testado. E por fim são apresentadas as conclusões e considerações para trabalhos futuros.

Palavras-chave: MODBUS, IEEE 802.15.4, SCADA, Sistema Supervisório, Comunicação sem Fio.

ABSTRACT

JAHNO, Pablo H. S. Implementation of the MODBUS protocol using wireless networks based on protocol IEEE 802.15.4. 2016, 53 f. Trabalho de Conclusão de Curso – Curso de Engenharia Elétrica, Universidade Tecnológica Federal do Paraná. Pato Branco, 2016.

This work presents the activities developed in the search for the development of a prototype of a gateway module MODBUS using wireless networks based on the IEEE 802.15.4 standard. Initially, is presented the importance of communication protocols in the industry and the use of wireless networks. On the sequence is made a brief theoretical revision of the protocols used to develop the prototype. From the concepts presented the objectives of this work are introduced and described the methodologies used to develop the project. After the implementation of gateway and slave prototypes, several tests were performed, varying the size of the transmitted packet and the distance between the devices. In this way, it was possible to identify a success rate for each scenario tested. And finally, are presented the conclusions and considerations for future works.

Keywords: MODBUS, IEEE 802.15.4, SCADA, Supervisory System, Wireless Communication.

LISTAS DE FIGURAS

Figura 1 - Comparativo entre os protocolos sem fio.....	12
Figura 2 - Frame IEEE 802.15.4.....	17
Figura 3 - Pilha de comunicação MODBUS.	19
Figura 4 - Tela inicial ScadaBR.....	24
Figura 5 - Tela inserir data source ScadaBR.....	25
Figura 6 - Tela inserir data point ScadaBR.....	26
Figura 7 - Diagrama de blocos do sistema proposto.	28
Figura 8 -Modelo do sistema proposto	28
Figura 9 - Modulo de rádio MRF24J48MA.	30
Figura 10 - Kit Freescale Freedom FRDMKL25Z.....	31
Figura 11 - Data source configurado no ScadaBR.....	33
Figura 12 - Opções de configuração do data source.....	34
Figura 13 - Lista de data points configurados no sistema ScadaBR.	35
Figura 14 - Watch List ScadaBR.	35
Figura 15 - Dispositivo de hardware.....	37
Figura 16 - ScadaBR tela de representação gráfica.....	39

LISTAS DE QUADROS

Quadro 1 - Principais características físicas do padrão IEEE 802.15.4.	16
Quadro 2 - Frame MODBUS.	20
Quadro 3 - Funções definidas pelo protocolo MODBUS.	21
Quadro 4 - Exemplo de função sendo enviada pelo protocolo MODBUS.	22
Quadro 5 - Exemplo de função sendo recebida pelo protocolo MODBUS.	22
Quadro 6 - - Modelo OSI aplicado ao projeto proposto	29
Quadro 7- Dados teste 1.	41
Quadro 8 - Dados coletados teste 2.	42

LISTA DE ACRÔNIMOS

CAN	<i>Controller Network Area</i> (controlador de área de rede)
FINEP	Financiadora de Estudos E Projetos
LAN	<i>Local Area Network</i> (rede de área local)
LOWPAN	<i>Low Power Wireless Personal Area Networks</i> (rede sem fio de área pessoa de baixa potência)
MAC	<i>Media Access Control</i> (controle de acesso ao meio)
PAN	<i>Personal Area Network</i> (rede de área pessoa)
SCADA	<i>Supervisory Control and Data Acquisition</i> (sistemas de supervisão e aquisição de dados)
SCADABR	<i>Supervisory Control And Data Acquisition Brasileiro</i> (sistemas de supervisão e aquisição de dados brasileiro)
UFSC	Universidade Federal de Santa Catarina

LISTA DE ABREVIATURAS

ASCII	<i>American Standard Code for Information Interchange</i> (código padrão americano para o intercâmbio de informação)
BRTOS	<i>Brazilian Real Time Operating System</i> (sistema operacional de tempo real brasileiro)
CCA	<i>Clear Channel Assessment</i> (avaliação de canais livres)
CLP	Controlador Lógico Programável
CRC	<i>Cyclic Redundancy Check</i> (verificação de redundância cíclica)
DSSS	<i>Direct Sequence Spread Spectrum</i> (sequência direta de espalhamento do espectro)
FFD	<i>Full Function Device</i> (dispositivo de função completa)
GPL	<i>General Public License</i> (licença pública geral)
IEEE	<i>Institute of Electrical and Electronics Engineers</i> (instituto de engenheiros eletricitas e eletrônicos)
IHM	Interface Homem-Máquina
LLC	<i>Logical Link Control</i> (controle de enlace lógico)
LQI	<i>Link Quality Indicator</i> (indicador de qualidade de enlace)
MIT	<i>Massachusetts Institute of Technology</i> (instituto de tecnologia de Massachusetts)
O-QPSK	<i>Offset Quadrature Phase-Shift Keying</i> (modulação por deslocamento de fase em quadratura deslocada)
OSI	<i>Open Systems Interconnection</i> (interconexão de sistemas abertos)
PHY	<i>Physical Layer</i> (camada física)
RFD	<i>Reduce Function Devices</i> (dispositivos de função reduzida)
RSSF	Redes de Sensores Sem Fio
RSSI	<i>Received Signal Strength Indicator</i> (indicador de intensidade de sinal recebido)
RTOS	<i>Real Time Operating System</i> (sistema operacional de tempo real)
RTU	<i>Remote Terminal Unit</i> (unidade de terminal remota)
USB	<i>Universal Serial Bus</i> (barramento serial universal)

SUMÁRIO

Capítulo 1	10
INTRODUÇÃO	10
1.1 OBJETIVO GERAL.....	13
1.2 OBJETIVOS ESPECÍFICOS	13
1.3 ORGANIZAÇÃO DO TRABALHO.....	13
CAPÍTULO 2	15
FUNDAMENTAÇÃO TEÓRICA.....	15
2.1 PROTOCOLO IEEE 802.15.4	15
2.2 PROTOCOLO MODBUS.....	18
2.3 SISTEMAS SCADA.....	22
CAPÍTULO 3	27
MATERIAIS E MÉTODOS.....	27
3.1 DESCRIÇÃO DO SISTEMA	27
3.2 MATERIAIS	29
3.2.1 COMUNICAÇÃO SEM FIO.....	30
3.2.2 MICROCONTROLADOR.....	30
3.2.3 SISTEMA OPERACIONAL	31
3.3 MÉTODOS	32
3.3.1 SOFTWARE EMBARCADO	32
3.3.2 INTERFACE COM O SCADABR	33
CAPÍTULO 4	37
RESULTADOS.....	37
4.1 HARDWARE.....	37
4.2 SCADABR	38
4.3 TESTES.....	39

CAPÍTULO 5	43
CONCLUSÕES	43
REFERÊNCIAS.....	45

CAPÍTULO 1

INTRODUÇÃO

Os recentes avanços nos processos industriais têm elevado a complexidade dos acionamentos de máquinas e equipamentos no chão de fábrica, devido a sua maior interação com sistemas em rede. Tal conexão em rede permite não só a comunicação entre os dispositivos e equipamentos, como também possibilita a integração destes com sistemas de gestão da produção.

A comunicação entre dispositivos geralmente está sujeita ao que se conhece por pirâmide da automação, passando pela utilização de protocolos determinísticos no nível inferior da pirâmide (nível de chão de fábrica) até a utilização de protocolos mais complexos no alto nível da pirâmide (nível de gerenciamento). Esses protocolos têm como principal função padronizar a comunicação nas indústrias, de forma que qualquer fabricante possa disponibilizar equipamentos que se comuniquem com compatibilidade de padrões (BOYER, 1993).

Existem diversos protocolos para comunicação de equipamentos industriais em nível de chão de fábrica, sendo que dentre os mais utilizados podemos destacar: *Profibus*, *CAN (Controller Area Network)*, *Fieldbus Foundation*, *Devicenet* e *MODBUS* (VLAD, 2013).

A comunicação entre os dispositivos neste trabalho utiliza o protocolo MODBUS. Esse é um protocolo de mensagens estruturadas desenvolvido pela Modicon® em 1979. Essa característica permite seu uso em diversas áreas de aplicação, bem como com equipamentos de diferentes fabricantes, mantendo compatibilidade na troca de mensagens. Segundo Li Hui et al. (2012), o número de dispositivos MODBUS em 2004, era de aproximadamente 800 milhões, sendo que apenas cerca de 25% desses dispositivos eram produtos da *Schneider Electric*®. Atualmente o protocolo é mantido pela *MODBUS Organization Inc.* e diversos equipamentos, tais como, inversores de frequência, CLP's (Controlador Lógico Programável, IHM's (Interface Homem Máquina) entre outros (MODBUS , 2015), são comercializados com o protocolo embarcado. Embora este protocolo seja robusto e

largamente utilizado, ele fica de certa forma limitado em aspectos físicos pois em geral, utiliza-se o MODBUS em redes cabeadas de par trançado, neste contexto este trabalho pretende integrar a utilização de redes de sensores sem fio para compor a camada física, substituindo assim redes cabeadas por transmissores sem fio.

Para definição do protocolo de comunicação sem fio foram estudados alguns dos principais protocolos utilizados atualmente. Após o estudo optou-se pela utilização do protocolo IEEE 802.15.4. Pois considerando outros protocolos de redes sem fio, como o protocolo IEEE 802.11, percebe-se que este foi concebido para altas taxas de transmissão de dados e para redes de longo alcance com milhares de dispositivos presentes na mesma rede. Já o protocolo IEEE 802.15.1 também conhecido como *bluetooth*, foi concebido originalmente para substituir cabos, formando uma rede com até 8 dispositivos e com alcance de poucos metros.

Portanto, o protocolo IEEE 802.15.4 é a alternativa adequada para essa aplicação, pois possui capacidade para integrar diversas topologias de redes com capacidade para vários dispositivos conectados (GUARESE, 2012). A Figura 1 faz um comparativo entre taxa de transferência, alcance e aplicações, para os protocolos citados. É possível perceber que os protocolos IEEE 802.11 são indicados para compor uma LAN (*Local Area Network*), já os protocolos IEEE 802.15.4 e IEEE 802.15.1 são mais indicados para compor uma PAN (*Personal Area Network*).

O protocolo IEEE 802.15.4, é um protocolo que define as regras de acesso ao meio e a camada de enlace de dados. Se for considerado o modelo OSI (*Open Systems Interconnection*), o protocolo IEEE 802.15.4 define as camadas 1 (física) e parcialmente a camada 2 (enlace), com a subcamada de acesso ao meio (MAC – *Medium access control*) (IEEE 802.15.4, 2006).

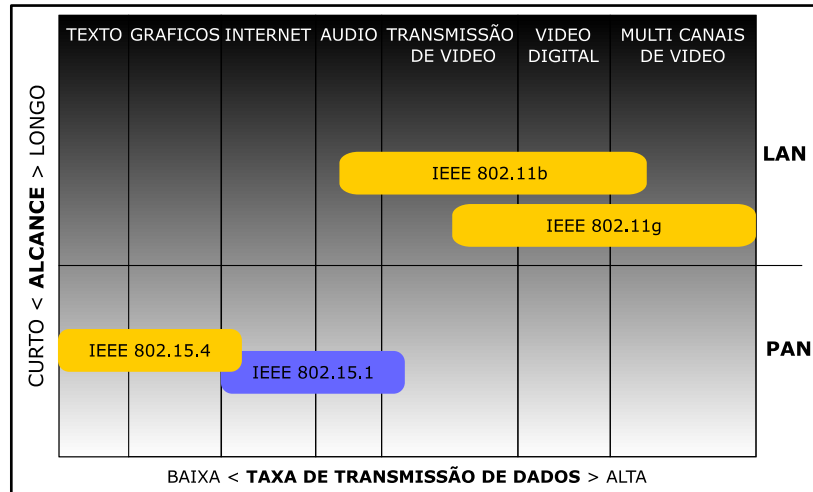


Figura 1 - Comparativo entre os protocolos sem fio.
Fonte: Aatoria própria, 2015.

Do ponto de vista da rede sem fio, a implementação do protocolo MODBUS forma uma rede ponto a ponto, pois a comunicação sempre ocorre entre o mestre e um escravo. Para ler e escrever dados em todos os nós escravos ocorre uma sequência de comunicações ponto a ponto. Em geral, a ordem de requisições segue o endereço numérico dos dispositivos escravos. Como o protocolo MODBUS especifica um tempo máximo de varredura de cada escravo, é possível afirmar que a comunicação terá comportamento determinístico.

O protocolo MODBUS é utilizado diversas vezes em conjunto com sistemas supervisórios de controle e aquisição de dados (SCADA - *Supervisory control and data acquisition*). Um *software* SCADA de fazer a aquisição de dados (de forma direta ou por comunicação com outros dispositivos), realiza tarefas de controle supervisório, apresentada dados de forma organizada (IHM – Interface Homem Máquina), gerencia eventos e alarmes e gerencia o histórico desses. Portanto é capaz de interpretar os dados vindos dos dispositivos conectados à rede e exibi-los ao usuário de forma rápida, clara e objetiva, sem que seja necessário conhecimento dos protocolos por parte do usuário. Ainda, é capaz de permitir o controle de dispositivos a distância, facilitando e agilizando o controle de equipamentos dentro da indústria (SCADABR, 2015) (SCADABR, 2010). Este trabalho pretende fazer a interconexão dos módulos sem fio com o *software* ScadaBR, por meio do protocolo MODBUS, para isso será desenvolvido um *gateway* que receberá os dados do computador e irá transmiti-los pela rede sem fio.

É possível perceber que é relevante a utilização de módulos MODBUS sem fio, pois há a possibilidade de utilização desse sistema em diversas aplicações, como por

exemplo em locais de difícil acesso ou de difícil instalação dos cabos convencionais. A utilização de módulos MODBUS escravos alimentados a bateria facilitaria a operação e manutenção da rede, além de poder reduzir custos com cabeamento, ou ainda dar a possibilidade de deslocamento.

1.1 OBJETIVO GERAL

O objetivo geral deste trabalho é desenvolver os protótipos de módulos MODBUS *gateway* e escravos, com comunicação sem fio baseada no protocolo IEEE 802.15.4. A comunicação entre o *software* ScadaBR e o módulo *gateway* MODBUS será por uma interface de comunicação serial *RS232*.

1.2 OBJETIVOS ESPECÍFICOS

Os principais objetivos específicos desse trabalho são:

- Realizar uma revisão bibliográfica e estudos acerca dos protocolos IEEE 802.15.4, MODBUS e do *software* ScadaBR;
- Implementar a camada física de enlace do padrão MODBUS a partir do protocolo IEEE 802.15.4;
- Implementar a camada de aplicação do protocolo MODBUS;
- Interconexão do módulo MODBUS mestre com o *software* ScadaBR;
- Implementar uma aplicação para a rede desenvolvida utilizando o *software* ScadaBR.

Realizar testes experimentais com o objetivo de comprovar a compatibilidade do sistema desenvolvido com dispositivos MODBUS, bem como verificar sua eficiência e confiabilidade.

1.3 ORGANIZAÇÃO DO TRABALHO

Este trabalho divide-se em cinco capítulos, de forma a englobar todo o conteúdo necessário para o desenvolvimento do projeto descrito. No primeiro capítulo,

foi apresentada uma breve introdução e justificativa aos temas relacionados, além de terem sido apresentados os objetivos que se pretende alcançar ao término do mesmo.

O segundo capítulo é composto por uma revisão bibliográfica de todo o conteúdo que se julga necessário para conclusão deste projeto. Nele são apresentadas as principais características sobre o padrão IEEE 802.15.4, sobre o protocolo MODBUS e sobre o sistema ScadaBR.

O terceiro capítulo apresenta a metodologia utilizada. É apresentada uma breve descrição dos materiais utilizados e feita uma descrição mais detalhada do sistema implementado.

O quarto capítulo apresenta os resultados e faz uma análise acerca dos testes efetuados com o sistema.

E por fim no capítulo cinco são apresentados os resultados e conclusões obtidos nesse trabalho, falhas e dificuldades, além de apresentar sugestões de melhorias que poderiam ser implementadas em trabalhos futuros.

CAPÍTULO 2

FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem o objetivo de apresentar uma revisão teórica dos principais tópicos utilizados neste trabalho, a primeira parte faz uma apresentação do protocolo IEEE 802.15.4, na sequência apresentando o protocolo MODBUS e por fim sendo apresentado uma revisão sobre sistemas SCADA com ênfase no *software* ScadaBR. Ao término deste capítulo espera-se que o leitor compreenda os aspectos necessários para compreensão do trabalho proposto.

2.1 PROTOCOLO IEEE 802.15.4

A criação de RSSF (Redes de Sensores Sem Fio) e a sua rápida expansão e utilização em um gama de aplicações, fizeram com que fosse necessário o desenvolvimento de um protocolo padrão, responsável por definir regras para compatibilização de dispositivos conectados a essas redes e é nesse contexto que o protocolo IEEE 802.15.4 foi desenvolvido.

O padrão IEEE 802.15.4 define a camada física e a subcamada MAC sendo complementada pela subcamada de LLC (*Logical Link Control*). Esta divisão da camada de enlace faz parte da especificação do padrão IEEE 802.15.4.

Redes *LowPan* (*Low Power Wireless Personal Area Networks*) apresentam dois tipos de dispositivos. Os FFD (*Full Function Devices*) e os RFD (*Reduce Function Devices*). Dispositivos FFD como o nome indica possuem todas as funções das pilhas de protocolo implementadas, portanto possuem maior complexidade e em geral maior custo e consumo de energia. Um dispositivo FFD pode se comunicar com outros dispositivos FFD e com dispositivos RFD, enquanto os dispositivos RFD podem comunicar-se apenas com dispositivos FFD, porém apresentam complexidade reduzida, menor consumo de energia e menor custo de implementação.

A partir destes dispositivos são definidos tipos de dispositivos lógicos presentes nas redes. Redes IEEE 802.15.4 apresentam 3 tipos de dispositivos lógicos, coordenador, roteador e dispositivo final.

Coordenador da rede – este é um dispositivo que cria e gerencia a rede mantendo parâmetros importantes como tabela de roteamento. Em uma rede deve existir apenas um coordenador de rede simultaneamente, estes dispositivos ainda podem se comunicar com redes externas, funcionando como um *gateway*, passando dados entre redes distintas.

Roteador – é o dispositivo que geralmente encaminha mensagens entre o coordenador e os dispositivos finais. É um dispositivo FFD.

Dispositivo final – são os dispositivos mais básicos conectados na rede, em geral são utilizados para monitorar variáveis do ambiente, são dispositivos RFD e são conectados em maior número em uma rede.

A existência de dispositivos roteadores em redes de baixo consumo é fundamental para que seja possível obter uma maior área de abrangência, formando assim uma rede com diversas topologias.

2.1.1 Camada Física

A camada física do protocolo IEEE 802.15.4 prove recursos como indicador de intensidade do sinal recebido (RSSI - *Received Signal Strength Indicator*), indicador de qualidade do enlace (LQI - *Link Quality Indicator*), verificação de atividade no canal (CCA - *Clear Channel Assessment*), além de especificar uma variedade de funcionalidades que beneficiam o baixo consumo e baixa sobrecarga dos pacotes de dados.

O padrão define duas especificações físicas. Estas especificações são apresentadas na Quadro 1.

Parâmetros	PHY 2,4 GHz	PHY 868	915 MHz
Banda (MHz)	2400 - 2483,5	868 - 868,3	902 - 928
Número de canais	16	1	10
Espaçamento dos Canais (MHz)	5	Não se aplica	2
Taxa de transmissão (kbps)	250	20	40
Modulação	DSSS utilizando O-QPSK	DSSS utilizando BPSK	DSSS utilizando BPSK

Quadro 1 - Resumo das principais características físicas do padrão IEEE 802.15.4.

Fonte: Adaptado de IEEE, 2007.

A estrutura dos pacotes ou *frames* enviados pela camada física no padrão IEEE 802.15.4 dividem-se basicamente em dois campos, o cabeçalho ou *header* e o

payload ou carga do pacote. O cabeçalho é subdividido em 3 partes, preâmbulo com tamanho de 4 *bytes*, utilizado para sincronização dos símbolos, na sequência o início do delimitador do pacote com tamanho de 1 *byte*, utilizado para sincronização do quadro e por fim o tamanho do quadro também com tamanho de 1 *byte*, especifica a quantidade de dados transmitidos pelo campo do *payload*. O segundo campo, *payload*, possui a carga útil de dados transmitidos pode conter até 127 *bytes* de dados. A estrutura descrita acima está representada na Figura 02.

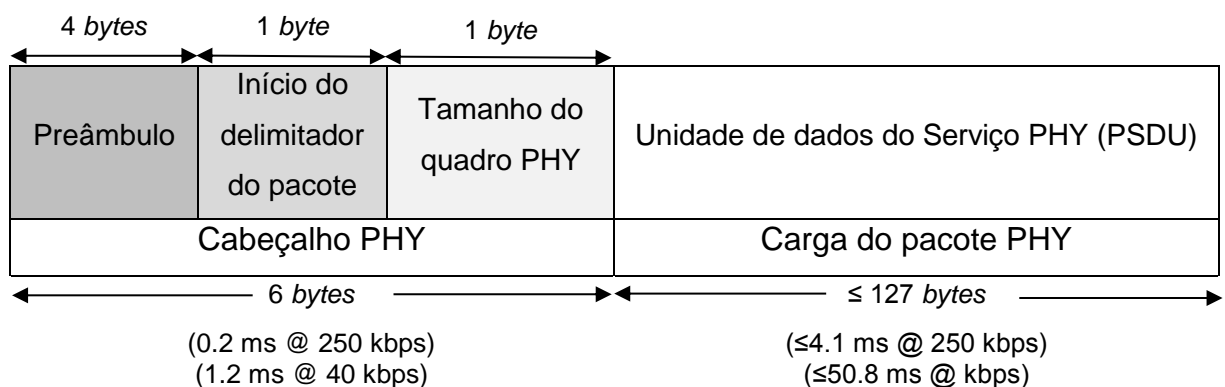


Figura 2 - Frame IEEE 802.15.4.
Fonte: Adaptado de IEEE 802.15.4

2.1.2 Camada de Enlace

a. Sub-camada MAC

A subcamada MAC como o nome mesmo sugere é responsável pelo controle de acesso ao meio de transmissão compartilhado, o principal objetivo desta camada é evitar as colisões de acesso de múltiplos dispositivos ao meio, nesta camada ainda são adicionados o endereço e informações para detecção de erro, montando um pacote e entregando para camada física proceder com a comunicação (TANENBAUM, 2003).

O padrão IEEE 802.15.4 prevê dois modos de operação da camada MAC, o modo *beacon-enabled* e o modo *beaconless*.

- *Beacon-enabled* – Este modo utiliza uma estrutura especial de dados chamada de *superframe*, utilizado para sincronização e transmissão de dados. Ainda neste modo pacotes de dados sinalizadores chamados de *beacons* são transmitidos periodicamente, em intervalos pré-determinados que variam de

15ms a 245s. Os *beacons* são transmitidos pelo coordenador a fim de sincronizar os dispositivos da rede, determinando quando estes devem transmitir ou receber mensagens. Este modo de operação aumenta o tráfego de dados na rede e conseqüentemente a latência, devido a necessidade de envios regulares dos pacotes de controle e devido a inclusão de dados na carga útil de cada pacote, reduzindo assim a quantidade de dados que podem ser enviados pela aplicação.

- *Beaconless* – Este modo não faz o uso de *beacons* e utiliza o protocolo CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*) para controle de acesso ao meio, neste modo os dispositivos ficam no modo de recepção à espera de transmissões e a comunicação ocorre de forma assíncrona.

b. Sub-camada LLC

A subcamada LLC ou controle de enlace lógico é responsável por isolar a camada de rede da camada física, tornado assim possível utilizar vários meios físicos sem que a camada de rede tenha que lidar com esses aspectos físicos.

2.2 PROTOCOLO MODBUS

A década de 70 foi marcada por avanços nas áreas de controle e automação industrial, com um aumento significativo na utilização de controladores lógicos que substituíam toda a lógica de utilização de relés. Com isso houve também a necessidade da criação de protocolos industriais para que fosse possível a comunicação e sincronização dos diversos dispositivos. Nesse contexto foi desenvolvido o protocolo MODBUS, um dos primeiros protocolos para comunicação entre CLP desenvolvido pela Modicon®, utilizado para controle e aquisição de sinais. Em 2004 houve a aquisição da Modcon® pela Schenider Electric® que transformou o protocolo em domínio público. Isso fez com que este protocolo seja utilizado por diversas empresas em milhões de equipamentos de uso industrial no mundo todo.

Originalmente o protocolo MODBUS foi implementado para ser utilizado com a camada física baseada no padrão RS232, RS422 e RS485. Mas atualmente o

protocolo pode ser encontrado sobre diversos meios físicos, como mostra a Figura 3 redes/barramentos:

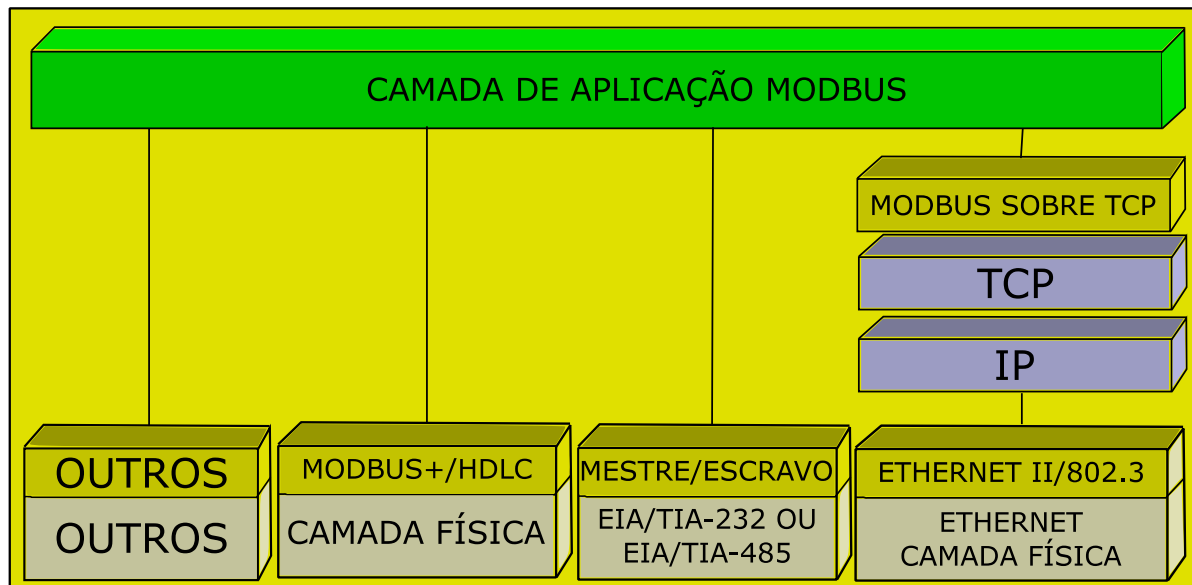


Figura 3 - Pilha de comunicação MODBUS.
 Fonte: Adaptado de MODBUS, 2012.

A comunicação do protocolo é sempre iniciada pelo mestre que faz requisições enviando mensagens aos escravos, que por sua vez retornam com a resposta solicitada. As mensagens que trafegam seguem uma estrutura lógica definida pelo protocolo, e será detalhada na sequência. O protocolo possui diversos comandos, para solicitar apenas o dado desejado.

O mestre pode solicitar dados para os escravos de duas maneiras, por meio de mensagens *unicast* e *broadcast*. Mensagens *unicast* são destinadas a apenas um escravo específico, que é identificado por seu ID único na rede e pode variar de 1 a 247. Mensagens *broadcast* são enviadas a todos os escravos da rede e são facilmente reconhecidas por serem enviadas para o endereço 0. O protocolo ainda define duas formas de envio serial. O modo RTU (*Remote Terminal Unit*) e o modo ASCII (*American Standard Code for Information Interchange*). O modo define como o dados é encapsulado e decodificado dentro dos pacotes transmitidos. Tanto o mestre como o escravo devem se comunicar no mesmo modo para que haja sucesso na comunicação.

Modo RTU – neste modo os dados transmitidos em conjuntos de 8 *bits*, contendo 2 caracteres no formato hexadecimal, o padrão prevê que todos os dispositivos devem suportar este formato.

Modo ASCII – neste modo os dados são transmitidos no formato ANSI ASCII de sete *bits*, a vantagem é que este formato gera mensagens que podem ser legíveis por pessoas. É um modo de transmissão opcional.

Toda comunicação feita no modo RTU é transmitida em pacotes, que seguem um *frame*, como o mostrado no Quadro 2. Que contém o endereço, o código da função, os dados e a verificação de consistência dos dados (CRC - *Cyclic redundancy check*).

Endereço do Escravo	Código da Função	Dados	CRC
1 Byte	1 Byte	0 a 255 Byte(s)	2 Bytes

Quadro 2 - Frame MODBUS.

Fonte: Adaptado de MODBUS, 2012.

O campo endereço é composto por 1 *byte* e pode conter endereços que variam de 0 até 255, entretanto alguns valores são reservados e não podem ser utilizados como endereço de escravos. O valor 0 é utilizado para representar mensagens *broadcast* e os valores 248 a 255 são reservados pelo protocolo.

O campo código da função também é composto por 1 *byte* e descreve a ação ou o dados que o mestre está solicitando. O quadro 3, apresenta as principais funções descritas pelo protocolo.

				Código da Função (decimal)	Subcódigo	Código da Função (hexadecimal)
Acesso a Dados	Acesso a Bit		Ler entradas discretas	02		0x02
			Ler Bobinas	01		0x01
			Escrever única bobina	05		0x05
			Escrever múltiplas bobinas	15		0x0F
	Acesso a 16 Bits		Ler registrador de entrada	04		0x04
			Ler registrador de retenção	03		0x03
			Escrever único registrador	06		0x06
			Escrever múltiplos registradores	16		0x10
			Ler/Escrever múltiplos registradores	23		0x17
			Mascarar escrita de registrador	22		0x16
			Ler fila FIFO	24		0x18
	Acesso a arquivos de registros		Leitura de arquivo de registro	20		0x14
			Escrita de arquivo registro	21		0x15
Diagnósticos		Ler status da exceção	07		0x07	
		Diagnostico	08	00-18,20	0x08	
		Ler contador de evento	11		0x0B	
		Ler log de evento	12		0x0C	
		Reportar id do servidor	17		0x11	
		Ler identificação do dispositivo	43	13,14	0x2B	
Outros		Interface de transporte encapsulada	43	13	0x2B	
		CAN referência geral	43	13	0x2B	

Quadro 3 - Funções definidas pelo protocolo MODBUS.
Fonte: Adaptado de MODBUS , 2012.

Na sequência, o Quadro 4 apresenta um exemplo de utilização do protocolo, para efetuar a leitura de entradas discretas. A leitura de 8 portas discretas é feita a partir do endereço 0, no dispositivo que possui o ID 10. Percebe-se que o *frame* tem início com o valor do endereço. Na sequência vem o número do comando de leitura para entradas discreta que é 2. A seguir são utilizados 2 *bytes* para definir o endereço da primeira porta a ser lida. Neste caso foi utilizado o endereço 0, portanto, tanto a parte alta, como a parte baixa são zero, na sequência o numero de portas lidas que também é composta por 2 *bytes* e por fim o CRC também composto por 2 *bytes*.

Endereço do escravo	Função	Endereço Inicial da porta (parte alta)	Endereço Inicial da porta (parte baixa)	Quantidade (parte alta)	Quantidade (parte baixa)	CRC (parte baixa)	CRC (parte alta)
0x0A	0x02	0x00	0x00	0x00	0x08	0x78	0xB7

Quadro 4 - Exemplo de função sendo enviada pelo protocolo MODBUS.

Fonte: Autoria própria, 2015.

A resposta fornecida pelo dispositivo seria um *frame* como o mostrado no Quadro 5, com início semelhante ao apresentado no quadro 4, onde o primeiro *byte* indica o endereço do escravo, na sequência a função, depois o número de *bytes* que compõe o quadro de dados e na sequência os valores obtidos no dispositivo e por fim o CRC.

Endereço do escravo	Função	Número de bytes que compõe o dado	Valor lido	CRC (parte baixa)	CRC (parte alta)
0x0A	0x02	0x01	0x00	0xA3	0xAC

Quadro 5 - Exemplo de função sendo recebida pelo protocolo MODBUS.

Fonte: Autoria própria, 2015.

Em caso de erros na resposta, como solicitação de um endereço não existente ou um número maior que o existente de portas, o escravo responderia com um código de erro, gerando assim uma exceção, que pode ser interpretada e tratada de acordo com a necessidade.

2.3 SISTEMAS SCADA

Os primeiros sistemas SCADA originaram-se na década de 1970, e derivaram dos sistemas de telemetria existentes na época. SCADA vem do inglês *Supervisory Control and Data Acquisition*, significa Controle Supervisório e Aquisição de Dados, os primeiros sistemas SCADA permitiam apenas visualizar variáveis de processos de forma centralizada em painéis compostos por lâmpadas e indicadores sonoros, e em geral contava com pelo menos um operador que efetuava o controle por meio de botões e alavancas [Moreira, 2011].

A partir da década de 80 com o avanço dos computadores foi possível a implantação gradativa de controles automatizados. Atualmente sistemas SCADA são plataformas avançadas com sistema supervisório e de controle completo, gerenciando e atuando em milhares de sensores e atuadores, contam ainda com conexão com

diversos *softwares* gerenciais, compõe gráficos, relatórios, entre outras funcionalidades todas centralizadas em apenas um computador e disponível para acesso e controle remoto. Atualmente sistemas SCADA mais avançados utilizam um grupo de servidores, para permitir maior segurança e confiabilidade.

Atualmente existe no mercado diversos *softwares* SCADA, dentre eles é possível destacar diversos *softwares* proprietários como LabView, Eclipse, Proficy-IFIX entre outros, e ainda diversos *softwares open-source* como é o caso do Mango, Argos Scada, Likindoy e o ScadaBR, que será utilizado neste trabalho.

O sistema ScadaBR, é um sistema de supervisão e aquisição de dados adaptado e traduzido para uso no Brasil, desenvolvido por uma parceria entre diversas empresas de Florianópolis/SC com patrocínio do FINEP (Financiadora de Estudos e Projetos) e apoio da UFSC (Universidade Federal de Santa Catarina), possui licença geral pública GPL (*General Public License*).

O projeto ScadaBR foi adaptado do projeto Mango desenvolvido no Canadá, e conta com diversas funcionalidades, dentre as quais destacamos a possibilidade de estender o projeto por meio de API em diversas linguagens de programação. Ainda conta com suporte a diversos protocolos como *MODBUS*, TCP/IP, DNP3 entre outros.

Possui uma interface amigável e de fácil utilização que é acessada por meio de um navegador web. A Figura 4 mostra a tela inicial do ScadaBR.

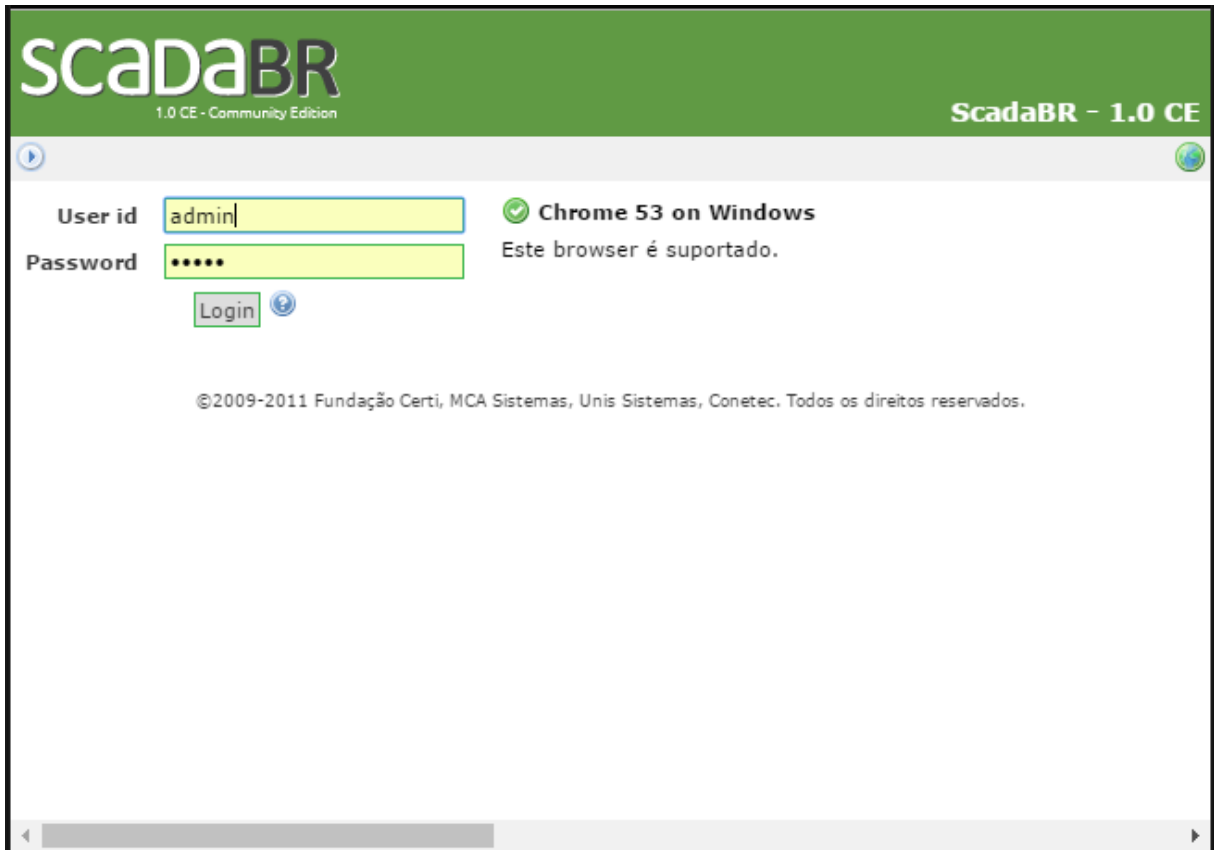


Figura 4 - Tela inicial ScadaBR.
Fonte: Autoria própria, 2016.

2.3.1 *Data Source*

Como o nome sugere, são fontes de dados, que em geral são equipamentos físicos conectados ao sistema, são parte fundamental para o funcionamento do sistema ScadaBR, por meio deste elemento que a abstração do protocolo é feita. Por padrão o sistema ScadaBR suporta diversos protocolos, dentre eles o MODBUS, TCP-IP, ASCII, DNP3 entre outros. A Figura 5 mostra a janela de seleção do protocolo para inclusão de um novo *data source*.

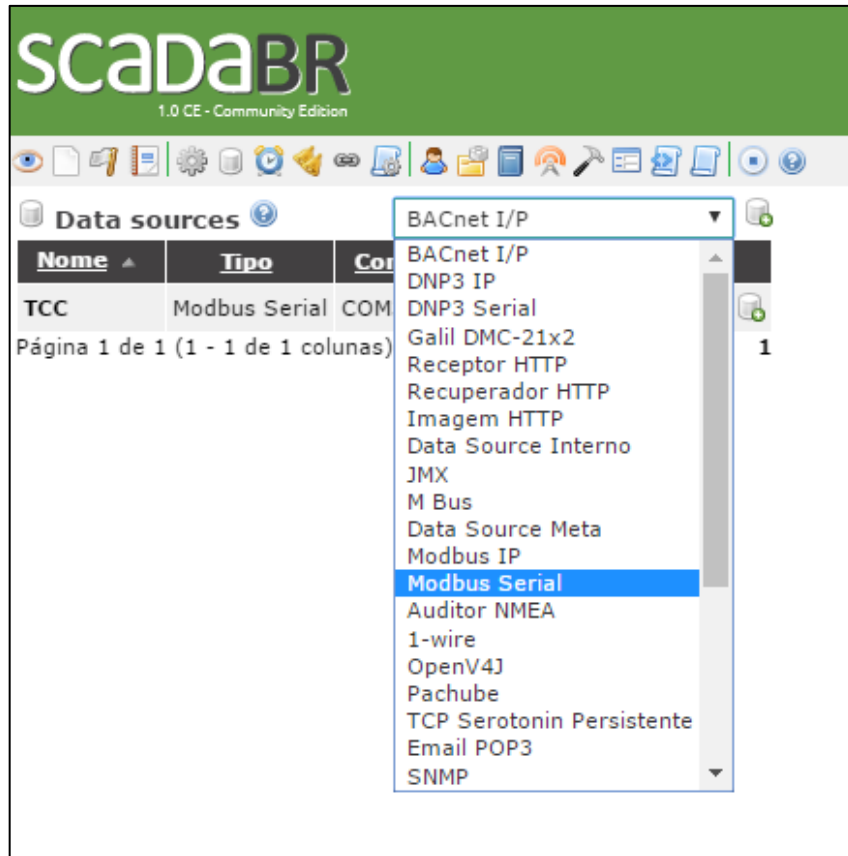



Figura 5 - Tela inserir *data source* ScadaBR.
Fonte: Autoria própria, 2016.

2.3.2 Data Point

Data point's como o nome sugere são pontos de dados, são variáveis digitais numéricas ou alfanuméricas que representam um valor físico de um dispositivo conectado ao sistema. Por meio dos *data point's* que o programa faz a abstração e controle dos processos. Cada *data point* tem um conjunto de informações que deve ser configurada. A Figura 6 mostra a janela de configuração.



Detalhes do data point	
Nome	<input type="text"/>
Export ID (XID)	<input type="text" value="DP_301070"/>
Id do escravo	<input type="text" value="1"/>
Faixa do registro	<input type="text" value="Status do coil"/>
Tipo de dados modbus	<input type="text" value="Status do coil"/>
Offset (baseado em 0)	<input type="text"/>
Bit	<input type="text" value="0"/>
Número de registradores	<input type="text" value="0"/>
Codificação de caracteres	<input type="text" value="ASCII"/>
Configurável	<input checked="" type="checkbox"/>
Multiplicador	<input type="text" value="1"/>
Aditivo	<input type="text" value="0"/>

Figura 6 - Tela inserir data point ScadaBR.
Fonte: Autoria própria, 2016.

Este capítulo apresentou uma breve revisão teórica dos assuntos que o autor considerou necessário para melhor entendimento deste trabalho. Na primeira parte o protocolo IEEE 802.15.4 foi apresentado com suas peculiaridades e principais divisões em camada física e MAC, na sequência o protocolo MODBUS foi apresentado, sendo mostrado a formato de um frame MODBUS, além das funções e ainda um exemplo de utilização. Por fim foi feita uma revisão sobre sistemas SCADA com ênfase no *software* ScadaBR.

CAPÍTULO 3

MATERIAIS E MÉTODOS

Este capítulo apresenta a metodologia adotada para desenvolvimento do projeto proposto, inicialmente é feita uma apresentação geral do sistema, bem como os principais componentes utilizados seguido da justificativa para sua utilização. Na sequência são apresentados aspectos relativos ao desenvolvimento da rede sem fio e do protocolo MODBUS, bem como a configuração do *software* ScadaBR e a integração das partes.

3.1 DESCRIÇÃO DO SISTEMA

Para melhor apresentação do trabalho desenvolvido, o sistema proposto está simplificado em um diagrama de blocos, apresentado na Figura 7, o primeiro bloco representa o computador com o sistema ScadaBR, no centro conectado ao computador, o dispositivo que opera como uma ponte ou *gateway*, e faz a conversão física dos dados vindos do computador por meio de um conversor RS232/USB, já embarcado no kit de desenvolvimento escolhido. Na sequência os dados são interpretados pelo *gateway*, o dispositivo analisa para qual endereço o pacote é destinado e encaminha os dados por meio da rede sem fio. O terceiro bloco por sua vez se comporta como um escravo, ao receber o pacote pela rede sem fio interpreta, executa a ação necessária e responde.

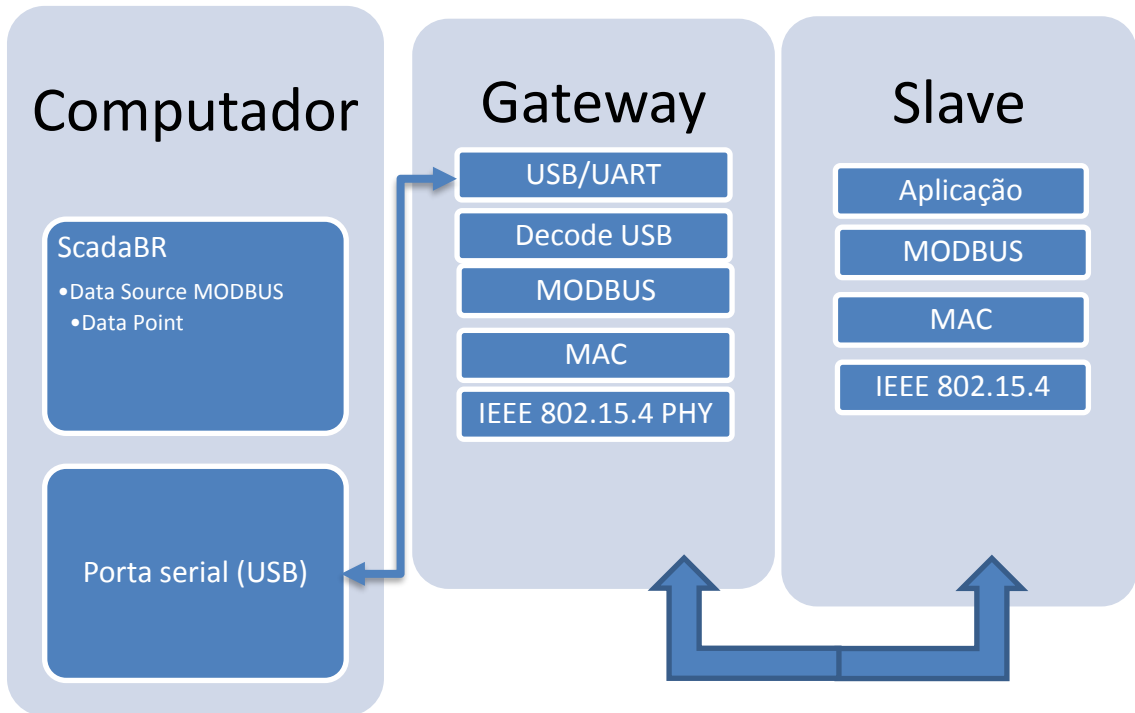


Figura 7 - Diagrama de blocos do sistema proposto.
Fonte: Autoria própria, 2016.

A Figura 8 apresenta um diagrama simplificado de como o sistema deverá operar. Pode-se perceber que o *gateway* MODBUS é o responsável pela conexão entre o computador e a rede sem fio para conexão com os escravos MODBUS.

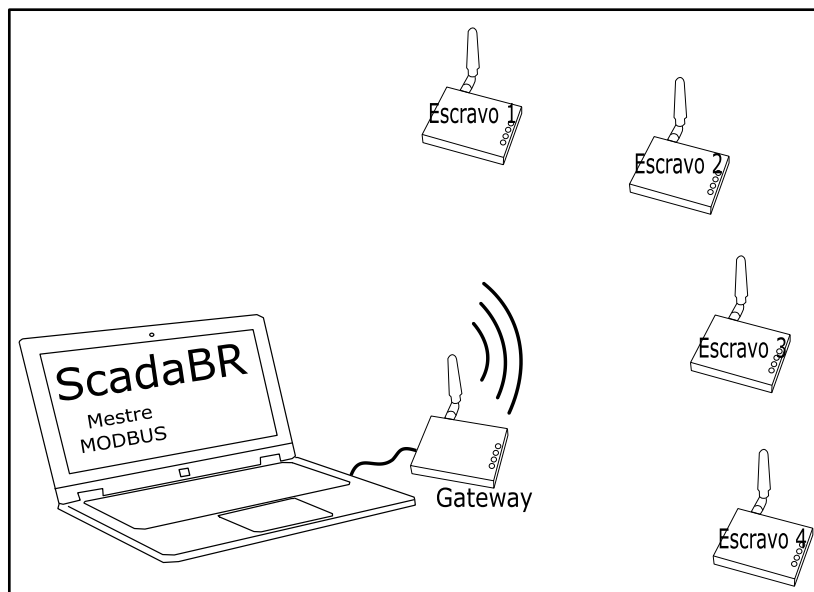


Figura 8 - Modelo do sistema proposto.
Fonte: Autoria própria, 2015.

A proposta desse trabalho é modificar o meio físico tradicionalmente utilizado pelo protocolo MODBUS para ondas de rádio, por meio do protocolo IEEE 802.15.4. O Quadro 6 apresenta o modelo OSI para uma aplicação MODBUS tradicional, e para a aplicação proposta.

CAMADA	MODELO OSI	MODELO MODBUS SERIAL	MODELO PROPOSTO
7	APLICAÇÃO	APLICAÇÃO MODBUS	APLICAÇÃO MODBUS
6	APRESENTAÇÃO		
5	SESSÃO		
4	TRANSPORTE		
3	REDE		
2	ENLACE	MODBUS SERIAL	IEEE 802.15.4 MAC
1	FÍSICA	RS 485	IEEE 802.15.4 2.4 GHz

Quadro 6 - - Modelo OSI aplicado ao projeto proposto

Fonte: Autoria própria, 2015.

A substituição da camada de enlace serial pela subcamada MAC representa um desafio do ponto de vista do determinismo, pois o fato do dispositivo *gateway* pertencer simultaneamente a duas redes faz com que o cumprimento de requisitos temporais seja um desafio. O *gateway* neste contexto deve assumir três papéis distintos e complementares, do ponto de vista da rede MODBUS o dispositivo pode ser visto como um mestre, pois tem o papel de redistribuir as solicitações feitas pelo sistema ScadaBR aos escravos. Do ponto de vista da rede IEEE 802.15.4, o dispositivo tem o papel de coordenador da rede, devendo gerir todo o tráfego de dados da rede sem fio, tabela de dispositivos e controle de acesso ao meio. E por fim do ponto de vista do sistema ScadaBR o dispositivo pode ser visto como um escravo MODBUS, pois possui todas as funcionalidades disponíveis nos demais dispositivos presentes na rede.

O fato de acumular diversas tarefas simultaneamente, e principalmente a necessidade de troca de informações entre duas redes distintas, faz com que aumente consideravelmente o tempo entre a troca de informações, desde a solicitação a partir do sistema ScadaBR até a resposta do cliente.

3.2 MATERIAIS

Após a apresentação do sistema proposto, é possível fazer um detalhamento de cada componente utilizado para compor os blocos. O primeiro passo para o desenvolvimento deste trabalho foi definir os elementos de *hardware*. Os dispositivos

devem apresentar capacidade física para interconexão e capacidade de processamento suficiente para suportar as pilhas dos protocolos e a aplicação.

3.2.1 COMUNICAÇÃO SEM FIO

Um dos principais componentes de *hardware* deste projeto é o módulo transceptor, pois este dispositivo é responsável por efetuar a comunicação sem fio. Os módulos transceptores utilizados são do modelo MRF24J40MA, mostrado na Figura 9, fabricados pela Microchip Technology Inc[®], este módulo é compatível com o padrão IEEE 802.15.4 opera na frequência de 2,4 GHz (com taxa de transmissão de até 250 kbps), que possui cristal e antena integrado, opera nas faixas de tensão de 2,4 a 3,6 volts, possui interface SPI (Serial Peripheral Interface) com frequência de operação de até 10MHz e opera através da modulação DSSS utilizando O-QPSK.



Figura 9 - Modulo de rádio MRF24J48MA.
Fonte: EKT, 2016.

3.2.2 MICROCONTROLADOR

Para o desenvolvimento do *firmware* o microcontrolador escolhido foi o kit de desenvolvimento *Freedom Board* KL25Z da fabricante Freescale[®] mostrado na Figura 10.

Este kit é composto por um microcontrolador Freescale FRDMKL25Z128VLK, que é um microcontrolador de 32 bits, possui CPU ARM Cortex M0+ com processamento de até 48MHz, 128 KB de memória *flash* e 16KB de memória RAM, conta ainda com três portas UART, 2 portas SPI, ADC de 16 bits, timers/PWM, DAC e 66 pinos de propósito geral. O kit ainda possui um sensor capacitivo, acelerômetro tri-axial de 12 *bits*, LED RGB e uma interface de depuração. O kit ainda conta com 53 pinos do microcontrolador mapeados nos barramentos.

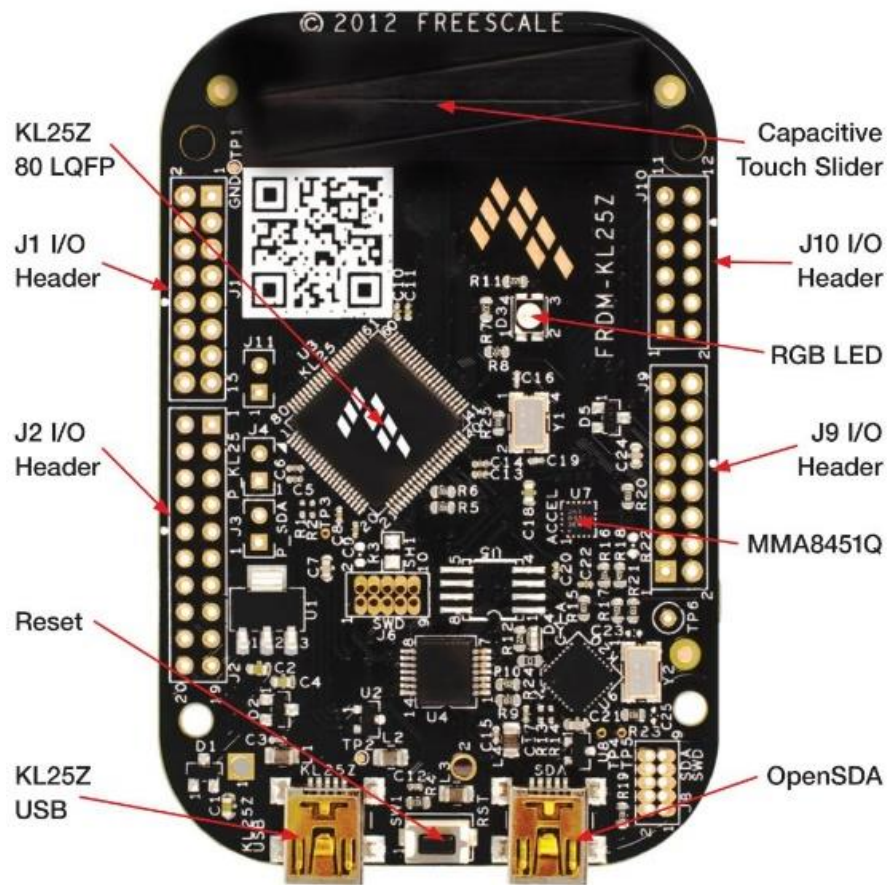


Figura 10 - Kit Freescale Freedom FRDMKL25Z.
Fonte: HAOYU, 2016.

3.2.3 SISTEMA OPERACIONAL

Devido à complexidade da implementação e a necessidade da execução de diversas tarefas concorrentes, fez-se necessária a utilização de um RTOS (*Real Time Operational System*), além disso um sistema operacional oferece diversas ferramentas que facilitam a implementação de sistemas embarcados.

Para a implementação do projeto optou-se pela escolha do BRTOS (*Brazilian Real Time Operating System*), este sistema possui licença MIT (*Massachusetts Institute of Technology*), desenvolvido no Brasil por Gustavo Weber Denardin e Carlos Henrique Barriquello. O BRTOS foi desenvolvido com o objetivo de ser um sistema operacional simples, possui baixo consumo de memória e carga computacional, além de contar com núcleo preemptivo. Ainda, possui diversas funcionalidades como

gerenciamento de memória, de tempo, sincronização de tarefas, semáforos, filas, mutexes (DENARDIN, 2016).

3.3 MÉTODOS

3.3.1 SOFTWARE EMBARCADO

O *firmware* desenvolvido para a aplicação é subdividido em tarefas. Cada tarefa tem uma função específica, para melhor compreensão de como foi executada a integração dos componentes do sistema, e como *software* e *hardware* interagem, a seguir será explicado como essas tarefas funcionam.

Gateway

O *firmware* do *gateway* é composto por 5 tarefas 3 destas tarefas são referentes à rede e são comuns aos dispositivos escravos, estas tarefas serão explicadas na sequência em uma seção própria. As duas tarefas restantes, tem a função de adquirir os dados vindos do PC (computador pessoal) e encaminhar os pacotes de dados para os escravos.

A primeira tarefa aguarda o recebimento de dados vindos do computador. Esta tarefa implementa a camada de enlace do protocolo MODBUS, interpreta os dados e da encaminhamento para o destino correto, por meio da rede sem fio.

A segunda tarefa faz a função inversa, aguarda o recebimento de dados pela rede sem fio, interpreta e encaminha para o computador.

Escravos

O *firmware* implementado para os dispositivos escravos possui 4 tarefas instaladas, dentre elas 3 são da rede sem fio e serão apresentadas em seção própria. A tarefa restante é responsável pela implementação da camada de enlace do protocolo MODBUS e pelo encaminhamento da mensagem para a camada de aplicação.

Rede sem fio

Como dito anteriormente ambos os *firmwares* possuem três tarefas que gerenciam a rede sem fio. Uma tarefa de gerenciamento de eventos físicos, outra para

gerenciamento da subcamada MAC e a última tarefa faz o gerenciamento da camada de rede.

A primeira tarefa foi implementada com o objetivo de verificar eventos físicos do rádio, como recepção ou envio de pacotes.

A segunda tarefa foi implementada com o objetivo de decodificar os pacotes recebidos pela rede, verificando os tipos de pacotes recebidos e dando o destino adequado, como por exemplo passando para a camada de rede.

A última tarefa tem a função de receber os dados da camada MAC e repassar para a camada de aplicação.

3.3.2 INTERFACE COM O SCADABR

Para que seja possível fazer a conexão da rede MODBUS ao sistema ScadaBR, inicialmente foi necessária a configuração da comunicação entre o *gateway* e o computador. Essa comunicação é feita por meio de um adaptador USB/RS232 integrado ao kit do microcontrolador, emulando uma porta serial, portanto para o *software* toda rede é vista como sendo uma rede serial tradicional. E para efetuar a configuração do *gateway* criou-se um *data source* com o protocolo MODBUS, a Figura 11 mostra os *data sources* ativos no sistema.

Nome	Tipo	Conexão	Status
TCC	Modbus Serial	COM3	On

Figura 11 - Data source configurado no ScadaBR.
Fonte: Autoria própria, 2016.

Após a criação do *data source* deve-se fazer configuração dos parâmetros da comunicação serial, como *baud rate*, paridade, codificação, taxa de atualização entre outros. A Figura 12 mostra a janela de ajuste dos parâmetros.

Propriedades do modbus serial

Nome: TCC

Export ID (XID): DS_558906

Período de atualização: 5 segundo(s)

Quantificação:

Timeout (ms): 500

Tentativas: 2

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: COM3

Baud rate: 57600

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

Figura 12 - Opções de configuração do *data source*.
Fonte: Autoria própria, 2016.

Após a comunicação serial estar configurada, deve-se configurar os *data points*, que são as variáveis de interesse presentes nos escravos da rede. A partir dos *data points* que é possível fazer a interação entre o *software* com os dispositivos físicos. Essas variáveis podem ser lidas ou escritas de acordo com sua função. Para efeitos de teste do sistema cada escravo foi configurado para possuir 8 entradas discretas, 8 saídas discretas e 3 entradas analógicas. A Figura 13 mostra a configuração de 3 LED's nos escravos 1 e 48, ainda, 3 entradas analógicas no dispositivo 1 e mais 1 entrada analógica no dispositivo 48.

Data points 						
Nome	Tipo de dado	Status	Escravo	Faixa	Offset (baseado em 0)	
S1_AD1	Numérico		1	Registrador holding	0	
S1_AD2	Numérico		1	Registrador holding	1	
S1_AD3	Numérico		1	Registrador holding	2	
S1_LED1	Binário		1	Status do coil	5	
S1_LED2	Binário		1	Status do coil	6	
S1_LED3	Binário		1	Status do coil	7	
S2_AD1	Numérico		48	Registrador holding	0	
S2_LED1	Binário		48	Status do coil	5	
S2_LED2	Binário		48	Status do coil	6	
S2_LED3	Binário		48	Status do coil	7	

Figura 13 - Lista de *data points* configurados no sistema ScadaBR.
Fonte: Autoria própria, 2016.

Após concluir a configuração dos *data points*, o sistema cria uma lista chamada *watch list*, mostrada na Figura 14, onde é possível acompanhar o *status* atual das variáveis. Todas as variáveis inseridas nesta lista estão disponíveis para utilização na página da interface gráfica.












































































Watch list 					(sem nome) 			
	TCC - S1_LED2	1	10:54:09					
	Valor do ponto pode não ser confiável 							
	TCC - S1_LED1	1	10:53:58					
	Valor do ponto pode não ser confiável 							
	TCC - S1_LED3	1	10:54:12					
	Valor do ponto pode não ser confiável 							
	TCC - S1_AD1	27456.0	12:01:35					
	Valor do ponto pode não ser confiável 							
	TCC - S1_AD2	27752.0	12:01:35					
	Valor do ponto pode não ser confiável 							
	TCC - S1_AD3	38100.0	12:01:35					
	Valor do ponto pode não ser confiável 							
	TCC - S2_AD1	27024.0	Out 19 18:16					
	Valor do ponto pode não ser confiável 							
	TCC - S2_LED1	1	Out 19 18:14					
	Valor do ponto pode não ser confiável 							
	TCC - S2_LED2	1	Out 19 18:14					
	Valor do ponto pode não ser confiável 							
	TCC - S2_LED3	1	Out 19 18:15					
	Valor do ponto pode não ser confiável 							

Figura 14 - Watch List ScadaBR.
Fonte: Autoria própria, 2016.

Este capítulo apresentou os materiais e a metodologia adotada para a implementação dos protótipos tanto do *gateway* como dos escravos. Na primeira seção uma descrição do sistema foi apresentada, expondo os principais pontos que serão trabalhados dentro do projeto proposto. Logo na sequência foi feita uma apresentação dos materiais utilizados e por fim são apresentados aspectos relativos a configuração do *software* ScadaBR e a integração das partes.

CAPÍTULO 4

Este capítulo apresenta os resultados obtidos após a implementação dos protótipos, inicialmente são apresentados os resultados do hardware implementado, logo após é apresentada a implementação desenvolvida no *software* ScadaBR e por fim são apresentados os resultados obtidos por meio de diversos testes, que tem o objetivo de verificar o desempenho do sistema implementado.

RESULTADOS

4.1 *HARDWARE*

A fim de facilitar a conexão física entre o módulo de rádio e o microcontrolador, foi desenvolvida uma placa de circuito impresso com pinos. O módulo de rádio foi soldado sobre a placa e dessa forma é possível encaixá-lo diretamente nos pinos do microcontrolador, como mostra a Figura 15.

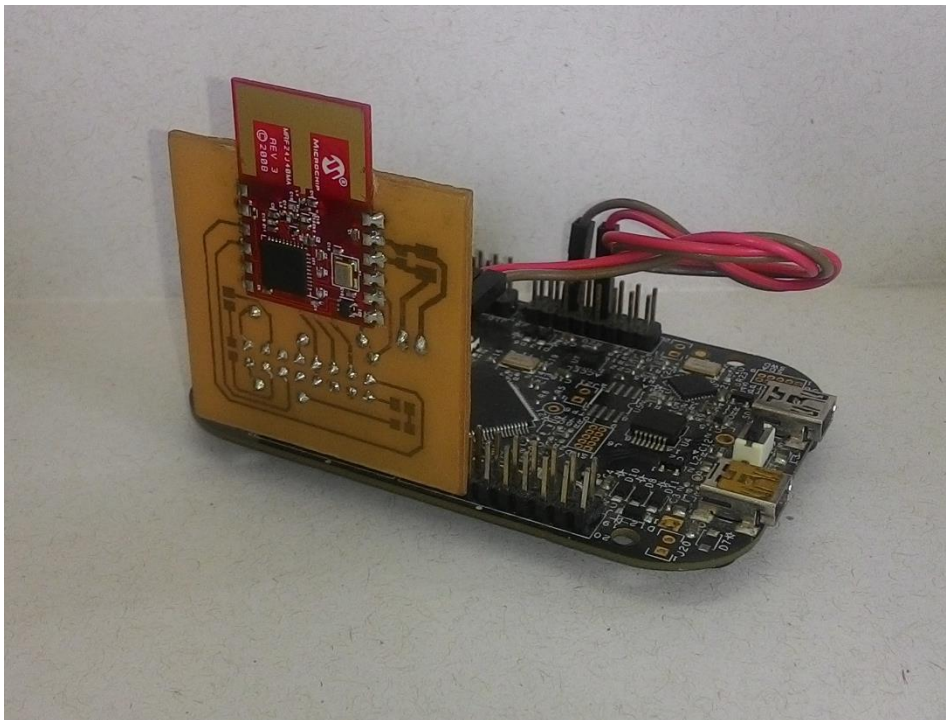


Figura 15 - Dispositivo de *hardware*.
Fonte: Autoria própria, 2016.

Para confecção da placa de circuito impresso foi utilizada uma placa de fenolite. Para fazer as trilhas foi aplicado o método térmico, que consiste na transferência de da impressão das trilhas em papel fotográfico para a placa de fenolite, por meio de aquecimento. Após a placa estar com a impressão, é submersa em solução de perclorato de ferro, as áreas de cobre da placa expostas são corroídas, ficando sobre a placa apenas a parte protegida pela impressão.

4.2 SCADABR

Após a execução da metodologia descrita na seção anterior, para inserir o *data source* e os *data points*, foi possível criar uma representação gráfica dos elementos configurados, dessa forma é possível interagir com os dispositivos de forma visual e interativa. A Figura 16 apresenta a tela criada, configurada com 18 *data points*, 6 de cada escravo. Os círculos vermelhos representam saídas digitais, e logo acima os botões possuem a função de possibilitar a interação ligando ou desligando a saída correspondente. Abaixo dos círculos vermelhos, são apresentados 3 tipos de mostradores analógicos, por meio deles é possível fazer a leitura de entradas dos conversores AD do microcontrolador.

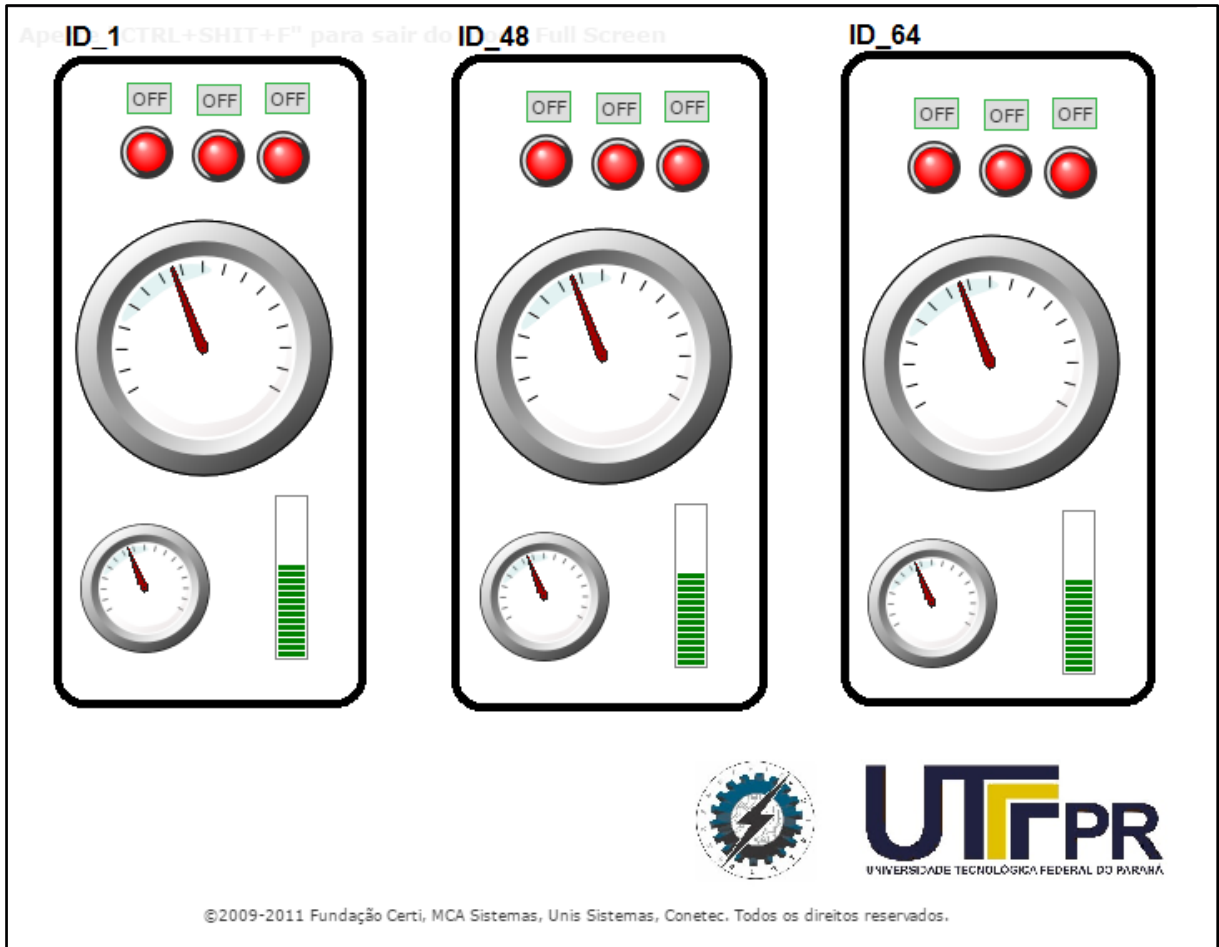


Figura 166 - ScadaBR tela de representação gráfica.
 Fonte: Autoria própria, 2016.

4.3 TESTES

A fim de avaliar o desempenho dos dispositivos desenvolvidos foram realizados testes, práticos. Dois tipos de métodos de testes foram utilizados. Em um primeiro momento avaliou-se a confiabilidade da rede sem fio contra falhas de comunicação, na sequência foi avaliada a integridade da entrega de dados a partir do protocolo MODBUS integrado a rede sem fio. Por fim foi avaliado o sistema operando por completo.

Após a implementação do protocolo MODBUS, o dispositivo *gateway* foi conectado ao computador e por meio do *software* MODBUS *Tester*, dados foram enviados para um dispositivo configurado como escravo. Foram feitos testes de envio das funções 1 (ler bobinas / entradas digitais) e 5 (escrever única bobina/saída digital) do protocolo

MODBUS. Cada função foi enviada 100 vezes para cada dispositivo, em todos os casos foram registrados o número de transmissões com sucesso e falhas.

1º Teste

O primeiro teste foi feito após a implementação do padrão IEEE 802.15.4. Foram enviados diversos pacotes com tamanhos variados, tanto dos dispositivos *router's* para o coordenador como do coordenador para o *router's*. Em cada teste foram enviados 1000 pacotes de 10, 50 e 90 *bytes*. Ainda a fim de verificar a influência da distância entre os dispositivos, os testes foram repetidos para as distâncias de menos de 1 metro, 10 metros, 50 metros e 100 metros. Cada um dos testes foi executado 10 vezes. Em todos os casos foram registrados o número de transmissões com sucesso e falhas para envio dos 1000 pacotes. Por fim os dados foram analisados e a média dos resultados obtidos é apresentada no quadro 07.

Tamanho 10 <i>bytes</i> de <i>payload</i>			
Distância	Pacotes enviados	Média de dados recebidos pelo computador	Taxa de sucesso na entrega (%)
<1m	1000	1000	100
10m	1000	1000	100
50m	1000	987	98,7
100m	1000	508	50,8
Tamanho 50 <i>bytes</i> de <i>payload</i>			
Distância	Pacotes enviados	Média de dados recebidos pelo computador	Taxa de sucesso na entrega (%)
<1m	1000	1000	100
10m	1000	1000	100
50m	1000	957	95,7
100m	1000	458	45,8
Tamanho 90 <i>bytes</i> de <i>payload</i>			
Distância	Pacotes enviados	Média de dados recebidos pelo computador	Taxa de sucesso na entrega (%)
<1m	1000	1000	100
10m	1000	997	99,7
50m	1000	805	80,5

100m	1000	389	38,9
------	------	-----	------

Quadro 7- Dados teste 1.

Fonte: Autoria própria, 2016.

Os dados apresentados no quadro 07 nos mostram uma queda na taxa de sucesso conforme a distância entre os dispositivos aumenta. Esse comportamento é esperado, pois a intensidade do sinal diminui.

2º Teste

Neste segundo teste o protocolo MODBUS foi integrado a rede sem fio e o *gateway* foi conectado ao computador para que fosse possível enviar solicitações ao escravo pela rede sem fio. Em um primeiro momento a função 4 (ler bobinas/entradas digitais), configurada para ler 3 registradores a partir do endereço 0 foi enviada. A resposta recebida foi analisada e verificada a fim de analisar se os dados estavam de acordo com os valores reais definidos no microcontrolador.

Para execução deste teste, foram enviados 1000 pacotes variando a distância entre os dispositivos. Os testes foram repetidos para as distâncias de menos de 1 metro, 10 metros, 50 metros e 100 metros. Cada um dos testes foi executado 10 vezes. Em todos os casos foram registrados o número de transmissões com sucesso e falhas para envio dos 1000 pacotes. Por fim, os dados foram analisados e a média dos resultados obtidos é apresentada no quadro 08.

Na sequência, a fim de analisar a interferência eletromagnética, um motor de ¼ de CV acionado por um inversor de frequência foi colocado entre os dispositivos. Os testes descritos anteriormente foram repetidos. Em todos os casos foram registrados o número de transmissões com sucesso e falhas e o tempo total para envio dos 1000 pacotes.

Teste utilizando MODBUS solicitando leitura de 3 registradores			
Distância	Pacotes enviados	Média de dados recebidos pelo computador	Taxa de sucesso na entrega (%)
<1m	1000	996	99,6
10m	1000	987	98,7
50m	1000	800	80
100m	1000	240	24
Teste utilizando MODBUS solicitando leitura de 3 registradores com motor ligado			

Distância	Pacotes enviados	Média de dados recebidos pelo computador	Taxa de sucesso na entrega (%)
<1m	1000	997	99,7
10m	1000	990	99
50m	1000	635	63,5
100m	1000	150	15

Quadro 8 - Dados coletados teste 2.

Fonte: Autoria própria, 2016.

Como verificado no primeiro teste, os dados apresentados no quadro 08 nos mostram uma queda na taxa de sucesso conforme a distância entre os dispositivos vai aumentando. Esse comportamento é esperado pois a intensidade do sinal vai diminuindo. Além disso é possível perceber que o número de falhas aumenta significativamente ao inserimos uma fonte de ruído eletromagnético, como é o caso do inversor entre os dispositivos.

CAPÍTULO 5

CONCLUSÕES

Antes de concluir o trabalho, cabe recordar os objetivos definidos neste trabalho. O objetivo geral consistia em desenvolver os protótipos de módulos MODBUS *gateway* e escravos, com comunicação sem fio baseada no protocolo IEEE 802.15.4. A comunicação entre o *software* ScadaBR e o módulo *gateway* MODBUS foi por uma interface de comunicação serial RS232.

Enquanto os principais objetivos específicos desse trabalho são: realizar uma revisão bibliográfica e estudos acerca dos protocolos IEEE 802.4.15, MODBUS e do *software* ScadaBR, implementar a camada física de enlace do padrão MODBUS a partir do protocolo IEEE 802.15.4, implementar a camada de aplicação do protocolo MODBUS, interconexão do módulo MODBUS mestre com o *software* ScadaBR, implementar uma aplicação para a rede desenvolvida utilizando o *software* ScadaBR e por fim realizar testes experimentais com o objetivo de comprovar a compatibilidade do sistema desenvolvido com dispositivos MODBUS, bem como verificar sua eficiência.

De forma geral, pode-se afirmar que tanto o objetivo geral como os objetivos específicos foram atingidos. Vale ressaltar neste momento a importância da revisão bibliográfica, em relação aos protocolos utilizados, visto que a implementação deve considerar os detalhes pormenorizados para que seja possível sua implementação sem que ocorram erros. O segundo capítulo apresentou um breve resumo a respeito de todo o embasamento teórico necessário para a execução deste projeto.

Todo o processo de realização do trabalho trouxe grandes conhecimentos agregados, e evidenciou falhas cometidas na realização de trabalhos com maior complexidade do que se está habituado na graduação. Por fim, é possível concluir a relevância deste trabalho para o meio acadêmico e as possibilidades de melhorias nos protótipos desenvolvidos. Dessa forma os principais pontos a serem trabalhados no futuro para melhoria do sistema são:

- Substituição da antena do módulo transceptor por uma de maior potência;

- Optimização do código desenvolvido visando redução de consumo;
- Desenvolvimento de uma placa de circuito impresso, com todos os componentes acoplados sem a necessidade de encaixes, reduzindo as dimensões dos protótipos;
- Desenvolvimento de uma embalagem protetora;
- Testes em ambiente severos e com grande quantidade de ruídos eletromagnéticos.

REFERÊNCIAS

BOYER, Stuart A. **SCADA, Supervisory Control and Data Acquisition**. 3ª Ed USA, ISA – Instrument Society of America, 1993.

DENARDIN, Gustavo W., **O que é BRTOS?**, disponível em <<https://brtosblog.wordpress.com/2010/10/06/o-que-e-o-brtos/>>, Acesso em 05 de ago. de 2016.

EKT Electronics,. **MRF24J40MA**, Disponível em <http://www.ekt2.com/Sitefinity/WebsiteTemplates/WebGreen/App_Themes/WebGreen/Images/Products/35000/412%20ZIGBEE%20MRF24J40MA-I-RM%20-2-.jpg>. Acesso em 05 de ago. de 2016.

GUARESE, Giuliano B. M. et. al. **Exploiting MODBUS Protocol in Wired and Wireless Multilevel Communication Architecture**. Porto Alegre, RS, Brasil. 2012

HAOYU, Electronics, **FRDM-KL25Z Kit**, Disponível em <<http://www.haoyuelectronics.com/ebay/FRDM-KL25Z/3.jpg>>. Acesso em 05 de ago. de 2016.

IEEE STANDARD ASSOCIATION. **IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003)**, IEEE Standard for Information Technology – Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs). 2006.

LI HUI, Zhang HAO, Peng Daogang. **Design and Application of Communication Gateway of EPA and MODBUS on Electric Power System**, International Conference on Future Electrical Power and Energy Systems, Shanghai, China, 2012.

MODBUS. **MODBUS application protocol specification**. Versão 1.1.b3. Massachusetts, USA, Abr. 2012.

MODBUS. **MODBUS PROTOCOL**, Disponível em <<http://www.modbus.org/specs.php>>. Acesso em 01 de maio de 2015.

MOREIRA, Pedro Manuel barbosa. **Sistema de Supervisão e Controle de Estruturas**. Dissertação (Mestrado) 53 f. Faculdade de Engenharia da Universidade do Porto - FEUP. Porto, Portugal, 2011.

SCADABR. **ScadaBr 0.7 Sistema Open-Source para Supervisão e Controle: Manual do Software**. Brasil. Out. de 2010.

VLAD. Cristian G. **Optimized SCADA System for Electrical Substations**. Bucharest, Romania. Maio de 2013.

TANENBAUM, Andrew S.. **Redes de Computadores**. 4^a.ed. Campus, Ed. Campus, São Paulo, 2003.

