

**UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE CIÊNCIAS EXATAS E DA NATUREZA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA**



**SISTEMA TELEMÉTRICO PARA MONITORAMENTO DE TRENS
ATRAVÉS DE REDES DE SENSORES SEM FIO
E PROCESSAMENTO EM SISTEMA EMBARCADO**

Jerry Lee Alves Dos Santos

João Pessoa, Março de 2010



PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA - UFPB

**SISTEMA TELEMÉTRICO PARA MONITORAMENTO DE TRENS
ATRAVÉS DE REDES DE SENSORES SEM FIO E
PROCESSAMENTO EM SISTEMA EMBARCADO**

Por

Jerry Lee Alves dos Santos

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal da Paraíba, como parte dos requisitos para a obtenção do título de Mestre em Informática.

Orientadores:

José Antônio Gomes de Lima

Francisco Antônio Belo

João Pessoa, Março de 2010

S237s Santos, Jerry Lee Alves dos
Sistema Telemétrico para Monitoramento de Trens Através
de Redes de Sensores sem Fio e Processamento em Sistema
Embarcado / Jerry Lee Alves dos Santos. – João Pessoa, 2010.
110f. :il.
Orientadores: José Antônio Gomes de Lima e Francisco
Antônio Belo.
Dissertação (Mestrado) – UFPB - CCEN
1.Informática . 2. Telemetria. 3. Sistemas Embarcados.

UFPB/BC

CDU: 004 (043)

Ata da Sessão Pública de Defesa de Dissertação de
Mestrado de **Jerry Lee Alves dos Santos**,
candidato ao Título de Mestre em Informática na
Área de Sistemas de Computação, realizada em 05
de março de 2010.

Aos cinco dias do mês de março do ano dois mil e dez, às 10 horas, no Auditório do Centro de Ciências Exatas e da Natureza da Universidade Federal da Paraíba, reuniram-se os membros da Banca Examinadora constituída para examinar o candidato ao grau de Mestre em Informática, na área de "*Sistemas de Computação*", na linha de pesquisa "*Processamento de Sinais e Sistemas Gráficos*", o Sr. Jerry Lee Alves dos Santos. A comissão examinadora foi composta pelos professores doutores: José Antônio Gomes de Lima (DI-UFPB), Primeiro Orientador e Presidente da Banca Examinadora, Francisco Antonio Belo (CT-UFPB), Segundo Orientador, Lucídio dos Anjos Formiga Cabral (DI-UFPB), Antônio Carlos Cavalcanti (DI-UFPB) como examinadores internos e Manoel Eusebio de Lima (UFPE), como examinador externo. Dando início aos trabalhos, o José Antônio Gomes de Lima, cumprimentou os presentes, comunicou aos mesmos a finalidade da reunião e passou a palavra ao candidato para que o mesmo fizesse, oralmente, a exposição do trabalho de dissertação intitulado "*Sistema Telemétrico para Monitoramento de Trens através de Redes de Sensores sem fio e Processamento em Sistema Embarcado*". Concluída a exposição, o candidato foi argüido pela Banca Examinadora que emitiu o seguinte parecer: "*aprovado*". Assim sendo, deve a Universidade Federal da Paraíba expedir o respectivo diploma de Mestre em Informática na forma da lei e, para constar, eu, professora Tatiana Aires Tavares, Coordenadora deste Programa, servindo de secretária lavrei a presente ata que vai assinada por mim mesmo e pelos membros da Banca Examinadora. João Pessoa, 05 de março de 2010.

Tatiana Aires Tavares
Tatiana Aires Tavares

Prof. Dr. José Antônio Gomes de Lima
Primeiro Orientador (DI-UFPB)

Prof. Dr. Francisco Antonio Belo
Segundo Orientador (CT-UFPB)

Prof. Dr. Antonio Carlos Cavalcanti
Examinador Interno (DI-UFPB)

Prof. Dr. Lucídio dos Anjos Formiga Cabral
Examinador Interno (DI-UFPB)

Prof. Dr. Manoel Eusebio de Lima
Examinador Externo (UFPE)

Dedicatória

Dedico este trabalho aos meus pais, Marlene Maria dos Santos e José Alves dos Santos, por sempre terem acreditado em mim e pelo grande esforço que fizeram para me dar as condições necessárias para estudar e alcançar meus objetivos.

As minhas irmãs, pelo apoio e carinho que me ajudaram a seguir em frente.

A minha tia Maria de Lourdes, por ter generosamente e pacientemente me acolhido em sua casa, possibilitando a minha vinda para João Pessoa.

Agradecimentos.

Agradeço em primeiro lugar a Deus, por sempre ter me guiado pelos melhores caminhos ao longo da vida.

Aos meus orientadores José Antônio e Francisco Belo, pelas orientações, ensinamentos e conhecimentos compartilhados, que foram fundamentais para o desenvolvimento deste trabalho. Pela amizade que entre nós surgiu durante a graduação e que se concretizou durante o mestrado.

Aos professores Lúcido, Antônio Carlos e Valéria, por todo apoio e incentivo recebidos.

Aos amigos Rômulo Araújo e Bruno Maia, pela contribuição essencial que deram para concretização deste trabalho.

A todos os amigos do LES e LASID, em especial a Eudisley Gomes, Abel Cavalcante e Ruan Delgado, pela amizade, companheirismo e pelos momentos de descontração que me ajudaram a seguir nesta longa jornada.

*“Renasce em ti mesmo.
Multiplica os teus olhos, para verem mais.
Multiplica os teus braços, para semeares tudo.
Destrói os olhos que tiverem visto.
Cria outros, para as visões novas.
Destrói os braços que tiverem semeado,
para se esquecerem de colher.
Sê sempre o mesmo.
Sempre outro.
Mas sempre alto.
Sempre longe.
E dentro de tudo.”*

Cecília Meireles

Resumo

Este trabalho apresenta o desenvolvimento de um sistema telemétrico que integra redes de sensores sem fio e processamento de dados em sistema embarcado para monitoramento de veículos em redes de transportes metroferroviários, possibilitando um acompanhamento em tempo real do movimento dos trens. Este sistema tem como principal objetivo permitir uma constante avaliação de vários fatores que influenciam o desempenho dos trens, como velocidade e pressão nas bolsas de ar, de forma a gerar informações que possibilitem estratégias de operação e manutenção, evitando assim uma degradação parcial ou total dos serviços de transporte prestados à população. Para realizar a telemetria em tempo real, redes de sensores sem fio com tecnologia ZigBee são montadas em toda a via de tráfego. Estas redes realizam a captação e envio dos dados coletados em cada trem até estações base onde um sistema de tratamento embarcado em um FPGA fica encarregado de receber os dados, realizar os cálculos necessários e enviar as informações obtidas, através de uma rede ethernet, a um sistema central de supervisão. O sistema embarcado atua como um servidor de dados disponibilizando informações para um sistema de supervisão instalado no Centro de Controle de Operações (CCO) e em outros setores que necessitem destas informações. A utilização de um dispositivo de processamento dedicado, como uma FPGA, proporciona ao sistema uma eficiência muito maior do que normalmente é encontrada em processadores de uso geral, além de permitir a customização do *hardware*, reduzindo o custo final do sistema. O sistema de supervisão instalado no CCO é responsável por gerenciar o armazenamento e a visualização dos dados recebidos. Ele exibe os dados em tabelas e gráficos em tempo real permitindo aos controladores de tráfego analisar o funcionamento de cada trem e detectar possíveis problemas. O sistema também armazena os dados em um banco de dados para realização de estudos posteriores.

Palavras-chave: Telemetria, Redes de Sensores Sem Fios, Protocolo ZigBee, Sistemas Embarcados, Processador Nios II, FPGA.

Abstract

The present study portrays the development of a telemetric system integrating wireless sensor networks and data processing into an embedded system for monitoring vehicles in metropolitan railway networks, enabling real-time monitoring of train movements. The main objective of this system is to allow constant evaluation of numerous factors which can influence train performance, such as speed and pressure in air bags. By so doing, it may become possible to generate information so as to facilitate intervention strategies, thereby preventing the partial or total degradation of the transport services provided to the population. In order to perform real-time telemetry, wireless sensor networks with ZigBee technology will be mounted alongside the entire route of the train traffic. These networks capture and transmit data collected in each train until they reach base stations where a processing embedded system in an FPGA is responsible for receiving the data, performing the necessary calculations and sending the obtained information to a central monitoring system through an Ethernet network. The embedded system functions as a data server providing information to a monitoring system installed in the Operational Control Center (OCC) and in other sectors requiring this information. The use of a dedicated processing device such as an FPGA gives the system far greater efficiency than that typically found in general purpose processors. Furthermore, it allows the customization of the *hardware*, thus reducing the final cost of the system. The monitoring system installed in the OCC is responsible for managing the storage and display of the data received. It will display the data in tables and graphs in real time, enabling the traffic controllers to analyze the operation of each train and also to detect possible problems. The system also stores the data in a database for further study.

Keywords: Telemetry, Wireless Sensor Networks, ZigBee Protocol, Embedded Systems, Nios II processor, FPGA.

Sumário

CAPÍTULO 1	16
Introdução	16
1.1. Motivação	16
1.2. Objetivo.....	18
1.3. Organização.....	19
CAPÍTULO 2	20
Fundamentação Teórica	20
2.1. Sistemas de Telemetria e de Tempo Real	20
2.2. Redes de Sensores Sem Fio	21
2.2.1. Classificação das RSSF.....	22
2.2.2. Áreas de aplicação.....	23
2.2.3. Topologias de RSSF	24
2.3. Protocolo ZigBee	25
2.3.1. Características do Protocolo ZigBee	26
2.3.2. Arquitetura protocolar.....	28
2.3.3. Comparação com outros protocolos.....	30
2.4. <i>Hardware</i> ZigBee para RSSF.....	31
2.5. Sistemas Embarcados	32
2.5.1. Arquitetura de Sistemas Embarcados	33
2.6. Computação Reconfigurável.	34
2.6.1. FPGA.....	35
2.6.2. Processador Nios II.....	36
2.7. Ferramentas de desenvolvimento.	38
2.7.1. Quartus II	38
2.7.2. SOPC Builder.....	39
2.7.3. IDE Nios II.....	39
2.7.4. LabVIEW.....	41
2.7.5. IDE NetBeans	42
CAPÍTULO 3	44
Sistemas de Monitoramento de Trens	44
3.1. Atual Sistema de Monitoramento do MetroRec.	44
3.2. Estado da Arte	45

3.2.1.	Sistemas de Monitoramento Ferroviários	45
3.2.1.1.	Sistema Para Monitoramento de Trens Baseado em Identificação por Rádio Frequência (RFID).....	46
3.2.1.2.	Sistema Wireless para monitoramento da velocidade de trens baseado em RCM3000	47
3.2.1.3.	Sistema baseado em comunicação sem fio para monitoramento do desempenho de Veículos ferroviários.....	47
3.2.1.4.	COMBAT – Dispositivo de rastreamento de trens através de comunicação sem fio.....	48
3.2.2.	Sistemas para monitoramento e controle com ZigBee	49
3.2.3.	Comparativo entre tecnologias.....	50
CAPÍTULO 4		52
Sistema Telemétrico para Monitoramento de Trens - RailBee		52
4.1.	Sistema de Captação de Sinais	53
4.1.1.	Montagem e configuração das redes ZigBee	54
4.1.2.	Componentes da rede.....	56
4.1.3.	Configuração dos módulos ZigBee	57
4.2.	Sistema de Tratamento de Dados.....	60
4.2.1.	Arquitetura do sistema	61
4.2.1.1.	Camada de <i>Hardware</i>	61
4.2.1.1.1.	Configuração do <i>Hardware</i> utilizado	65
4.2.1.2.	Camada de Sistema de <i>Software</i>	66
4.2.1.3.	Camada de Aplicação	67
4.2.2.	Desenvolvimento do sistema embarcado.....	68
4.2.3.	Análise e projeto do sistema embarcado.....	68
4.2.3.1.	Requisitos do sistema	68
4.2.3.2.	Modelagem UML.....	70
4.2.3.2.1.	Diagrama de Atividades	71
4.2.3.2.2.	Diagrama de Componentes	72
4.2.3.2.3.	Diagrama de Seqüência.....	73
4.2.4.	<i>Software</i> de tratamento de dados	73
4.2.5.	<i>Software</i> servidor de dados.....	79
4.3.	Sistema de Supervisão	81
4.3.1.	Análise e projeto do sistema de supervisão	82
4.3.1.1.	Diagrama de Casos de Uso	82
4.3.1.2.	Diagrama de Classes.....	84

4.3.2.	Desenvolvimento do sistema de supervisão	85
4.3.3.	Interfaces projetadas para o sistema	86
CAPÍTULO 5		91
Testes e resultados obtidos		91
5.1	Instalação da rede para testes.	91
5.2	Resultados.....	95
CAPÍTULO 6		100
Conclusão e trabalhos futuros		100
Referências Bibliográficas		102
Glossário		108

Lista de Figuras

Figura 2.1: Topologia de rede em malha.....	25
Figura 2.2: Topologias de rede ZigBee	27
Figura 2.3: Arquitetura do protocolo ZigBee.....	28
Figura 2.4: Módulo RF ZigBee da Maxstream.....	31
Figura 2.5: arquitetura interna de um FPGA	35
Figura 2.6: Exemplo de Sistema Nios II	37
Figura 2.7: Abstração de <i>Hardware</i> do Sistema de Bibliotecas HAL a partir de uma Aplicação de Usuário	41
Figura 2.8: Diagrama de blocos de um programa no LabVIEW	42
Figura 4.1: Visão geral do sistema proposto	53
Figura 4.2: Mapa das linhas do MetroRec.....	55
Figura 4.3: Localização das redes na linha Recife-Jaboatão	56
Figura 4.4: Distribuição dos nós da rede ZigBee em um trecho da via.....	57
Figura 4.5: <i>Software</i> para configuração dos nós da rede ZigBee.....	58
Figura 4.6: Modelo arquitetural de um sistema embarcado.....	61
Figura 4.7: Placa de desenvolvimento Nios II	62
Figura 4.8: Arquitetura do sistema de tratamento	63
Figura 4.9: Arquitetura projetada no SOPC Builder.....	64
Figura 4.10: Camada de sistemas de <i>software</i>	67
Figura 4.11: Diagrama de Atividades do sistema de tratamento de dados.....	71
Figura 4.12: Diagrama de Componentes de <i>software</i> do sistema de tratamento de dados..	72
Figura 4.13: Diagrama de sequência do sistema de tratamento de dados	73
Figura 4.14: Processamento e transformação de um pacote ZigBee em um Pacote RailBee75	
Figura 4.15: Diagrama de casos de uso do sistema de supervisão.....	83
Figura 4.16: Diagrama de classes do sistema de supervisão.....	84
Figura 4.17: Componentes de interface do LabVIEW.	86
Figura 4.18: Interface do sistema para visualização dos dados de um trem	87
Figura 4.19: Painel de controle de um TUE.	87
Figura 4.20: Interface inicial do sistema RailBee.....	88

Figura 4.21: Interface para visualização dos dados de vários trens simultaneamente.	89
Figura 4.22: Interface para visualização dos dados de um trem específico.....	89
Figura 4.23: Interface do sistema de supervisão para consulta de dados	90
Figura 5.1: Vista aérea do trecho da via onde a rede de testes foi implantada.....	92
Figura 5.2: Estação Base que foi instalada na Estação Werneck.....	93
Figura 5.3: Local da instalação da EM dentro do TUE	93
Figura 5.4: Nó final instalado na cabine do trem.	94
Figura 5.5: Local da instalação do roteador na cabeceira leste da Estação Werneck	95
Figura 5.6: Gráfico da medida de velocidade do TUE 11 indo de WEK a LUZ.	96
Figura 5.7: Gráfico da medida de velocidade do TUE 11 indo de LUZ a WEK.	97
Figura 5.8: Gráfico do cálculo do número de passageiros do TUE 11 indo de WEK a LUZ.	98
Figura 5.9: Gráfico do cálculo do número de passageiros do TUE 11 indo de LUZ a WEK.	98
Figura 5.10: Gráfico do número de passageiros do TUE 11 na estação WEK, com o trem na via 2 durante quase todo o dia.	99
Figura 5.11: Gráfico do número de passageiros do TUE 11 na estação WEK, com o trem na via 1 durante quase todo o dia.	99

Lista de Tabelas

Tabela 2.1: Comparação entre os protocolos para redes sem fio mais populares.	30
Tabela 2.2: Pinos do módulo Xbee da Maxstream.....	32
Tabela 4.1: Características da arquitetura projetada para o sistema embarcado.....	66
Tabela 4.2: Descrição dos campos de um pacote ZigBee.	76
Tabela 4.3: Descrição dos campos do pacote RailBee gerado pelo sistema de tratamento.	77

CAPÍTULO 1

Introdução

Este capítulo apresenta as motivações para o desenvolvimento do sistema apresentado neste trabalho, expõe os objetivos a serem alcançados com a pesquisa e o desenvolvimento do sistema, e finaliza descrevendo a organização e estrutura desta dissertação.

1.1. Motivação

A transmissão de dados remota é um recurso fundamental para empresas que necessitam de informações em tempo real, como fator estratégico para tomadas de decisões. No segmento de transporte metroferroviário, a falta de informações precisas sobre o movimento dos trens dificulta o trabalho de engenheiros e técnicos e impede que importantes decisões relacionadas ao tráfego e à manutenção sejam tomadas de forma rápida e eficiente. Surgiu então a necessidade de desenvolver um sistema que fosse capaz de monitorar de forma contínua e em tempo real os trens em movimento e fornecer um conjunto de informações detalhadas, eliminando as dificuldades causadas pela falta destas informações.

Este trabalho desenvolve tecnologias que integram redes de sensores sem fio e processamento em sistema embarcado para monitoramento de trens em um sistema de metrô. O sistema em foco tem a aplicação da tecnologia ZigBee para a telemetria dos TUE (Trens Unidades Elétricas) do Metrô do Recife (MetroRec) e de computação reconfigurável e pretende otimizar os processos de comunicação entre os operadores nas cabines dos trens e o CCO (Centro de Controle Operacional), aperfeiçoando o monitoramento de vários fatores que medem o desempenho dos TUE, tais como: tensão de alimentação, corrente de armadura dos motores de tração, pressão do encanamentos principal e geral, localização, velocidade real e permitida pelo sistema de sinalização no trecho.

Os sistemas de monitoramento de trens até hoje propostos na literatura, empregam tecnologias de alto custo e/ou muito limitadas, se restringindo a medição de poucas

variáveis, em geral, velocidade e posição. O sistema aqui proposto emprega uma tecnologia de baixo custo, de fácil instalação e que permite o monitoramento de múltiplos fatores.

O advento do protocolo de comunicação ZigBee tornou possível o desenvolvimento de um sistema de telemetria metroviária eficiente, de fácil implementação e de baixo custo. O protocolo ZigBee tem como principais características o fato de ser um protocolo para aplicações de baixo consumo e baixo custo. Estas duas características o tornam ideal para sistemas de monitoramento e sensoriamento como é o caso dos sistemas de automação doméstica, segurança, rastreamento de veículos, etc. É uma tecnologia bastante recente e possui determinadas características que o fazem absolutamente distinto dos restantes, sustentando os motivos que levaram à sua criação.

Aliada a rede de sensores com tecnologia ZigBee é utilizado um dispositivo lógico reconfigurável (um FPGA), com um processador embarcado soft-core Nios II para processamento dos dados captados da rede. Como este sistema será instalado em locais das vias onde o uso de um computador convencional não é viável, fez-se necessário utilizar um dispositivo de processamento compacto que pudesse ser facilmente acoplado a rede e instalado na via. Dentre as opções de dispositivos possíveis, optou-se pelo FPGA pela sua característica de reconfigurabilidade. A capacidade de reconfiguração do FPGA permite desenvolver um *hardware* customizado de acordo com as necessidades do sistema, reduzindo seu tamanho e o custo final.

Um sistema de telemetria metroviária como este, possibilita o monitoramento das principais informações dos TUE de forma constante e em tempo real, trazendo benefícios aos diagnósticos e ações preventivas de operação e manutenção do sistema MetroRec, como também favorece a tomada de decisões preditivas, com base em prognósticos auferidos das medidas em tempo real, de forma a possibilitar estratégias de intervenção que possam evitar a degradação parcial ou total dos serviços de transporte prestados à população. Algumas das vantagens apresentadas por este sistema são:

- Dinamismo, otimização e flexibilidade nos processos de planejamento e controle de tráfego dos TUE;
- Menor tempo de resposta às ações, aos procedimentos de rotina e eventos emergenciais, possibilitando decisões mais rápidas e confiáveis;
- Dados, bem como alarmes serão mostrados em tempo real nos consoles do CCO.
- Comunicação precisa e padronizada entre o Centro de Controle e os TUE, trazendo consistência nas informações a disposição dos Operadores do TUE.
- Para o Controlador do CCO, uma visão unificada do TUE, mesma visão dos controles e instrumentos do Operador do TUE.
- Disponibilização de dados históricos.

1.2. Objetivo

O objetivo deste trabalho é apresentar o desenvolvimento de um sistema para telemetria em tempo real dos TUE do MetroRec que seja eficiente e que possua um custo relativamente baixo. Este sistema é subdividido em três subsistemas compostos por uma rede de sensores sem fio para captação dos dados, um sistema embarcado reconfigurável para tratamento e distribuição dos dados obtidos e um sistema de supervisão para armazenamento e visualização das informações recebidas.

Os objetivos específicos do projeto são:

- **Montagem e configuração de uma rede de sensores sem fio com tecnologia ZigBee.**

Esta rede é responsável pela captação dos dados nas cabines dos TUE e pelo envio dos dados até o sistema de tratamento. Um requisito importante dessa rede é a confiabilidade, ou seja, ela deve ser desenvolvida de forma a minimizar a possibilidade de ocorrer falhas que interrompam seu funcionamento. A rede de sensores também deve oferecer de forma simples, possibilidades de expansão, permitindo que novos nós sejam adicionados facilmente à rede. As principais tarefas desta etapa consistem em determinar e montar uma topologia adequada para aquisição dos dados, configurar um firmware para cada tipo de módulo que compõe a rede e desenvolver uma placa customizada para suporte aos módulos dos nós finais, roteadores e coordenadores.

- **Desenvolvimento de um sistema embarcado para tratamento dos dados captados da rede de sensores e distribuição dos dados processados através de uma rede ethernet.**

Este sistema possui uma alta capacidade de processamento e apresenta flexibilidade para atualização de seu *software* e arquitetura. Um dispositivo de processamento dedicado com computação reconfigurável (FPGA), aliado ao poder de um processador embarcado foram utilizados de forma a atuarem satisfatoriamente no projeto. O sistema embarcado na FPGA realiza a decodificação, processamento e empacotamento dos dados recebidos da rede de sensores sem fios e paralelamente atua como um servidor de dados para o sistema de supervisão ao qual estará conectado através de uma rede ethernet local.

- **Desenvolvimento de um Sistema de Supervisão.**

Este sistema foi instalado em um computador no Centro de Controle Operacional e no Centro de Informação e Manutenção. Ele se conecta ao sistema de tratamento embarcado nos FPGA e solicita o envio das medidas realizadas. Estas medidas deverão ser exibidas assim que recebidas em gráficos e tabelas e devem ser armazenadas em um banco de dados para futuras análises. O sistema deve apresentar também possibilidades de consulta e exportação dos dados.

1.3. Organização

Esta dissertação é composta por seis capítulos. No Capítulo 2 é apresentada uma fundamentação teórica para introduzir os principais conceitos que são utilizados neste trabalho e servem como base para o seu entendimento. O Capítulo 3 mostra como é o atual sistema de monitoramento de trens no MetroRec e faz uma análise do estado da arte, sendo descritos alguns dos sistemas de monitoramento metroferroviários encontrados na literatura e as tecnologias utilizadas em tais sistemas. No capítulo 4, o sistema proposto é descrito. Neste capítulo são apresentados os três subsistemas que compõem o sistema de telemetria. São explicados os processos de configuração e montagem das redes de sensores, a arquitetura e os *softwares* projetados para o sistema embarcado são demonstrados e também é feita a descrição do sistema de supervisão. Este capítulo também apresenta o projeto através da Modelagem UML feito para o sistema embarcado e para o sistema de supervisão. O capítulo 5 demonstra os testes efetuados e uma análise dos resultados obtidos para mostrar a viabilidade do projeto. Por fim, no capítulo 6 é apresentada a conclusão e os trabalhos a serem realizados no futuro.

CAPÍTULO 2

Fundamentação Teórica

Este capítulo apresenta os principais conceitos técnicos e teóricos necessários ao desenvolvimento deste trabalho. Aqui são descritas as diversas tecnologias utilizadas, desde dispositivos de *hardware*, passando por protocolos e padrões adotados e finalizando com as ferramentas de programação utilizadas.

2.1. Sistemas de Telemetria e de Tempo Real

Telemetria é a transmissão, através de rede fixa ou sem fio, de dados providos de múltiplas máquinas ou dispositivos remotos distribuídos sobre uma área geográfica de forma pré-determinada, para uma central de monitoramento e controle. Em resumo, um sistema de telemetria é um sistema que possibilita a aquisição de dados de equipamentos à distância (França, 1997).

A necessidade de instalação de sistemas que possibilitem a transmissão de sinais e dados de automação e monitoramento industrial entre diversos pontos e à distância, que em certos casos pode chegar a algumas dezenas de quilômetros, pode tornar-se um problema, principalmente em áreas onde não há infra-estrutura disponível ou com dificuldades para sua implantação, como também devido a custos altos, prazos de execução e outros fatores. Com o surgimento e desenvolvimento de novas tecnologias para transmissão de dados via frequência de rádio, que vem se tornando cada vez mais baratas e populares, o desenvolvimento de sistemas de monitoramento e controle de processos remotos têm se tornado mais fácil e econômico, uma vez que diversos custos podem ser reduzidos ou mesmo eliminados como os que envolvem instalações, cabeamento, mão-de-obra e o tempo gasto no processo.

O uso de um sistema de telemetria pode trazer inúmeros benefícios para as empresas que o utilizam. Para uma empresa que necessite realizar o monitoramento de máquinas, por exemplo, um sistema telemétrico pode detectar defeitos de operação e indicar as razões do mau funcionamento, permitindo que sejam enviados os profissionais

mais adequados para solucionar problemas específicos. Além disso, sistemas de telemetria auxiliam na redução de taxas de seguro referentes ao mau funcionamento das máquinas e, conseqüentemente, a perda de produção.

Sistemas de telemetria, nas últimas décadas, vêm sendo aplicados em diferentes áreas como medicina (Torres-Pereira et al., 1997; Kumar e Rahman, 2004), construção civil (Iraqi et al., 1999), transportes (Southworth et al., 1989; Santos et al. 2009; Araújo et al., 2009), dentre outras.

Em muitos casos sistemas de telemetria também atuam como sistemas de tempo real. Nestes sistemas, para que o funcionamento do sistema seja correto não basta que as ações sejam corretas, elas devem ser executadas dentro do intervalo de tempo especificado, ou seja, os requisitos de tempo são um fator crítico para determinar a eficiência do sistema.

Em sistemas de tempo real destinados à automação e controle, o não cumprimento dos requisitos de tempo do sistema pode torná-lo inútil, pois pode resultar na quebra de máquinas e/ou em graves acidentes.

Em sistemas telemétricos de tempo real para supervisão ou monitoramento, as restrições de tempo podem não ser tão severas como nos sistemas de controle. Estes sistemas têm como objetivo realizar a aquisição de dados de equipamentos remotos e entregá-los a uma central no menor tempo possível. Dependendo do tipo de monitoramento que estes sistemas realizam, a perda de alguns dados ou um pequeno atraso na entrega podem não afetar drasticamente o sistema. Neste caso o sistema é tolerável a pequenas falhas nos requisitos temporais.

2.2. Redes de Sensores Sem Fio

Uma rede de sensores sem fio (RSSF), em sua forma mais simples, pode ser definida como uma rede de dispositivos chamados nodos ou nós que podem monitorar um ambiente e enviar as informações colhidas através de comunicação sem fio. Os dados são enviados, possivelmente através de múltiplos saltos de transmissão, para um nó sorvedouro (chamado às vezes de controlador ou monitor) que pode ser usado localmente ou estar conectado a outras redes (em geral a *internet*) por meio de um *gateway*. Os nós podem ser estacionários ou se moverem. Podem estar cientes de sua localização ou não e podem ou não ser homogêneos (Verdone et al., 2007). Este tipo de rede é apropriado para situações onde não se pode ter uma instalação com fios e se deseja ter acesso contínuo a determinadas informações.

Redes de Sensores Sem Fio tem sido viabilizadas pela rápida convergência de três tecnologias: microprocessadores, comunicação sem fio e micro sistemas eletromecânicos (MEMS - Micro Electro-Mechanical Systems) (Loureiro et al., 2002). Uma RSSF pode ser usada para monitorar e, eventualmente, controlar um ambiente. Este tipo de rede é formado geralmente por centenas ou milhares de dispositivos autônomos que tendem a ser projetados com pequenas dimensões (cm^3 ou mm^3) chamados nós sensores. Os principais componentes de um nó sensor são: transceptor para comunicação sem fio, fonte de energia, unidade de sensoriamento, memória e processador. O componente lógico de um nó sensor é o *software* que executa no processador (Loureiro et al., 2003). Existem casos em que uma RSSF também pode ser composta de dispositivos chamados atuadores que permitem ao sistema controlar parâmetros do ambiente monitorado.

As RSSF diferem de redes de computadores tradicionais em vários aspectos. Normalmente essas redes possuem um grande número de nodos distribuídos, têm restrições de energia, e devem possuir mecanismos para autoconfiguração e adaptação devido à problemas como falha de comunicação e perda de nodos. Uma RSSF tende a ser autônoma e requer um alto grau de cooperação para executar as tarefas definidas para a rede. Isto significa que algoritmos distribuídos tradicionais, como protocolos de comunicação e eleição de líder, devem ser revistos para esse tipo de ambiente antes de serem usados diretamente. Os desafios e considerações de projeto de RSSF vão muito além das redes tradicionais (Loureiro et al., 2002).

Segundo (Tilak et al, 2002), as redes de sensores sem fio prometem revolucionar o sensoriamento para uma vasta gama de domínios de aplicação devido a suas características de confiabilidade, precisão, flexibilidade, relação custo-benefício favorável e facilidade de implantação.

2.2.1. Classificação das RSSF

A classificação de uma RSSF depende de seu objetivo e área de aplicação. A aplicação influenciará diretamente nas funções exercidas pelos nós da rede, assim como na arquitetura desses nós (processador, memória, dispositivos sensores, fonte de energia, transceptor), na quantidade de nós que compõem a rede, na distribuição inicialmente planejada para a rede, no tipo de disposição dos nós no ambiente, na escolha dos protocolos da pilha de comunicação, no tipo de dado que será tratado, no tipo de serviço que será provido pela rede e conseqüentemente no tempo de vida dessa rede (Ruiz, 2003).

Segundo (Ruiz, 2003), as RSSF podem ser classificadas de acordo com sua configuração, o sensoriamento e o tipo de comunicação. Uma RSSF também pode ser diferente segundo o tipo de processamento que executa.

Quanto à configuração, as RSSF podem diferir na composição do *hardware* dos nós sensores, na organização dos nós dentro da rede, pela mobilidade e pela forma de distribuição e densidade dos nós em um local.

Quanto ao sensoriamento, as RSSF podem diferir pela forma como realizam a coleta dos dados. Esta coleta pode se dar de forma periódica, ou seja, em intervalos de tempo predeterminados, de forma contínua, em tempo real coletando a maior quantidade de dados possíveis no menor intervalo de tempo, ou podem coletar dados apenas quando ocorrerem eventos de interesse ou quando solicitado pelo observador.

Segundo a comunicação, as RSSF podem ser classificadas pela forma como disseminam os dados coletados, pelo alcance dos nós da rede, pelas características da transmissão e recepção de mensagens e pela forma de alocação do canal de comunicação.

2.2.2. Áreas de aplicação

As RSSF têm sido alvo de estudo de vários pesquisadores e estão sendo usadas como solução para diversos tipos de aplicação. Alguns exemplos de utilização das RSSF encontrados na literatura são:

Monitoramento Ambiental: Aplicações para monitoramento ambiental têm uma importância crucial para a comunidade acadêmica e para a sociedade como um todo (Cardell-Oliver et al., 2004; Szewczyk et al., 2004; Santoni et al., 2006). Com estas aplicações, pode-se monitorar áreas com milhares de quilômetros quadrados durante anos. Microsensores em rede tornam possível a obtenção de informações detalhadas sobre áreas naturais onde não é possível se monitorar através de métodos convencionais. Frequentemente nessas aplicações, a rede fornece soluções para medidas de segurança e vigilância; desastres naturais como enchentes e terremotos podem ser percebidos antecipadamente instalando-se um sistema embarcado em rede em lugares onde esses fenômenos podem ocorrer (Verdone et al., 2007).

Medicina: Estas aplicações envolvem telemonitoramento de dados fisiológicos humanos, rastreamento e monitoramento de médicos e pacientes dentro de um hospital, administração de remédios em hospitais, etc. (Scott, 1998; Stankovic et al., 2005) juntaram tecnologia de sensores sem fio em um hospital com aplicações para medicina facilitando a vida de médicos, deficientes físicos e pacientes.

Posicionamento de objetos ou rastreamento de animais: Outra importante área de aplicação está relacionada ao posicionamento (Simon Wong et al., 2005). Uma RSSF poderia ser utilizada para avaliar a posição exata, por exemplo, de pessoas ou objetos em uma sala. Uma aplicação interessante relacionada a esta área é o rastreamento de animais (Guo et al., 2006). O uso de nós sensores em animais, além de possibilitar sua localização, poderia facilitar outras atividades de monitoramento como: detecção do período de calor do corpo dos animais que pode indicar o dia ideal para sua. Eventualmente, pode-se detectar outros problemas como uma perna quebrada ou doenças.

Transportes: Uma RSSF pode ser utilizada para monitorar o movimento de veículos obtendo dados como velocidade e localização. Estas aplicações têm como objetivo gerar medidas de segurança, manutenção, analisar o desempenho e prover melhorias de serviço aos usuários dos meios de transportes monitorados (Santos et al., 2009; Araújo et al., 2009; Svensson, 2005).

2.2.3. Topologias de RSSF

RSSF, dependendo da tecnologia que empregam para comunicação, podem ser definidas com diferentes topologias. Para RSSF que empregam o protocolo ZigBee por exemplo, três tipos de topologias são suportadas: Estrela, Cluster-Árvore, e Malha.

Geralmente em uma RSSF, os nós podem entrar ou deixar a rede dinamicamente resultando em mudanças imprevisíveis no tamanho e na topologia da rede. Isto pode ser ocasionado por falhas na comunicação entre nós ou por defeitos no *hardware* dos dispositivos (Verdone et al., 2007). Em consequência deste fato, as redes em malha podem ser, em muitos casos, a melhor opção para RSSF.

Uma rede em malha é uma rede *ad hoc* de alta confiabilidade. Este tipo de rede é composta de vários nós/roteadores, que passam a se comportar como uma única e grande rede, possibilitando que os nós clientes se conectem em qualquer um destes nós. Os nós fazem a função de repetidores e cada nó está conectado a um ou mais nós (figura 2.1). Desta maneira é possível transmitir mensagens de um nó a outro por diferentes caminhos.

Redes em malha apresentam como vantagem características como: autoformação e autocura. A autoformação está relacionada à capacidade dos nós se associarem automaticamente para formar e otimizar o tráfego de dados na rede. Isto permite que novos nós possam ser adicionados a rede de forma bastante simples. A autocura refere-se à capacidade da rede de recuperar-se da perda de determinados nós ajustando e mantendo a comunicação entre os nós restantes, tornando a rede bastante tolerante a falhas (Ananda et

al., 2006). Além das vantagens citadas acima, as redes em malha apresentam baixo custo e fácil implantação.

A rede *mesh* ou rede em malha pode ser considerada um bom modelo para redes de sensores sem fios de larga escala, distribuídas sobre uma ampla área geográfica (como uma cidade). Aplicações típicas de redes em malha são encontradas atualmente em sistemas de segurança e vigilância de ambientes ou veículos, aplicações de respostas a emergências e sistemas de gerenciamento de tráfego.

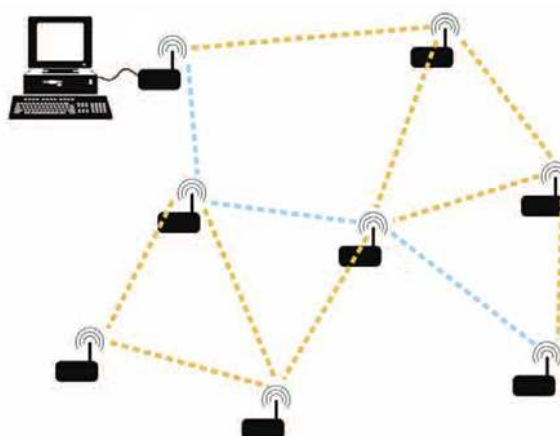


Figura 2.1: Topologia de rede em malha

2.3. Protocolo ZigBee

A ampla demanda de sistemas na área de monitoramento e controle coligada aos avanços tecnológicos ocorridos na área da microeletrônica, em especial dos dispositivos de transmissão de dados sem fio, e a disponibilidade de sensores de baixo custo com capacidade computacional, fizeram crescer o interesse de pesquisadores e empresas pelo desenvolvimento de uma grande variedade de aplicações, tais como: monitoramento de temperatura ou umidade de ambientes, detecção de incêndio ou incidentes, rastreamento de veículos, monitoramento ambiental, automação residencial, biotecnologia, monitoramento e controle industrial, segurança pública e de ambientes em geral, dentre outras aplicações. Nesse contexto, a tecnologia ZigBee surgiu como uma alternativa para atender o desenvolvimento destas aplicações.

O protocolo ZigBee foi criado no ano de 2002. Trata-se de um padrão de protocolo para redes de sensores sem fio, desenvolvido pela ZigBee Alliance junto ao IEEE. A ZigBee Alliance é uma associação de empresas que trabalham em conjunto para desenvolver

tecnologias, baseadas em uma padrão aberto global, que possibilite comunicação sem fio confiável, com baixo consumo de energia e baixo custo para aplicações de monitoramento e controle.

Recentemente, o protocolo ZigBee tem sido considerado um bom candidato para uso em ambientes industriais (Edgan, 2005), sendo utilizado em aplicações de monitoramento e controle de diversos tipos.

2.3.1. Características do Protocolo ZigBee

O protocolo ZigBee opera em três segmentos de frequências livres: 868 MHz na Europa, 915 MHz nos Estados Unidos e 2.4 GHz nos outros lugares do mundo. Ele utiliza a definição 802.15.4 do IEEE para implementar as camadas físicas (PHY) e de controle de acesso ao meio (MAC). As demais camadas do protocolo são definidas pela ZigBee Alliance.

Ele apresenta algumas vantagens importantes sobre os demais protocolos de comunicação como o Wi-Fi e o Bluetooth. Seu consumo de energia é geralmente mais baixo que nas redes com outros protocolos. Uma característica importante do ZigBee que o diferencia dos outros é que ele permite expandir uma rede através de milhares de nós. Teoricamente, uma rede ZigBee pode conter cerca de 65.536 nós, embora, na prática não seja recomendável exceder 3000 nós em uma rede (Ascariz, 2007). Outra vantagem dos dispositivos ZigBee é a velocidade com que novos nós podem ser adicionados a rede: 30 ms; enquanto que um nó que estava em estado de sono pode ser acordado em 15 ms, e então começar a comunicação com outros nós da rede

Numa rede ZigBee são identificados dois tipos de dispositivos: FFD (*Full Function Device*), Dispositivos de Funções Completas, e RFD (*Reduced Function Device*), Dispositivos de Funções Reduzidas. Numa topologia de rede ZigBee, os FFDs podem assumir o papel de Coordenadores, Roteadores ou mesmo de um Dispositivo Final e podem se comunicar com quaisquer outros dispositivos da rede. Os RFD podem atuar somente como Dispositivos Finais e só podem se comunicar com os FFD.

O Coordenador é responsável pela inicialização, distribuição de endereços, manutenção da rede e reconhecimento de todos os nós, entre outras funções, podendo servir como ponte entre várias outras redes ZigBee.

O Roteador só pode ser implementado por intermédio de um dispositivo FFD. Tem as características de um nó normal na rede, mas com poderes extras de também exercer a função de roteador intermediário entre nós sem precisar do Coordenador. Por meio de um roteador, uma rede ZigBee pode ser expandida e assim ter mais alcance.

O Dispositivo Final (*End Device*) - É onde os atuadores ou sensores serão hospedados. Pode ser implementado utilizando-se um dos dispositivos FFD ou RFD.

Uma rede ZigBee suporta três topologias de rede: estrela, malha e cluster-árvore. A figura 2.2 demonstra estas três topologias.

Uma rede em estrela é formada por um nó coordenador e múltiplos nós finais. O coordenador é usado para organizar a rede e rotear as informações. Os nós finais se comunicam com os outros nós através do nó coordenador (Cao et al.). As aplicações que se encaixam neste tipo de topologia, são: automação residencial, periféricos de computador pessoal, jogos e aplicações médicas (Golmie et al.).

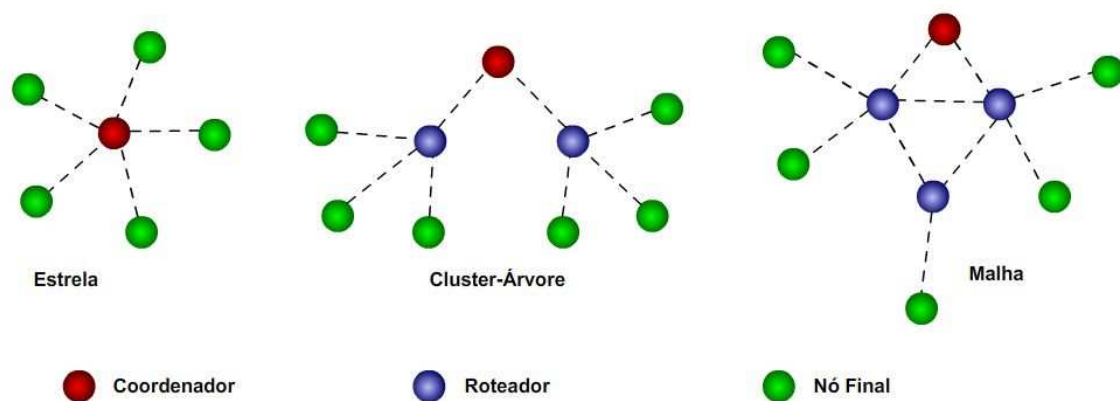


Figura 2.2: Topologias de rede ZigBee

Uma rede em cluster-árvore é composta de múltiplas redes em estrela organizadas por um coordenador. Os nós finais não se comunicam apenas com o nó coordenador, mas também com cada nó FFD com função de roteador. Os nós roteadores não podem se comunicar diretamente com todos os outros roteadores. Essa comunicação é feita via coordenador.

Uma rede em malha é uma rede *ad hoc* que provê alta confiabilidade. Um roteador em uma rede em malha, ao contrário de uma rede cluster-árvore, pode se comunicar com qualquer outro diretamente. Nessa topologia a rede pode se ajustar automaticamente, tanto na sua inicialização como na entrada ou saída de dispositivos na rede. A rede se auto-organiza para otimizar o tráfego de dados. Com vários caminhos possíveis para a comunicação entre os nós, esse tipo de rede pode abranger, em extensão, uma ampla área geográfica (Cao et al.).

2.3.2. Arquitetura protocolar

O protocolo ZigBee é baseado no padrão 802.15.4 definido pela IEEE (IEEE Std 802.15.4™-2006). Este padrão define um protocolo de interconexão para dispositivos de comunicação de dados usando baixa taxa de transmissão, baixa potência, baixa complexidade e transmissões de rádio frequência de pequeno alcance em uma rede sem fio (WPAN).

A arquitetura do ZigBee foi desenvolvida em blocos ou camadas. Cada camada executa serviços específicos para servir a camada acima: a entidade de dados provê dados para o serviço de transmissão e a entidade de gerência fornece informação para todos os outros serviços. Cada entidade de serviço expõe uma interface para a camada superior através do ponto de acesso ao serviço (SAP) e cada SAP suporta um número de primitivas de serviço para ativar a funcionalidade solicitada.

A arquitetura do ZigBee, observada na figura 2.3, é baseada no modelo OSI (*Open Systems Interconnection*) de sete camadas, mas define apenas as camadas relevantes para obter a funcionalidade desejada. O padrão IEEE 802.15.4 define as duas camadas mais baixas: a camada física (PHY) e a sub-camada de controle de acesso ao meio (MAC). A ZigBee Alliance trabalha sobre essas camadas para desenvolver a camada de rede (NWK) e o *framework* para a camada de aplicação, que inclui a camada de aplicação para suporte às sub-camadas (APS), os objetos de dispositivo ZigBee (ZDO) e os objetos de aplicação definidos pelo fornecedor (*Application Object*). A figura abaixo demonstra a organização das camadas no protocolo ZigBee (ZigBee Specification, 2007).

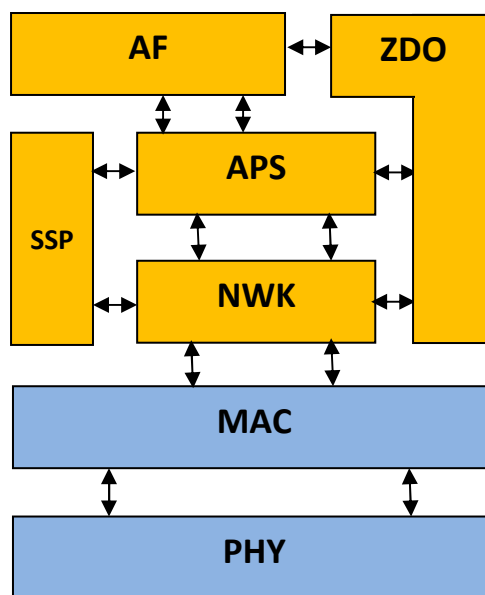


Figura 2.3: Arquitetura do protocolo ZigBee.

A camada Física (PHY), padrão IEEE 802.15.4-2006, define duas camadas PHY que operam em duas faixas de frequência distintas: 868/915 MHz e 2.4 GHz. A frequência mais baixa cobre a banda europeia de 868 MHz e a banda de 915 MHz usadas em países como EUA e Austrália. A frequência mais alta da camada PHY é utilizada virtualmente no mundo inteiro.

A camada de Acesso ao Meio (MAC) (IEEE 802.15.4-2006) controla o acesso ao canal de rádio usando o mecanismo CSMA-CA. Sua responsabilidade inclui transmitir *frames* de sinalização, sincronização e prover um mecanismo confiável de transmissão.

A camada de Rede (NWK) inclui mecanismos usados na conexão e desconexão de dispositivos na rede, mecanismos de aplicação de segurança aos *frames* e roteamento para seus destinos. Adicionalmente, a camada de rede inclui a descoberta e manutenção de rotas entre dispositivos envolvidos na rede NWK. A camada de rede também realiza a descoberta e armazenamento das informações pertinentes da vizinhança. O coordenador da camada NWK é responsável por iniciar uma nova rede, sempre que apropriado e atribuir endereços para os novos dispositivos associados.

A camada de Aplicação é composta pela subcamada de suporte a aplicação (APS), pelo *framework* de aplicações (AF), pelo objeto de dispositivo ZigBee (ZDO) e pelos objetos de aplicação definidos pelos fornecedores. A responsabilidade da subcamada APS inclui a manutenção das tabelas para *binding*, que possibilita combinar dois dispositivos baseado em seus serviços e suas necessidades, enviando mensagens entre os dispositivos combinados. O ZDO define a regra para o dispositivo na rede (coordenador ou dispositivo), iniciando e/ou respondendo às requisições de *binding* e estabelecendo conexões seguras na rede. O ZDO também é responsável por descobrir dispositivos na rede e determinar que tipos de serviços estes dispositivos provêm. O AF é o ambiente em que os objetos da aplicação são hospedados nos dispositivos ZigBee. Esse ambiente possui perfis de aplicação que definem acordos para as mensagens, formatos de mensagens e processamento de ações que permitem aos desenvolvedores criar aplicações interoperáveis e distribuídas empregando entidades que residem em dispositivos separados. Estes perfis tornam as aplicações capazes de enviar comandos, requisitar dados e processar comandos e requisições.

O provedor de serviços de segurança (SSP) é uma subcamada que oferece mecanismos de segurança para as camadas de rede e aplicação. A segurança geral do sistema é definido ao nível do perfil da aplicação, que define o tipo de segurança implementada dentro de uma rede específica.

2.3.3. Comparação com outros protocolos

O ZigBee possui características que o tornam absolutamente distinto de outros protocolos. A tabela abaixo mostra um comparativo das principais características deste protocolo com os protocolos de redes sem fio mais populares.

	ZigBee	Bluetooth	Wi-Fi
Taxa de transferência	250 Kbps	1 Mbps	54 Mbps
Modulação	DSSS	FHSS	DSSS/OFDM
Alcance	300 m	10 m	100 m
Tempo de acesso à rede	30 ms	3 s	3 s
Tempo de transição para o estado ativo	15 ms	3 s	----
Corrente de transmissão	30 mA	40 mA	400 mA
Corrente em <i>standby</i>	3 uA	200 uA	20 mA
Duração de baterias	1000 dias	1 a 7 dias	0,5 a 5 dias
Número de nós	65536	7	32

Tabela 2.1: Comparação entre os protocolos para redes sem fio mais populares.

Dentre estas características, destacam-se a grande escalabilidade das redes ZigBee, o alcance máximo entre os nós e os tempos de associação e de transição dos nós em uma rede ZigBee.

A escalabilidade de uma rede refere-se à capacidade de expansão pela inclusão de novos nós na rede. Ao contrário dos outros protocolos, o ZigBee possibilita a formação de redes com até 65536 nós (Ascariz, 2007). Isto torna este protocolo ideal para redes de sensores que precisam abranger áreas muito grandes. Outra característica que contribui para uso deste protocolo no desenvolvimento de amplas redes de sensores é o seu alcance. Embora tenha sido projetado para redes de pequeno alcance, pode-se encontrar hoje no mercado módulos ZigBee com alcances que variam de 30m a 1600m (Digi Internaitonal, 2008).

Outra característica relevante do ZigBee é o tempo bastante curto que os dispositivos gastam para se associar a uma rede e o tempo em que um dispositivo que se encontrava em *standby* leva para voltar ao estado ativo. Para aplicações industriais onde o tempo é um fator crítico (sistemas de tempo real), esta característica é de grande importância.

2.4. Hardware ZigBee para RSSF

Para desenvolvimento de redes de sensores sem fio com protocolo ZigBee, existem atualmente no mercado inúmeras empresas que produzem diferentes módulos com características e funcionalidades variadas. Dentre as principais e mais conhecidas empresas que trabalham junto à ZigBee Alliance podemos citar: Maxstream, Atmel, Radiocrafts, Texas Instruments, Freescale, dentre outras.

Neste projeto são utilizados módulos RF da Maxstream. Esta empresa produz diversos tipos de módulos que podem ser usados no desenvolvimento de uma grande variedade de aplicações.

O modelo escolhido para a montagem da rede foi o XBee-Pro Znet mostrado na figura 2.4. Estes módulos foram projetados para permitir o desenvolvimento simples e rápido de redes em malha. Eles possuem 20 pinos sendo que 4 desses pinos podem ser usados como entradas analógicas (com conversores analógicos/digitais embutidos) e 12 como entrada ou saída digitais. A tabela 2.2 mostra uma descrição dos pinos do XBee com as possíveis configurações que eles podem assumir.

A Maxstream fornece gratuitamente um *software*, chamado X-CTU, para configuração dos módulos. Este *software* vem com um conjunto de firmwares para diferentes tipos de módulos Xbee e com diferentes versões que permitem definir e configurar a função de cada módulo. Para os módulos XBee Znet, o X-CTU possui várias versões de um firmware para montagem de uma rede em malha. Esses firmwares quando descarregados em um módulo podem atribuir a ele a função de coordenador de uma rede em malha, de roteador ou de um nó final. Para isso é preciso, além de descarregar o firmware adequado no dispositivo de acordo com a função que ele deverá atuar, configurar uma série de parâmetros para fazer com que os nós possam se comunicar e exercer seu papel dentro da rede.



Figura 2.4: Módulo RF ZigBee da Maxstream

Pino	Nome	Direção	Descrição
1	VCC	-	Alimentação
2	DOUT	Saída	Saída de dados da UART
3	DIN / CONFIG	Entrada	Entrada de dados UART
4	DIO12	Bidirecional	Saída digital 12
5	RESET	Entrada	Reinicia o módulo
6	PWM0 / RSSI/DIO10	Bidirecional	Saída PWM 0 /Indicador De Sinal RSSI/Digital E/S 10
7	PWM / DIO11	Bidirecional	Digital IO 11
8	[Reservado]	-	Não conectado
9	DTR / SLEEP_RQ/ DI8	Bidirecional	Pino de controle de espera ou entrada digital 8
10	GND	-	Terra
11	DIO4	Bidirecional	entrada/saída Digital 4
12	CTS / DIO7	Bidirecional	Controle de fluxo ou entrada/saída digital 7
13	ON / SLEEP / DIO 9	Bidirecional	Indicador de status do módulo ou E/S digital 9
14	[Reservado]	-	Não conectar
15	Associate / DIO5	Bidirecional	Indicador de Associação, entrada/saída digital 5
16	RTS / / DIO6	Bidirecional	Controle de Fluxo, ou entrada/saída digital 6
17	AD3 / DIO3	Bidirecional	Entrada analógica 3 ou entrada/saída digital 3
18	AD2 / DIO2	Bidirecional	Entrada analógica 2 ou entrada/saída digital 2
19	AD1 / DIO1	Bidirecional	Entrada analógica 1 ou entrada/saída digital 1
20	AD0 /DIO0	Bidirecional	Entrada analógica 0 ou entrada/saída digital 0

Tabela 2.2: Pinos do módulo Xbee da Maxstream

2.5. Sistemas Embarcados

Um sistema embarcado, ou sistema embutido, é um sistema microprocessado no qual o computador é inteiramente encapsulado ou dedicado ao dispositivo ou sistema que ele controla. Diferente de computadores de finalidade geral, como o computador pessoal, um sistema embarcado realiza um conjunto de tarefas predefinidas, geralmente com requisitos peculiares e não se destina a ser programado pelo usuário final.

Em um sistema embarcado, o usuário final pode fazer escolhas relativas à funcionalidade, mas não pode alterar a funcionalidade do sistema adicionando ou substituindo *software*. Com um computador, isto é exatamente o que um usuário pode fazer: em um momento, o PC pode atuar como um processador de texto e logo depois pode ser uma máquina para jogos simplesmente alterando o *software* em execução. Um sistema

embarcado é projetado para executar uma tarefa específica, embora com diferentes escolhas e opções (Heath, 2003).

Os sistemas embarcados estão inseridos em milhares de dispositivos comuns utilizados no dia-a-dia como em eletrodomésticos, aparelhos de áudio e vídeo, celulares e outros (Gupta, 2002). Algumas das aplicações mais importantes dos sistemas embarcados são: Sistema de controle para aviação, como sistemas de controle inercial, controle de vôo e outros sistemas integrados nas aeronaves (como sistemas de orientação de mísseis), Sistemas automotivos, Sistemas de monitoramento de ambientes e Sistemas de comunicação: redes de computadores, telefonia e TV digital.

A rápida proliferação de sistemas embarcados vem exigindo um aumento do número de engenheiros treinados no desenvolvimento de sistemas baseados em microcontroladores, conceitos de sistemas de tempo real, conceitos de projeto de *hardware* e *software*, processamento distribuído e integração *hardware-software* (Wolf et. al, 2000).

2.5.1. Arquitetura de Sistemas Embarcados

A arquitetura de um sistema embarcado é uma abstração dos seus dispositivos, o que significa que é uma generalização do sistema que normalmente não mostra informações detalhadas de implementação, como o código fonte do *software* ou os projetos de circuitos do *hardware*. No nível arquitetural, os componentes de *hardware* e *software* em um sistema embarcado são representados como composição de alguns elementos que interagem entre si. Elementos são representações de *hardware* e/ou *software* cujos detalhes de execução tenham sido abstraídos, deixando apenas comportamentos de interdependência de informação. Eles também podem ser integrados internamente dentro do dispositivo embutido, ou existir fora do sistema e interagir com elementos internos. Em resumo, uma arquitetura embarcada inclui elementos do sistema, elementos que interagem com o sistema, as propriedades individuais de cada um dos elementos e as relações entre os elementos interativos. (Noegard, 2005).

A abordagem da arquitetura de sistemas embarcados se torna bastante importante pela sua capacidade de rapidamente e informalmente explanar um projeto para pessoas que possuam ou não um conhecimento técnico na área, podendo servir como fundamento para o planejamento do projeto. Como ela claramente esboça os requisitos do sistema, a arquitetura pode atuar como uma base para a análise e teste de qualidade de um dispositivo e de seu desempenho sobre várias circunstâncias.

De acordo com (Noegard, 2005), se entendida e criada, uma arquitetura pode ser usada para estimar com precisão e reduzir os custos através da demonstração dos riscos envolvidos na execução dos vários elementos, permitindo a redução destes riscos. Finalmente, as diversas estruturas de uma arquitetura podem depois ser aproveitadas para a concepção de futuros produtos com características similares, permitindo assim, reutilizar o conhecimento sobre o projeto, levando a uma diminuição da futura concepção e dos custos de desenvolvimento

2.6. Computação Reconfigurável.

Segundo (Athanas, 1993) e (Olukotun, 1994) a principal característica da computação reconfigurável (*reconfigurable computing* – RC) é a presença de um *hardware* que pode ser reconfigurado para implementar uma funcionalidade particular mais apropriada e sob medida, e não um processador de propósito geral. Sistemas de RC unem os microprocessadores e o *hardware* programável com a finalidade de combinar o potencial do *hardware* e do *software* e ser utilizado em aplicações que vão desde um sistema embarcado a sistemas de alto desempenho computacional.

O termo computação reconfigurável, atualmente refere-se a sistemas que provêm alguma forma de programabilidade do *hardware* e adaptação de sua arquitetura de acordo com necessidades computacionais específicas (Compton et al, 2000) (Bondalapati et al., 2002). Esta característica, aliada a capacidade de implementar funções em *hardware* para aumentar o desempenho sem grandes perdas no desempenho, faz da computação reconfigurável uma ótima alternativa para desenvolvimento de SoCs (Holanda, 2007).

Arquiteturas reconfiguráveis, devido a sua adaptabilidade, permitem explorar o paralelismo disponível em aplicações apresentando vantagens no desempenho bastante significativas se comparadas às arquiteturas convencionais. (Bondalapati et al., 2002)

Embora os conceitos básicos tenham sido propostos na década de 60, a RC só veio ser praticável recentemente. A computação reconfigurável surge como um novo paradigma que busca um meio termo entre flexibilidade e desempenho. Durante a última década um grande número de sistemas de RC desenvolvidos pela comunidade científica tem demonstrado o potencial para atingir alto desempenho para uma diversidade de aplicações.

2.6.1. FPGA

Field Programmable Gate Arrays (FPGA) são circuitos integrados digitais que contêm blocos lógicos reconfiguráveis (reprogramáveis), chamados de *CLB* (*Configuration Logical Blocks*) que são formados por portas lógicas e *flip-flops* que implementam funções lógicas, e interconexões entre eles que também podem ser rearranjadas. Engenheiros de projetos podem reconfigurar estes dispositivos para uma enorme variedade de tarefas. Dependendo da maneira como é implementada, alguns FPGA podem ser programados somente uma vez, enquanto outras podem ser reconfigurados diversas vezes. (Maxfield, 2004).

O termo "*Field programmable*" refere-se a possibilidade de um programador alterar o funcionamento e a estrutura lógica do dispositivo, diferente de circuitos integrados típicos que têm suas funcionalidades definidas no momento da fabricação. Já o termo "*gate array*" é uma referência a arquitetura interna básica que torna a reprogramação possível. A figura 2.5 apresenta um esquema da arquitetura interna de um FPGA.

FPGA são caracterizados principalmente pelo tamanho da lógica (medido tanto pelo número de transistores, ou, mais comumente, pelo número de blocos lógicos fundamentais que contêm), pela sua estrutura lógica e funcionalidades de processamento, e pela sua velocidade e consumo de energia (Pellerin, 2005).

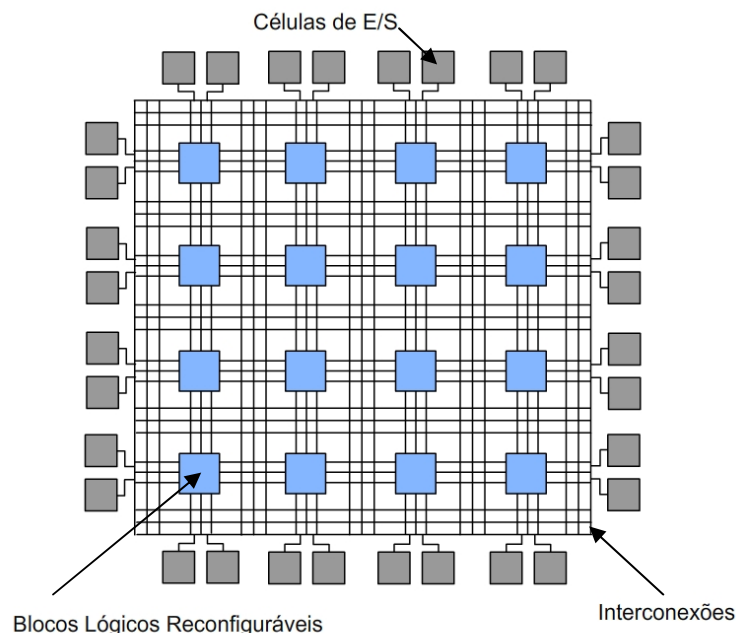


Figura 2.5: arquitetura interna de um FPGA

Atualmente, encontram-se no mercado inúmeras opções de plataformas de *hardware* com diferentes recursos para desenvolvimento de diversos tipos de sistemas. Algumas

delas são: ASIC (*Application Specific Integrated Circuits*), PICs (*Programmable Microprocessors*), PLD (*Programmable Logic Devices*) ou CPLD (*Complex Programmable Logic Devices*). A escolha do FPGA para implementação de um determinado sistema deve ser baseada nos requisitos do sistema e não em preferências pessoais de um projetista.

Se o projeto requer um dispositivo programável que necessite de muitas mudanças no projeto, e utilize algoritmos complexos usando muitas operações de multiplicação/divisão e *loopings*, então pode fazer mais sentido usar um dispositivo de processamento de sinais dedicado como um DSP que pode ser programado e reprogramado facilmente utilizando C ou alguma outra linguagem de alto nível. Se a velocidade não é um forte requisito e uma plataforma compacta e barata é necessária, um microprocessador de propósito geral, como um microprocessador PIC seria a escolha ideal. Finalmente, se os requisitos de *hardware* requerem um maior nível de desempenho, com velocidade de *clock* da ordem de 100 MHz por exemplo, um FPGA apresenta-se como uma boa opção (Wilson, 2007).

Nos casos em que um projeto exige pequenos e simples blocos de *hardware*, então um PLD ou CPLD pode ser a melhor opção (compacto, simples lógica programável), no entanto, se o projeto tem múltiplas funções, ou uma combinação de controles complexos e funções específicas de *hardware*, então o FPGA é a melhor alternativa a se usar. Deve-se ainda considerar a necessidade de execução de funções em paralelo e de otimização do *hardware*. Estes requisitos podem ser mais facilmente implementados em FPGA (Wilson, 2007).

2.6.2. Processador Nios II

O Nios II consiste em um processador RISC de 32 bits de propósito geral, projetado como uma plataforma para dispositivos embarcados e sua síntese pode ser feita nas famílias de FPGA da Altera. Suas principais características são: conjunto de instruções, espaço de endereçamento e *data path* de 32 bits; 32 registradores de propósito geral; 32 fontes de interrupções externas; instruções dedicadas ao cálculo de multiplicações com 64 bits e 128 bits; acesso a uma variedade de periféricos *on-chip*, e interfaces para acesso a memórias e periféricos *off-chip*; dispõe de cerca de 2 GBytes de espaço de endereçamento; e permite a customização de até 256 instruções.

O Nios II é um processador fornecido como um *soft-core*, ou seja, não é uma pastilha de silício pronta. Este processador é representado através de um projeto codificado em uma linguagem de descrição de *hardware*, como VHDL ou Verilog e pode ser implementado em qualquer FPGA da família Altera que o suporte (Peron, 2007).

Um sistema com processador Nios II é equivalente a um microcontrolador ou um SoC (*system-on-chip*) que inclui uma CPU e uma combinação de periféricos e memória em um único chip. O termo “sistema do processador Nios II” se refere a um core do processador Nios II, um conjunto de periféricos *on-chip*, memória interna e interfaces para memória externa, tudo implementado em um único chip da Altera (figura 2.6). Do mesmo modo que uma família de microcontroladores, todos os sistemas do processador Nios II usam um conjunto de instruções e um modelo de programação consistentes (Peron, 2007). Uma das principais diferenças entre o processador Nios II e os microcontroladores consiste em seu conjunto flexível de periféricos. Devido à natureza *soft-core* do processador Nios II, os projetistas podem desenvolver sistemas sob demanda com o conjunto exato de periféricos necessários para as aplicações desejadas.

O processador Nios II se comunica com um barramento de interconexão denominado Avalon. O Avalon é um barramento especial que prioriza a velocidade de transmissão de dados, permitindo conexões em paralelo. Este barramento é responsável pela integração do núcleo de processamento e os demais dispositivos, como memória, temporizadores, periféricos de entrada e saída, e outros. A figura 2.6 apresenta um exemplo de um sistema Nios II.

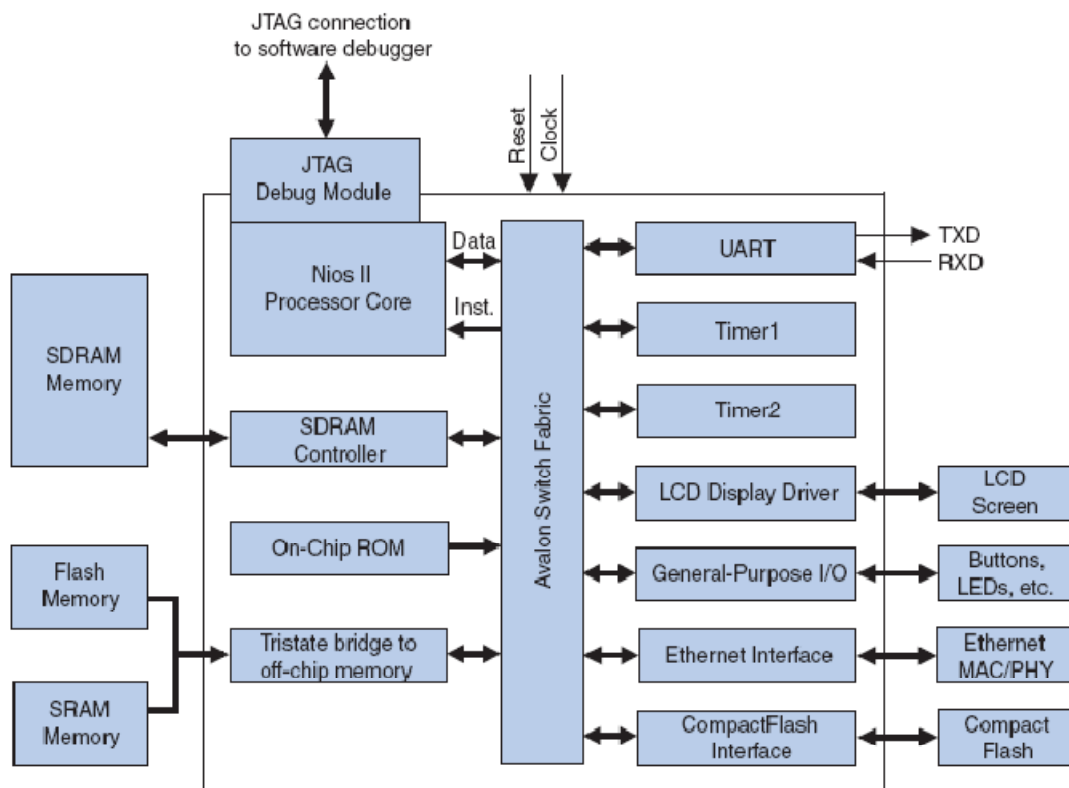


Figura 2.6: Exemplo de Sistema Nios II

A característica de configurabilidade do Nios II não implica necessariamente que o projetista do sistema tenha que configurar um novo processador Nios II para cada projeto. A Altera fornece núcleos prontos do processador Nios II que os projetistas podem usar sem que seja necessário fazer nenhuma alteração, caso algum destes núcleos atendam aos requisitos do seu projeto.

Atualmente a Altera fornece três núcleos do processador Nios II. Nios II/f (versão rápida), Nios II/e (versão econômica) e Nios II/s (versão padrão). As versões Nios II/s e Nios II/f oferecem ainda, respectivamente, 5 e 6 estágios de *pipeline* e predições de salto estático e dinâmico. Ambas possuem *cache* de instruções e somente a versão Nios II/f possui *cache* de dados, todas parametrizáveis. A versão Nios II/e não possui muitas características para o aumento de desempenho, entretanto possui um tamanho menor em elementos lógicos, podendo ser utilizado em quantidade maior em uma FPGA para o aumento do desempenho (Carvalho, 2007).

2.7. Ferramentas de desenvolvimento.

Os tópicos a seguir apresentam uma breve descrição das ferramentas utilizadas para desenvolvimento do sistema embarcado e do sistema de supervisão. Para o sistema embarcado foram utilizadas as ferramentas da Altera. Para o sistema de supervisão inicialmente foi utilizado o ambiente de programação gráfica LabVIEW da National Instruments e posteriormente a IDE NetBeans da Sun Microsystems.

2.7.1. Quartus II

O Quartus II é uma ferramenta desenvolvida pela Altera que possibilita o desenvolvimento de projetos para uma ampla variedade de famílias de FPGA da Altera. A descrição de um projeto pode ser feita através de linguagens de programação de *hardware* como Verilog, System Verilog, VHDL ou AHDL.

O *software* Altera Quartus II fornece um completo ambiente de projeto multiplataforma que se adapta facilmente às necessidades específicas de concepção do sistema. Trata-se de um ambiente global para projetos de sistema em chips programáveis (*system-on-a-programmable-chip* - SOPC). O *software* Quartus II versão 8.0, inclui soluções para todas as fases de projetos de FPGA e CPLD.

2.7.2. SOPC Builder

O SOPC Builder é uma poderosa ferramenta de desenvolvimento da Altera que vem integrada ao *software* Quartus II e que permite definir e gerar um sistema completo em um chip programável (*system-on-a-programmable-chip* - SOPC), em muito menos tempo do que utilizando métodos tradicionais.

O SOPC Builder integra uma biblioteca de componentes pré-fabricados (incluindo processadores Nios II, controladores de memória, interfaces e periféricos) e uma interface personalizada para adição de componentes desejados. As interconexões entre os componentes da arquitetura são feitas automaticamente pela ferramenta através do barramento Avalon.

Através do SOPC Builder, um projetista pode facilmente modificar a arquitetura do sistema e configurar os dispositivos de *hardware* para atuarem da forma desejada.

2.7.3. IDE Nios II

O ambiente de desenvolvimento integrado (IDE) Nios II® é a principal ferramenta para o desenvolvimento de *software* para a família de processadores embarcados Nios II. O IDE Nios II é baseado no IDE Eclipse utilizando as ferramentas de desenvolvimento C/C++ CDT (*C/C++ Development Tools*). (www.eclipse.org). Ele oferece todas as características esperadas de um ambiente de desenvolvimento de projetos de *software* profissional (gestor de projeto, editor código, depurador, etc.), mas também oferece recursos adicionais para aumentar a produtividade em FPGA baseado no desenvolvimento do sistema. Estas incluem: (Zammattio, 2006)

- Importação de todos os parâmetros de *hardware* relevantes para o ambiente de *software*.
- Geração de uma biblioteca ANSI "C" que é customizado para o sistema de *hardware*.
- Inclusão automática e configuração dos controladores de periféricos Funções ANSI "C" e UNIX para auxiliar classes de dispositivo padrão.
- Um *framework* pré-definido para a criação de controladores de periféricos que são captadas a partir de dependências de *hardware*.

Quando o sistema de *hardware* é criado, a ferramenta SOPC Builder grava todos os parâmetros do sistema em um formato de arquivo de texto simples (*.ptf). Este arquivo contém informações sobre a configuração de cada componente e como ele tem sido

conectado dentro do sistema. Quando o desenvolvedor de *software* cria um novo projeto a IDE solicita a localização do arquivo *.ptr* no sistema. A informação contida neste arquivo é então usada para criar um ambiente de desenvolvimento de *software* adaptado para corresponder ao processador e a configuração do sistema.

Um arquivo chamado *system.h* é criado após a análise do arquivo *.ptr*. O *system.h* contém toda a informação relevante de *hardware* para o processador, periféricos e configuração do sistema, sob a forma de declarações *#define* claramente especificadas para cada parte do sistema. Isto significa que essa biblioteca pode ser utilizada ao longo de todo o sistema de *software* para abstrair todos os detalhes de *hardware* a partir da aplicação desenvolvida. Por exemplo, o IDE constrói uma biblioteca ANSI 'C', com base no código fonte da *newlib*, que foi adaptado para corresponder à configuração do processador; isto é muito importante para características como sistema de inicialização, interromper *handlers* e gerenciamento de código em *cache* (Zammattio, 2006).

A fim de delimitar claramente a aplicação e o código do sistema de *hardware* específico, o Nios II IDE cria dois subprojetos, um projeto de aplicação e um projeto de biblioteca do sistema. O projeto de aplicação é usado pelo desenvolvedor para adicionar e gerenciar os arquivos usados para criar a aplicação. O sistema de biblioteca contém todo o *hardware* específico para as funções para abstrair todos os detalhes de *hardware* do desenvolvedor, por isso é conhecida como a biblioteca de sistema *Hardware Abstraction Layer* (HAL). Esta biblioteca oferece um consistente ambiente de programação C / C++, independentemente das características de *hardware* subjacentes ao sistema e separa o código da aplicação do código dos dispositivos de *hardware*. (Anjos, 2009)

O sistema de bibliotecas HAL é um ambiente leve que proporciona uma interface simples do controlador de dispositivos para os programas se comunicarem com a camada de *hardware*. A interface do programa de aplicação (API) HAL está integrada com o padrão ANSI C e permite o acesso à biblioteca e dispositivos periféricos e arquivos usando biblioteca e funções familiares de C, tais como *printf ()*, *fopen ()*, *getch ()*, etc. Um exemplo do sistema de bibliotecas pode ser visto na figura 2.7.



Figura 2.7: Abstração de Hardware do Sistema de Bibliotecas HAL a partir de uma Aplicação de Usuário

O sistema de bibliotecas HAL provê os seguintes serviços:

- Integração com a biblioteca *newlib* do padrão ANSI C (Funções Familiares da biblioteca padrão);
- Drives de dispositivos periféricos (Acesso a todos os dispositivos periféricos do sistema);
- API HAL (Provê um consistente padrão de estilos de interfaces UNIX para os serviços HAL);
- Inicialização do sistema (Executa as tarefas de inicialização para o processador);
- Inicialização de dispositivos (Instancia e inicializa os dispositivos no sistema antes do `main()`).

2.7.4. LabVIEW

O LabVIEW é um ambiente de desenvolvimento criado pela empresa National Instruments para possibilitar o desenvolvimento rápido de sistemas voltados principalmente para as áreas de medição e automação. No LabView, a programação de um sistema é feita através de uma linguagem gráfica que utiliza ícones para representar funções e fios para determinar o fluxo dos dados (ver figura 2.8). Em contraste às linguagens de programação baseadas em texto, em que instruções determinam a execução do programa, o LabVIEW utiliza programação baseada em fluxo de dados, onde este fluxo é que determina a execução.

Desenvolvida pela Sun Microsystems, e mantida pela comunidade, esta IDE é considerada uma das melhores IDEs livres e de código aberto do mercado.

O NetBeans fornece uma base sólida para a criação de projetos e módulos, possui um grande conjunto de bibliotecas, módulos e API's (*Application Program Interface*), um conjunto de rotinas, protocolos e ferramentas para a construção de aplicativos de *software*) além de uma documentação vasta, inclusive em português, bem organizada.

CAPÍTULO 3

Sistemas de Monitoramento de Trens

Este capítulo expõe o atual sistema de monitoramento utilizado no MetroRec e apresenta alguns trabalhos sobre sistemas de monitoramento de trens encontrados na literatura. O capítulo também apresenta alguns trabalhos relacionados ao uso do protocolo ZigBee em sistemas de monitoramento e no final é feito um comparativo das tecnologias empregadas nos sistemas citados.

3.1. Atual Sistema de Monitoramento do MetroRec.

Atualmente, o monitoramento dos metrôes do MetroRec é feito de forma periódica utilizando registradores oscilográficos portáteis (figura 3.1). Estes oscilógrafos são equipamentos de aquisição de dados que podem realizar medidas de sensores e armazená-las em uma memória interna.



Figura 3.1: Oscilógrafo utilizado para aquisição de dados

O monitoramento ocorre da seguinte forma: O oscilógrafo é colocado em um determinado trem e tem suas entradas analógicas conectadas aos sensores existentes no armário da cabine deste trem podendo então realizar as medidas dos fatores que se deseja

monitorar (velocidade, pressão, corrente de armadura e tensão de alimentação). O oscilógrafo utilizado é o modelo OR100E da YOKOGAWA, que dispõe de 4 canais analógicos e 4 digitais permitindo a aquisição de dados de até quatro sensores em paralelo.

O oscilógrafo permanece realizando a aquisição dos dados durante um determinado percurso. Ao final da viagem, o oscilógrafo é retirado da cabine do trem e levado até o CCO onde os dados são extraídos e armazenados em um computador para serem então processados e analisados pelos engenheiros e técnicos.

Métodos convencionais de monitoramento e manutenção baseados em inspeção periódica, como este utilizado pelo MetroRec, possuem várias limitações e podem não ser suficientes para evitar alguns defeitos operacionais (Aboelela et al, 2006). Se o intervalo de tempo entre as inspeções for relativamente longo, muitos defeitos podem não ser detectados até que os danos causados por eles sejam reconhecíveis ou tenham sérias consequências para o equipamento (Nejtkovsky e Keller, 2000).

Além de ineficientes, os equipamentos utilizados neste tipo de monitoramento apresentam um custo bastante elevado. Surgiu então a necessidade de pesquisar e desenvolver mecanismos que possibilitem um monitoramento mais eficiente e que apresentem uma relação custo/benefício mais favorável que o atual método utilizado.

3.2. Estado da Arte

Para o desenvolvimento deste trabalho foi realizado uma pesquisa a respeito das principais tecnologias utilizadas nos atuais sistemas de monitoramento ferroviários com o intuito de analisar suas vantagens e desvantagens e obter conhecimento sobre o estado da arte nesta área. Tomando como base alguns sistemas de monitoramento de trens propostos na literatura, pode-se observar que muitos desses sistemas empregam tecnologias de comunicação sem fio de longo alcance como GPS (Sistema de Posicionamento Global) e GSM (Sistema Global de Telefonia Móvel). Alguns trabalhos encontrados propõem o uso de redes de sensores sem fio aliados a algum tipo de microcontrolador ou microcomputador. Foi realizado também uma pesquisa sobre o estado da arte de sistemas para monitoramento e controle com tecnologia ZigBee, a fim de analisar sua viabilidade e eficiência para esses tipos de sistemas.

3.2.1. Sistemas de Monitoramento Ferroviários

Esta seção descreve alguns dos sistemas para monitoramento de trens encontrados na literatura. São descritos quatro sistemas que empregam princípios semelhantes ao sistema proposto neste trabalho. Ao final deste capítulo as tecnologias são discutidas.

3.2.1.1. Sistema Para Monitoramento de Trens Baseado em Identificação por Rádio Frequência (RFID)

Sharma et AL (2005), propõe um sistema para monitoramento de vagões de trens utilizando módulos de transmissão por rádio frequência RFID, microcontroladores e comunicação GSM, cujo objetivo é rastrear e registrar automaticamente a localização dos vagões eliminando os erros causados pelo rastreamento e registro manual dessas informações. Este sistema é composto por módulos RFID *tags* acoplados aos trens e módulos RFID *readers* instalados na ferrovia a distâncias fixas. Cada RFID *reader* fica conectado a um computador por meio de uma interface Serial. Este computador, através de uma interface serial, se conecta a um celular para envio dos dados coletados para as outras estações de coleta.

Sistemas RFID tipicamente operam em faixas de frequência entre 30 KHz a 500 KHz (baixa frequência), 850 a 950 MHz e 2.4 a 2.5 GHz (ambos considerados de alta frequência).

O autor utilizou neste trabalho, módulos RFID ID10 de baixo custo, produzidos pela empresa 'Innovative Devices'. Estes dispositivos operam com baixa frequência – 125 KHz, e tem uma área de alcance limitada a 12 cm.

Os microcontroladores utilizados foram o AT89S52. Este microcontrolador possui um processador CMOS de baixo consumo com 8 Kbytes de memória EPROM, memória RAM interna de 256x8-bit, 32 linhas de E/S programáveis, 2 temporizadores/contadores de 16 bits, 8 fontes de interrupção, um canal serial para programação. Um cristal com frequência de 11.0592 MHz fornece o sinal de clock para o microcontrolador.

O sistema proposto atua da seguinte forma: Os RFID *tags* acoplados nos vagões dos trens enviam as informações do vagão (como número de identificação e tipo) para os RFID *readers* sempre que o trem passa de frente a um RFID *reader* fixado nas vias. A informação recebida pelos RFID *readers* é processada por um microcontrolador, é exibida em uma pequena tela LCD e simultaneamente enviada para o computador que armazena a informação em um banco de dados. O computador é conectado a um celular que transmite os dados recebidos para as outras estações através do serviço de mensagens curtas (SMS) do GSM. Este sistema restringe-se ao monitoramento apenas da posição dos trens e não realiza qualquer outro tipo de medida.

3.2.1.2. Sistema Wireless para monitoramento da velocidade de trens baseado em RCM3000

Zeng-Quiang et al (2006), propõe um sistema para monitoramento sem fio da velocidade de trens que integra tecnologias de sensores, redes ethernet e GPRS. Este sistema realiza a aquisição de dados em tempo real, o transporte dos dados através de redes sem fio e o gerenciamento dos dados medidos dos trens.

O sistema é composto de um sistema de controle central e muitas estações de monitoramento remotas. Cada estação tem um endereço IP especial e pode monitorar os trens que passam em sua via. Os dados de velocidade dos trens medidos por cada estação são enviados para a central de controle através de uma rede GPRS e de um *gateway* GGSN. Depois de recebidos, os dados podem ser armazenados no servidor da central de controle e consultados por usuários autorizados.

As estações remotas são compostas por dois sensores de velocidade, uma placa com circuito para regulagem do sinal e uma placa contendo um microcontrolador RCM3000 e um modem sem fio (wavecom Q2403A). Os dois sensores são fixados na rodovia a uma determinada distância. Os sinais de saída dos sensores que são produzidos quando o trem passa por eles sofrem vários tipos de distúrbios sendo necessário um circuito para regular e ajustar o sinal para que ele possa ser lido com precisão. Quando cada roda do trem passa pelos dois sensores o número serial e a velocidade são registrados e a velocidade média pode ser calculada depois que o trem passa completamente. A velocidade é calculada com base na diferença de tempo entre os pulsos nos dois sensores. Os dados obtidos são enviados para o modem através de uma porta serial do RCM3000 e então transportados para o centro de controle por meio de uma rede GPRS. Os funcionários no centro de controle, através de um sistema de controle central, podem receber, mostrar, imprimir essas informações e monitorar o estado dos trens que passam na estação de monitoramento.

3.2.1.3. Sistema baseado em comunicação sem fio para monitoramento do desempenho de Veículos ferroviários

Nejikovsky e Keller (2000) propõem um sistema baseado em uma combinação de computação embarcada, processamento de sinais digitais, comunicação sem fio, GPS, e tecnologia de SIG. O sistema inclui plataformas instaladas dentro de cada trem monitorado e uma estação central localizada em um escritório. Cada plataforma *onboard* detecta vários eventos dos veículos em movimento, colocam o tempo e a localização junto com os dados e os enviam para a estação central através de comunicação sem fio. Elas são compostas por

sensores para uma determinada aplicação, um GPS com uma antena para comunicação e um módulo eletrônico encapsulados em uma caixa.

A estação central registra os dados em um banco de dados e mostra a localização e o estado de cada veículo, como também detecta eventos em um mapa. Formas de onda de todos os sensores podem ser enviadas a cada evento e podem ser vistas pelo operador na estação central.

O sistema provê duas formas de comunicação entre a estação central e a plataforma móvel dentro do trem. Dependendo dos requisitos de cobertura, esta comunicação é feita via satélite ou através de redes sem fio associada com transmissão por celular. Entretanto os dois métodos apresentam alguns problemas como: a comunicação por redes sem fio possui um baixo custo, mas seu alcance também é muito limitado. A Comunicação via satélite provê a melhor cobertura, mas o custo é mais alto. O uso de redes sem fio combinadas com transmissão por celulares analógicos provê um amplo alcance a um custo médio.

A estação Central recebe os pacotes de dados dos veículos e armazena em um banco de dados. Para visualização dos dados é utilizado um sistema de informações geográficas (SIG). A Interface do SIG apresenta um mapa da rede de ferrovias e mostra no mapa a localização reportada por cada estação remota.

3.2.1.4. COMBAT – Dispositivo de rastreamento de trens através de comunicação sem fio.

Nishibori et al.(2001), propõe um sistema, denominado COMBAT (*Computer and Microwave Balise Aided Train detection*), para monitorar a posição dos trens em uma ferrovia através de sensores instalados nas vias e comunicação sem fio. O sistema é composto de uma unidade de transmissão de ondas de rádio (que é constituído por: um interrogador, um respondedor instalado na via e dois respondedores embarcados no trem) e uma unidade de processamento. Os interrogadores e os respondedores da via possuem um rádio transmissor/receptor que opera na frequência de 2450 MHz e são instalados um de frente para o outro em lados opostos na via.

Os sensores mantêm uma comunicação constante através do envio de sinais de um para outro. Quando um trem passa entre o interrogador e o respondedor da via a comunicação é interrompida e o sistema assume então que há um trem no local sendo possível determinar a localização deste trem através da localização do equipamento.

Nesse momento os dois rádios respondedores embarcados no trem enviam para o interrogador fixo na via o código de identificação do trem e o código de sua localização

dentro do trem, que foram previamente definidos e armazenados em sua memória. Esses respondedores ficam instalados dentro da cabine a uma certa distância de forma que cada respondedor se comunicará com o rádio interrogador em momentos diferentes. De acordo com esta ordem de comunicação é possível detectar a direção do movimento do trem.

3.2.2. Sistemas para monitoramento e controle com ZigBee

Desde o anúncio de sua primeira versão no fim de 2004, o protocolo ZigBee vem sendo utilizado no desenvolvimento de diversas aplicações voltadas para monitoramento e controle. Existem atualmente na literatura, vários artigos, dissertações e teses que demonstram o uso deste protocolo nesses tipos de sistemas e outros tantos que analisam sua eficiência.

Em (Cao et al.,2008) é demonstrado o uso do ZigBee no desenvolvimento de um sistema para realizar a telemetria em tempo real do consumo de energia, água, gás, temperatura, dentre outras medidas, de casas, empresas ou indústrias. Esse sistema é composto de uma rede de sensores sem fio em malha com ZigBee, onde os nós finais da rede estariam conectados a sensores instalados dentro do lugar a ser monitorado realizando as medidas das grandezas citadas anteriormente. As medidas realizadas são transmitidas pelos nós da rede até alcançarem um nó gateway que coleta os dados da rede, transforma os pacotes ZigBee em pacotes TCP/IP e depois os envia para um computador servidor que armazena os dados recebidos em um banco de dados. Através do sistema proposto por (Cao et al.,2008) é possível monitorar os gastos de uma empresa ou de uma casa podendo ser usado por departamentos de gerência de condomínios ou por empresas fornecedoras de água, gás e eletricidade.

Em (Wang, 2008) é apresentado o uso de uma rede de sensores com tecnologia ZigBee no desenvolvimento de um sistema de monitoramento do estado físico de pacientes em um hospital. Nesse sistema, o paciente pode ter preso ao seu corpo diversos sensores como: sensor de som cardíaco, sensor de pressão arterial, sensor de pulso, sensor de temperatura e sensor de aceleração de movimento. Estes sensores são conectados a um módulo ZigBee que envia os dados coletados para um centro de monitoramento e controle hospitalar onde um *software* de diagnóstico processa e analisa os dados em tempo real, fornecendo meios auxiliares para o diagnóstico correto do médico e ajudando no tratamento preventivo de doenças.

(Zhou et. al, 2009) utilizou a tecnologia ZigBee no estudo e desenvolvimento de um sistema para monitoramento em tempo real de um armazém de grãos. O sistema de Zhou

tem como objetivo monitorar e controlar a temperatura, a umidade e ventilação do ambiente a fim de garantir a qualidade dos grãos e evitar sua degradação.

Em (Liu et al.,2008; Sottile et al.,2008) são demonstrados estudos de técnicas de localização em tempo real de objetos ou pessoas através de redes de sensores com ZigBee. A localização destes objetos é feita através da medida da Qualidade do Link do Sinal (RSSI) entre os nós finais e o coordenador da rede. Estes estudos mostram que através do valor do RSSI é possível determinar a posição com um erro consideravelmente pequeno.

(Umirov, 2008) mostra a utilização do ZigBee em um sistema de controle de motores e analisa sua eficiência no desenvolvimento deste sistema. Neste trabalho o autor analisa alguns problemas relacionadas a perda de dados na transmissão e propõe possíveis soluções.

Em (Santos, 2007) é feito um estudo da aplicação do protocolo ZigBee em redes de monitoramento e controle. O autor faz uma comparação do ZigBee com outros protocolos e analisa as características deste protocolo através de testes realizados sobre diferentes cenários. O estudo foi focado nos atributos interferência interna, confiabilidade, latência de taxa de dados de uma rede ZigBee, nas camadas inferiores correspondentes ao padrão IEEE 802.15.4.

3.2.3. Comparativo entre tecnologias

Sistemas que utilizam GPS (Zeng-Quiang, 2006; Nejkovsky e Keller, 2000) apresentam como vantagem o grande alcance de cobertura, o que possibilita o monitoramento através de uma vasta área geográfica. A desvantagem do uso de GPS para monitoramento de trens consiste em seu alto custo e no fato destes equipamentos estarem sujeitos à falhas causadas por interferências eletromagnéticas, como as que ocorrem ao se passar por dentro de túneis, o que inviabiliza seu uso para monitoramento de trens subterrâneos. Além disso, o funcionamento dos GPS é dependente de um conjunto de satélites proprietários, não havendo garantias que estes continuarão a funcionar no futuro.

Uma outra tecnologia utilizada para monitoramento de trens por (Sharma, 2005) é o RFID aliado a comunicação por GSM. Os módulos RFID utilizados apresentam um alcance bastante limitado e funções também bastante reduzidas realizando apenas a identificação do trem e determinando sua localização em pontos específicos. A utilização de comunicação via GSM torna o sistema muito vulnerável a falhas uma vez que este é dependente da qualidade do sinal do serviço de telefonia disponível no local, além de apresentar um custo elevado.

O presente trabalho utiliza módulos RF com tecnologia ZigBee para desenvolver um sistema de monitoramento que evita as falhas causadas por interferências eletromagnéticas observadas em sistemas que utilizam receptores GPS. O sistema com ZigBee apresenta ainda como vantagem a possibilidade de realizar o monitoramento contínuo de vários fatores simultaneamente e enviar os dados adquiridos remotamente até distâncias relativamente longas sem necessitar de tecnologias auxiliares para a transmissão destes dados; característica que não é observada nos sistemas propostos por (Sharma, 2005; Nishibori et al.,2001).

Analisando as características das tecnologias acima citadas, chegou-se a conclusão que usando a tecnologia ZigBee, pode-se desenvolver um sistema contínuo e autônomo com uma relação custo/benefício bem mais favorável que os sistemas demonstrados.

CAPÍTULO 4

Sistema Telemétrico para Monitoramento de Trens - RailBee

O sistema de telemetria de sinais de trens apresentado neste trabalho é composto por três subsistemas: Um sistema de redes sem fios para captação de sinais nas vias do metrô, um sistema embarcado para tratamento dos dados recebidos pelo sistema de captação e um sistema de supervisão para recebimento, armazenamento e visualização dos dados processados pelo sistema de tratamento. A união desses três subsistemas compõe um sistema que foi denominado RailBee, capaz de obter dados de sensores remotos dentro da cabines do TUE, processá-los e enviá-los até centrais de controle e monitoramento.

Este sistema integra redes de sensores sem fio com tecnologia ZigBee e sistemas embarcados em FPGA para processamento e distribuição dos dados coletados pela RSSF, possibilitando o monitoramento e a distribuição de informações sobre o desempenho dos trens nos trechos da via onde se encontra instalado.

A obtenção de informações como velocidade, pressão nas bolsas de ar, tensão de alimentação e corrente de armadura dos trens em movimento, é de grande valia para os controladores de tráfego, engenheiros e técnicos da manutenção. Através da visualização e análise contínua destes dados, problemas podem ser identificados na hora em que acontecem permitindo que ações sejam tomadas de forma mais rápida. Além disso, a análise dos dados pode indicar a necessidade da tomada de ações que visem melhorar a qualidade dos serviços prestados. Pela medida da pressão, por exemplo, é possível calcular uma estimativa do número de passageiros dentro de cada trem em diferentes horários, identificando os horários e trechos da via em que os trens encontram-se superlotados. Tendo esta informação, os controladores podem tomar medidas para melhorar o nível de conforto dos passageiros, como a inclusão de mais trens trafegando no horário de pico.

Nas próximas seções, cada subsistema é discutido em detalhes. A figura 4.1 demonstra uma visão geral do sistema proposto.

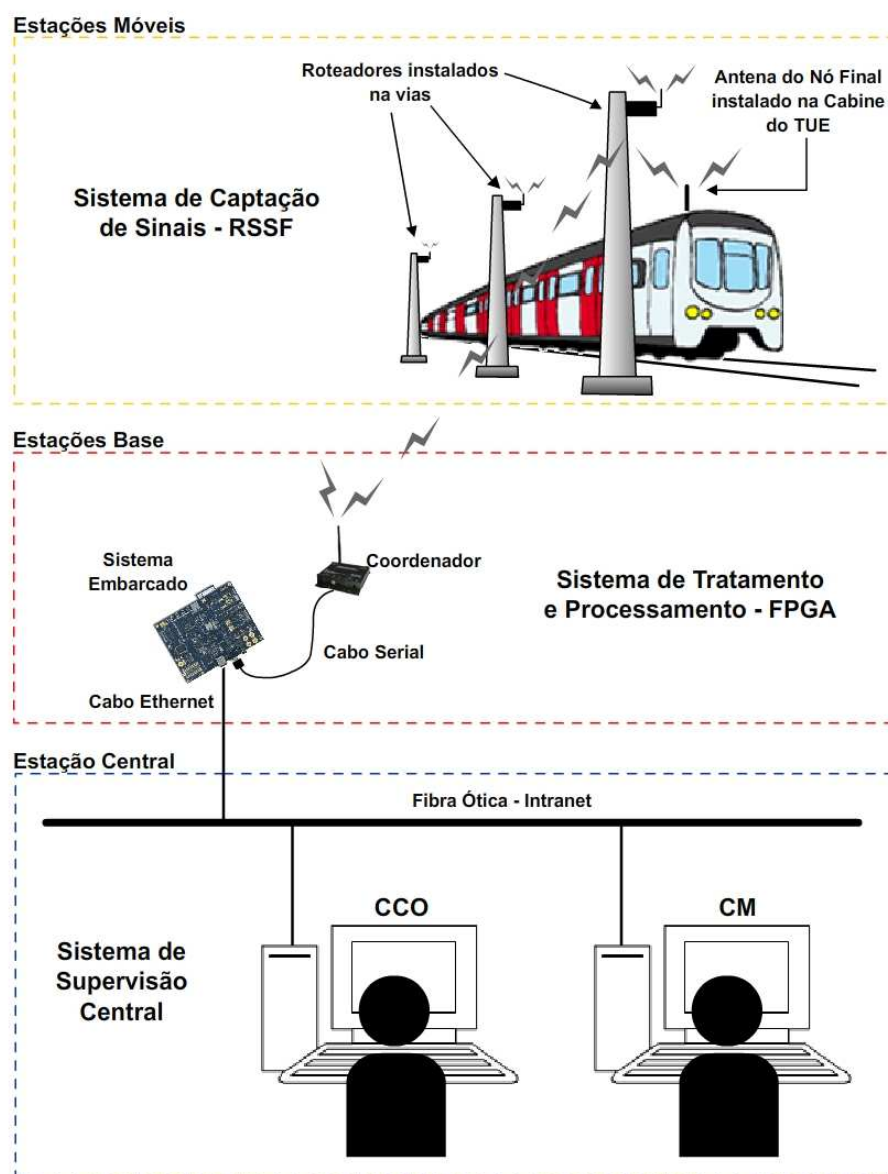


Figura 4.1: Visão geral do sistema proposto

4.1. Sistema de Captação de Sinais

O sistema de telemetria para captação dos sinais é constituído por sensores e módulos de transmissão por rádio frequência instalados nos trens e nas vias formando várias rede de sensores sem fios com tecnologia ZigBee que possibilitam o monitoramento dos trens durante todo seu percurso na área abrangida pelas redes. Cada rede é responsável pela captação dos dados obtidos pelos sensores e pelo envio destes dados até uma Estação Base (EB) onde um sistema embarcado trata e envia os dados tratados para o CCO e CIM (Centro de Informação e Manutenção).

A figura 4.2 mostra um mapa das linhas atualmente existentes no MetroRec. O sistema foi instalado inicialmente na linha vermelha mostrada na figura, que liga Recife a Jaboatão. Ao todo, são 14 estações de passageiros entre Recife e Jaboatão, separadas por uma distância de aproximadamente 1,2 Km. Nesta linha, operam comercialmente das 5h às 23h, nos períodos de maior fluxo de passageiros pela manhã (5h20 às 7h50) e tarde/vespertino (17h às 19h30), 14 TUE que circulam com um intervalo mínimo de 5 minutos. No restante do dia, circulam 9 TUE com um intervalo de 7,5 minutos. São realizadas 300 viagens durante o dia, ou seja, 150 partidas da Estação Recife, 75 partidas da Estação Jaboatão e 75 partidas da Estação Camaragibe.

Nessa linha está localizado o Edifício Operacional Administrativo (EOA) e o Centro de Manutenção de Cavaleiro. No EOA, que está situado próximo a estação Werneck, encontra-se o CCO.

As próximas seções descrevem como é feita a montagem e configuração das redes ZigBee.

4.1.1. Montagem e configuração das redes ZigBee

Para este trabalho foram utilizadas redes com topologia em malha. Este tipo de rede foi escolhido porque os nós finais, que ficam instalados nas cabines dos TUE, estão em constante movimento e tem que se conectar dinamicamente com qualquer um dos roteadores instalados ao longo das vias férreas. Estes roteadores, por sua vez, retransmitem os dados recebidos dos nós finais para outros nós roteadores até que os dados alcancem o coordenador da rede.

Projetou-se a instalação de seis redes que abrangeriam toda a linha de Recife a Jaboatão, como mostrado na figura 4.3. Haverá seis estações Base (Estações receptoras de dados) em estações de passageiros alternadas, começando pela estação Joana Bezerra até a estação Engenho Velho. Cada rede é identificada pela sigla da estação na qual está instalada - JOA(Joana Bezerra), MAN (Mangueira), WEK (Werneck), TEJ (Tejipió), CAV (Cavaleiro) e ENG (Engenho Velho).

A necessidade de utilizar roteadores para aumentar o alcance da comunicação entre os nós finais e os coordenadores da rede pode atribuir ao sistema um problema de confiabilidade pois uma falha que ocasionasse a parada do funcionamento de um destes roteadores comprometeria a entrega dos dados coletados. Para evitar este problema, os nós roteadores são dispostos na via de forma que cada roteador tenha ao seu alcance pelo menos dois outros roteadores. Isto faz com que haja caminhos alternativos para o tráfego

dos dados caso algum dos roteadores pare de funcionar. Desta forma, pelas propriedades da topologia em malha, garante-se a confiabilidade do sistema.

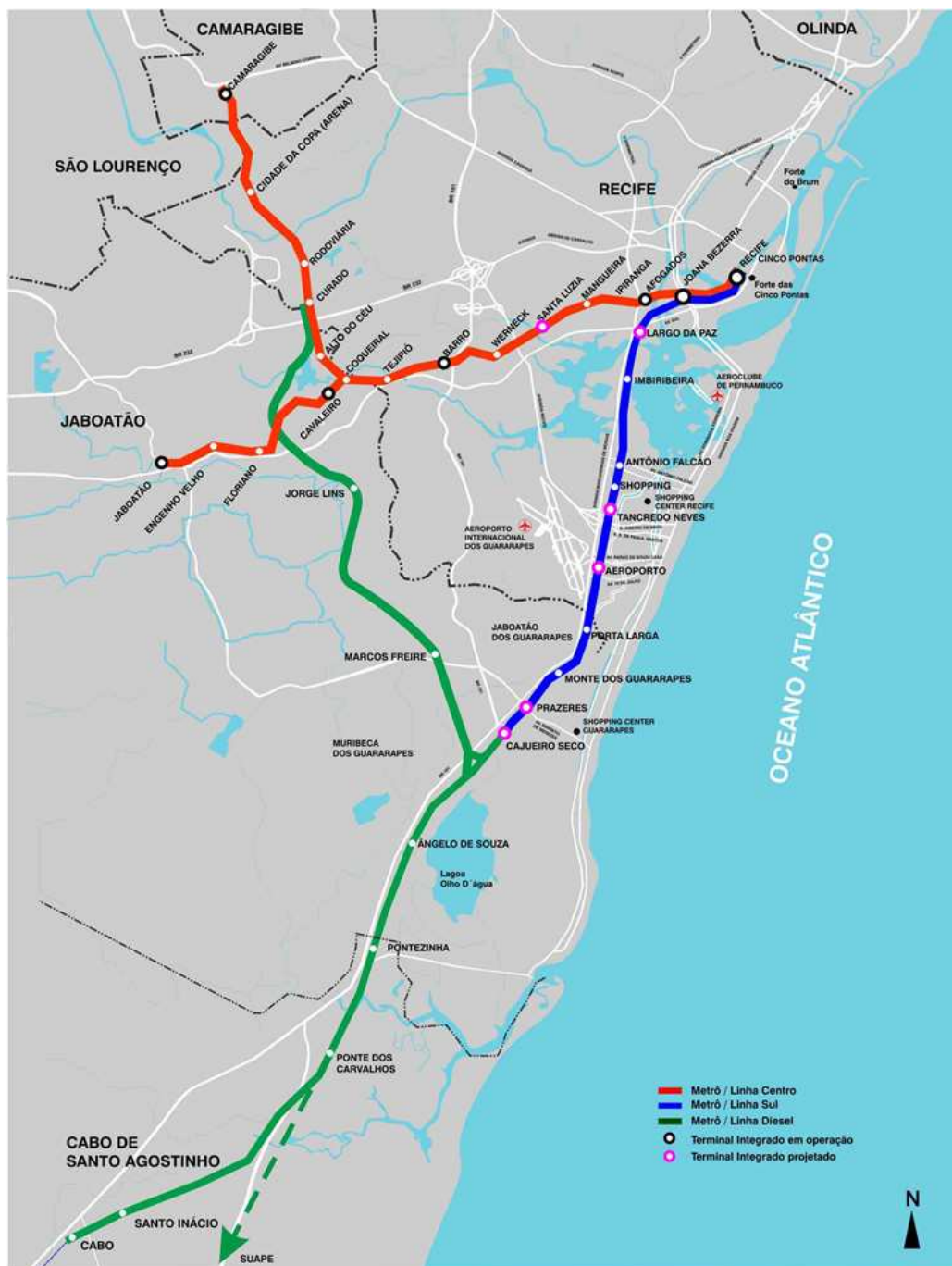


Figura 4.2: Mapa das linhas do MetroRec



Figura 4.3: Localização das redes na linha Recife-Jaboatão

4.1.2. Componentes da rede

Cada rede de sensoriamento sem fio que é constituída por três tipos de módulos ZigBee, chamados de Estações Móveis (EM), Estações Repetidoras (ER) e Estações Base (EB). Cada um desses módulos realiza uma função específica dentro da rede, que é definida pelo tipo e pelas configurações do firmware instalado no módulo.

Estação Móvel (EM): Cada estação móvel é constituída por um dispositivo ZigBee configurado para atuar como nó final. Este dispositivo fica localizado dentro do armário da cabine do trem e tem as suas quatro entradas analógicas conectadas aos sensores de velocidade, pressão, corrente de armadura e tensão de alimentação. A antena deste dispositivo fica instalada na parte superior externa da cabine do TUE para prover um maior alcance e qualidade na transmissão dos dados. A EM pode transmitir os dados para as ER instaladas na via ou diretamente para a EB se estiver ao seu alcance.

Estação Roteadoras (ER): É composta de um dispositivo ZigBee programado como roteador e de um circuito para suporte e alimentação do dispositivo. As ER são instaladas nos armários de sinalização de via (cases), nas Cabines de Seccionamento e Paralelismo (CSP) e nas Subestações Retificadoras (SSR). Estas estações retransmitem os dados recebidos das EM para outras ER ou diretamente para as EB.

Estação Base (EB): Constitui-se de um dispositivo ZigBee configurado para atuar como nó coordenador da rede conectado a um FPGA através de uma interface serial. O coordenador recolhe os dados emitidos pelas EM e através da porta serial, os dados são transferidos do

coordenador para o FPGA onde um sistema embarcado processa os dados recebidos e os envia para o CCO. Este sistema é descrito na seção 4.2

A figura abaixo demonstra uma visão de como estes componentes estarão dispostos na ferrovia.

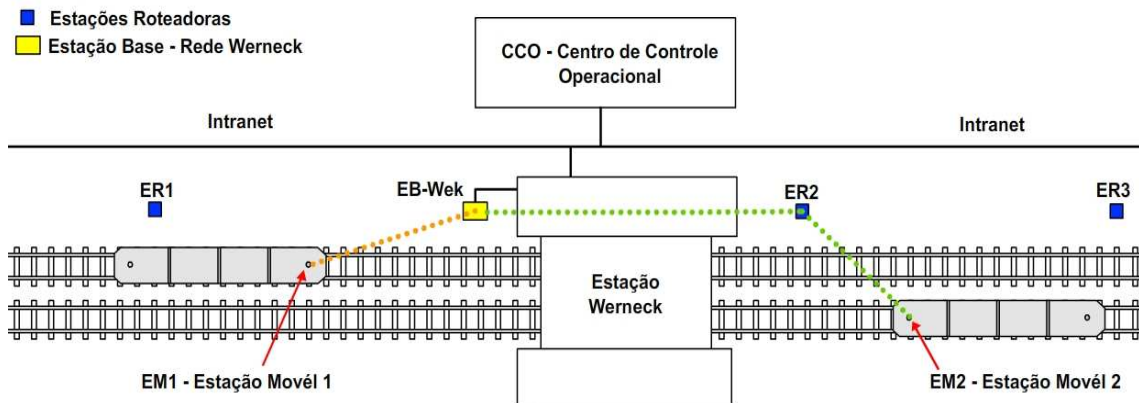


Figura 4.4: Distribuição dos nós da rede ZigBee em um trecho da via.

4.1.3. Configuração dos módulos ZigBee

Para montar a rede em malha é preciso primeiramente programar e configurar os nós finais que ficarão instalados nas cabines dos metrô, os módulos que serão usados como roteadores e os nós coordenadores. A Maxstream fornece uma variedade de firmwares para diferentes tipos de módulo ZigBee que o projetista da rede pode utilizar para configurar os nós de acordo com o tipo de aplicação desejada.

Primeiramente foi realizado um estudo sobre o protocolo ZigBee, sobre o *hardware* dos módulos ZigBee (XBee Znet da Maxstream) utilizados no projeto e sobre os diferentes firmwares para definir o melhor a ser instalado em cada tipo de módulo.

Após a escolha dos firmwares, iniciou-se o processo de configuração de cada módulo para formar a rede. Cada firmware possui diversos parâmetros de entrada que precisam ser corretamente configurados para que os dispositivos possam se associar e se comunicar corretamente dentro da rede. Através do *software* X-CTU fornecido pela Maxstream (figura 4.5), é possível descarregar o firmware nos módulos ZigBees e configurar seus parâmetros.

Para cada nó na rede ZigBee pode-se configurar parâmetros de rede, de endereçamento, de interface serial, de segurança, de configuração dos pinos de Entrada e

Saída, dentre outros. A figura 4.5 mostra uma interface do X-CTU apresentando alguns dos parâmetros para a configuração de um nó coordenador para a rede.

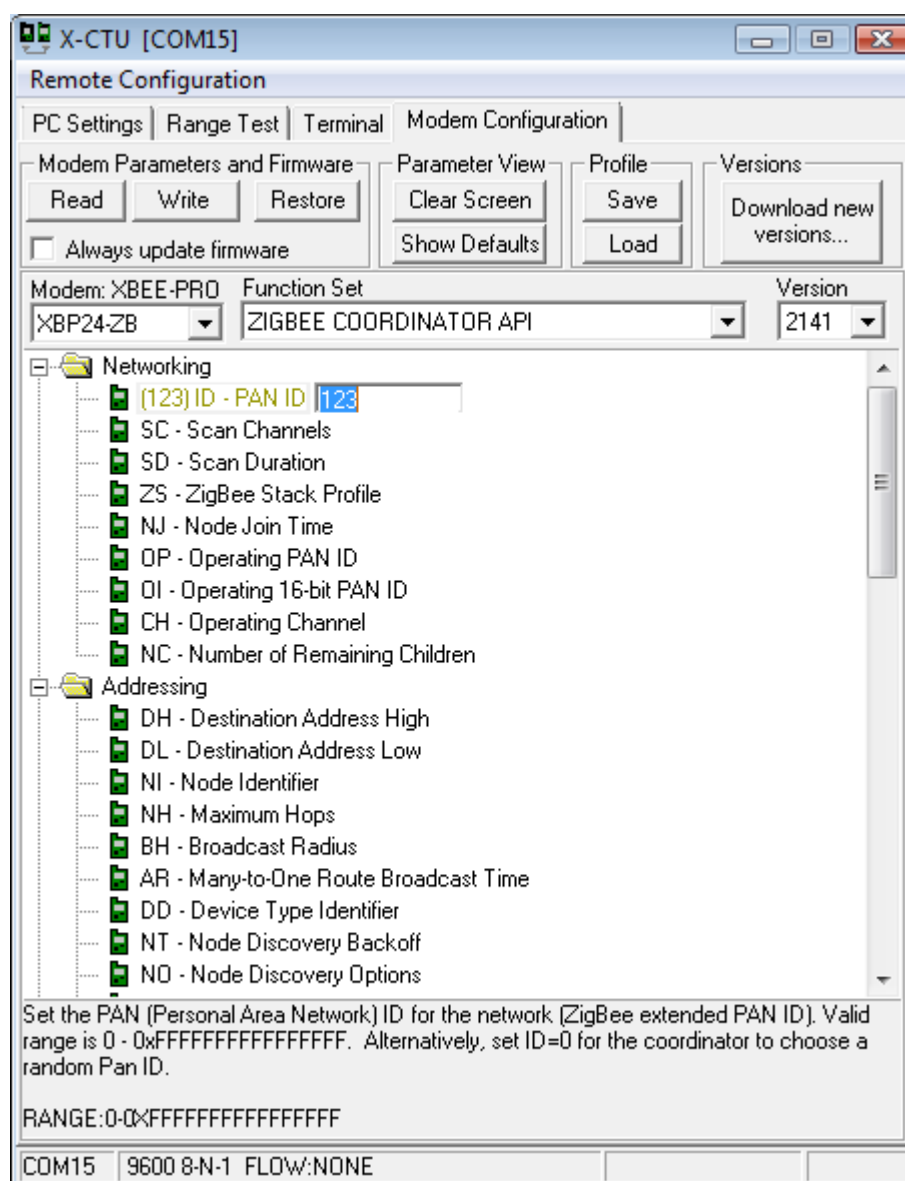


Figura 4.5: Software para configuração dos nós da rede ZigBee

Para o coordenador, os principais parâmetros configurados foram o identificador da rede (ID - PAN ID), o tempo para junção de nós a rede (NJ - *Node Join Time*), a definição da lista de canais a serem verificados (SC – *Scan Channels*) e do tempo para verificação desses canais (SD – *Scan Duration*). Como o nó coordenador ficará conectado a um FPGA, através de uma interface serial, é necessário também configurar os parâmetros que definem as configurações da comunicação serial com o meio externo, como *baud rate* e *bits* de paridade.

Cada rede ZigBee possui um ID único que a identifica, permitindo que os nós se associem a ela. Desta forma, diferentes redes podem atuar em uma mesma área sem que haja interferências entre elas.

O parâmetro SC foi configurado para verificar a disponibilidade de conexão com todos os 16 canais que ele permite, desta forma, um coordenador pode gerenciar até 16 conexões simultaneamente de nós roteadores ou nós finais. O parâmetro NJ (*Node Join Time*) foi configurado com o valor máximo permitido, fazendo com que os nós roteadores e finais possam se conectar a qualquer instante, ou seja, a rede se torna aberta para inclusão rápida de novos nós a qualquer momento.

Para os nós que atuam como estações roteadoras, as configurações de rede são semelhantes as do coordenador com a inclusão de um parâmetro adicional para verificação do canal. Este parâmetro foi habilitado para que os nós roteadores, sempre que ligados ou reiniciados, verifiquem a existência de um coordenador no mesmo canal de frequência certificando-se que estão operando em um canal válido. Assim como o nó coordenador, cada roteador também tem capacidade para gerenciar conexões de até 16 outros nós.

Alguns roteadores poderão ser instalados em pontos da via onde o fornecimento de energia para alimentá-los não é possível, sendo necessário o uso de baterias. Estes roteadores serão configurados para entrarem em estado de sono (*standby*) quando não estiverem transmitindo ou recebendo dados, a fim de economizar energia e prolongar o tempo de vida dessas baterias. Na maior parte da via, os nós roteadores poderão ser ligados através da rede elétrica do metrô, não sendo necessário o uso de baterias.

Para os nós finais, a configuração é um pouco mais trabalhosa, pois é necessário configurar um maior número de parâmetros para cada nó. Como existirão várias redes ZigBee e cada nó final instalado na cabine dos trens terá que se associar a elas em momentos diferentes, nas configurações de rede de cada nó, o parâmetro ID foi configurado para obter o endereço da rede de forma dinâmica no momento da sua inicialização.

O parâmetro NJ (*Rejoin Policy*) foi configurado para permitir que o nó final, ao perder o sinal do coordenador ao qual estava associado, procure por um novo coordenador e se associe a este novo coordenador caso o encontre. Isto permite, por exemplo, que um nó final que estava conectado na rede ZigBee de Werneck ao sair da área abrangida por ela, possa se conectar à rede ZigBee de Tejipió que será a próxima rede no sentido Recife-Jaboatão.

Nas configurações de endereçamento, o endereço de destino dos dados é definido com o valor 0 que corresponde ao endereço do nó coordenador da rede.

Dentro da Cabine dos trens, cada nó final tem os pinos 20, 19, 18 e 17 conectados aos sensores de velocidade, pressão, corrente e tensão de alimentação respectivamente. É

necessário portanto habilitar os parâmetros de entrada destes pinos para atuarem como conversores A/D para possibilitar a medição e conversão dos sensores conectados. Esta configuração é feita através dos parâmetros do firmware D0 (pino 20), D1 (pino 19), D2 (pino 18) e D4 (pino 17).

Para operação do nó final, é preciso ainda configurar a frequência de amostragem de Entrada e Saída de dados. O Parâmetro IR permite definir o intervalo de tempo em milissegundos entre a aquisição de amostras consecutivas. Para este projeto, o parâmetro IR foi configurado com o valor em hexadecimal 7D (125 em decimal), o que significa que a cada 125 ms uma amostra será adquirida e enviada, resultando em uma taxa de amostragem de 8Hz (8 amostras por segundo).

Após a configuração de todos os módulos a rede pode ser então montada e posta em operação. A formação da rede começa quando o nó coordenador da EB é ligado, neste momento o coordenador envia uma mensagem em broadcast com informações sobre a configuração da rede para todos os nós das ER (nós roteadores) e EM (nós finais) que estão ao seu alcance para agregá-los a rede. A rede é então inicializada e as Estações Móveis conectadas podem enviar os dados obtidos dos sensores nas cabines dos trens para a EB. Os dados podem ser enviados diretamente para a EB caso a EM esteja ao seu alcance ou através das ER quando não estiverem.

4.2. Sistema de Tratamento de Dados

O sistema de tratamento de dados é o subsistema que compõe as EB e é responsável pela recepção dos dados do coordenador da RSSF, por seu processamento e envio dos dados processados para o Centro de Controle Operacional (CCO) e para o Centro de Manutenção (CM).

O *hardware* deste sistema é composto por um módulo ZigBee coordenador e uma placa com um sistema embarcado, conectados através de uma interface serial, que serão encapsulados em uma caixa e instalados em seis pontos estratégicos da via para possibilitar que o coordenador possua a melhor qualidade de conexão com os nós da rede, de forma a diminuir o número de roteadores necessários e evitar perda de dados.

A necessidade de se ter o nó coordenador da rede instalado na via, inviabiliza a utilização de um computador comum junto ao nó coordenador para implementar o sistema de tratamento. Como alternativa optou-se pela implementação de um sistema embarcado em FPGA, que é um dispositivo compacto de processamento dedicado suficientemente robusto para gerenciar os processos de tratamento e distribuição dos dados tratados. O uso de uma FPGA para implementação do sistema de tratamento, além de possibilitar sua

instalação em qualquer ponto da via, atribui ao sistema flexibilidade na atualização de *softwares*, oferece possibilidades simples de expansão e facilita sua manutenção. Como o sistema de tratamento embarcado é dedicado a tarefas específicas, através de engenharia pode-se aperfeiçoar o projeto reduzindo tamanho, recursos computacionais e custo final do produto.

4.2.1. Arquitetura do sistema

O modelo arquitetural utilizado para este trabalho foi apresentado por Noergaard (2005). Este padrão define uma formação típica de um sistema embarcado dividido em camadas interdependentes: camada de *hardware*, camada de *software* de sistema e camada de aplicação, como pode ser visto na figura 4.6. Para qualquer sistema embarcado a camada de *hardware* deve estar presente. Entretanto algumas modelagens abstraem as camadas de *software* e aplicação tornando essas camadas opcionais para determinadas implementações. Embora dispensáveis como na arquitetura de Noergaard, essas camadas são bastante importantes para se obter as vantagens da modelagem arquitetural. As camadas do sistema são descritas a seguir.

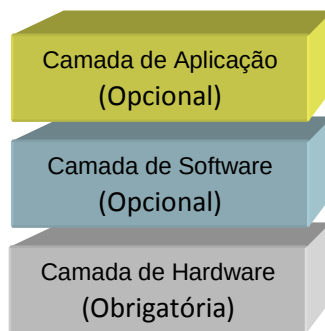


Figura 4.6: Modelo arquitetural de um sistema embarcado

4.2.1.1. Camada de Hardware

Para desenvolvimento do projeto, foi utilizado um kit de desenvolvimento Nios II, edição Stratix. Este kit consiste em uma placa que provê uma plataforma de *hardware* e *software* para o desenvolvimento de sistemas embarcados baseados nos dispositivos da família Stratix da Altera. Os principais dispositivos presentes nesta placa são:

- Um FPGA Stratix™ EP1S10F780C6
- 8 Mbytes de memória *flash*

- 1 Mbyte de RAM estática
- 16 Mbytes de SDRAM
- Duas portas seriais RS232 DB9
- Interface Ethernet (MAC/PHY)
- Conectores JTAG para conexão com dispositivos da Altera.
- Circuito oscilador para *clock* de 50 MHz.

Além dos componentes citados acima, a placa ainda possui 41 pinos de I/O aos quais podem ser conectados LEDs, display de sete segmentos e botões. A figura 4.7 demonstra uma placa semelhante com todos os componentes da placa de desenvolvimento utilizada.



Figura 4.7: Placa de desenvolvimento Nios II

O FPGA presente nesta placa possui 10570 elementos lógicos e 920448 bits de memória *on-chip*. Esta placa de desenvolvimento traz consigo projetos de referência do processador Nios II para que projetistas de *hardware* possam usá-los como exemplo para desenvolvimento de seus próprios projetos. Projetistas de *software* podem usar um projeto com um processador pré-programado na placa para começar a desenvolver *software* rapidamente. (Altera, 2007).

Para implementar o sistema embarcado no FPGA, primeiramente é necessário definir a arquitetura especificando e configurando sua estrutura interna e suas interfaces de entrada e saída para acesso aos periféricos necessários para a aplicação. Esta tarefa é realizada utilizando-se a ferramenta SOPC Builder.

Foi tomado como base para o desenvolvimento, o modelo de arquitetura *full_featured* disponibilizado pela Altera, que possui todos os componentes necessários ao

desenvolvimento do sistema e foi projetada para ter alto desempenho. A arquitetura *full_featured* possui os seguintes componentes: Uma CPU Nios II/f (*fast*), memória *On-Chip*, memória *Flash*, memória SDRAM e SRAM, um controlador DMA, PIO para LED, *display* de sete segmentos e Botões, um *display* LCD, uma interface JTAG, uma interface Ethernet e uma porta serial.

Esta arquitetura foi modificada e reduzida para atender as necessidades do projeto. Foram retirados dela todos os componentes que não seriam necessários como LED, botões, *display* de sete segmentos, *display* LCD e Memória SDRAM.

A figura 4.8 apresenta a organização da arquitetura planejada para o sistema e a figura 4.9 demonstra o projeto desta arquitetura no SOPC Builder.

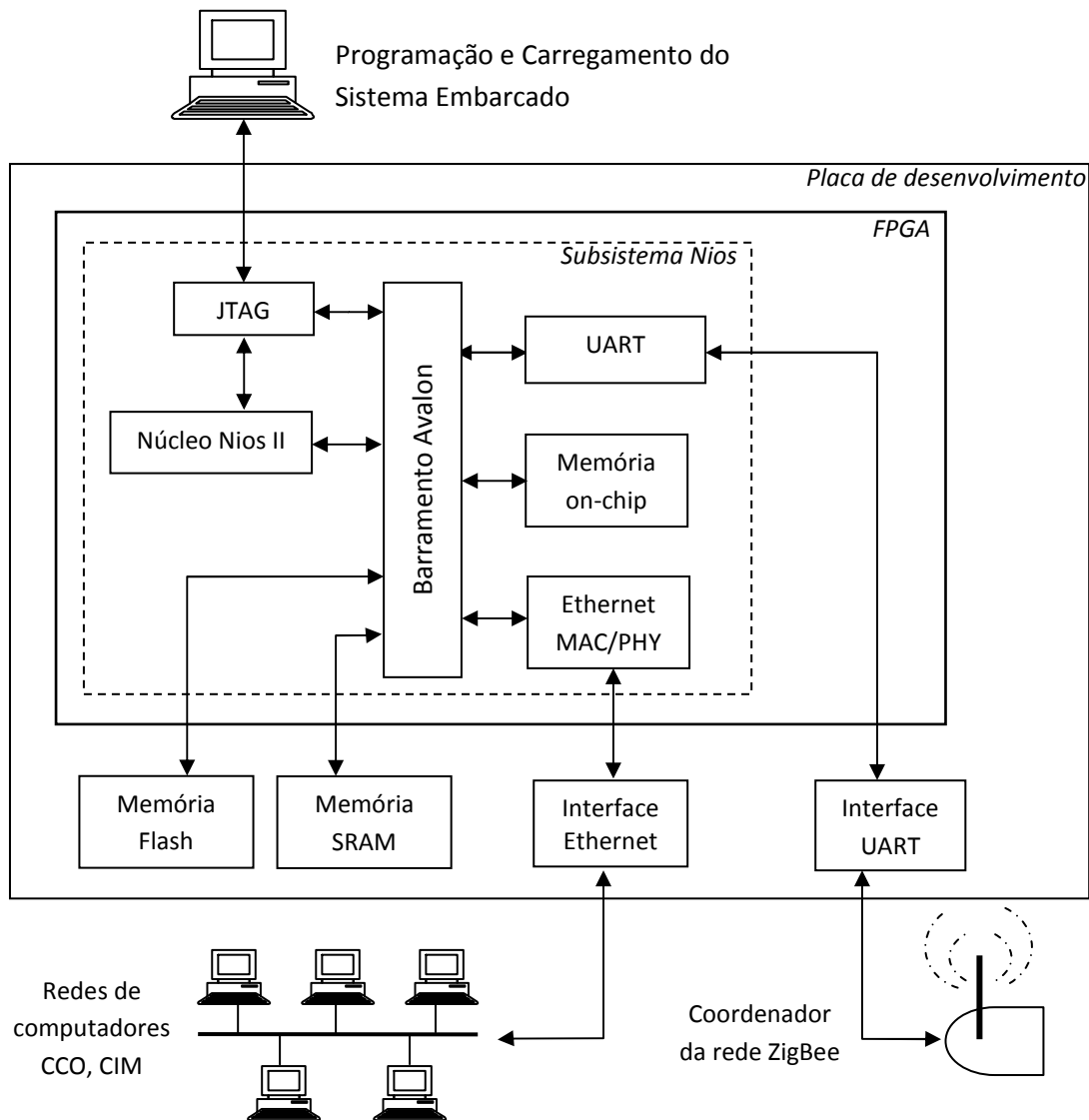


Figura 4.8: Arquitetura do sistema de tratamento

Os principais componentes incluídos na arquitetura são:

- **Uma CPU Nios II** – Está CPU foi configurada para usar como base o núcleo *fast* (Nios II/f) projetado para prover alto desempenho. A CPU se conecta através do barramento Avalon a dois blocos de memória TCM *on-chip* para *cache* de dados e instruções.
- **Uma interface UART** - Esta interface foi configurada com as mesmas configurações da porta serial do módulo ZigBee para permitir a comunicação e o recebimento dos dados deste dispositivo.
- **Uma interface Ethernet** - Está interface permite a comunicação com a rede local e possibilita que o sistema envie os dados processados para o sistema de supervisão.
- **Memória On-chip** – Pequeno componente de memória utilizado para armazenamento de instruções e dados para acesso e processamento rápido.
- **Memória Flash e SRAM** – A memória *flash* é usada para armazenar permanentemente os arquivos de inicialização do sistema fazendo com que não seja necessário descarregar e compilar o *software* na FPGA através da JTAG a cada vez que o sistema precise ser reiniciado. A SRAM é a memória volátil utilizada para armazenar o sistema carregado do flash na inicialização e durante todo o funcionamento do sistema.

Use	...	Module Name	Description
☒		+ cpu	Nios II Processor
☒		+ tightly_coupled_instruction_memory	On-Chip Memory (RAM or ROM)
☒		+ tightly_coupled_data_memory	On-Chip Memory (RAM or ROM)
☒		+ dma	DMA Controller
☒		+ ext_ram_bus	Avalon-MM Tristate Bridge
☒		+ ext_flash	Flash Memory (CFI)
☒		+ ext_ram	IDT71V416 SRAM
☒		+ onchip_ram_64_kbytes	On-Chip Memory (RAM or ROM)
☒		+ lan91c111	LAN91C111 Interface
☒		+ sys_clk_timer	Interval Timer
☒		+ high_res_timer	Interval Timer
☒		+ jtag_uart	JTAG UART
☒		+ uart1	UART (RS-232 Serial Port)
☒		+ sysid	System ID Peripheral
☒		+ pll	PLL

Figura 4.9: Arquitetura projetada no SOPC Builder.

Após a definição e configuração de todos os componentes, a arquitetura do sistema Nios II foi então gerada. O SOPC Builder gera como resultado alguns arquivos contendo informações sobre a arquitetura projetada. Esses arquivos são:

- Um arquivo com a extensão .ptf que armazena o conteúdo do sistema de *hardware* criado pelo SOPC Builder. Este arquivo é solicitado pela IDE Nios II para criar e compilar *software* para um *hardware* específico.
- Arquivos de linguagem de descrição de *Hardware* (HDL) - Estes são os arquivos com o projeto do *hardware* descritos no sistema do SOPC Builder e que são utilizados pelo Quartus na compilação do projeto.

Depois de gerar o sistema para o Nios II utilizando o SOPC Builder, é necessário integrá-lo ao projeto global do Quartus II. Utilizando o *software* Quartus, são realizadas todas as tarefas necessárias para criar o projeto final para o *hardware* da FPGA. São atribuídos pinos de I/O para sinais e pode-se ainda especificar requisitos de tempo e aplicar outras restrições. Finalmente, o projeto é compilado para produzir um arquivo objeto SRAM (.sof) para configurar o FPGA.

O arquivo de configuração (.sof) gerado na compilação do projeto no Quartus pode ser então carregado na placa através da interface JTAG utilizando um cabo USB-Blaster. Após a configuração, o FPGA se comporta conforme o especificado pelo projeto de *hardware* definido, que neste caso é um sistema processador Nios II.

4.2.1.1.1. Configuração do Hardware utilizado

O projeto da arquitetura foi definido de forma a prover uma arquitetura eficiente para atender aos requisitos do sistema, mas que ao mesmo tempo, consumisse poucos recursos do *hardware* utilizado para sua prototipagem. Desta forma, procurou-se reduzir ao máximo o número de periféricos e interfaces diminuindo assim o espaço ocupado do *hardware*. O objetivo de reduzir os recursos de *hardware* utilizados é tornar viável a instalação deste sistema em FPGA com recursos mais limitados e mais baratos, como os FPGA do tipo Cyclone, ou em outros tipos de circuitos, para reduzir os gastos na implantação do sistema.

A tabela 4.1 apresenta as características da configuração definida para o *hardware*.

Características da Arquitetura do Sistema	
Versão do Quartus II:	8.0 build 215 07/30/2009 SJ Web Edition
Nome de Revisão:	Servidor_RailBee
Nome da entidade de alto nível:	Servidor_RailBee
Família:	Stratix
Dispositivos:	EP1S10F780C6
Modelos de Temporização:	Final
Requisitos de tempo encontrados:	Sim
Total de elementos lógicos:	4.091 / 10.570 (46%)
Total de pinos:	104/ 427 (24%)
Total de pinos virtuais:	0
Total de bits de memória:	687.680 / 920.448 (75%)
Blocos DSP de 9 bits:	8 / 48 (17%)
Total de PLLs (phase-locked loop):	2 / 6 (33%)
Total de DLLs (delay-locked loop):	0 / 2 (0%)
Frequência do clock	50 MHz
Total de bytes do software	320 Kb

Tabela 4.1: Características da arquitetura projetada para o sistema embarcado.

4.2.1.2. Camada de Sistema de Software

Seguindo o modelo de referência apresentado, a camada de sistema de *software* (Figura 4.10) é subdividida em *drives* de dispositivos, sistema operacional e biblioteca de funções de sistema. Inicialmente, cada periférico da plataforma Nios II deve fornecer um driver para que o *software* saiba utilizá-los corretamente, formando o conjunto dos *drivers* de dispositivos. A Altera oferece um alto nível de abstração ao fornecer uma biblioteca de funções em C denominada HAL (*Hardware Abstraction Layer*) que padroniza o uso dos

drivers de tal forma que o programador possa utilizá-los pelo simples manuseio de funções GNU C. (Anjos, et. al., 2008).

Para o gerenciamento dos vários processos de *software* que necessitam ser executados concorrentemente, a arquitetura proposta utiliza o sistema operacional μ C/OS-II que é fornecido em uma versão especializada ao processador Nios II. Chamado de RTOS (*Real Time Operating System*), esse sistema operacional é descrito como portátil, escalável, possui um *kernel* de tempo real e um ambiente multitarefa (Labrosse, 2002).

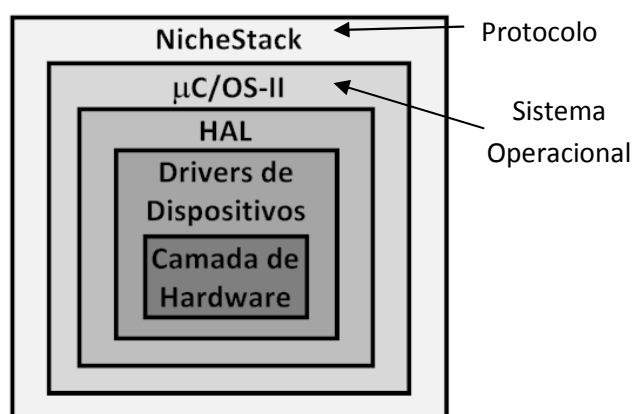


Figura 4.10: Camada de sistemas de software

A interconexão com redes locais através da interface ethernet é realizada utilizando a pilha de protocolo NicheStack TCP/IP que é uma implementação reduzida do protocolo TCP/IP.

4.2.1.3. Camada de Aplicação

A camada de aplicação corresponde ao *software* para tratamento dos dados e distribuição dos dados tratados. As funcionalidades do sistema são implementadas em três módulos de *software* que são executados paralelamente.

O primeiro módulo realiza os processos de recepção dos dados da porta serial e de separação e verificação dos pacotes ZigBee.

O segundo é responsável pelo reconhecimento dos pacotes, extração dos valores digitais, conversão de valores de digital para analógico, cálculo das medidas realizadas (velocidade, pressão, tensão de alimentação e corrente de armadura) e montagem dos pacotes com os dados processados.

O terceiro módulo atua como um sistema servidor de dados realizando a comunicação com uma rede local através da interface ethernet. Este módulo gerencia as conexões de sistemas clientes e envia os dados processados para estes sistemas. O desenvolvimento destes módulos é descrito em maiores detalhes no tópico a seguir.

4.2.2. Desenvolvimento do sistema embarcado

O desenvolvimento do sistema foi dividido em três etapas: Análise dos requisitos, projeto da arquitetura e desenvolvimento do sistemas de tratamento e servidor de dados .

A etapa de análise dos requisitos é descrita nos próximos tópicos deste capítulo. Esta análise serve como base para identificar a necessidades do sistema indicando os recursos de *hardware* e *software* que precisam ser desenvolvidos.

Após a análise, foi iniciado o processo de elaboração da arquitetura do sistema, que foi mostrada no tópico 4.2.1.1, utilizando a ferramenta SOPC Builder do Quartus II.

Após a elaboração da arquitetura deu-se início ao desenvolvimento do sistema de tratamento de dados utilizando o ambiente de programação Nios II, que é demonstrado nos tópicos a seguir.

4.2.3. Análise e projeto do sistema embarcado

Esta seção descreve os requisitos do sistema de tratamento que será implementado e apresenta o seu projeto através de diagramas UML. Os requisitos são apresentados sob os aspectos funcionais e não funcionais. Os requisitos não funcionais correspondem às características que o sistema deve apresentar, enquanto que os funcionais correspondem às funcionalidades as quais o sistema deve dar suporte.

4.2.3.1. Requisitos do sistema

O sistema de tratamento tem como função receber os dados captados pelo coordenador da rede ZigBee, processar estes dados e disponibilizá-los rapidamente para um sistema de supervisão. Para realizar tal trabalho, o sistema deve atender aos seguintes requisitos:

Requisitos não funcionais:

- **Processamento rápido e dedicado** – O sistema deve ser capaz de processar os dados de forma rápida e otimizada, garantindo o envio das informações no menor tempo possível.
- **Processamento multitarefa** – O sistema deve ser capaz de gerenciar diferentes tarefas em paralelo, como o tratamento dos dados recebidos do nó coordenador e o gerenciamento de conexões dos sistemas de supervisão.
- **Flexibilidade para atualização** – O sistema deve permitir que atualizações sejam feitas em seu *software* para atender a novos requisitos.
- **Confiabilidade** – O sistema deve garantir a entrega dos dados tratados ao sistema de supervisão. Para isto, o sistema deverá ser desenvolvido de forma a diminuir as possibilidades de falhas no funcionamento e na comunicação com sistemas externos.
- **Segurança** – O sistema de tratamento deve aceitar apenas conexões de máquinas onde o sistema de supervisão esteja instalado, não permitindo o acesso às informações por usuários não autorizados.

Requisitos funcionais:

- **Extrair os dados da porta serial** – O primeiro passo do sistema de tratamento é recuperar os dados do buffer da porta serial. Cada *byte* extraído deve ser convertido para hexadecimal para verificação dos pacotes.
- **Separar e verificar os pacotes ZigBee** – Cada pacotes de dados deve ser separado e verificado.
- **Extrair dos pacotes as informações necessárias** – Os dados referentes às medidas dos sensores são separadas.
- **Calcular medidas** – Os dados digitais devem ser convertidos em seus correspondentes valores analógicos e as medidas calculadas.
- **Empacotar dados processados** – As informações obtidas devem ser empacotadas em um formato específico.
- **Receber conexões da rede local** – O sistema deve esperar e aceitar(ou recusar) conexões do sistema de supervisão.
- **Validar conexões** – As conexões recebidas devem ser validadas através de senha.
- **Enviar dados processados para clientes conectados** – Os dados processados e empacotados de vem ser enviados para o sistema de supervisão conectado.

4.2.3.2. Modelagem UML

A Linguagem de Modelagem Unificada (UML) fornece uma linguagem gráfica para visualização, especificação, construção e documentação dos artefatos de um sistema de *software*. A elaboração de diagramas UML facilita a concepção do projeto por qualquer integrante do grupo de desenvolvimento, principalmente facilitando o entendimento de áreas que não são de sua compreensão. Os diagramas UML também podem ser usados em engenharia reversa e se tornam úteis para uma apresentação do projeto ao público em geral (Fowler, 2005). Para o sistema de tratamento foram projetados os diagramas de Componentes, Atividades e Seqüência.

4.2.3.2.1. Diagrama de Atividades

O diagrama de atividades representa os fluxos conduzidos por processamentos. É essencialmente um gráfico de fluxo, mostrando o fluxo de controle de uma atividade para outra. Comumente isso envolve a modelagem das etapas sequenciais em um processo computacional. O diagrama abaixo apresenta o processo de aquisição dos dados desde o a rede sem fio, as tarefas envolvidas no processamento destes dados e seu envio para o sistema de supervisão.

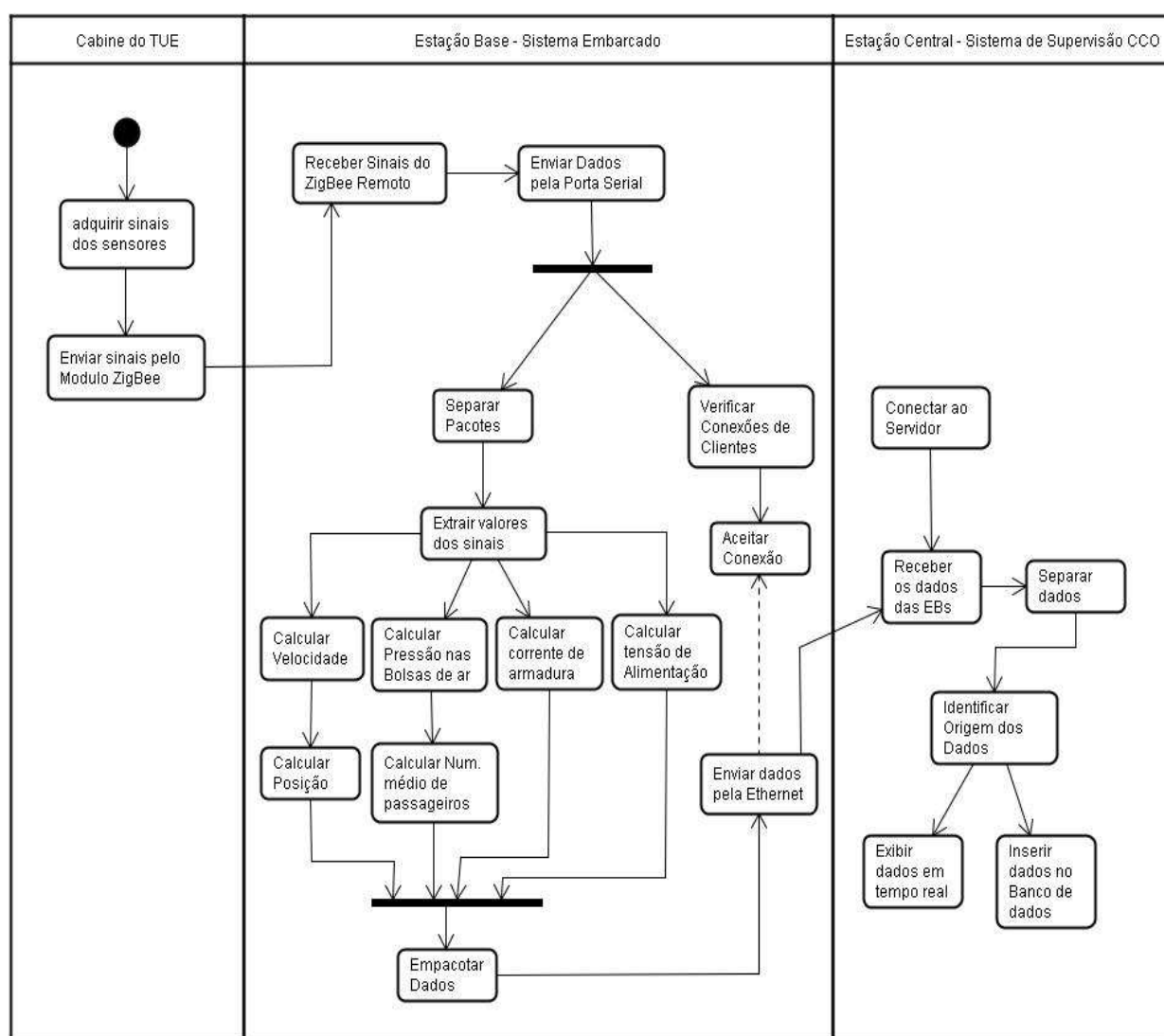


Figura 4.11: Diagrama de Atividades do sistema de tratamento de dados

4.2.3.2.2. Diagrama de Componentes

Este diagrama mostra a organização entre arquivos de código fonte, bibliotecas, tabelas de banco de dados, etc. A relação mais usada é a dependência, mostrando como um arquivo de código fonte depende de outro que ele inclui, e como, por exemplo, um executável depende de uma biblioteca. Um componente é uma parte física do sistema. Muitas vezes um componente mostra um arquivo específico do sistema.

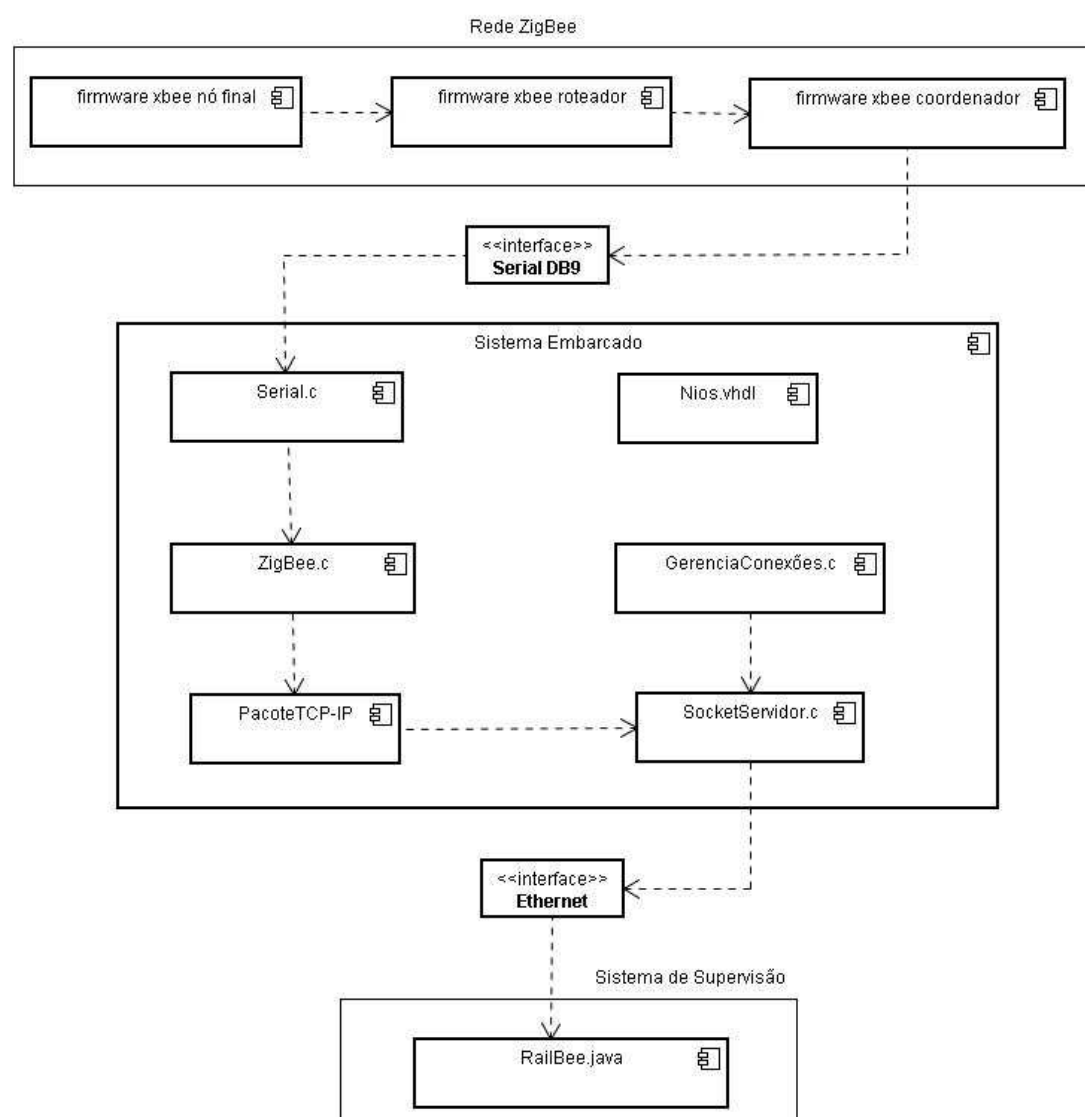


Figura 4.12: Diagrama de Componentes de software do sistema de tratamento de dados

4.2.3.2.3. Diagrama de Seqüência

Este diagrama fornece uma visão orientada para o tempo das interações, mais especificamente, de mensagens passadas entre objetos, para realizar um objetivo comportamental do sistema.

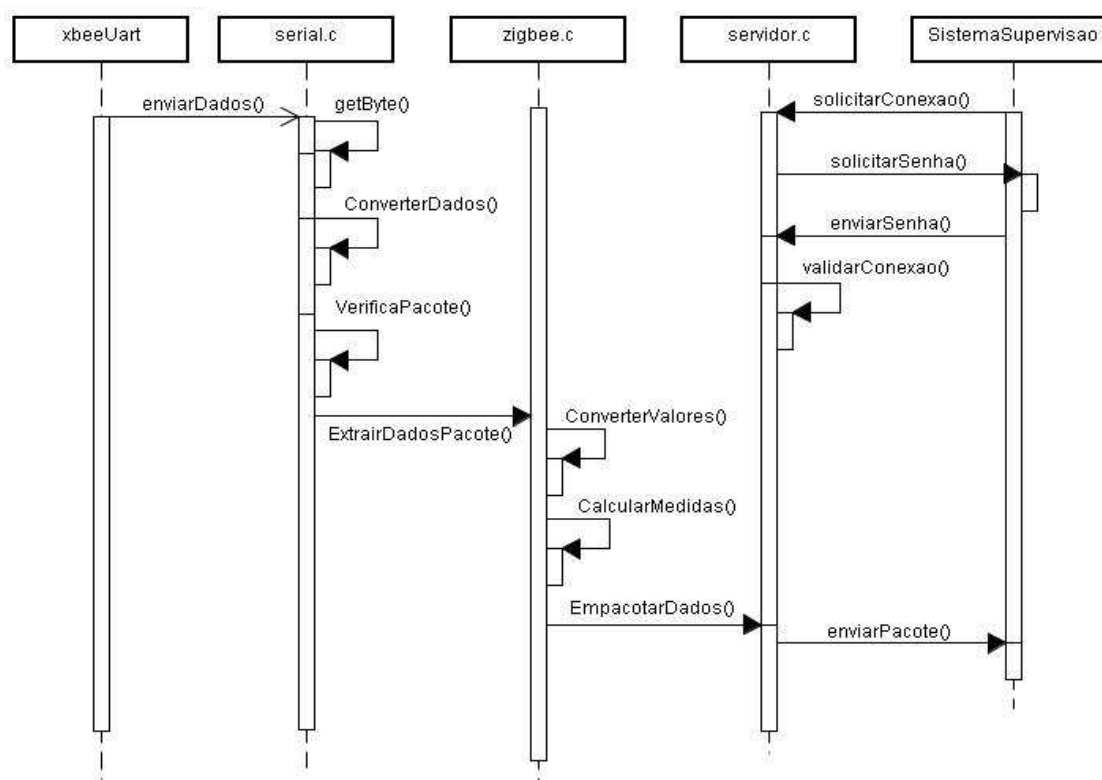


Figura 4.13: Diagrama de seqüência do sistema de tratamento de dados

4.2.4. Software de tratamento de dados

Para o tratamento das informações colhidas da rede ZigBee, foram desenvolvidos dois módulos de *software*: *serial.c* e *zigbee.c*. O módulo *serial.c* contém a implementação de funções para recuperação dos dados da serial e funções para a conversão e verificação destes dados.

Os dados são extraídos do buffer da UART *byte a byte*. Cada *byte* é extraído como um valor inteiro. Este valor é então convertido para o seu valor hexadecimal correspondente para permitir a verificação do pacote. Esta verificação consiste em analisar o cabeçalho do pacote, permitindo assim que eles sejam separados e analisar o tamanho de cada pacote

para verificar se houve ou não perda de algum *byte*. É possível ainda fazer uma verificação de cada pacote através do cálculo do *checksum* (que consiste na soma de alguns de seus dados) e comparar o resultado com o *checksum* contido no pacote para detectar se houve dados corrompidos na transmissão.

Cada pacote ZigBee contém uma série de informações que não são necessários para o sistema, desta forma, para diminuir o tamanho do pacote a ser enviado para o sistema de supervisão, o sistema embarcado extrai dos pacotes ZigBee apenas os dados necessários como o endereço de 64 bits do nó final e os valores dos canais analógicos habilitados. Os valores dos canais analógicos são processados pelo sistema para fornecer os valores das medidas realizadas pelos sensores como: velocidade, pressão, corrente e tensão.

Pelos dados de velocidade e pressão, o sistema pode calcular a posição aproximada do trem identificado pelo endereço da estação Móvel (endereço de 64 bits do nó final) e realizar uma estimativa do número de passageiros dentro do trem.

A figura 4.14 demonstra um esquema do processamento e transformação de um pacote ZigBee em um pacote RailBee, que é o pacote resultante do processamento realizado pelo sistema embarcado. Os dados destacados em azul no pacote ZigBee correspondem ao endereço da EM e aos quatro canais analógicos habilitados. Os dados destacados em azul no Pacote RailBee correspondem ao endereço da EM extraído do pacote ZigBee e aos valores das medidas obtidas no canais analógicos da EM. A tabela 4.2 apresenta a descrição de cada campo do pacote ZigBee e a tabela 4.3 apresenta os campos que compõem o pacote RailBee.

A seguir são apresentados os protótipos e descrição das principais funções implementadas para o sistema de tratamento.

void uartTask();

Esta função corresponde a tarefa do sistema responsável pelo processamento dos dados recebidos na porta serial. Dentro desta função, serão chamadas todas as funções necessárias para o tratamento dos pacotes ZigBee. É a tarefa que possui a maior prioridade de execução.

void converteParaHexa(unsigned char, char*);

Esta função converterá cada *byte* extraído da porta serial para o seu correspondente valor em hexadecimal. A função recebe como parâmetro o *byte* lido da porta serial e um arranjo no qual o valor convertido será atribuído.

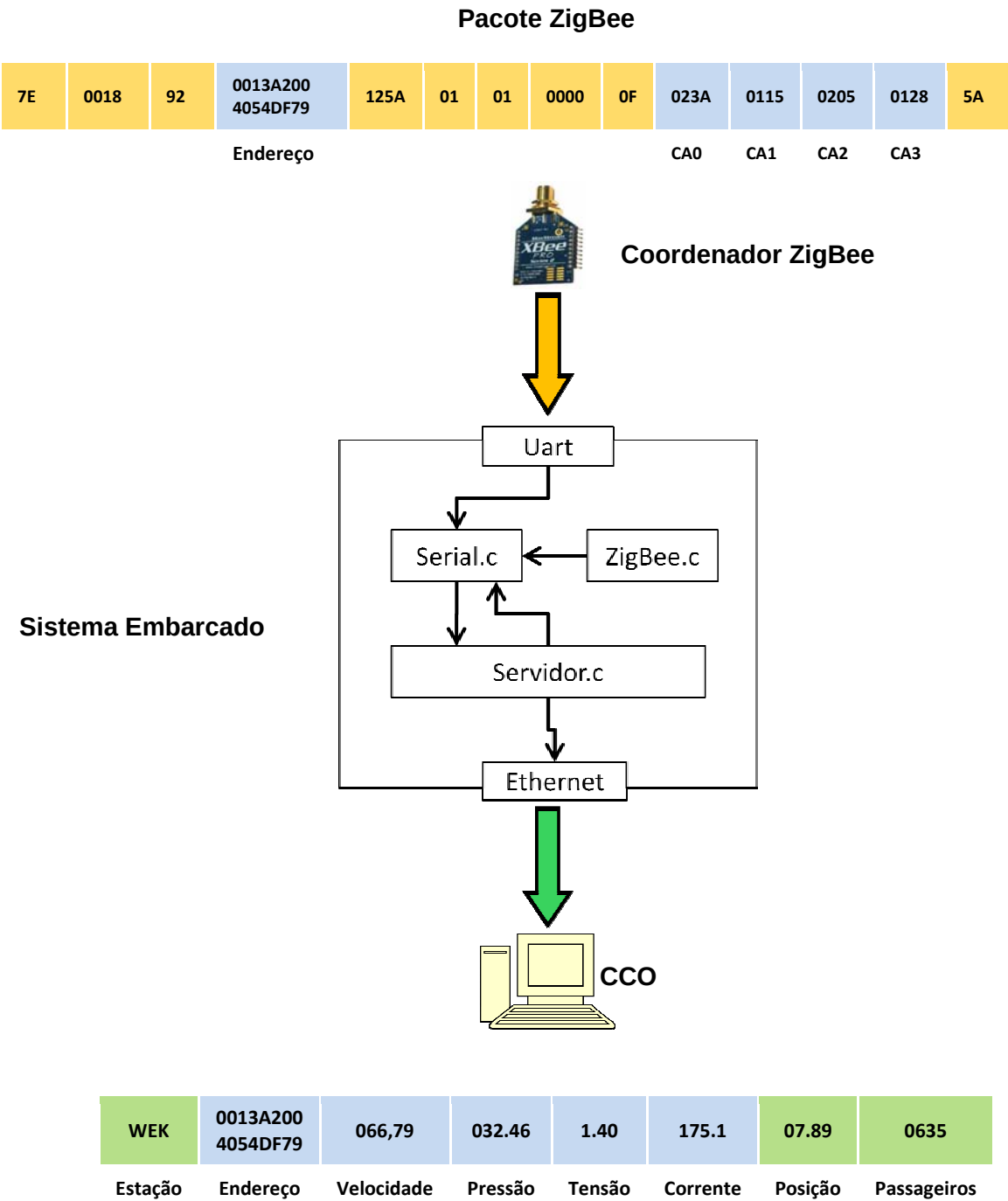


Figura 4.14: Processamento e transformação de um pacote ZigBee em um Pacote RailBee

7E	Delimitador de início de pacote
0018	Quantidade de <i>bytes</i> de um pacote, excluindo o <i>byte</i> de <i>checksum</i> , o delimitador de início de pacote e os <i>bytes</i> deste campo.
92	Identificador de operação da API (ZigBee <i>IO Data Sample Rx Indicator</i>). Ao receber um pacote com este identificador, o coordenador envia os dados recebidos para a porta serial do dispositivo.
0013A200 4054DF79	Endereço de 64 bits do dispositivo. Cada dispositivo possui um endereço de 64 bits que já vem configurado de fábrica, não podendo ser modificado
125A	Endereço de 16 bits. Este endereço é atribuído dinamicamente e aleatoriamente no momento da inicialização dos dispositivos. É usado para comunicação entre os nós.
01	Opção de recebimento de Pacote. Este <i>byte</i> indica se o pacote recebido é um pacote reconhecido(01) ou se é um pacote em broadcast.(02)
01	Este campo indica o número de amostras contidas no pacote.
0000	Este campo indica quais canais digitais estão habilitadas no dispositivo remoto. O valor 0000 indica que não há entradas analógicas habilitadas.
0F	Indica quais canais analógicos do dispositivo remoto estão habilitados. O valor 0F indica que os quatro canais analógicos estão habilitados
023A	Valor recebido na entrada analógica do canal 0. Este valor representa o valor analógico 0,6679, referente a tensão do sensor, convertido para digital no formato hexadecimal.
0115	Valor recebido na entrada analógica do canal 1. Este valor representa o valor analógico 0,3246, referente a tensão do sensor, convertido para digital no formato hexadecimal.
0205	Valor recebido na entrada analógica do canal 2. Este valor representa o valor analógico 0,6058 convertido para digital no formato hexadecimal.
0128	Valor recebido na entrada analógica do canal 3. Este valor representa o valor analógico 0,3757 convertido para digital no formato hexadecimal.
5A	<i>Checksum</i> dos dados do pacote

Tabela 4.2: Descrição dos campos de um pacote ZigBee.

WEK (Estação)	Corresponde à sigla da estação base que coletou e processou os dados do pacote ZigBee recebido. Este campo é constituído por 3 caracteres, como por exemplo: WEK, que indica que os dados contidos neste pacote foram coletados de trens nas proximidades da estação Werneck.
0013A2004054DF79 (Endereço do Xbee de Origem.)	Este campo contém o endereço de uma estação móvel instalada dentro dos trens e serve para identificar de qual trem os dados são providos. É formado por 16 caracteres que representam o endereço de um nó final pré-configurado. Exemplo: 0013A2004054DF79.
066,79 (Velocidade – Km/h)	Este campo corresponde ao cálculo da velocidade do trem. O valor pode variar de 000.00 a 100.00 Km/h
032,46 (Pressão – psi)	Este campo corresponde ao cálculo da pressão nas bolsas de ar e pode ser utilizado para calcular o peso do trem e estimar a quantidade de passageiros dentro de cada trem. O valor pode variar de 000.00 a 100.00 psi
1,40 (Tensão – Kv)	Este campo corresponde à medida da tensão de alimentação do trem. O valor pode variar de 0.00 a 3.00 Kv
175,1 (Corrente – A)	Corresponde à medida da corrente de armadura do trem. O valor pode variar de 000,00 a 300,00 A
07.89 (Posição – Km)	Este campo contém o valor resultante do cálculo da posição para o trem especificado pelo endereço de origem. Utiliza os valores da posição e velocidade do pacote anterior do mesmo trem, que foram previamente calculados e armazenados no sistema embarcado. A distancia varia entre 0 e 16,000 Km
0635 (Estimativa de Passageiros)	Representa o valor resultante do cálculo da estimativa de passageiros dentro do trem correspondente. Este cálculo é baseado na medida da pressão. A lotação máxima de cada trem é de aproximadamente 1500 passageiros.

Tabela 4.3: Descrição dos campos do pacote RailBee gerado pelo sistema de tratamento

char converteParaChar(int);

Função para converter um valor inteiro em um caractere que representa um dígito hexadecimal. É utilizada dentro da função toHexa.

int toDecimal(char);

Converte um digito hexadecimal em seu correspondente valor em decimal. É utilizada em conjunto com a função obterValoresCanais.

int detectaCabeçalho(char*, char*);

Esta função verifica os *bytes* lidos da porta serial para identificar o começo de um pacote. É utilizada pela *uartTask* para separar os pacotes ZigBee.

int verificaPacote(char*);

Esta função verifica se o pacote ZigBee extraído contém todos os devidos *bytes* e realiza a verificação do *checksum* dos dados.

Após a extração e verificação de um pacote, o sistema utiliza as funções definidas no módulo *zigbee.c* para obtenção das informações necessárias. De cada pacote são extraídos o endereço da estação móvel que o enviou, e os valores dos canais analógicos. Estes valores correspondem às medidas realizadas pelos sensores na cabine do trem onde a EM ZigBee estava acoplada. Através do endereço desta EM é possível identificar de qual trem os dados estão vindo.

Os valores dos canais analógicos (que chegam ao sistema como valores digitais) são convertidos do formato hexadecimal para o valor decimal correspondente a tensão medida pelo sensor e então armazenados em variáveis utilizadas nos cálculos de velocidade, pressão, corrente de armadura e tensão de alimentação. Os dados de velocidade e pressão calculados são então utilizados para determinação da posição do trem e para estimativa do número médio de passageiros presentes no trem em determinado momento. As principais funções que serão desenvolvidas para este sistema serão:

void obterEndereço(char*, char*);

Função que extrai o endereço do nó final do pacote ZigBee e o armazena em um *array*. Este endereço é utilizado pelo sistema de supervisão para identificar de qual trem foram obtidos os dados.

void obterValoresCanais(char*, float*);

Esta função recebe como entrada um pacote ZigBee, calcula os valores de tensão de cada canal analógico presente no pacote e os armazena em um *array* passado como parâmetro para função. O valor de cada canal varia de 0 a 1,2 V.

As quatro funções definidas abaixo realizam os cálculos das medidas de Velocidade, Pressão, Tensão e Corrente. Elas recebem como entrada o valor do canal referente ao

sensor da medida a ser calculada e um fator de ponderação para o cálculo, retornando o valor calculado.

```
float calcularVelocidade(float, float);  
float calcularPressão(float, float);  
float calcularTensão(float, float);  
float calcularCorrente(float, float);
```

```
float calcularPosicao(float, float);
```

Com base nas medidas de velocidade calculadas, esta função calcula a posição de cada trem na via. Os dois parâmetros de entrada da função representam a velocidade anterior e atual de um determinado trem.

```
float mediaPassageiros(float, float);
```

Recebe como entrada a pressão calculada e um fator de ponderação que varia dependendo do trem ao qual a pressão se refere. Esta função calcular uma estimativa do número de passageiros dentro de cada trem no momento em que eles partem das estações.

```
void pacoteRailbee(char *);
```

Ao final de todos os cálculos, as medidas calculadas serão, através desta função, organizadas e colocadas em uma string para serem enviadas ao sistema de supervisão.

Todos os dados calculados são então colocados em um novo pacote, seguindo um esquema como mostrado na figura 4.14, definido para a troca de informações entre o sistema servidor e os sistemas clientes (Sistema de supervisão instalado no CCO e CIM). Assim que processados, estes pacotes são enviados imediatamente para o sistema de supervisão.

4.2.5. Software servidor de dados

O outro módulo que compõem o sistema é o módulo servidor de dados – servidor.c. Ele é composto de funções para aceitar e validar conexões, receber dados enviados pelo sistema de supervisão e enviar os pacotes com os dados processados pelo módulo de tratamento ao sistema de supervisão.

Este módulo é executado paralelamente ao módulo de tratamento. Ele é capaz de receber solicitações de conexão a qualquer momento, verificando e validando estas conexões através de uma senha pré-definida no sistema. O Sistema é capaz de aceitar até 16 conexões, entretanto, inicialmente, o sistema só permitirá duas conexões simultâneas correspondentes ao sistema de supervisão instalado no CCO e na CM.

Após estabelecidas as conexões com o sistema do CCO e/ou CM, os dados tratados pelo módulo de tratamento podem ser enviados a esses locais. A comunicação entre o sistema de tratamento e o Sistema de Supervisão é feita através do protocolo TCP/IP usando a pilha de protocolo NicheStack. Para cada conexão foi criada uma tarefa responsável por gerenciar e enviar os dados para o sistema cliente solicitante.

O sistema servidor de dados além de enviar os dados processados, pode receber e efetuar alguns comandos enviados pelo sistema de supervisão como: solicitação para encerrar o envio de dados ou para recomençar, desconexão do sistema, dentre outras ações semelhantes. Abaixo seguem os protótipos e uma pequena descrição de cada função do sistema servidor.

void servidorTask();

Esta função corresponde a tarefa do sistema responsável pelo processamento das conexões do sistema de supervisão. Dentro desta função, serão chamadas todas as funções necessárias para prover a comunicação através da rede ethernet.

void aceitarConexao(int, Con*);

Processa os pedidos de conexão vindos do sistema de supervisão. Para se conectar, o sistema de supervisão envia para o servidor uma senha que é analisada através desta função para validar a conexão.

void receberDados(Con*);

O sistema de supervisão pode enviar alguns dados ou instruções para o servidor. Esta função recebe esses dados e dependendo do tipo de dado enviado chama funções específicas para executar determinadas ações.

void executarComando(Con*);

Executa instruções enviadas pelo sistema de supervisão como: conectar servidor, parar envio dos dados, reiniciar envio ou encerrar conexão. Esta função é usada em conjunto com a função receberDados.

void enviarDados(Con*, char*);

Envia os pacotes com todos os dados calculados para o sistema de supervisão.

void encerrarConexao(Con*);

Função para encerrar as conexões com o sistema de supervisão.

4.3. Sistema de Supervisão

Um sistema supervisor permite que sejam monitoradas e rastreadas informações de um processo produtivo ou instalação física. Tais informações são coletadas através de equipamentos de aquisição de dados e, em seguida, manipuladas, analisadas, armazenadas e posteriormente apresentadas ao usuário. Estes sistemas também são conhecidos como SCADA (*Supervisory Control and Data Aquisition*). O objetivo deste sistema é propiciar uma interface de alto nível do operador com o processo, informando-o em tempo real de todos os eventos de importância da planta (Santos, 2007).

O sistema de supervisão RailBee apresentado neste trabalho, fica instalado em um PC no Centro de Controle Operacional localizado próximo a estação Werneck e também no Centro de Informação e Manutenção. No CCO trabalham os controladores de tráfego e engenheiros responsáveis por gerenciar o tráfego dos trens em todas as linhas do MetroRec. O sistema de supervisão desenvolvido proporciona ao controlador uma visão do funcionamento e movimento de cada trem em operação em tempo real, facilitando a tomada de decisões. Ele recebe os dados enviados pela intranet de todas as EB, extrai os dados referentes às medidas obtidas e mostra em tempo real estas informações, possibilitando ao controlador de tráfego ter a mesma visão que o operador do trem na cabine.

O RailBee recebe as medidas de velocidade, pressão nas bolsas de ar, número estimado de pessoas, posição, tensão de alimentação e corrente de armadura de cada trem em movimento nas vias. Todas estas medidas já foram previamente calculadas nas EB, liberando o sistema do processamento destes dados e deixando-o responsável pelo gerenciamento do armazenamento e visualização dos dados recebidos.

O RailBee armazena os dados recebidos em um banco de dados e apresenta diferentes opções de interfaces que permitem ao usuário visualizar as medidas de todos os trens simultaneamente, ou para cada trem especificamente. Este foi inicialmente desenvolvido usando o Ambiente de programação LabVIEW.

Sabendo da velocidade e da localização, o controlador de tráfego pode se comunicar com o operador de cada trem e solicitar o aumento ou diminuição da velocidade a fim de agilizar o tráfego e reduzir o tempo de espera dos passageiros. É possível ainda monitorar

sinais de alerta ou emergência, como por exemplo: excesso de velocidade, aproximação perigosa, parada em local indevido ou tempo de parada na estação excedido.

Através da medida da pressão nas bolsas de ar, o controlador de tráfego o sistema informa os horários e locais em que ocorre superlotação dos trens. Isto permite a tomada de ações que visem otimizar o tráfego desses trens para evitar a superlotação, aumentando assim o conforto dos passageiros.

O sistema exibe gráficos dos dados obtidos em tempo real, como também dos dados armazenados no banco de dados permitindo aos engenheiros realizar estudos e gerar relatórios que ajudam a tomar medidas preventivas e corretivas.

A seguir são apresentados os requisitos deste sistema identificados através de um diagrama de casos de uso e sua estrutura é demonstrada por meio de um diagrama de classes. O último tópico desta seção apresenta o sistema através das interfaces desenvolvidas.

4.3.1. Análise e projeto do sistema de supervisão

Este tópico apresenta a descrição do projeto especificado para o sistema de supervisão através de diagramas UML. Os requisitos do sistema de supervisão foram descritos através da modelagem e descrição dos casos de uso do sistema que são apresentados no tópico seguinte. Foram elaborados um diagrama de casos de uso que apresenta as interações dos usuários com o sistema e um diagrama de classes que representa a estrutura interna do *software*.

4.3.1.1. Diagrama de Casos de Uso

O diagrama de Casos de Uso é usado para modelar o modo como as pessoas esperam usar um sistema. O diagrama descreve quem serão os usuários relevantes e os serviços que eles exigem do sistema. O diagrama de casos de uso pode ser usado para identificar os requisitos funcionais do sistema.

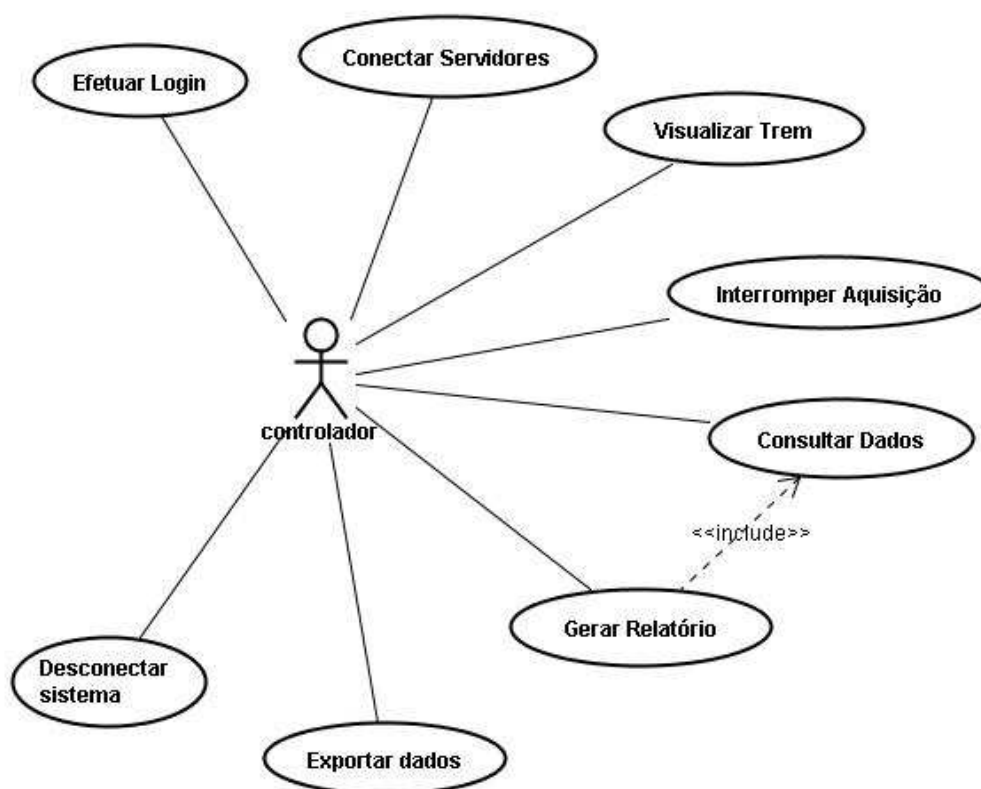


Figura 4.15: Diagrama de casos de uso do sistema de supervisão.

Cada caso de uso do diagrama mostrado acima apresenta um requisito funcional do sistema supervisão.

Efetuar login: O sistema deve permitir apenas o acesso de um usuário autorizado.

Conectar Servidores: Ao se logar no sistema, o controlador no CCO, pode conectar o sistema aos servidores embarcados nas EB instaladas na via para receber os dados processados.

Visualizar trem: A interface principal do sistema de supervisão consiste em uma tabela mostrando os dados de todos os trens simultaneamente. O sistema deve possibilitar ao controlador visualizar os dados de um trem específico em uma interface a parte.

Interromper Aquisição: O controlador poderá interromper o recebimento dos dados dos servidores. Neste caso, o sistema de supervisão envia um comando para o sistema servidor embarcado que interrompe a tarefa responsável pelo envio dos dados.

Consultar Dados: O sistema deve propiciar ao usuário diferentes formas de consultas sobre os dados armazenados no banco de dados. Esta consulta pode ser na forma de gráfico ou de tabelas.

Gerar Relatório: A partir de uma consulta realizada pelo usuário, o sistema poderá gerar um relatório sobre os dados consultados.

Exportar Dados: O sistema deve possibilitar a exportação dos dados do banco para arquivos permitindo ao usuário utilizar os dados em outros programas ou realizar backup dos dados.

Desconectar Sistema: O usuário poderá desconectar o sistema dos servidores caso seja necessário.

4.3.1.2. Diagrama de Classes

O diagrama de classes modela os recursos usados para montar e operar o sistema em termos de sua estrutura, relacionamentos e comportamentos. Ele serve como base para orientar a implementação do sistema.

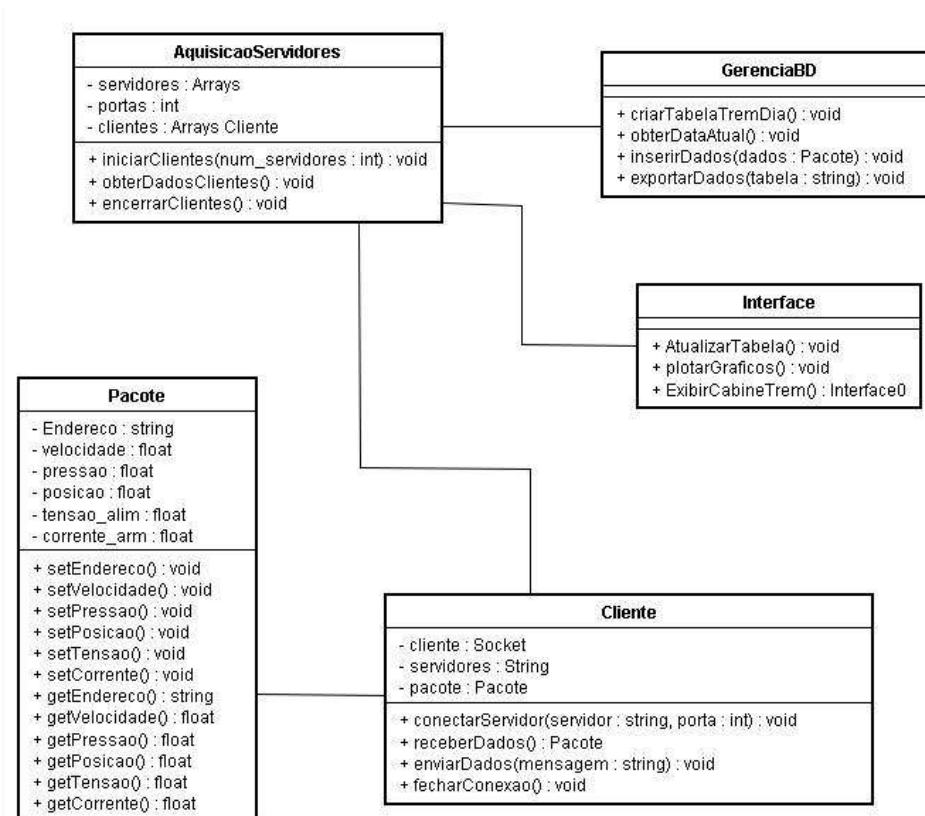


Figura 4.16: Diagrama de classes do sistema de supervisão.

O diagrama acima demonstra as classes projetadas para o sistema. Cada classe apresenta um conjunto de atributos e métodos. A classe Cliente implementa os métodos para conexão com os servidores, para recepção dos dados processados e para envio de dados para o servidor. Esta classe é instanciada dentro da classe principal do programa, a classe AquisiçãoServidores. A classe AquisiçãoServidores define os métodos IniciarCliente e ObterDadosCliente que evocam os métodos do objeto Cliente para se conectarem aos servidores e iniciar a aquisição dos dados.

Os dados recebidos do servidor vêm em pacotes como mostrado na Figura 4.14. O método recebeDados da classe Cliente recebe os pacotes enviados pelo servidor, separa cada dado (velocidade, pressão, tensão e corrente) e atribui estes dados aos atributos de um objeto da classe Pacote. O objeto Pacote contém um atributo para cada campo extraído do pacote e disponibiliza métodos *set* e *get* para que outros objetos possam alterar e recuperar os dados armazenados nele. Esse objeto é passado como parâmetro para o método que realiza a inserção dos dados no Banco de Dados na classe GerenciarBD, que é invocado dentro do método obterDados da classe principal, e também é utilizado para atualizar os dados exibidos na interface principal do programa.

4.3.2. Desenvolvimento do sistema de supervisão

O desenvolvimento do sistema de supervisão se baseou na definição dos casos de uso, que mostraram as necessidades dos controladores de tráfego e apontaram as funcionalidades que deveriam ser desenvolvidas. Dentre os requisitos do sistema, os de maior prioridade consistiam na visualização simultânea dos dados em tempo real de todos os trens circulando nas vias e na visualização das informações de cada trem em particular, permitindo aos controladores saber o estado de funcionamento desses trens, bem como ter a visão das ações tomadas pelo operador na cabine do TUE.

Desta forma, o desenvolvimento do sistema se iniciou com a definição e construção das interfaces para apresentação das informações aos controladores de tráfego e posteriormente com a implementação do código para comunicação dos sistemas em rede e armazenamento no banco de dados.

Inicialmente, optou-se pelo desenvolvimento do sistema de supervisão utilizando-se o ambiente de programação LabVIEW 8.6. O LabVIEW agiliza a construção de interfaces pois disponibiliza para o desenvolvedor uma série de componentes prontos que podem ser facilmente combinados gerando rapidamente interfaces bastantes complexas, que demorariam muito mais tempo para serem implementadas através de linguagens de

programação baseadas em texto. A figura 4.17 apresenta uma tela com alguns dos componentes disponibilizados pelo LabVIEW.

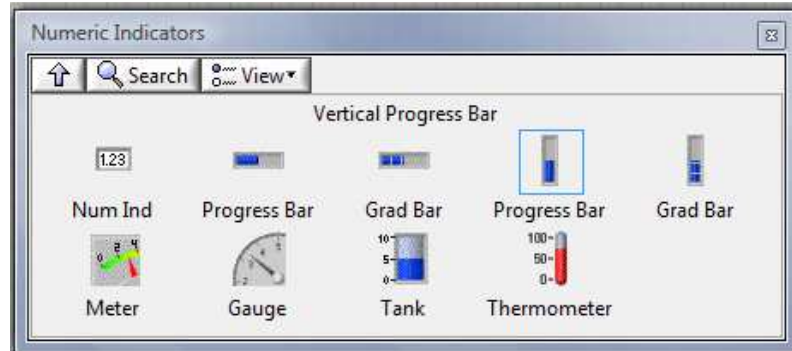


Figura 4.17: Componentes de interface do LabVIEW.

Além das facilidades para a construção das interfaces, o LabVIEW proporciona um ambiente de programação concorrente que permite o desenvolvimento simples de aplicações com vários processos sendo executados em paralelo e dispõe também de vários componentes para desenvolvimento de sistemas em rede.

Embora o LabVIEW possua diversas vantagens, ele apresenta algumas desvantagens que fizeram com que optássemos por substituí-lo por um outro ambiente de desenvolvimento. Além de ser uma ferramenta proprietária, o LabVIEW não tem, em sua versão original, ferramentas para suporte a banco de dados. A Nacional Instruments possui um kit de ferramentas para manipulação de banco de dados, mas ele não é disponibilizado gratuitamente, sendo necessário comprar a licença para utilizá-lo.

Devido a isso, optou-se por desenvolver o sistema em linguagem Java utilizando a IDE NetBeans. Esta IDE pode ser baixada gratuitamente no site da Sun Microsystems e possui uma série de recursos avançados para o desenvolvimento de sistemas de diversos tipos.

4.3.3. Interfaces projetadas para o sistema

A figura 4.18 demonstra uma interface para testes desenvolvida para o sistema usando o LabVIEW. Esta interface tenta imitar o painel de controle real de um trem (figura 4.19) dando ao usuário a possibilidade de ter a mesma visão dos controles que o operador na cabine de um TUE. Além disto, ela fornece informações adicionais que não estão disponíveis aos operadores na cabine dos TUE, como informações sobre quantidade de passageiros e gráficos das medidas realizadas.

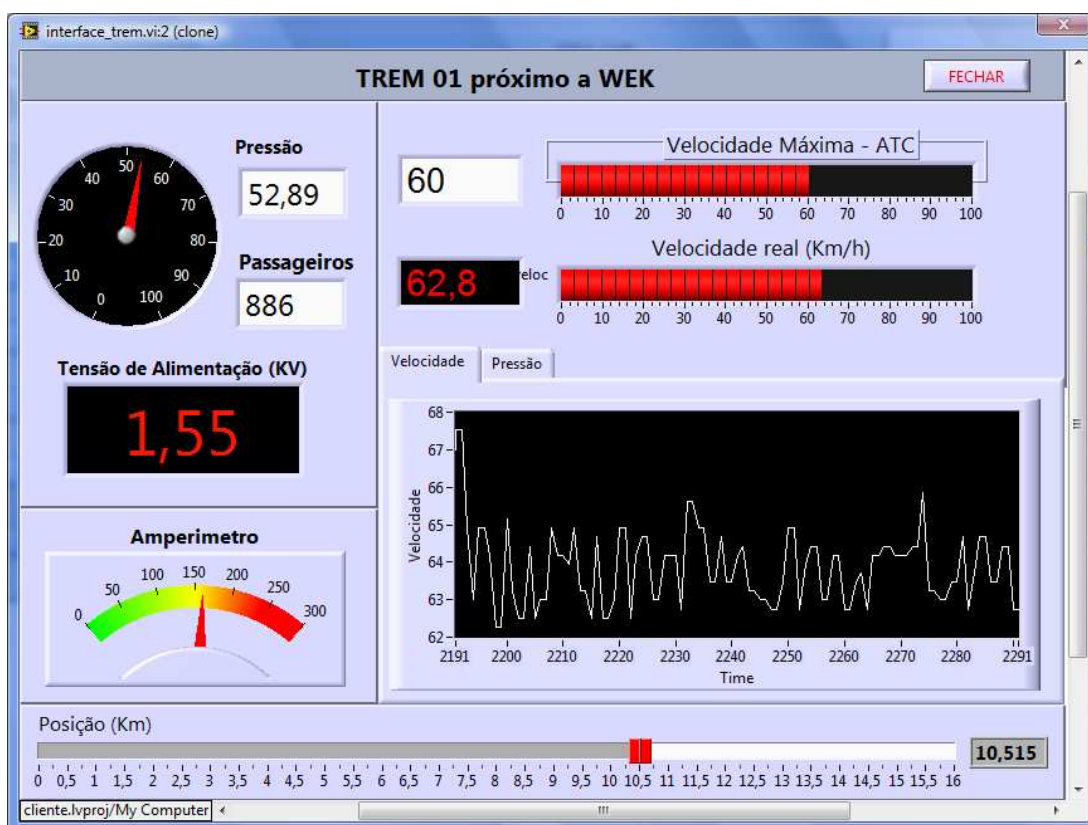


Figura 4.18: Interface do sistema para visualização dos dados de um trem



Figura 4.19: Painel de controle de um TUE.

Através desta interface o controlador de tráfego consegue visualizar o funcionamento de um trem específico, podendo avaliar o desempenho do trem e de seu operador. A interface é composta por um amperímetro para medida da corrente de armadura, um componente para medida da pressão nas bolsas de ar, sendo exibido em um campo o número de passageiros calculado com base no valor da pressão medida, duas barras de progresso que demonstram a velocidade real e a velocidade máxima permitida, possui ainda uma barra de progresso para indicar a posição do trem e gráficos para as medidas de velocidade e pressão.

Após os testes com o sistema desenvolvido com o LabVIEW, iniciou-se a implementação do sistema final em linguagem Java usando a IDE NetBeans. A figura 4.20 apresenta a interface inicial deste sistema.



Figura 4.20: Interface inicial do sistema RailBee.

Para visualização dos dados de todos os trens em movimento, o sistema dispõe de uma interface composta por uma tabela onde são exibidas as informações referentes às medidas de cada trem como mostrado na figura 4.21. Ao lado de cada linha da tabela há um botão que ao ser clicado abre a interface mostrada na figura 4.22 para visualização dos dados de um trem individualmente. A interface apresentada na figura 4.22 é semelhante a interface projetada em LabVIEW mas foi desenvolvida em Java com os componentes gráficos de uma API denominada *JFreeChart* que pode ser encontrada facilmente para

download na internet. Desta forma, foi possível desenvolver um sistema com as interfaces requeridas usando ferramentas totalmente gratuitas.

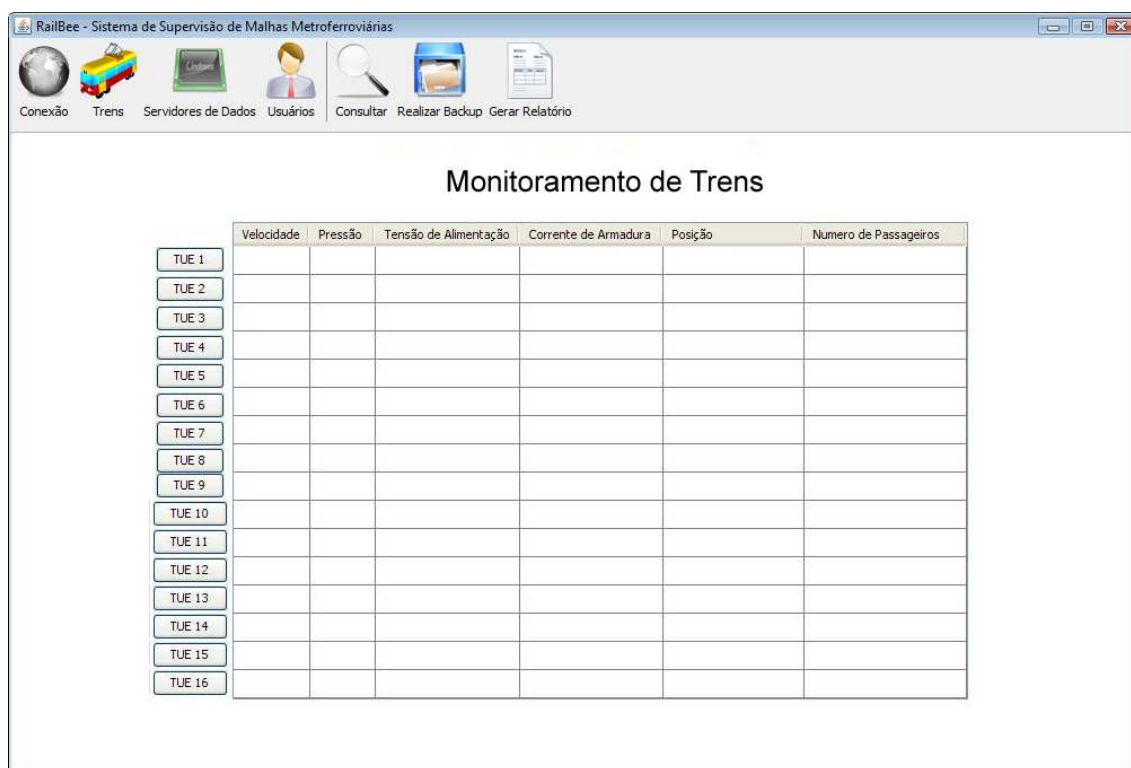


Figura 4.21: Interface para visualização dos dados de vários trens simultaneamente.

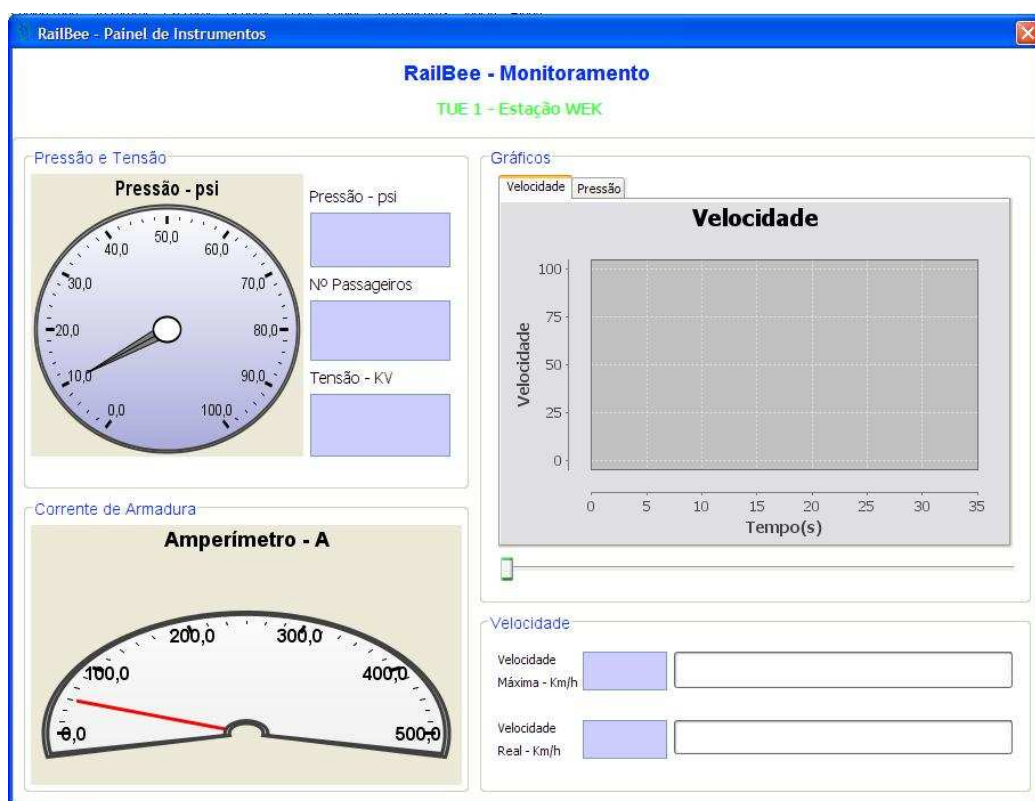


Figura 4.22: Interface para visualização dos dados de um trem específico.

Além de exibir os dados coletados em tempo real, os dados coletados também são salvos em um banco de dados, onde existe uma tabela para armazenar as informações de cada trem separadamente. O banco também armazena informações sobre os servidores e módulos RF instalados nos trens. O sistema permite ao usuário realizar consulta sobre dados históricos possibilitando analisar e estudar o comportamento de cada trem durante seu funcionamento não só no momento atual, mas também durante longos períodos anteriores. Com base na análise desses dados históricos, pode se gerar estatísticas que indicarão ações a serem tomadas.

A figura 4.23 apresenta uma interface projetada para realização de algumas consultas. Na interface mostrada, o usuário escolhe o trem que deseja consultar, digita a data que desejada e a hora inicial e final para o filtro das informações. O usuário pode definir também que parâmetros farão parte da consulta, marcando nas *checkboxs* os parâmetros escolhidos. Os dados retornados da consulta são exibidos em gráficos, sendo gerado um gráfico para cada parâmetro escolhido. A interface oferece ainda a opção de visualização dos dados em tabelas e possibilita a exportação destes dados no formato de planilha com as extensões txt e xls, possibilitando sua manipulação por qualquer *software* processador de planilhas.

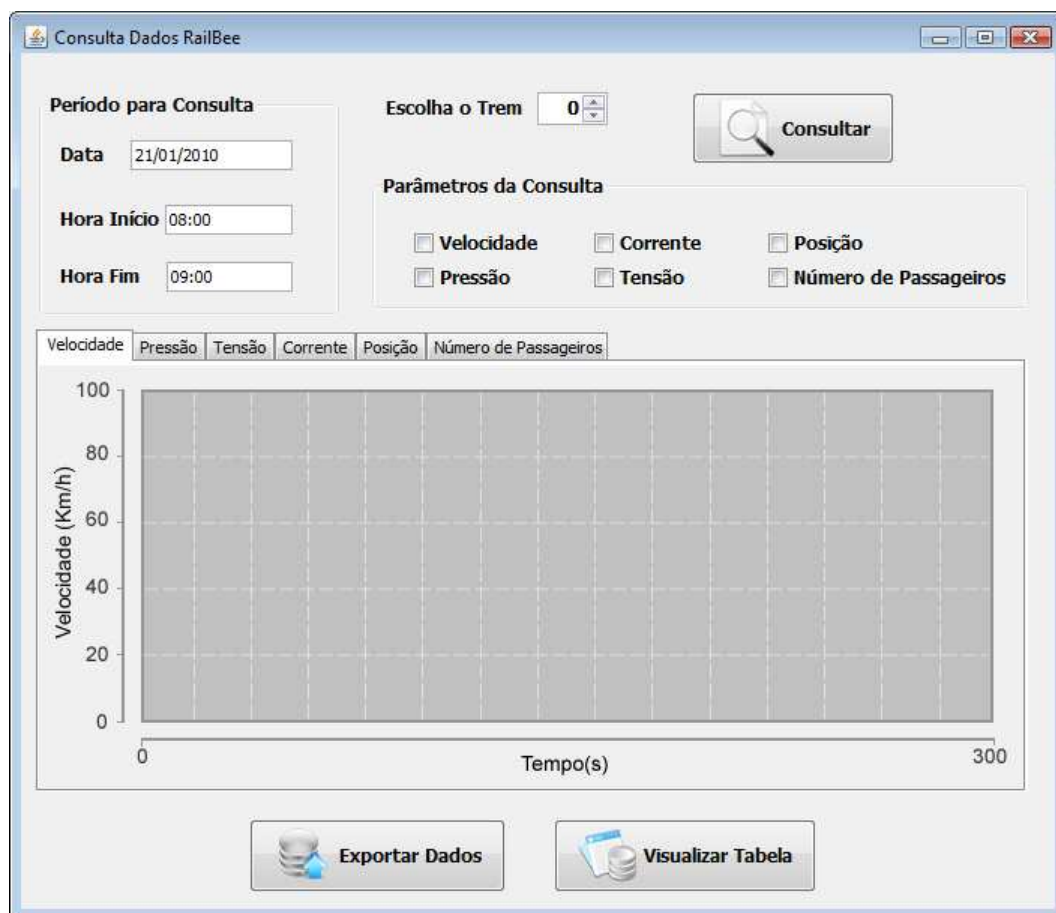


Figura 4.23: Interface do sistema de supervisão para consulta de dados

CAPÍTULO 5

Testes e resultados obtidos

Este capítulo descreve alguns dos trabalhos realizados no desenvolvimento do projeto e algumas etapas de testes que foram efetuadas para instalação das redes ZigBee, do sistema embarcado e de supervisão, realizando aquisição de dados para análise e validação das medidas obtidas. Após o desenvolvimento dos três subsistemas que compõem o sistema RailBee e de vários testes realizados em laboratório, um protótipo da rede foi instalada em campo para realização de testes no ambiente real em que o sistema será implantado.

Os tópicos a seguir demonstram como foi montada a rede para testes em um trecho da via e apresentam os resultados do monitoramento dos trens nesse trecho.

5.1 Instalação da rede para testes.

Para validar a telemetria via ZigBee, testar o processamento dos dados pelo sistema embarcado e o sistema de supervisão, foi implantada uma pequena rede ZigBee para monitorar o trecho da via entre as estações de passageiros Werneck(WEK) e Santa Luzia(LUZ) situadas na linha Centro do MetroRec. A figura 5.1 apresenta uma visão aérea do trecho da via onde foram realizados os testes.

A rede instalada constituía-se de uma EB, uma ER e uma EM. A estação base composta pelo sistema embarcado e pelo nó coordenador foram instalados na extremidade Oeste da plataforma da estação Werneck. O sistema embarcado e o nó coordenador foram colocados cada um em uma caixa preta (figura 5.2), estando conectados através de um cabo serial. Um cabo ethernet foi levado até o sistema embarcado conectando-o a rede ethernet local para comunicação com o sistema de supervisão.

A ER foi instalada na extremidade leste da plataforma da estação Werneck a uma distância de aproximadamente 60 metros da EB, aumentando o alcance da rede no lado leste da plataforma, em direção a estação Santa Luzia. A EM foi instalada no armário da cabine de Comando do TUE 11, no local mostrado pela figura 5.3. e 5.4. Em um computador

no CCO foi instalado o sistema de supervisão para recepção dos dados enviados pela EB montada na Estação Werneck.



Figura 5.1: Vista aérea do trecho da via onde a rede de testes foi implantada.

Dois canais analógicos do nó final foram conectados aos sensores de velocidade e pressão nas bolsas de ar que ficam na Unidade de Processamento Central no armário do TUE, como mostrado na figura 5.4. Devido a alguns problemas relacionados aos sensores de corrente de armadura e tensão de alimentação, estas medidas não foram realizadas nestes testes.

Os registros foram realizados em vários horários diferentes com o TUE 11. Para validação dos dados obtidos pela rede ZigBee, um oscilógrafo foi colocado dentro do trem para realizar a aquisição dos dados no mesmo tempo que a rede ZigBee. Como o oscilógrafo é um instrumento com alta precisão, os dados medidos por ele podem ser comparados com os dados obtidos pelos ZigBees determinando o nível de erro das medidas realizadas.

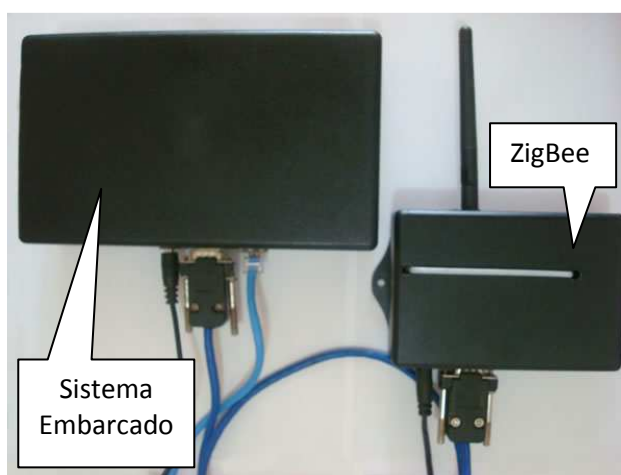


Figura 5.2: Estação Base que foi instalada na Estação Werneck



Figura 5.3: Local da instalação da EM dentro do TUE

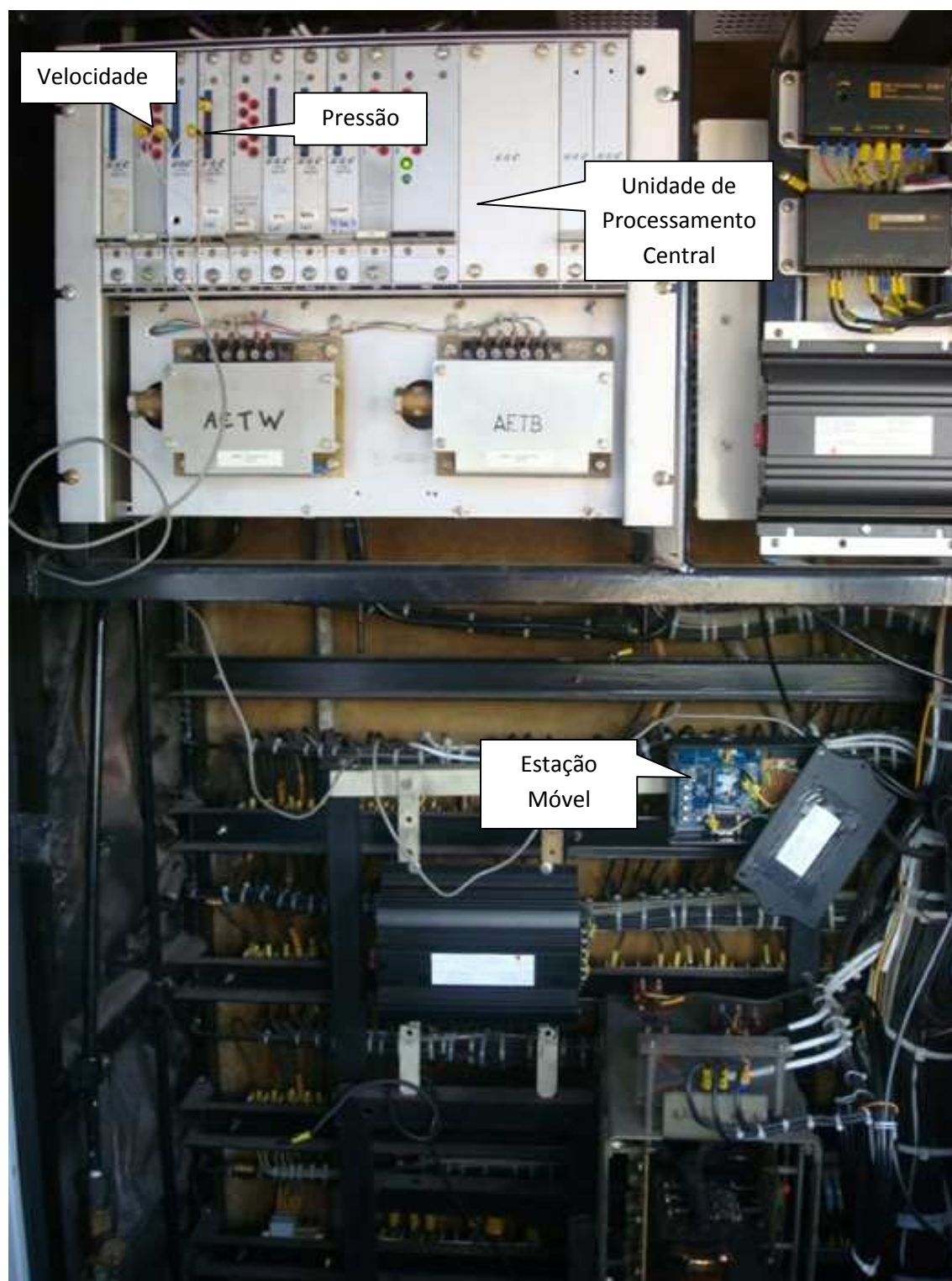


Figura 5.4: Nó final instalado na cabine do trem.

O módulo ZigBee configurado como nó roteador (ER), foi colocado em uma caixa e fixado na extremidade leste da plataforma, no ponto mostrado pela figura abaixo. A EB foi instalada na extremidade oposta a qual foi instalada a ER em um ponto semelhante ao mostrado nesta figura.



Figura 5.5: Local da instalação do roteador na cabeceira leste da Estação Werneck

5.2 Resultados

O sistema foi implantado e realizou a coleta de dados no trecho entre as estações Santa Luzia e Werneck (LUZ/WEK), com o TUE 11, no horário da operação comercial, em situação real de tráfego do trem, durante a primeira semana de dezembro de 2009.

Para validar os dados coletados pela rede ZigBee foi usado um Oscilógrafo conectado aos mesmos sensores que a EM, realizando as medidas simultaneamente e na mesma frequência. No CCO, o sistema de supervisão coletou as medidas enviadas pela EM do TUE 11 e o oscilógrafo coletou as medidas dentro do trem.

Após a coleta dos dados do TUE 11, os dados obtidos pela rede foram comparados aos dados do oscilógrafo. Na comparação das curvas geradas a partir dos dados registrados nos dois instrumentos, verificou-se uma semelhança na sua resposta dos

parâmetros velocidade real e pressão média das bolsas de ar. O fator de correlação das curvas foi 0,994 para velocidade e 0,992 para a pressão.

Através destes dados é possível definir uma fórmula para calibrar as medidas da rede ZigBee reduzindo possíveis erros.

O gráfico apresentado na figura 5.6 demonstra uma curva das medidas de velocidade do TUE 11 coletados pelo sistema no trecho no sentido de WEK a LUZ. Os dados foram coletados no dia 2 de dezembro de 2009, durante cerca de 3 minutos, das 09h08min as 09h11min. Já o gráfico da figura 5.7, apresenta a curva de velocidade deste mesmo trem na viagem de volta, no sentido de LUZ a WEK, às 10h57min.

Através do gráfico da figura 5.6 podemos verificar a área de abrangência do monitoramento pela rede sem fios. No trecho de WEK a LUZ os dados foram captados até uma distância de 1000 metros entre o roteador e o nó final na cabine do TUE 11. Este trecho da via é uma linha praticamente reta e sem obstáculos, como pode ser observado na figura 5.1, o que possibilitou um grande alcance entre os nós roteador e final. No trecho a oeste de WEK a distância máxima alcançada foi de cerca de 300 metros. A via a oeste da estação Werneck é sinuosa e possui alguns obstáculos que interferem na comunicação entre os nós da rede.

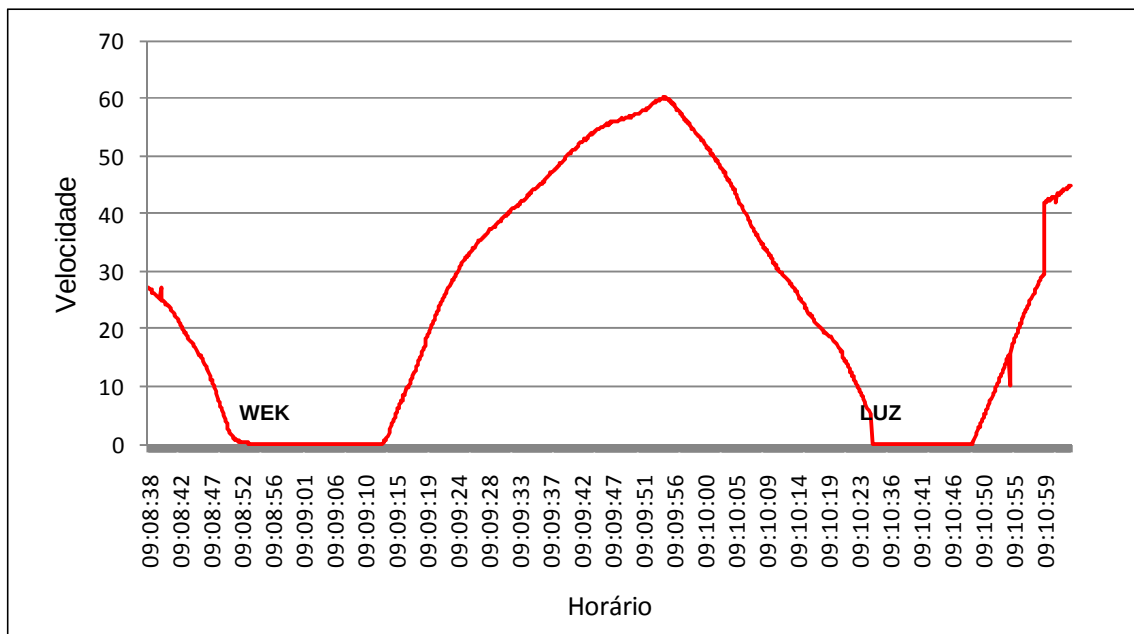


Figura 5.6: Gráfico da medida de velocidade do TUE 11 indo de WEK a LUZ.

Através da medida da velocidade nesse trecho é possível saber os picos de velocidade atingidos, o tempo em que o trem ficou parado na estação e a distância percorrida pelo trem.

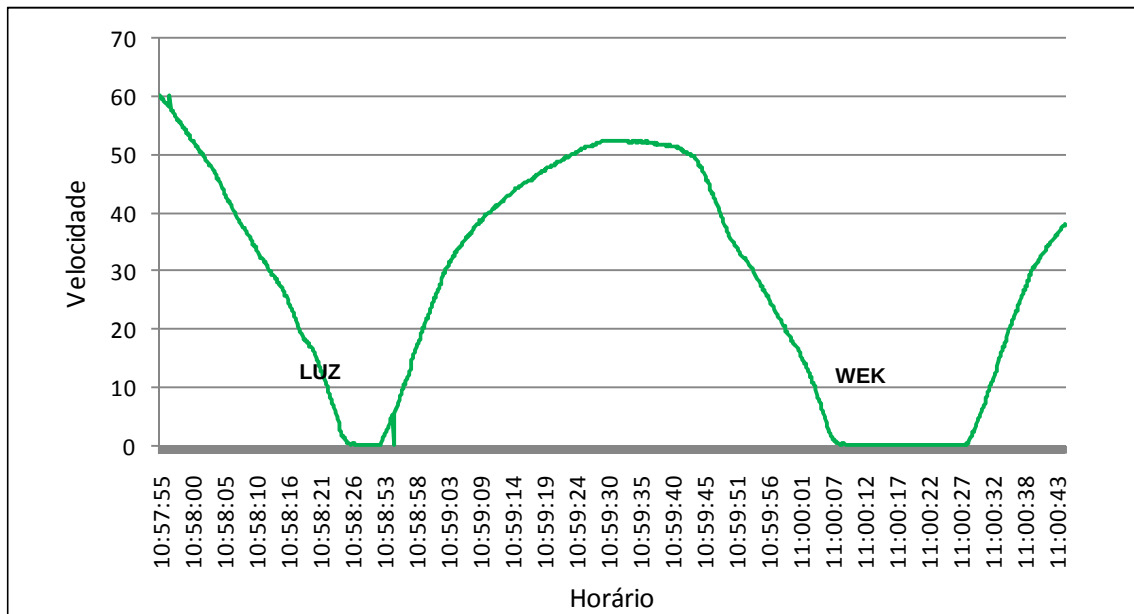


Figura 5.7: Gráfico da medida de velocidade do TUE 11 indo de LUZ a WEK.

Pela análise da variação da pressão nas bolsas de ar é possível saber se houve aumento ou diminuição de passageiros e estimar uma média para o número de passageiros presentes em cada trem. Como o trem sofre vibrações devido ao seu movimento e a movimentação dos passageiros dentro do trem, o sinal da pressão medido apresenta-se bastante ruidoso.

Para calcular o número médio de passageiros dentro do trem é necessário obter e identificar o valor da pressão nas bolsas de ar no momento em que o trem se encontra estável. Este momento ocorre quando ele se encontra parado nas estações de passageiros, com as portas fechadas e quando dá a partida. Neste momento o sistema embarcado da EB identifica a mudança de velocidade e realiza o cálculo do número de passageiros com o valor da pressão medido.

O gráfico da figura 5.8 demonstra a estimativa do número de passageiros do TUE 11 se movimentando na via 1, no sentido de LUZ a WEK. O gráfico da figura 5.9 apresenta a estimativa do número de passageiros com o trem vindo na via 2, sentido de WEK a LUZ. Através dos dois gráficos, podemos perceber um aumento no número de passageiros quando o trem sai da estação Werneck e uma diminuição quando sai da Estação Santa Luzia. Com o sistema instalado em toda a via, será possível identificar os pontos com maior lotação de passageiros para todas as estações, permitindo definir estratégias de operação por trecho.

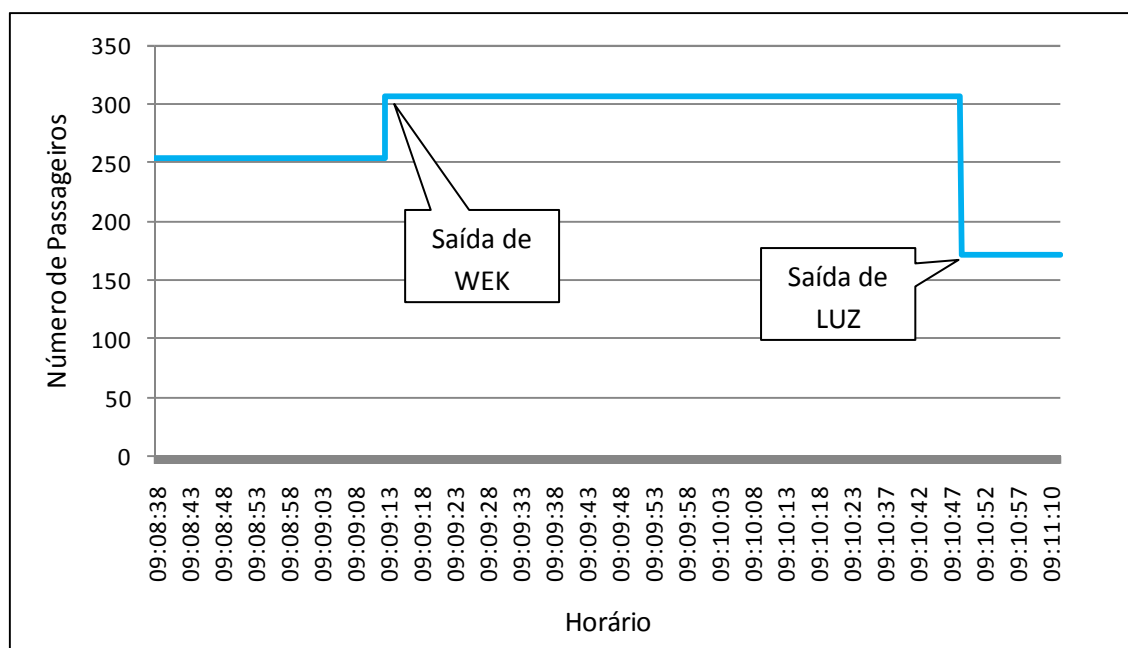


Figura 5.8: Gráfico do cálculo do número de passageiros do TUE 11 indo de WEK a LUZ.

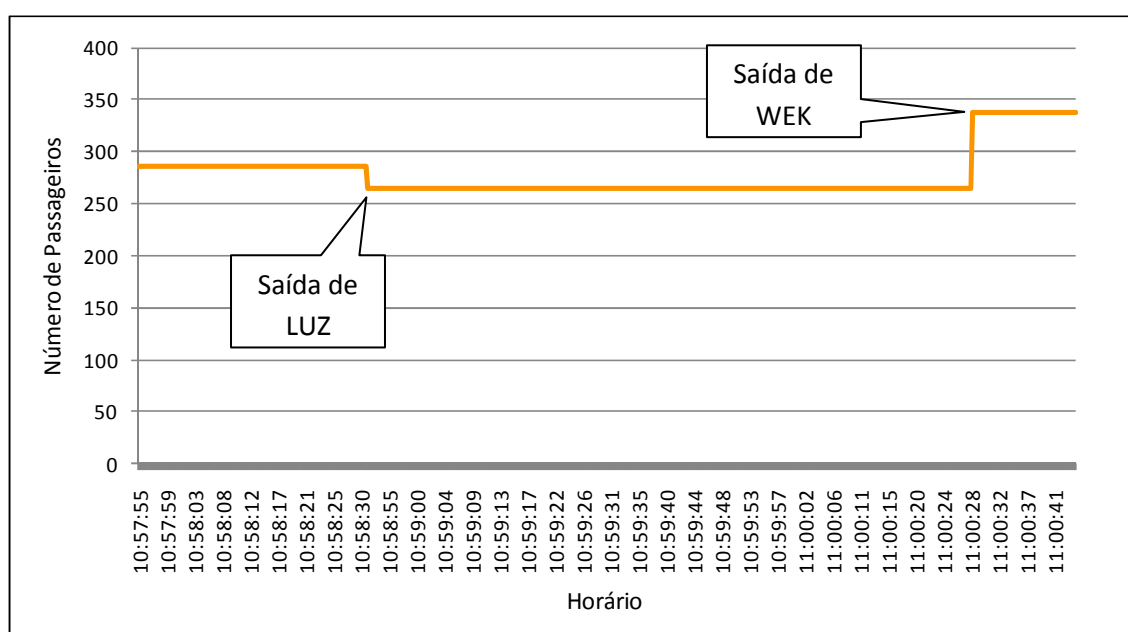


Figura 5.9: Gráfico do cálculo do número de passageiros do TUE 11 indo de LUZ a WEK.

Com o monitoramento constante da pressão nas bolsas de ar do trem, é determinada a variação diária do fluxo de passageiros em cada estação como mostrado nos gráficos das figuras 5.10 e 5.11. Através destes gráficos, os controladores de tráfego podem identificar os horários em que ocorre maior lotação dos trens, podendo elaborar estratégias para evitar a superlotação. No gráfico da figura 5.10, pode-se observar que o horário de pico, na estação WEK com o trem na via 2 (sentido Jaboatão - Recife), ocorre no período da manhã, no horário entre as 6:00h e 7:00h.

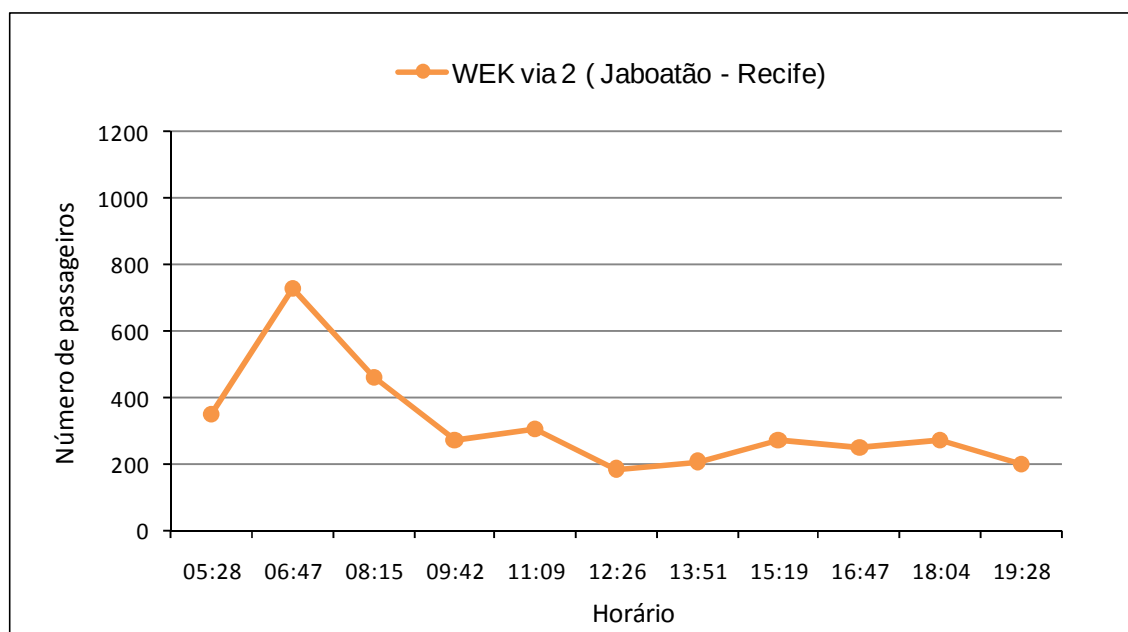


Figura 5.10: Gráfico do número de passageiros do TUE 11 na estação WEK, com o trem na via 2 durante quase todo o dia.

Com o trem se movimentando no sentido contrário (via 1, Recife - Jaboatão) a maior lotação de passageiros ocorre no horário após as 18:00h, como pode se observar no gráfico da figura 5.11.

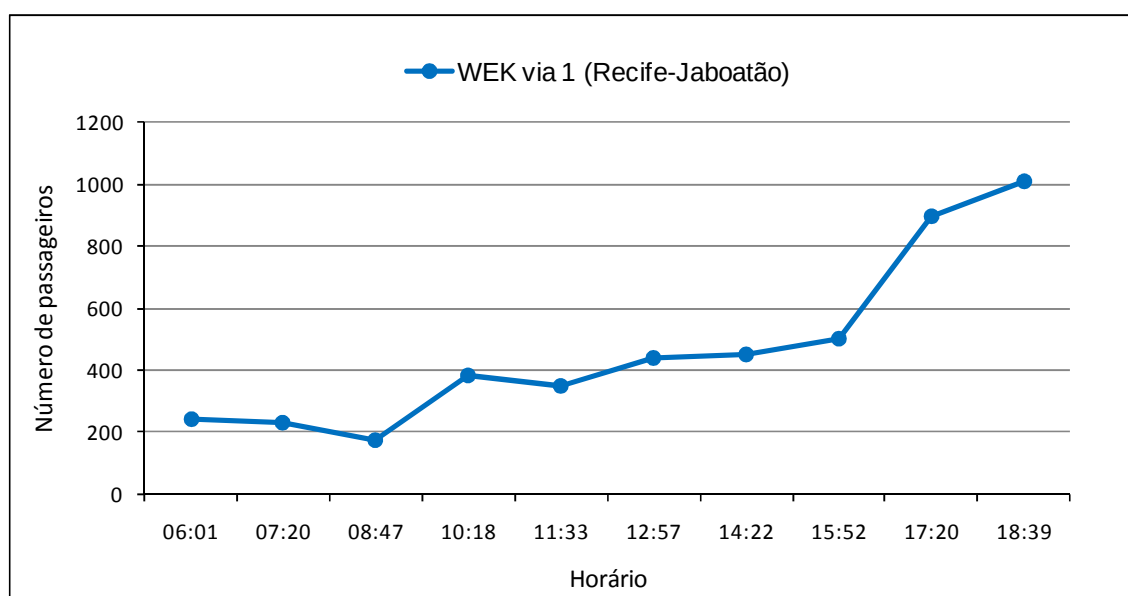


Figura 5.11: Gráfico do número de passageiros do TUE 11 na estação WEK, com o trem na via 1 durante quase todo o dia.

CAPÍTULO 6

Conclusão e trabalhos futuros

Para muitas empresas que necessitam de informação em tempo real como um fator estratégico para a tomada de decisões, o uso de transmissão de dados remota e sem fio é de extrema importância. Notadamente em empresas cujas decisões afetam não só o bem estar mas principalmente a segurança das pessoas, como é o caso do segmento de transporte metroferroviário, fatores como a precisão e qualidade das informações são de extrema relevância. Este trabalho, portanto, tem um grande impacto nesse segmento, pela proposta de desenvolvimento de tecnologia inovadora, integrando redes de sensores sem fio e processamento em sistemas embarcados que sejam capazes de trazer bons resultados, facilitando e aperfeiçoando os mecanismos de manutenção e possibilitando a otimização do tráfego de metrô.

Quanto ao sistema de uma forma geral, percebe-se um crescente investimento na área de tecnologia sem fio, que evita vários inconvenientes de operação nos sistemas atuais e abre portas para novos sistemas impossíveis de serem implementados através de fios. A tecnologia utilizada permite o desenvolvimento de um sistema com um custo consideravelmente mais baixo e com maior capacidade de expansão que os atuais sistemas de monitoramento de trens existentes no mundo.

Comparado ao atual sistema de monitoramento utilizado no MetroRec, o sistema telemétrico aqui proposto apresenta uma relação custo/benefício bem mais favorável. Além de empregar tecnologia de custo muito mais baixo que o sistema atual, ele permite a aquisição e envio de dados em tempo real com os TUE em movimento, permitindo assim que os dados possam ser analisados de forma rápida e prática.

O sistema de telemetria desenvolvido é um sistema inovador e já se encontra patentado (Araújo et al., 2008). O sistema foi publicado em alguns congressos e revistas (Santos et al., 2009; Araújo et al., 2009; Araújo et al., 2010), e participou de duas edições do Prêmio Alston de Tecnologia em 2008 (Araújo et al., 2009) e 2009 (Araújo et al., 2010), saindo-se vencedor na edição de 2009.

Como trabalhos a serem desenvolvidos, podemos citar a instalação da rede abrangendo a totalidade da via entre Recife e Jaboatão para possibilitar o monitoramento dos trens entre várias estações e permitir uma melhor análise do sistema. Para realização de tal atividade é necessário a aquisição de mais equipamentos para instalação das redes e das estações base.

Será realizado um estudo mais aprofundado sobre a rede sem fio para identificar possíveis problemas relacionados à transmissão dos dados. Este estudo poderá indicar medidas a serem tomadas para tornar a rede mais tolerante a falhas e garantir a confiabilidade na entrega das informações coletadas.

Serão estudadas formas para realizar o monitoramento remoto do consumo de energia dos sensores, permitindo a utilização controlada de baterias para alimentação dos roteadores na via, nos pontos onde não for possível usar a rede elétrica da ferrovia.

O sistema embarcado será testado em outros chips menores verificando a possibilidade de sua implementação em FPGA mais simples ou mesmo em outros dispositivos com capacidade para suportá-lo.

Para o sistema de supervisão, serão desenvolvidas interfaces de consultas mais complexas. O sistema de supervisão será aperfeiçoado para ser capaz de analisar os dados coletados e sugerir com base nessa análise, estratégias e ações preventivas de operação e manutenção, evoluindo para se torna um sistema inteligente de apoio à decisão.

Por fim, procurar parcerias com entidades buscando financiamentos para instalação completa do sistema RailBee na CBTU/METROREC.

Referências Bibliográficas

- Aboelela, V. E., Edberg, W., Papakonstantinou, C., Vokkarane, V., "Wireless sensor network based model for secure railway operations", Performance, Computing, and Communications Conference, 2006. International, p. 623-628.
- Altera Corporation, 2007, Nios Development Board Reference Manual, Stratix Edition. Disponível em: www.altera.com, acesso em 15 de abril de 2009.
- Ananda, A., Chan, C. M.; Mobile, wireless, and sensor networks: technology, applications, and future directions. New Jersey: Wiley Interscience, 2006.
- Anjos, E. G. Sistema Embarcado Reconfigurável para Automação de Unidades de Bombeamento de Petróleo através de Redes de Sensores sem Fios. João Pessoa: PPGI-UFPB, 2009.
- Anjos, E. G., Belo, F. A., Lima, J. A. G., Costa, Y. G. G.. Embedded system architecture for oil pump using reconfigurable computing. Brazilian Workshop on Real-Time and Embedded Systems – Simpósio Brasileiro de Redes de Computadores. Rio de Janeiro – RJ, 2008.
- Araújo, R. C. C., Belo, F. A., Santos. J. L. S., Lima Filho, A. C., Lima, J. A. G., 2008, "Sistema Telemétrico Dinâmico De Monitoramento e Automação De Redes Ferroviárias". Patente INPI RePB 000240, UFPB, 01 dez. 2008 (Depósito).
- Araújo, R. C., Santos J. L. A., Anjos, E. G., Lima Filho, A. C., Belo, F. A., Lima, J. A. G; "RailBee", Sistema de instrumentação virtual de veículos em malhas metroferroviárias, Seminário Negócios nos Trilhos. Prêmio Alston de Tecnologia, Revista Ferroviária, 2009.
- Araújo, R. C., Santos, J. L. A., Belo, F. A., Lima, J. A. G; Utilização do Sistema Telemétrico Dinâmico RailBee para estimativa em tempo real do número de passageiros e análise do desempenho do TUE. Prêmio Alston de Tecnologia, Revista Ferroviária, 2010.

- Araújo, R. C., Santos, J. L. A., Morais, B. M., Belo, F. A., Lima, J. A. G.; Telemetria dinâmica de veículos rodoferroviários em malhas metroferroviárias, 17º Congresso Brasileiro de Transportes e Trânsito, Curitiba – PR, 2009.
- Ascariz, J. M. R., Boquete L., System for Measuring Power Supply Parameters with ZigBee Connectivity. *em Proc. 2007 IEEE Instrument and Technology Conf.*, pp. 1-2.
- Bondalapati, K.; Prassanna, V. K. Reconfigurable Computing Systems. *In: Proceedings of the IEEE*, 2002, p. 1201-1217.
- Cao L., Jiang W., Zangh Z. Networked wireless Meter Reading System Based on ZigBee Technology". *em Proc. 2008 IEEE Control and Decision Conf.* pp 3456
- Cardell-Oliver, R., Smettem, K., Kranz, M. & Mayer, K. (2004). Field testing a wireless sensor network for reactive environmental monitoring. *In Proceedings of the Intelligent Sensors, Sensor Networks and Information Processing Conference* , December 14–17, pp. 7–12.
- Carvalho M. G.; Implementação *Hardware/Software* da Estimacão de Movimento Segundo O Padrão H.264, Dissertação de Mestrado, Programa de Pós-Graduação em Sistemas e Computação, UFRN, 2007.
- Digi Internacional, 2008. XBee/XBee-PRO ZB OEM RF Modules. Disponível em: www.digi.com, acesso em 21 de fevereiro de 2009.
- Egan D., The Emergence of ZigBee in Building Automation and Industrial Controls. *IEEE Computing & Control Engineering*, Vol. 16, No. 2, pp. 14-19, April/May 2005.
- Fowler, M. UML essencial: um breve guia para a linguagem padrão de modelagem de objetos / Martin Fowler; Trad. João Tortello. – 3. ed – Porto Alegre: Bookman, 2005.
- França, G. A. T., Estudo e especificação de um Sistema de Telemetria usando a tecnologia spread spectrum, Campinas-SP, 1997.
- Golmie, N., Cypher, D., Erebalá, O. Performance Evaluation of Low Rate WPANS for Medical Applications. *National Institute of Standards and Technology* (2004).
- Guo, Y., Corke, P., Poulton, G., Wark, T., Bishop-Hurley, G. & Swain, D. (2006). Animal behaviour understanding using wireless sensor networks, ICN. *Proceedings, 2006, 31st IEEE Conference on Local Computer Networks* , 2006, pp. 607–614.

- Gupta, R. K. Introduction to Embedded Systems. Disponível em: <<http://www.ics.uci.edu/~rgupta/ics212/w2002/intro.pdf>>. Acesso em: 03 de maio. 2008. UCLA, 2002.
- Heath, S.. Embedded Systems Design. 2. ed. Oxford: Newnes, 2003. 541 p.
- Holanda, J. A. M. Projeto de um estimador de potência para o processador Nios II. Dissertação de Mestrado, ICMC-USP, 2007.
- IEEE Std 802.15.4™-2006, Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs).
- Iraqi, A.; Morawski, R.Z.; Barwicz, A.; Bock, W.J.; Distributed data processing in a telemetric system for monitoring civil engineering constructions. *IEEE Transactions on Instrumentation and Measurement*, vol. 48, no. 3, june 1999.
- Kumar, A.; Rahman, F.; System for wireless health monitoring; Sensors for Industry Conference, 2004. Proceedings the ISA/IEEE 2004 Page(s):207 – 210.
- Labrosse, J. J.. MicroC/OS-II: the real-time kernel. St. Louis: Focal Press, 2002. 605 p.
- Liu, C.; Fan, C.; ZigBee - Research into Integrated Real-Time Located Systems. Asia-Pacific Services Computing Conference, 2008. APSCC '08. IEEE Publication Date: 9-12 Dec. 2008, page(s): 942-947.
- Loureiro, A. A., Ruiz, L. B., Nogueira, J. M. S., and Mini, R. A. (2002). Rede de sensores sem fio. In Porto, I. J, editor, *Simpósio Brasileiro de Computação, Jornada de Atualização de Informática*, pages 193.234.
- Loureiro, A. A. F., Nogueira, J. M. S., Ruiz, L. B., de Freitas Mini, R. A., Nakamura, E. F., and Figueiredo, C. M. S. (2003). Redes de sensores sem fio. In *Simpósio Brasileiro de Redes de Computadores*, pages 179. 226.
- Maxfield Max C. The Design Warrior's Guide to FPGAs. USA, 2004.
- Noergaard, T. Embedded Systems Architecture: a comprehensive guide for engineers and programmers. Oxford: Elsevier, 2005. 656 p.
- Nejikovsky, B.; Keller, E.; Wireless communications based system to monitor performance of rail vehicles, *Proceedings of the 2000 ASME/IEEE Joint Railroad Conference*, 2000.

- Nishibori, N.; Sasaki, T.; Kawai, O.; Ogushi, H.; Kasai, T.; Toyoda, A.; Development of tracking train detection device (COMBAT) by using wireless communication, *Proceedings of the 2001 IEEE/ASME Joint Railroad Conference*, 2001.
- Olukotun, K. A. et al. A *Software-Hardware* Cosynthesis Approach to Digital System Simulation. *IEEE Micro*. v. 14, p. 48-58. Nov. 1994.
- Pellerin, D.; Thibault, S.; Practical FPGA Programming in C. Prentice Hall, 2005.
- Peron, R. V. Otimização de código fonte C para o processador embarcado Nios II. 2007. Dissertação de Mestrado. Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo, 2007
- Ruiz, L. B. (2003). *MANNA: Uma Arquitetura para o Gerenciamento de Redes de Sensores Sem Fio*. Tese de doutorado, Universidade Federal de Minas Gerais.
- Santoni, T. A., Santucci, J. F., De Gentili, E. & Costa, B. (2006). Using wireless sensor network for wildfire detection. A discrete event approach of environmental monitoring tool. *Proceeding of ISEIMA 06*, pp. 115 – 120.
- Santos, J. L. A., Morais, B. M., Araújo, R. C. C., Anjos, E. G., Lima, J. A. G.; Sistema Telemétrico para Monitoramento de uma Rede Metroferroviária Utilizando o Protocolo ZigBee, InfoBrasil, Fortaleza – CE, 2009.
- Santos, S. T., Redes de Sensores Sem Fio em Monitoramento e Controle, Dissertação de Mestrado, COPPE/UFRJ, Engenharia Elétrica, 2007.
- Scott, J. C. (1998). Applications of telecommunications and information technology for humanitarian health initiatives. *Proceedings of the Medical Technology Symposium*, 17–20 August, pp. 197–204.
- Sharma, V.K.; Vaidya, Yogita. M.; Radio Frequency Identification Based Rail Wagon Monitoring System, In *4th Proceedings of IEEE International Conference on Mechatronic*, Kumamoto Japan, 8-10 May 2007.
- Simon Wong, K. F., Tsang, I. W., Cheung, V., Gary Chan, S.H., Kwok, J. T. Position estimation for wireless sensor networks. In *Proceedings of IEEE Globecom 2005*, St. Louis, November 28, December 2, 2005.

- Sottile, F.; Giannantonio, R.; Spirito, M.A.; Bellifemine, F.L.; Design, deployment and performance of a complete real-time ZigBee localization system. *Wireless Days*, 2008. WD '08. 1st IFIP 24-27 Nov. 2008, Page(s):1 - 5
- Southworth, F.; Chin, S.-M.; Cheng, P.D.; A telemetric monitoring and analysis system for use during large scale population evacuations; *Road Traffic Monitoring*, 1989., Second International Conference on 7-9 Feb 1989 Page(s):99 – 103
- Stankovic, J. A., Cao, Q., Doan, T., Fang, L., He, Z., Kiran, R., Lin, S., Son, S., Stoleru, R. & Wood, A. (2005). Wireless sensor networks for in-home healthcare: Potential and challenges. *Proceedings of High Confidence Medical Device Software and Systems (HCMDSS) Workshop*, Philadelphia, PA, June 2–3, 2005.
- Svensoon, A. Executive Summary – The Safe Traffic Project, February, 2005.
- Szewczyk, R., Osterweil, E., Polastre, J., Hamilton, M., Mainwaring, A. & Estrin, D. (2004). Habitat monitoring with sensor networks. *Communications of the ACM*, 47 (6), pp. 34 – 40.
- Tilak S., Abu-Ghazaleh N., and Heinzelman W. A taxonomy of wireless micro-sensor network models. *ACM SIGMOBILE Mobile Computing and Communications Review*, 6(2):28–36, April 2002.
- Torres-Pereira, L.; Ruivo, P.; Torres-Pereira, C.; Couto, C.; A noninvasive telemetric heart rate monitoring system based on phonocardiography; *Industrial Electronics*, 1997. ISIE '97., Proceedings of the IEEE International Symposium on 7-11 July 1997 Page(s):856 - 859 vol.3.
- Umirov, U.R.; Seong-Hyun Jeong; Jung-Il Park; Applicability of ZigBee for real-time networked motor control systems. *ICCAS 2008. International Conference on Control, Automation and Systems*. 14-17 Oct. 2008 Page(s):2937 – 2940.
- Verdone, R.; Dardari, D.; Mazzini, G.; Conti, A.; *Wireless Sensor and Actuator Networks: Technologies, Analyses and Design*. Burlington: Elsevier, 2007.
- Wang, P; The Real-Time Monitoring System for In-Patient Based on Zigbee. *Second International Symposium on Intelligent Information Technology Application*, 2008. IITA '08. Volume 1, 20-22 Dec. 2008 Page(s):587 – 590
- Wilson, P. *Design Recipes for FPGAs*. Oxford: Elsevier, 2007

Wolf W., Madsen J., "Embedded systems education for the future," *Proceedings of the IEEE*, 88(1), January 2000, pp. 23-30.

Zeng-Qiang Ma; Zhan-Feng Gao; Yan Yan; Wireless Monitoring System of Train Speed Based on RCM3000, *In IEEE International Conference on Machine Learning and Cybernetics*, 2006.

Zammattio, S. Delivering a Consistent *Software Development Environment* on a Changing Processor Platform, 2006. Disponível em: <http://www.fpgajournal.com/whitepapers_2006/q1_altera.htm>. Acesso em: 27 de Maio. 2009.

Zhou H.; Zhang F.; Liu J.; Zhang F.; Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09. 5th International Conference on 24-26 Sept. 2009 Page(s):1 - 4

ZigBee Specification, ZigBee Document 053474r17, 2007

Glossário

Avalon: Barramento de comunicação utilizado pelo processador Nios II que permite a comunicação com controladores de dispositivos agilizando e facilitando o desenvolvimento de aplicações.

CCO: *Centro de Controle Operacional*. É o local onde os controladores de tráfego dos trens do MetroRec trabalham.

CIM: *Centro de Informação e Manutenção*. Local onde trabalham os engenheiros e técnicos responsáveis pela manutenção dos trens do MetroRec.

Estação Base (EB): Termo atribuído a ao *hardware* do sistema de tratamento que se compõe de um nó coordenador ZigBee conectado a um FPGA através de uma interface serial.

Estação Roteadora (ER): Termo atribuído aos nós roteadores ZigBee instalados na ferrovia.

Estação Móvel (EM): Refere-se aos nós finais da rede ZigBee instalados nas cabines dos TUE.

FPGA: É um dispositivo semicondutor que é largamente utilizado para o processamento de informações digitais. É composto de *arrays* de células lógicas e suporta a implementação de circuitos de médio e grande porte podendo ser customizados de acordo com o projeto do cliente.

GPS (*Global Positioning System*): Sistema de Posicionamento Global. Inclui um conjunto de satélites e um sistema de informação eletrônico que fornece, via rádio, a um aparelho receptor móvel, a posição do mesmo com referência às coordenadas terrestres.

GSM: *Global System for Mobile Communications*. O Sistema Global para Comunicações Móveis (GSM: originalmente, *Groupe Special Mobile*) é uma tecnologia móvel e o padrão mais popular para telefones celulares do mundo.

LUZ: Sigla da estação de passageiros Santa Luzia.

MetroRec: Superintendência de Trens Urbanos do Recife.

Módulo RF: Dispositivo utilizado para comunicação sem fios utilizando ondas de rádio com comprimento de onda maior e frequência menor que a transmissão infravermelha.

Nios II: Segunda geração de processadores embarcados *soft-core* da Altera.

Nó Coordenador: Dispositivo responsável pela formação e gerenciamento da rede ZigBee.

Nó Final: Dispositivo que realiza a aquisição de dados dos sensores e os envia para o nó coordenador.

Nó Roteador: Dispositivo que atua como um repetidor dos dados recebidos dos nós finais permitindo estender a rede por uma longa distância.

Processador Embarcado: Processador utilizado em sistemas embarcados. Podem ser separados em duas categorias: microprocessadores e microcontroladores.

RailBee: Denominação dada ao sistema proposto neste trabalho. Este sistema é composto de três subsistemas: Uma rede ZigBee para captação de dados, um sistema embarcado para processamento e distribuição dos dados coletados e um sistema central de supervisão.

RSSF: *Rede de Sensores Sem Fio*. É uma rede composta por vários nós sensores sem fio que podem monitorar eventos ocorridos em um ambiente.

SoPC: *System-on-a-Programmable-Chip*. São sistemas que podem ser implementados em um chip programável, como um FPGA por exemplo.

Sistema Embarcado: Um sistema embarcado, ou sistema embutido, é um sistema microprocessado no qual o computador é completamente encapsulado ou dedicado ao dispositivo ou sistema que ele controla.

Telemetria: É a transmissão, através de rede fixa ou sem fio, de dados oriundos de múltiplas máquinas ou dispositivos remotos distribuídos sobre uma área geográfica de forma pré-determinada, para uma central de monitoramento e controle.

TUE: É um trem de tração elétrica composto por 4 carros, sendo 2 carros-motor e 2 carros-reboque, destinados ao transporte de passageiros de forma confortável e segura.

WEK: Sigla da estação de passageiros Werneck.

ZigBee: Tecnologia de redes sem fio baseada no padrão IEEE 802.15.4, comparada as redes *Wi-fi* e *Bluetooth*, de pequeno consumo e utilizado principalmente em automação industrial e doméstica.