UNIVERSIDADE DE TAUBATÉ
Paulo Rogério da Silva

Proposta de um Sistema Computacional para Verificação do Estado de Funcionamento de Escadas Rolantes à Distância

Taubaté – SP
2008

UNIVERSIDADE DE TAUBATÉ
Paulo Rogério da Silva

Proposta de um Sistema Computacional para Verificação do Estado de Funcionamento de Escadas Rolantes à Distância

Dissertação apresentada para obtenção do Título de Mestre pelo Curso de Pós-Graduação do Departamento de Engenharia Mecânica da Universidade de Taubaté.
Área de Concentração: Automação
Orientador: Prof. Dr. Eduardo Hidenori Enari

Taubaté – SP
2008

S586p

Silva, Paulo Rogério da.

Proposta de um Sistema Computacional para Verificação do Estado de Funcionamento de Escadas Rolantes à Distância / Paulo Rogério da Silva.. – Taubaté: Unitau, 2008.

80 f. :il;30 cm.

Dissertação (Mestrado) – Universidade de Taubaté. Faculdade de Engenharia Mecânica. Curso de Engenharia Mecânica

Orientador: Eduardo Hidenori Enari.

1. Escada Rolante. 2. Bluetooth. 3. Comunicação sem fio. I. Universidade de Taubaté. Departamento de Engenharia Mecãnica. II. Título.

CDD(21) 621.384

PAULO ROGÉRIO DA SILVA

PROPOSTA DE UM SISTEMA COMPUTACIONAL PARA VERIFICAÇÃO DO ESTADO DE FUNCIONAMENTO DE ESCADAS ROLANTES Á DISTÂNCIA

Dissertação apresentada para obtenção do Título de Mestre pelo Curso de Pós- Graduação do Departamento de Engenharia Mecânica da Universidade de Taubaté.
Área de Concentração: Automação

Data : _____

Resultado : _____

BANCA EXAMINADORA

Prof. Dr. Eduardo Hidenori Enari Universidade _____

Assinatura _____

Prof. Dr. Daniel Massaru Katsurayama Universidade_____

Assinatura _____

Prof. Dr. Luiz Eduardo Nicolini do Patrocínio Nunes Universidade_____

Assinatura _____

Dedico este trabalho à minha esposa e ao meu filho pela compreensão, carinho e apoio e que souberam entender a minha ausência para me dedicar a este trabalho

Aos meus Pais e irmãos pelos conselhos construtivos.

AGRADECIMENTOS

A Deus.

Ao Professor Dr. Eduardo Hidenori Enari pela orientação e apoio fornecidos durante a realização deste trabalho.

A todos colegas que de alguma forma ajudaram durante o projeto.

RESUMO

O objetivo do presente trabalho é apresentar uma proposta de sistema computacional para verificação do estado de funcionamento de escadas rolantes por meio de comunicação sem fio. O sistema computacional proposto neste trabalho é composto de uma interface instalada diretamente no painel de controle da escada rolante com o objetivo de coletar informações que permitam identificar o estado de funcionamento desses equipamentos e transmiti-las para um computador remoto por meio de uma estrutura de rede sem fio construída sob o padrão *Bluetooth*. O sistema proposto possibilita que a inspeção periódica do estado de funcionamento seja feita sem a interrupção do serviço ao público, a qual ocorre somente quando a leitura de dados informar alguma não conformidade ou em períodos pré-estabelecidos pelo fabricante, para troca de peças por fadiga. Dessa forma, possibilita-se um ganho de qualidade e produtividade das equipes de manutenção, as quais poderão realizar uma quantidade maior de inspeções com menor tempo de execução do trabalho e um aumento do tempo de disponibilidade das escadas rolantes aos usuários. O presente trabalho apresenta a arquitetura básica de *hardware* e *software* desenvolvidos e discute questões de segurança e confiabilidade do sistema proposto, os ganhos de produtividade para as equipes de manutenção, bem como o aumento do tempo de disponibilidade dos serviços prestados por escadas e esteiras rolantes frente a indicadores mundiais de qualidade de serviço.

Palavras Chave: Escada rolante, *Bluetooth*, sistema de monitoração, manutenção e produtividade.

PROPOSAL OF A COMPUTER SYSTEM FOR VERIFICATION OF THE STATE OF OPERATION OF ESCALATORS AT THE DISTANCE

ABSTRACT

The objective of the present work is to present a proposal of computer system for verification of the state of operation of escalator through communication without thread. The computer system proposed in this work it is composed of an interface installed directly in the panel of control of the escalator with the objective of collecting information that allow to identify the state of operation of those equipments and to transmit them for a remote computer through a net structure without thread built under the pattern *Bluetooth*. The proposed system makes possible that the periodic inspection of the operation state is made without the interruption of the service to the public, which only happens when the reading of data informs some non conformity or in periods established by the manufacturer, for change of pieces for fatigue. In that way, it is made possible a quality earnings and productivity of the maintenance teams, which can accomplish a larger amount of inspections with smaller time of execution of the work and an increase of the time of readiness of the escalator to the users. The present work presents the basic architecture of hardware and *software* developed and it discusses subjects of safety and reliability of the proposed system, the productivity earnings for the maintenance teams, as well as the increase of the time of readiness of the services rendered by escalator front to world indicators of service quality.

Keywords: Escalator*,* Bluetooth, monitor system, maintenance and productivity**.**

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Nos dias atuais, a escada rolante é amplamente utilizada no transporte de pessoas entre pavimentos, como os encontrados em terminais rodoviários, ferroviários, aeroportos, metrôs, shoppings e supermercados, promovendo uma circulação ágil e segura dos usuários além do aproveitamento das áreas de circulação. A capacidade de transporte de uma escada rolante pode chegar a 13500 pessoas por hora podendo variar de acordo com a sua largura e velocidade, conforme a NBR 8900 (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 1995, p.5).

O perfeito funcionamento da escada rolante está relacionado à sua manutenção preventiva. Ajustes periódicos garantem a segurança e conforto dos usuários além de aumentar a durabilidade do equipamento e principalmente o seu período de funcionamento.

A qualidade da manutenção preventiva da escada rolante e a sua periodicidade estão diretamente relacionadas a alguns fatores: (1) disponibilidade do equipamento para o controle de fluxo de pessoas, (2) a segurança dos usuários e (3) o custo de mão de obra. Uma boa manutenção preventiva diminui a possibilidade e periodicidade de uma parada da escada rolante por falha, conseqüentemente diminuindo o número de chamados do cliente para a empresa conservadora.

O índice fundamental para medir a disponibilidade do equipamento bem como o custo de mão de obra é o tempo médio entre chamada, o qual é caracterizado como o intervalo de tempo que a escada rolante fica disponível ao usuário sem falhas, também conhecido como *Mean Time Between Callbacks* (MTBC). Quanto maior o MTBC, maior é o tempo da escada rolante em funcionamento e, conseqüentemente

menor o gasto com mão de obra e sobressalentes necessários na manutenção corretiva.

O MTBC das escadas rolantes no Brasil é de 4400 horas o que é considerado muito baixo em relação a outros países como o Japão, onde o MTBC atinge a marca de 8600 horas[1] (informação verbal). Os principais fatores que contribuem para o índice de MTBC seja insatisfatório são:

- a qualidade da manutenção preventiva devido a rotatividade e conseqüentemente a inexperiência dos técnicos;
- o tempo médio de uma hora gasto para a execução da manutenção preventiva;
- a necessidade de paralisação e isolamento da escada rolante para a manutenção preventiva gerando transtorno no tráfego das pessoas dificultando o aumento da periodicidade nas manutenções preventivas;
- a falta de conhecimento dos técnicos na interpretação dos códigos de falhas emitidos pela placa de comando da escada rolante;
- a falta de mão de obra adequada a demanda.

O aumento na quantidade de inspeções de rotina na escada rolante é essencial para constatar irregularidades e prevenir uma futura paralisação, porém, o procedimento de inspeção é um processo demorado devido à necessidade da interrupção e isolamento da escada rolante para a visualização da Interface homem máquina (IHM) localizada na placa de comando acarretando transtorno no tráfego de pessoas no local.

[1] Informação fornecida pela Empresa Elevadores Atlas Schindler SA, em São Paulo, em Junho 2007.

1.1 Objetivo do Trabalho

Visando aumentar o MTBC, diminuindo o custo de mão de obra e o tempo de paralisação da escada rolante em inspeções de rotina, este trabalho tem como objetivo principal desenvolver um sistema computacional com interface amigável para a inspeção de escadas rolantes e orientação dos técnicos quanto à ação a ser tomada, de forma mais ágil e precisa, com base na codificação gerada pela placa de comando da escada rolante. A ferramenta permitirá o aumento de inspeções sem a necessidade da interrupção da escada rolante e conseqüente transtorno para usuários. Para tanto, o presente trabalho visa gerar um sistema de comunicação entre a placa de comando da escada rolante e um sistema supervisório externo, por meio de transmissão de dados utilizando tecnologia de comunicação sem fio (*wireless*). Esse sistema de comunicação deve permitir que o sistema supervisório conecte-se com a escada rolante e faça a sua inspeção sem a necessidade de seu isolamento e conseqüente interrupção de seu funcionamento. Os dados transmitidos serão recebidos e decodificados por meio de um computador no qual está instalado um sistema de leitura e decodificação dos dados enviados pela placa de comando da escada rolante. Dessa forma, o técnico irá interferir e interagir com a escada rolante somente quando realmente for necessário, uma vez que a placa de comando além de gerar os códigos de falhas também gera códigos de avisos de alertas e os mesmos serão apresentados aos técnicos durante a inspeção.

Há diversas tecnologias que estão sendo expandidas para conexões e redes sem fios, principalmente as tecnologias infravermelha (IrDA), HomeRF, IEEEA 802.11/WI-FI e a tecnologia *Bluetooth*. No presente trabalho, considerando o grande de número de usuários da aplicação proposta, optou-se pela utilização de tecnologia *Bluetooth*,

principalmente pelos seguintes benefícios: ser uma tecnologia de baixo custo, ter um baixo consumo de energia, não precisar manter os dispositivos em linha de visão, possuir um protocolo que emula a comunicação serial RS232 e pelo fato desta tecnologia já fazer parte na maioria dos dispositivos móveis (celulares, PDAs e *Laptops*). Dessa forma, a escada rolante poderá ser monitorada, por meio de uma interface com qualquer outro dispositivo que também possua a tecnologia *Bluetooth* incorporada ou integrada. A principio, o presente trabalho utiliza um computador portátil modelo *Laptop*, com o sistema supervisório desenvolvido sobre a linguagem Delphi da Borland, mas o padrão de comunicação escolhido permite que o desenvolvimento do sistema supervisório possa ser executado em dispositivos ainda mais compactos como, por exemplo, um computador de mão ou um telefone celular com suporte à tecnologia *Bluetooth*.

1.2 Etapas do trabalho

A fim de verificar a viabilidade do projeto proposto foi adotada a seguinte metodologia de trabalho:

- Estudo da escada rolante, com o objetivo de conhecer sua estrutura básica de funcionamento bem como suas características técnicas e os recursos disponíveis no painel de controle, com especial atenção às conexões para a comunicação de dados.

- Estudo e análise das tecnologias existentes para a comunicação sem fio com a finalidade de identificar a mais adequada para a comunicação entre o sistema supervisório alternativo e a escada rolante, visando gerar um sistema eficiente de baixo custo de implantação e uso.
- Estudo e análise do protocolo de comunicação da escada rolante incluindo o desenvolvimento de uma metodologia de extração de protocolo. Nesta etapa será possível determinar a codificação dos dados transmitidos pela escada e fazer um mapeamento entre códigos e os respectivos significados (falhas e alertas).
- A implementação de um protótipo de sistema supervisório da escada rolante e da estrutura de comunicação entre o sistema supervisório e a escada rolante para realização de testes. O protótipo incluirá a interface que irá converter o sinal padrão da escada rolante em um sinal de radio freqüência (RF) com finalidade de comunicação sem fio.
- Análise e discussão dos resultados obtidos.

1.3 Estrutura do texto

Neste primeiro capítulo foi apresentada uma breve exposição dos motivos e desafios desta proposta de trabalho, os seus escopos principais, metodologia utilizada, bem como uma breve visão do contexto tecnológico.

O capítulo dois apresenta uma descrição geral dos principais elementos envolvidos neste trabalho: a escada rolante e a tecnologia de comunicação sem fio baseado

nas principais referências bibliográficas existentes. Nesse mesmo capítulo aborda-se com mais ênfase a tecnologia *Bluetooth*.

O capítulo três tem como objetivo descrever o funcionamento e características de todos os elementos que são parte integrante do sistema supervisório desenvolvido neste trabalho, bem como de todas as ferramentas utilizadas para o seu desenvolvimento.

No capítulo quatro será apresentado a metodologia desenvolvida para a extração do protocolo de comunicação da escada rolante.

O capítulo cinco apresenta a simulação do procedimento atual e do sistema proposto com o objetivo de comparar e constatar a viabilidade e as vantagens do sistema proposto.

E, finalmente, no capítulo seis apresenta-se a conclusão e as principais oportunidades de pesquisa e de trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

2.1 Escada Rolante

A escada rolante surgiu em 1891, quando Jesse Reno criou uma grande novidade para os transportes: uma escada em ângulo com 25 graus que elevava passageiros. Em 1900 tal invento foi apresentado em uma exposição em Paris, onde foi denominado pela primeira vez como Escada Rolante, e redesenhado posteriormente por Charles Seeberger, adquirindo a configuração que mais se aproxima da que conhecemos hoje. (ABOUT ESCALATORS, 2008, p. 2).

A nova geração de escada rolante presente no mercado possui a mais alta tecnologia eletrônica e mecânica desenvolvida para o transporte vertical. Nesta seção, será apresentada uma breve descrição dos principais componentes, dos dispositivos de segurança e o do comando eletrônico.

Acompanhando a evolução tecnológica digital, todas as escadas rolantes fabricadas atualmente possuem uma arquitetura eletro-mecânico comandado por um sistema eletrônico microprocessado, permitindo o atendimento das exigências globais de qualidade e principalmente de segurança. A escada rolante possui em média aproximadamente vinte pontos de segurança acionados por contatos mecânicos e sensores os quais são monitorados constantemente pela placa de comando que garantem a segurança dos usuários em casos de falha elétrica ou mecânica.

Por meio de uma IHM existente nas placas de comando, é possível monitorar, operar, configurar e inclusive diagnosticar falhas nos diversos componentes da escada rolante agilizando o processo de manutenção. As placas de comando normalmente ficam na parte inferior da escada e/ou na parte superior, podendo variar de acordo com o fabricante.

A figura 1 ilustra as principais partes e componentes de uma escada rolante as quais as essenciais serão descritas brevemente.

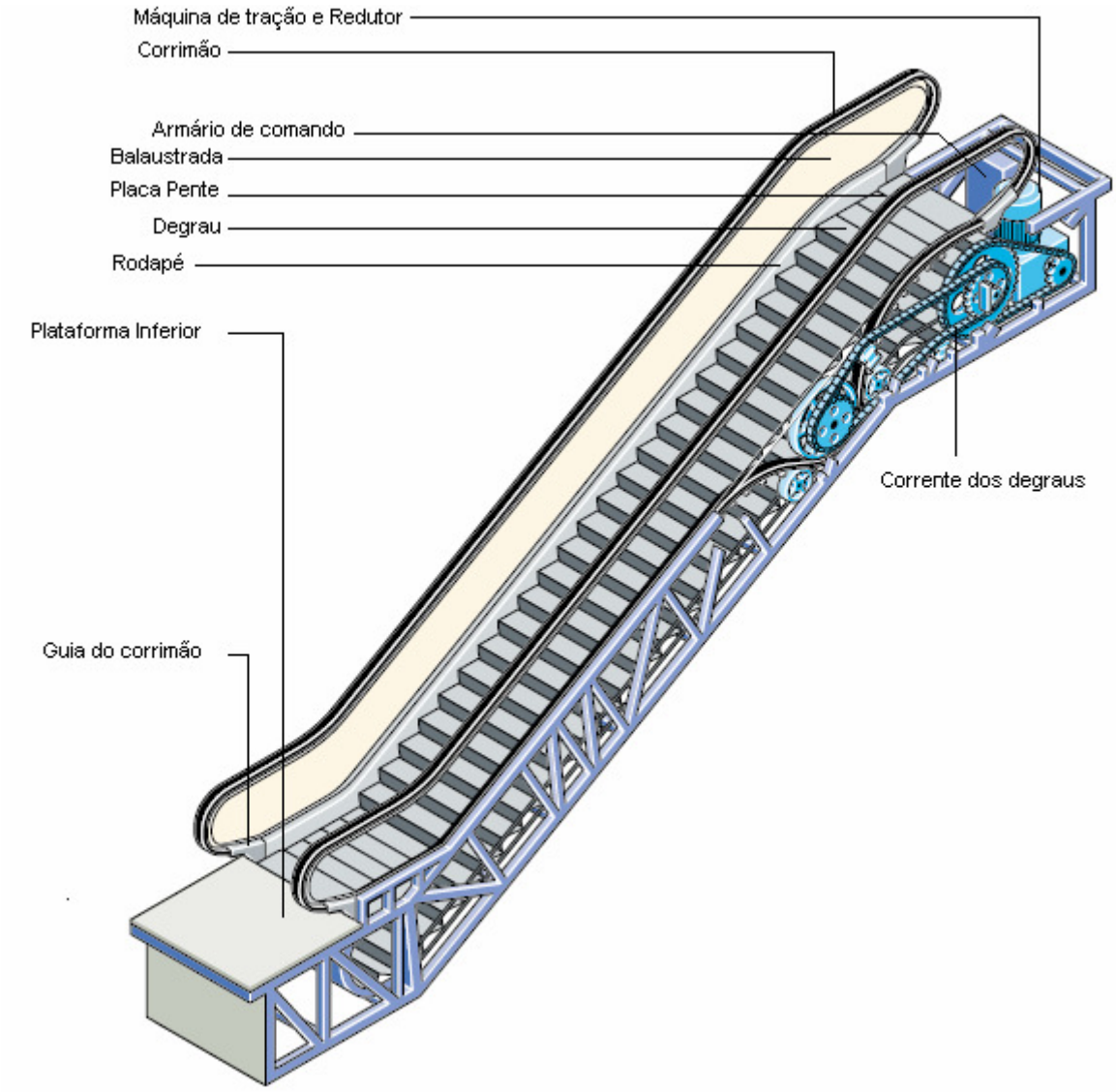

Figura 1 - Escada Rolante

2.1.1 Descrição dos principais componentes da Escada Rolante

As descrições dos principais componentes de uma escada rolante transcritas a seguir são informações baseadas em catálogos técnicos disponibilizados pelos principais fabricantes e pela NBR 8900.

2.1.1.1 Máquina de tração e Redutor

A máquina de tração é localizada no compartimento, em frente ao conjunto dos degraus e seu acesso é bem simples. O motor flangeado ao redutor transmite a potência através da corrente ao eixo principal. O conjunto é instalado na treliça, possibilitando um deslocamento longitudinal, o que permite um reajuste adequado da corrente de acionamento, conforme podemos observar na figura 2. A posição do redutor é assegurada por um dispositivo excêntrico.

Em geral são usados motores de tração com ventilação externa com potências de 5,5 / 7,5 / 11 / 15KW e possuem uma velocidade de 1200 RPM à 60Hz.

Figura 2 – Máquina de Tração

2.1.1.2 *Degraus*

Atualmente os degraus são de alumínio à prova de corrosão construídos em blocos únicos fundidos sob pressão dando maior segurança e pesos menores comparados com os antigos degraus multi-compostos. Possuem uma tensão de ruptura de aproximadamente 18kN e uma carga máxima de deformação de 5000N. Os patins guias de deslizamento são acoplados ao degrau.

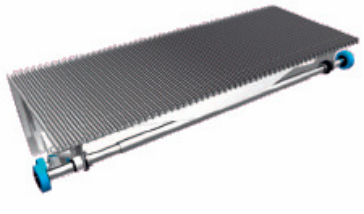

Figura 3 - Degrau

2.1.1.3 Corrente dos degraus

As correntes usadas para movimentar os degraus são de fabricação especial. Os rolos têm rolamentos que são revestidos com um material elástico resistente a lubrificantes, garantindo a ausência de ruídos. A cada três pinos da corrente um é do tipo especial, servindo para fixação do eixo do degrau.

2.1.1.4 Corrimão

O corrimão é fabricado com um tecido de reforço de várias camadas, pré-tensionadas. As forças de tração são absorvidas por uma malha de cabo de aço

colocada entre as camadas de tecido. Possui um tensão de ruptura mínimo de 25 kN. O corrimão é acionado em ambos os lados por polias de fricção.

A camada deslizante é de material sintético e a camada externa é de borracha.

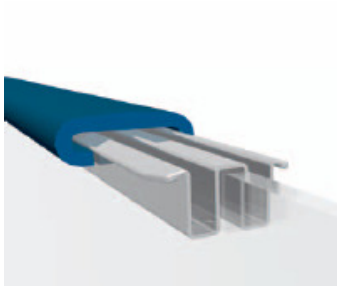

Figura 4 - Corrimão

2.1.1.2 *Freio*

Durante toda a operação da escada rolante o freio permanece aberto. Ele atua, quando ocorre uma parada de emergência, na interrupção de qualquer contato de segurança ou na falta de energia.

A abertura do freio é feita por um solenóide que atua sobre um sistema de molas. Na figura 5 podemos observar a arquitetura básica deste sistema.

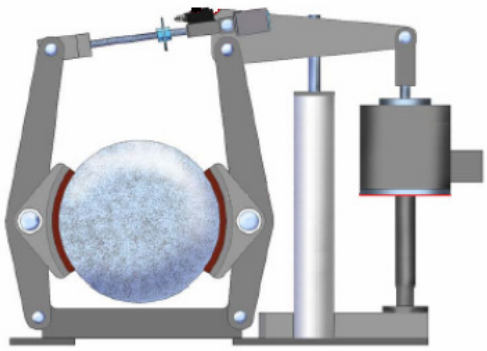

Figura 5 - Freio

2.1.2 Dispositivos de segurança

Os contatos de segurança existentes nas escadas rolantes são fundamentais para garantir a segurança dos usuários os quais interrompem imediatamente o funcionamento da escada rolante em situações de falhas mecânicas e elétricas. Uma escada rolante possui em média vinte contatos de segurança. Na figura 6 podemos observar a localização de alguns contatos de segurança.

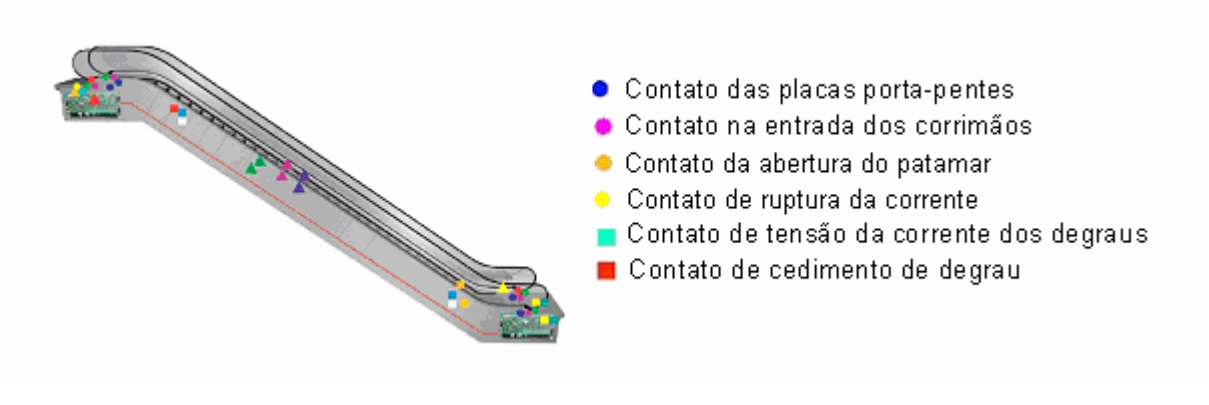

Figura 6 – Pontos de segurança

- Contato de ruptura da corrente de acionamento: No caso de ruptura da corrente de tração a alavanca que desliza sobre a corrente cai e o acionamento desliga o contato atuando em seguida o freio de segurança.
- Contato das placas porta-pentes: A placa porta-pentes é montada sobre guias laterais, podendo mover nos sentidos horizontal e vertical no caso de interferência com corpos estranhos. Com uma determinada força ocorre um deslocamento que aciona o contato e interrompe o circuito de segurança.

Figura 7 – Contato das placas porta-pentes

- Contato de tensão da corrente dos degraus: O contato de controle da tensão das correntes dos degraus é ativado no emperramento de um degrau, no bloqueio, no alongamento anormal ou na ruptura da corrente dos degraus.

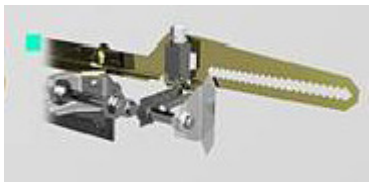

Figura 8 – Contato de tensão da corrente dos degraus

- Contato de cedimento de degrau: Se um degrau for rebaixado devido a ruptura, deformação, ou defeito nos rolos o contato é desligado por meio de apalpadores interrompendo o circuito de segurança.

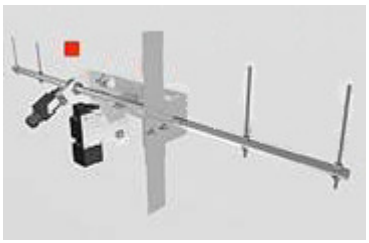

Figura 9 – Contato de cedimento de degrau

- Contatos na entrada dos corrimãos: Se algum objeto ficar preso entre o corrimão e a guarnição de borracha esta guarnição recua interrompendo o circuito de segurança por meio de atuação do contato.

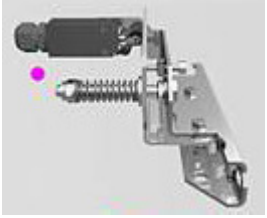

Figura 10 – Contato na entrada dos corrimãos

- Contato da cobertura do patamar: Com qualquer parte da cobertura do patamar levantada não é possível colocar a escada rolante em funcionamento.

2.1.3 Comando eletrônico da escada rolante

O comando eletrônico da escada rolante é composto por uma placa microcontrolada normalmente localizada na parte inferior da escada rolante. A placa é equipada com um *display* e teclas para navegação (IHM). O *display* pode mostrar, o estado operacional, informação de falha e dados memorizados codificados.

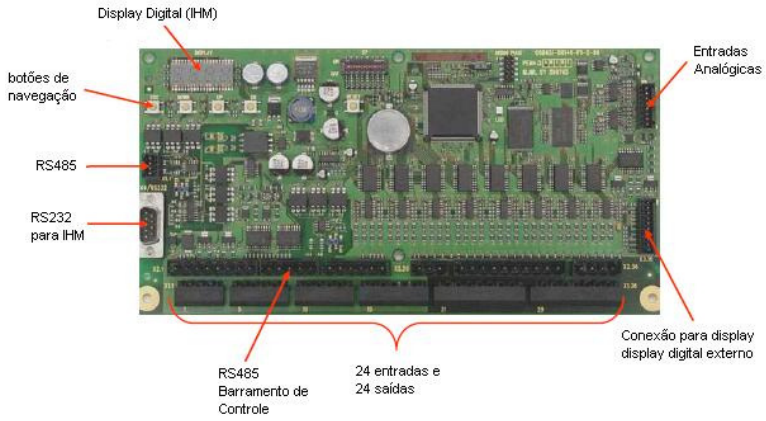

Figura 11– Comando Eletrônico da Escada Rolante

2.2 Tecnologias de comunicação sem fio

Esta seção examina as principais tecnologias sem fio com o objetivo de conhecer e identificar as suas principais vantagens e desvantagens. A tecnologia *Bluetooth* terá mais ênfase por ser a tecnologia definida na aplicação no trabalho proposto.

2.2.1 Tecnologia infravermelha (IrDA)

A *Infrared Data Association* (IrDA) estabeleceu um padrão para as conexões infravermelhas que foi adotado por mais de 160 empresas diferentes e é amplamente utilizado. O recurso de transmissão por raio infravermelho pode ser encontrado em mais de 150 milhões de dispositivos de computação, com as vendas desses dispositivos crescendo a uma taxa anual de 40% (MILLER, 2001).

A IrDA utiliza raio infravermelho para transmitir sinais de dados de um dispositivo para o outro, lembrando que raios infravermelho não são ondas de rádio.

Considerando que o raio não pode atravessar objetos sólidos, os dispositivos que se conectam por meio de IrDA têm que estar em linha de visão direta um com o outro tendo como limitações o alcance de até um metro e um ângulo de contato de 30 graus e inclusive é provável que qualquer movimento de um dispositivo interrompa a conexão infravermelha.

Em compensação às limitações, o padrão IrDA transmite dados a uma velocidade de processamento de 4Mbps. As limitações quanto a linha de visão, alcance e ângulo tornam a IrDA uma maneira muito segura para transmitir dados. Como os sinais são baseados em raios de luz em vez de ondas de rádio, os problemas decorrentes da interferência são reduzidos.

A IrDA é uma tecnologia de substituição sem fio ideal desde que os dois dispositivos estejam posicionados próximos um do outro.

As principais vantagens da IrDA são:

- Rapidez (4Mbps)
- Baixo custo (dois dólares ou menos por dispositivo)
- Baixo consumo de energia

- Segura
- Não suscetível à interferência de RF
- Amplamente utilizada

Porém, a IrDA não é uma boa tecnologia para rede local. A velocidade não é tão rápida quanto as tecnologias Ethernet ou 802.11 sem fios, discutidas ainda neste capítulo, e as limitações de conexão são muitos grandes. Também não é adequada para conexões públicas *ad hoc*[2].

As principais desvantagens da tecnologia IrDA são:

- Alcance limitado (um metro)
- Ângulo de conexão limitado (30 graus)

2.2.2 A tecnologia HomeRF

A *HomeRF* é uma tecnologia de rede sem fio projetada exclusivamente para uso em residência e em pequenas redes comerciais. Como a tecnologia *Bluetooth*, a *HomeRF* utiliza a banda 2.4Ghz. O *HomeRF Working Group* baseou a tecnologia *HomeRF* no *Shared Wireless Access Protocol* (SWAP)[3]. A *HomeRF*/SWAP permite os seguintes canais de transmissão:

- Seis canais de voz baseados no padrão *Digital Enhanced Cordless Telecommunications* (DECT)[4].

---

[2] O termo "*ad hoc*" é geralmente entendido como algo que é criado ou usado para um problema específico. Do latin, *ad hoc*, significa literalmente "para isto". Geralmente, numa rede *ad hoc* não há topologia predeterminada, e nem centralizada. Redes *ad hoc* não requerem uma infra-estrutura configurados antecipadamente. Os nós numa rede ad hoc se comunicam sem conexão sem conexão física entre eles (WIKIPEDIA, 2008).

[3] SWAP - Protocolo de acesso sem fio compartilhado. Umas das características deste protocolo é permitir a comunicação por voz com alta qualidade (ROSS, 2008)

[4] DECT é uma norma muito utilizada em telefones portáteis. Os postos de conversação DECT utilizam comunicação digital sem fios. A norma DECT também pode ser utilizada para comunicações sem fios de dados digitais (WIKIPEDIA, 2008).

- Um canal de dados, baseado na especificação Ethernet sem fio IEEE 802.11

A mais recente versão da *HomeRF* possui taxas de transmissão de até 10Mbps, que são comparáveis às tecnologias Ethernet ou 802.11 sem fios.

Como a tecnologia *Bluetooth*, a *HomeRF* utiliza a tecnologia de alargamento de banda de salto de freqüência – *frequency hopping spread spectrum* (FHSS)[5] para reduzir a interferência e melhorar a segurança, o salto de freqüência ocorre a 50 saltos por segundo, em intervalos de 1Mhz.

A *HomeRF* é uma alternativa mais barata para as redes sem fios 802.11 mais caras. As redes *HomeRF* não exigem hardware de ponto de acesso dedicado, são conexões ponto-a-ponto feitas entre dispositivos na rede.

A *HomeRF* foi criada visando as redes residenciais sem fios. Destaca-se as seguintes vantagens da *HomeRF*:

- Rapidez (10Mbps com SWAP 2.0; apenas 1 a 2Mbps com SWAP 1.0)
- Baixo custo (de 70 a 200 dólares por dispositivo)
- Fácil de instalar
- Não exige ponto de acesso dedicado
- Permite até 127 dispositivos por rede
- Permite diversas redes no mesmo local físico
- Os saltos de freqüência reduzem a interferência com aparelhos eletrônicos portáteis e residenciais.

Embora a tecnologia *HomeRF* faça sucesso em casa, o seu desempenho não é um dos melhores quando utilizada em um escritório maior ou ambiente corporativo.

Abaixo são relatadas as principais desvantagens:

[5] FHSS – é também chamada de Salto de Freqüência. O objetivo desta tecnologia é transmitir dados sempre em um canal diferente, tendo um tempo de mudança de canais e um tempo de transmissão nos canais. (WIKIPEDIA, 2008)

- Alcance limitado (de 25 a 60 metros) quando comparada à tecnologia 802.11
- Dificuldade para integrar-se a redes sem fios existentes
- Menos estável do que as conexões de rede baseadas nas tecnologias 802.11 ou Ethernet
- Alto consumo de energia (não adequada para utilização em portáteis).

2.2.3 A tecnologia IEEE 802.11 / WI-FI

A *Wireless Ethernet Compatibility Alliance* (WECA) adotou uma tecnologia de rede sem fio consistente baseada em uma especificação desenvolvida pelo *Institute of Eletronic and Electrical Engineers* (IEEE). A especificação IEEE 802.11 está no núcleo de uma tecnologia que foi denominada Wi-Fi, de *Wireless Fidelity* (fidelidade sem fio). A tecnologia Wi-Fi é direcionada a redes corporativas, à medida que é uma tecnologia mais cara e de desempenho mais alto (MILLER, 2001).

Ao contrário das tecnologias *Bluetooth* e *HomeRF*, que utilizam a tecnologia de saltos de freqüência a tecnologia 802.11 utiliza a tecnologia de alargamento de banda de seqüência direta (DSSS, *direct sequence spread spectrum*). A diferença entre a FHSS e a DSSS é que onde os sinais da FHSS saltam por 79 freqüências diferentes espaçadas em intervalos de 1Mhz, os sinais da DSSS são fixos dentro de um canal de 17Mhz, mas encobertos com muito ruído criado para reduzir a interferência e melhorar a segurança.

O resultado de utilizar a DSSS em vez da FSSS é que a tecnologia 802.11 é rápida, com taxas de transmissão de dados de até 11Mbps. Nesse aspecto, a 802.11 é uma substituta aceitável para a Ethernet que tem velocidade de transmissão semelhante.

Uma rede 802.11 exige o uso de hardware de ponto de acesso, ou seja, uma estação base que pode somar-se ao custo da rede. Estações base custam em torno de 250 a mais de 1200 dólares (MILLER, 2001).

As principais vantagens da 802.11 / Wi-Fi incluem:

- Rapidez (11Mbps)
- Conexões consistentes e confiáveis
- Longo alcance (100 metros ou mais)
- Facilidade de integração com redes Ethernet existentes

As principais desvantagens são:

- Alto custo
- Exige pontos de acesso físicos
- Dificuldade de configuração e manutenção
- Não fornece suporte para voz e telefonia
- Possíveis problemas de compatibilidade entre dispositivos de fabricantes diferentes.

2.2.4 A tecnologia *Bluetooth*

É uma tecnologia que facilita as conexões sem fios de curto alcance e as comunicações entre vários dispositivos eletrônicos utilizando sinais de rádio freqüência para estabelecer transferências de dados ponto-a-ponto e ponto-a-multiponto dentro de um raio de até cem metros.

Para se comunicarem entre si, dois dispositivos devem conter um rádio *Bluetooth*. Esse rádio é extremamente pequeno consumindo pouca energia. Cada rádio

Bluetooth se adapta exatamente às mesmas especificações para transmitir e receber sinais, para que possa ser usado em qualquer lugar do mundo sem modificação.

A tecnologia *Bluetooth* foi projetada para reduzir a complexidade de conectar dois ou mais dispositivos, criando um padrão de comunicação que é adotado por todos os dispositivos *Bluetooth* conectando-se por meio de um conjunto específico de freqüências de rádio, substituindo completamente os problemas da conexão física. O padrão *Bluetooth* determina os protocolos precisos utilizados para transmitir e receber dados por meio da conexão sem fio possibilitando que todos os dispositivos *Bluetooth* utilizem os mesmos protocolos (MILLER, 2001).

2.2.4.1 *Arquitetura Bluetooth*

De acordo com Miller (2001) um dispositivo *Bluetooth* é qualquer produto eletrônico completo que incorpora um rádio *Bluetooth*. Em termos práticos, um dispositivo *Bluetooth* poderia ser um telefone móvel, um PDA *(Personal Digital Assistant*), um computador portátil, uma impressora, um scanner. Desde que o produto completo incorpore a tecnologia *Bluetooth* (na forma de um rádio *Bluetooth* e do *software* operacional correspondente, como mostra a figura 12), o produto pode ser denominado um dispositivo *Bluetooth*. Naturalmente, qualquer dispositivo *Bluetooth* incorpora mais tecnologia do que apenas o *Bluetooth*.

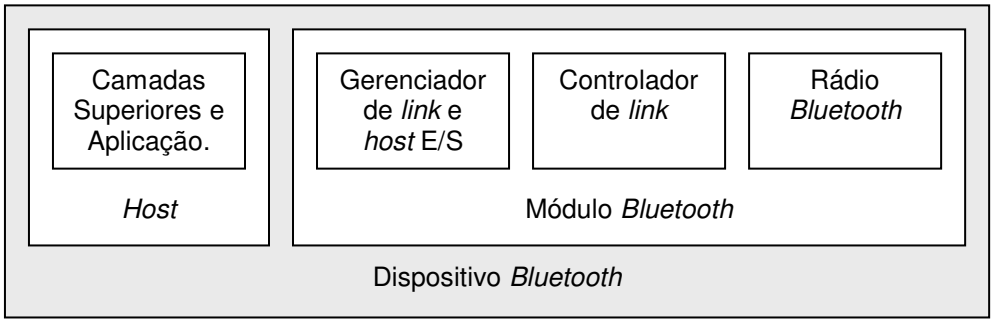

Figura 12 – A arquitetura de um dispositivo *Bluetooth*

Na terminologia *Bluetooth*, a parte não *Bluetooth* de um dispositivo é denominada *host* e todos os componentes *Bluetooth* (*hardware* e *software*) são combinados no módulo *Bluetooth*. As comunicações entre o *host* e o módulo *Bluetooth* são manipuladas pelo *software* gerenciador de *link* e pelo controlador *host* do módulo *Bluetooth*. A seguir serão abordados os componentes principais dessa arquitetura e também os tipos de canais de comunicação que podem ser estabelecidos entre dois dispositivos *Bluetooth*.

2.2.4.2 *Controlador Host*

O Controlador *Host* é a parte do módulo *Bluetooth* que gerencia toda a comunicação e interação entre o módulo *Bluetooth* e o dispositivo *Host*, interpretando os dados recebidos do *host* e os direcionando ao componente apropriado do módulo *Bluetooth* e vice-versa.

Para assegurar a interoperabilidade de módulos *Bluetooth* procedentes de vários fabricantes, a especificação *Bluetooth* define uma interface padrão (e o protocolo de comunicações) que pode ser usada por todos os módulos *Bluetooth* e por todos os dispositivos *host* que incorporam a tecnologia *Bluetooth*. Por meio desta interface (*Host Controller Interface* – HCI) um dispositivo *host* pode, por exemplo, instruir seu módulo *Bluetooth* para criar uma conexão com um dispositivo *Bluetooth* específico, executar procedimentos para verificar se existem outros dispositivos *Bluetooth* dentro de sua faixa de alcance, requerer autenticação, passar uma chave de segurança de conexão, solicitar ativação do modo de operação em baixo consumo de energia, etc (BISDIKIAN, 2001). O HCI não será abordado em detalhes, maiores informações podem ser obtidas em (*BLUETOOTH* SIG-a, 2001).

2.2.4.3 *O Rádio Bluetooth*

O principal componente de qualquer dispositivo *Bluetooth* é o rádio, que se caracteriza pelo seu pequeno tamanho e baixo consumo de energia. O rádio *Bluetooth* opera na freqüência 2.4Ghz, utilizando as tecnologias de salto de freqüência em banda larga FHSS e a *Time Division Duplexing* (TDD)[6].

Bluetooth é primeira tentativa de se fazer um rádio de um único chip que pode operar na banda 2.4GHz. A oferta de processamento de *front-end* de RF integrado com o módulo de bandabase é única (BHAGWAT, 2001). A integração em um só chip baixa o custo da interface de rede, e o tamanho pequeno tornam fácil embutir chips *Bluetooth* em dispositivos como telefones celulares e PDAs.

A especificação *Bluetooth* prevê três classes diferentes de rádios, conforme a potência de saída e conseqüentemente o seu alcance. Dessa forma se admite que diferentes dispositivos, previstos para suportar diferentes taxas de transferência de dados, podem ter diferentes necessidades de energia. O tipo mais comum de dispositivo é o de classe 3, que pode funcionar com alcance de cerca de 10m. Enquanto que um dispositivo classe 1 poderá alcançar até 100m.

Conforme Baatz (2001), a banda de 2.4GHz na qual o *Bluetooth* opera está disponível globalmente sem custo de licença de uso. Europa e EUA alocam 83.5MHz para esta banda, mas a Espanha, França e Japão alocaram menor quantidade.

Como o *Bluetooth* usa um sistema de salto de freqüência em banda larga, isto significa que o rádio salta através de toda a banda usando uma seqüência de salto

[6] TDD é uma técnica de implementação de comunicação *full-duplex* sobre um canal *half-duplex*. Consiste na aplicação de multiplexagem temporal aos sinais de comunicação em ambos os sentidos (WIKIPEDIA, 2008).

pseudo-randômica, a uma taxa de 1600 saltos por segundo, sendo que cada canal é usado para transmissão por 625µs (slot) antes que haja um salto para outro canal e assim sucessivamente, conforme ilustrado pela figura 13.

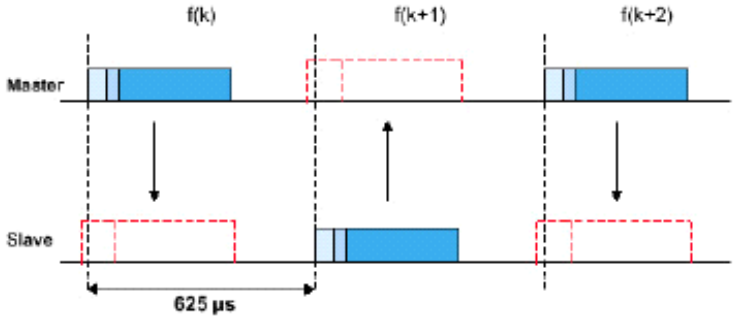

Figura 13 – Salto de freqüência por divisão de tempo (BLUETOOTH SIG-a, 2001)

Esta técnica de salto de freqüência é bastante apropriada para implementações de rádio de baixo custo e baixo consumo de energia. A principal vantagem da especificação *Bluetooth* foi à escolha de parâmetros: uma alta taxa de saltos por segundos, combinado com a escolha de um tamanho de pacote pequeno, deve garantir uma boa imunidade contra interferências de outras fontes na banda de 2.4Ghz (AU-SYSTEM, 2000). Essa banda, aberta a qualquer sistema de rádio, é bastante utilizada para operação de telefones sem fio, controles de porta de garagem e forno de microondas; sendo este último a fonte de interferência mais forte. Dessa forma, se uma transmissão é misturada com outra fonte, por exemplo, um forno microondas, a probabilidade de interferência no próximo canal saltado é muito baixa.

A velocidade máxima de *link* é 1Mbps, que é facilmente alcançada usando uma técnica de modulação (*Gaussian Frequency Shift Keying* – GFSK)[7]. Uma técnica de modulação mais complexa poderia alcançar taxas mais elevadas, mas GFSK é

[7] Na modulação GFSK os dados são codificados na forma de variações de freqüência em uma portadora. Os pulsos antes de entrarem no modulador é passado por um filtro gaussiano (para reduzir a largura espectral) (WIKIPEDIA, 2008).

interessante por que permite manter o projeto de rádio simples e de baixo custo (BRAGWAT, 2001).

2.2.4.4 Canais de Comunicação

Conforme Held (2001), o protocolo Banda Base (permite a conexão física via RF entre dois dispositivos) empregado pelo *Bluetooth* representa uma combinação da tecnologia de comutação de circuitos e de pacotes. *Bluetooth* pode suportar um canal de dados assíncrono, até três canais de voz simultâneos, ou um canal que simultaneamente suporta dados assíncronos e voz síncrona. Cada canal de voz suporta taxa de operação de 64Kbps em PCM[8]. O canal de dados assíncrono pode suportar a transferência assimétrica de dados a uma taxa máxima de 721Kbps numa direção, enquanto permite uma taxa de 57,6 Kbps na direção oposta. Além disso, o canal assíncrono pode suportar operações simétricas a uma taxa de 432,6 Kbps.

2.2.4.5 *A pilha de Protocolos*

De modo similar a outras tecnologias modernas de comunicação, *Bluetooth* inclui uma pilha de protocolos em sua especificação. A especificação *Bluetooth* divide sua pilha de protocolos conforme ilustrado na figura 14.

[8] PCM – É uma representação digital de um sinal analógico no qual a magnitude do sinal é obtida em intervalos regulares e então transformados em código binário (WIKIPEDIA, 2008).

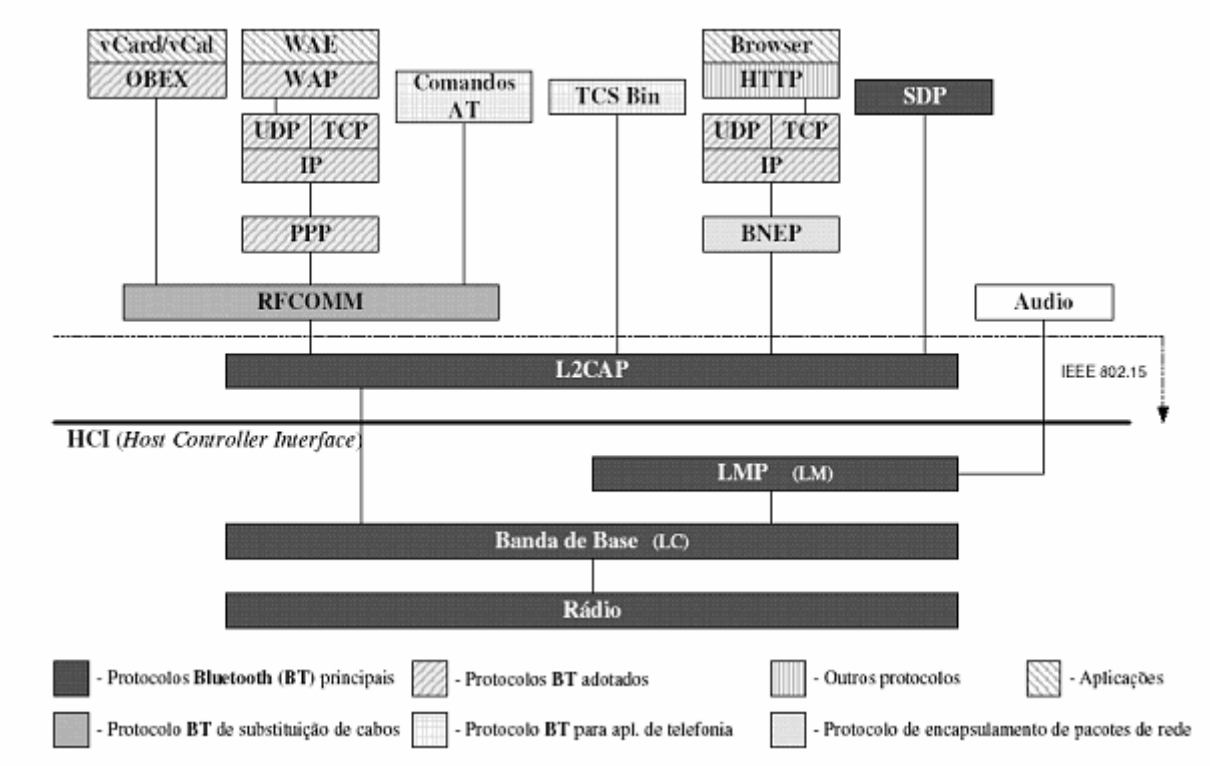

Figura 14 – Pilha de protocolos *Bluetooth*

No projeto da pilha de protocolos *Bluetooth* a estrutura buscou maximizar a reutilização de protocolos existentes nas camadas superiores, ao invés de reinventar protocolos similares. Os protocolos principais, que são listados na tabela 1, mais a transmissão de rádio *Bluetooth*, são requeridos pela maioria dos equipamentos *Bluetooth*, enquanto que os demais protocolos são usados somente quando necessário. Para melhor entender as funcionalidades da tecnologia *Bluetooth*, serão examinadas as diferentes camadas da pilha de protocolos e alguns dos protocolos mostrados na tabela 1.

Tabela 1 - Protocolos e camadas na pilha de protocolos *Bluetooth* (HELD, 2001)

Camada de Protocolo	Protocolos
Protocolos Principais *Bluetooth*	Banda Base, LMP, L2CAP, SDP
Protocolo de Substituição de Cabo	RFCOMM
Protocolos de Controle de Telefonia	TCS Binary, AT Commands
Protocolos Adotados	PPP, TCP/UDP, IP, OBEX, WAP, vCard, vCal, IrMC, WAE

2.2.4.6 *Protocolos principais Bluetooh*

Os protocolos principais *Bluetooth* são utilizados em todos os perfis *Bluetooth* e fornecem funções de transporte e gerenciamento de *link* a todas as aplicações. A seguir temos uma breve descrição dos seguintes protocolos:

- **Banda Base**: O protocolo Banda Base (*Baseband*) permite a conexão de freqüência de rádio física, denominada *link*, entre as duas ou mais unidades *Bluetooth* que formam uma *piconet*. Este protocolo também sincroniza o salto de transmissão de freqüência e os clocks dos dispositivos individuais *Bluetooth* em uma *piconet*.

  Há dois tipos diferentes de *link*s físicos fornecidos pelo protocolo *Baseband*. Com um *link Synchronous Connection-Oriented* (SCO), os pacotes podem conter uma combinação de áudio e dados ou apenas áudio. Com um *link Asynchronous Connection-less* (ACL), os pacotes são reservados apenas para dados (Miller, 2001).

- **Protocolo gerenciador de *Link***: Exatamente sobre o protocolo *Baseband* na pilha está o protocolo gerenciador de *Link* (LMP - *Link Manager Protocol*). O LMP é responsável pela configuração e controle do *link* entre dois ou mais dispositivos *Bluetooth*. Isso inclui diversos aspectos de segurança, como autenticação e criptografia, e o controle e negociação de tamanhos de pacotes da *Baseband*. O LMP também controla os modos de potência e ciclos de tarefas do rádio *Bluetooth*, assim como o estado da conexão do dispositivo *Bluetooth* quando ligada a uma *piconet* (Miller, 2001).

- **Protocolo para Descoberta de Serviço**: Os serviços *Discovery* permitem que dois dispositivos *Bluetooth* diferentes reconheçam e estabeleçam

conexões entre si e forneçam a base de cada perfil individual *Bluetooth*. O protocolo *Service Discovery Protocol* (SDP) permite que um dispositivo consulte o outro sobre informações, serviços e características daqueles serviços. Ele também permite o estabelecimento de uma conexão entre aqueles dois dispositivos. (Miller, 2001)

- **Protocolo de Adaptação e Controle de *Link* Lógico:** O protocolo *Logical Link Control and Adaptation Protocol* (L2CAP) funciona em paralelo com o LMP para transferir dados de nível mais alto para a camada *Baseband* e vice-versa. A grande diferença entre o L2CAP e o LMP é que o L2CAP fornece serviços à camada mais alta, o que o LMP não faz (Miller, 2001).

2.2.4.7 *Protocolo Substituição de Cabos*

A especificação *Bluetooth* inclui apenas um protocolo que trata da emulação sem fio de dados normalmente enviados por *link*s baseados em fios – o RFCOMM (AU-SYSTEM, 2000). O RFCOMM é um protocolo que emula uma conexão serial RS-232 entre dois dispositivos. Em linguagem simples, esse é o protocolo de substituição de cabo. O RFCOMM leva em consideração a emulação de controle RS-232 e sinais de dados pelo *Baseband Bluetooth* e também fornece recursos de transporte para serviços de nível superior que do contrário utilizariam uma conexão serial como seu mecanismo de transporte.

Embora, o protocolo RFCOMM tenha sido desenvolvido por engenheiros do *Bluetooth* SIG, ele está baseado em um subconjunto de um protocolo existente (MILLER, 2001).

2.2.4.8 *Protocolos de Controle de Telefone*

Os protocolos *Telephony Control* permitem que dispositivos *Bluetooth* manipulem chamadas de voz e dados de dispositivos compatíveis com a tecnologia *Bluetooth*. Para um dispositivo *Bluetooth* funcionar com um telefone ou um modem, um dos dois protocolos *Telephony Control* deve ser implementado na pilha de protocolos de um perfil.

O protocolo *Telephony Control Speficication – Binary* (TCS-BIN) define a sinalização de controle de chamada necessária para estabelecer chamadas de voz e dados entre dispositivos *Bluetooth.*

Todos os telefones e modens são controlados por um conjunto de comandos de áudio e telefonia (AT). Os comandos AT são normalmente utilizados para controlar todas as funções capazes de serem desempenhadas por um telefone ou modem de dados e são comuns entre vários dispositivos e fabricantes.

Os comandos AT da tecnologia *Bluetooth* são utilizados quando um perfil exige que um dispositivo *Bluetooth* seja empregado como um telefone ou modem quando se conecta a um sistema de telefonia celular ou fixa. O conjunto de comandos AT utilizado no protocolo *Bluetooth* está baseado em comandos estabelecidos pelo *European Telecommunications Standards Institute* (ETSI) e pela *International Telecommunication Union – Telecommunications* (ITU-T) (MILLER, 2001).

2.2.4.9 *Os protocolos Adotados*

A quarta camada de protocolo, protocolos adotados, inclui uma mistura de diversos protocolos existentes que são suportados e foram designados para rodar sobre

RFCOMM. Enquanto a maioria dos protocolos listados na tabela 1 (PPP, TCP/UDP, IP, OBEX, WAP, vCard, vCal, IrMC, WAE) pode parecer familiar, alguns poucos merecem menção. Dentre os protocolos que merecem menção inclui-se OBEX, vCard e vCalendar(vCal).

O protocolo OBEX representa um protocolo de sessão desenvolvido pela *Infrared Data Association* (IrDA) para a troca de objetos. OBEX usa o modelo cliente-servidor para a troca de objetos e é similar em funcionalidade ao http (*Hypertext Transmission Protocol*), mas opera em um modo muito mais simples, OBEX também define uma pasta tipo lista de objetos que é usada para navegar no conteúdo das pastas do dispositivo remoto.

Os protocolos vCard e vCal são especificações abertas controladas pelo Internet *Mail Consortium*. Ambas especificações definem o formato para troca eletrônica de cartão comercial e entradas de calendário pessoal e informações de organização e agenda. Essas especificações não definem mecanismos de transporte, mas apenas os formatos de dados para troca de informações. O transporte das informações nesses formatos é feito pelo OBEX.

Os demais protocolos dessa camada são padrões bem conhecidos no mercado e basicamente estão presentes nos cenários de uso do *Bluetooth* que envolvem o acesso aos recursos e serviços da Internet, em diversas maneiras possíveis, conforme já previstas nos modelos de uso definidos pela *Bluetooth* SIG (MILLER, 2001).

2.2.4.10 *Estados de conexão*

Um dispositivo *Bluetooth* pode operar em qualquer dos dois estados principais *Connection* e *Standby*. O dispositivo está no estado *Connection* se ele está conectado a outro dispositivo e envolvido em atividades correntes. Se o dispositivo não estiver conectado ou se estiver conectado, mas não envolvido ativamente com outros dispositivos então, ele automaticamente opera no estado *Standby*.

A criação de um estado *Standby* (espera) foi concebida com uma maneira para economizar energia em dispositivos *Bluetooth*. Se um dispositivo não precisa participar ativamente em determinado momento, não há nenhuma razão para que consuma energia em níveis máximos.

Quando um dispositivo está em estado *Standby*, “monitora”, a cada 1,28 segundos, as mensagens de outros dispositivos. Cada sessão de monitoramento ocorre pelo conjunto de 32 saltos de freqüências definido para aquele tipo de unidade (MILLER, 2001).

Uma vez que um dispositivo sai do estado *Standby* e entre no estado *Connection*, pode ser colocado em um dos quatro modos possíveis de conexão:

- ***Active*** – Diz-se que um dispositivo *Bluetooth* está em modo *Active* quando está participando ativamente na *piconet*, transmitindo ou recebendo. Unidades escravas ativas são automaticamente mantidas sincronizadas com a unidade mestra da *piconet*.
- ***Sniff*** – Quando um dispositivo for colocado em modo *Sniff*, monitora a *piconet* a uma taxa reduzida, diminuindo assim seu consumo de energia. A taxa *Sniff* é programável e varia de uma aplicação para outra.

- ***Hold*** – Dentro de uma *piconet*, as unidades mestras podem colocar as unidades escravas em modo *Hold*. Esse modo de economia de energia é utilizado quando nenhum dado precisa ser transmitido. Quando um dispositivo for colocado em modo *Hold*, apenas seu timer interno permanece ativo. Esse é um modo popular para dispositivos de baixo consumo de energia com necessidades de transferência de dados relativamente simples, como sensores de temperatura.
- ***Park*** – Quando um dispositivo precisa permanecer conectado a uma *piconet*, mas não necessita participar do tráfego de dados em progresso, esse dispositivo pode ser colocado em modo *Park*. No modo *Park*, o dispositivo permanece sincronizado à *piconet*, mas suspende seu endereço MAC. Ao estacionar dispositivos inativos, uma *piconet* pode realmente incluir mais de sete unidades escravas.

2.2.4.11 *Solicitação de paginação*

Todos os dispositivos *Bluetooth* não conectados iniciam no estado de baixo consumo de energia *Standby*. Quando uma unidade percebe outro dispositivo *Bluetooth* na área, um procedimento de conexão é iniciado. Nesse momento, o primeiro dispositivo assume o papel de unidade mestre no que irá se tornar em breve uma mini rede.

Um dispositivo *Bluetooth* pode emitir dois tipos diferentes de comandos para iniciar um procedimento de conexão. O primeiro comando é denominado um comando de solicitação. Um comando de solicitação é emitido quando o número de identificação ou endereço, do outro dispositivo ainda não é conhecido. Uma vez que o endereço

do dispositivo seja conhecido, um comando página é emitido. O comando página serve para despertar a outra unidade e estabelecer uma conexão plena entre os dois dispositivos.

2.2.4.12 *As piconets e scatternets*

Quando dois dispositivos *Bluetooth* estabelecem uma conexão, eles criam um tipo de rede pessoal denominada *piconet*. Cada *piconet* pode conter até oito dispositivos *Bluetooth* diferentes. Pode-se dizer, portanto, que *piconet* é um conjunto de dispositivos *Bluetooth* que compartilham um mesmo canal (BAATZ, 2002).

Dentro de cada *piconet*, um dispositivo serve como o mestre (*master*), enquanto os outros sete dispositivos funcionam como escravos (*slaves*), apresentando, portanto, uma configuração em estrela. Qualquer dispositivo individual pode pertencer, simultaneamente, a diversas *piconets*. A figura 15 mostra uma *piconet* típica.

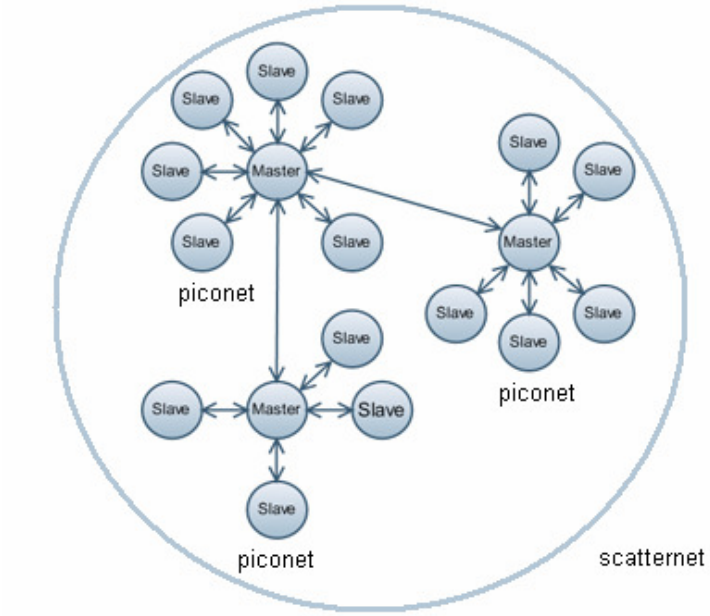

Figura 15 - Três *piconets* formando uma *scatternet*

Não há nenhuma diferença de hardware ou *software* entre um dispositivo mestre e um escravo, portanto qualquer dispositivo pode ser um mestre. O dispositivo que estabelece a *piconet* torna-se o dispositivo mestre (AU-SYSTEM, 2000). Os papéis em uma *piconet* podem mudar, mas nunca pode haver mais de um mestre.

A dispositivo exercendo o papel de mestre controla todo o tráfico numa *piconet*. Ele aloca capacidade para os *links* síncronos (*SCO*) e gerencia um esquema de *polling* para os *links* assíncronos (*ACL*). Dispositivos escravos somente podem enviar dados no *slot* escravo-para-mestre depois de terem sido endereçados no precedente *slot* mestre-para-escravo. Se um mestre não tem informação para enviar no *slot* mestre-para-escravo, um pacote contendo somente o *header* e o código de acesso é enviado. Ou seja, cada dispositivo escravo é endereçado em uma ordem especifica, num esquema de *polling,* e pode enviar dados somente após ter sido endereçado. Desse modo, é eliminada qualquer possibilidade de colisão de pacotes sendo enviados por diferentes dispositivos escravos.

Todos os dispositivos em uma *piconet* compartilham o mesmo canal de salto de freqüência, que é estabelecido na medida em que os escravos sincronizam os seus *clocks* internos ao *clock* da unidade mestre (MILLER, 2001). Isto permite que todas as unidades em uma *piconet* saltem de freqüência para freqüência na mesma seqüência - e estabelece uma identidade única para essa *piconet*. Como *piconets* diferentes têm identidades diferentes baseadas em canais de salto de freqüência diferentes, diversas *piconets* podem compartilhar o mesmo espaço físico sem interferir entre si.

Até sete *escravos* podem estar ativos e sendo atendidos simultaneamente pelo mestre. Se o dispositivo mestre necessita se comunicar com mais do que 7 dispositivos, ele poderá fazê-lo desde que primeiro instrua outros dispositivos

escravos ativos para que comutem para o modo *park* de operação em baixa energia, e então poderá convidar outros dispositivos em modo *park* para tornarem-se ativos na *piconet*. Esse procedimento pode ser repetido, o que permite ao mestre atender um grande número de escravos. Na verdade, uma mesma *piconet* poderá ter em teoria até um máximo de 255 dispositivos em modo *park* (BHAGWAT, 2001).

A maioria das aplicações *Bluetooth* imaginadas envolvem a comunicação local entre pequenos grupos de dispositivos. Uma configuração de *piconet* consistindo de dois, três, ou até oito dispositivos é idealmente adaptada para atender às necessidades de comunicação de tais aplicações. Quando muitos grupos de dispositivos necessitam estar ativos simultaneamente, cada grupo pode formar uma *piconet* separada. Os nós escravos em cada *piconet* permanecem sincronizados com o *clock* do mestre e saltam de acordo com a seqüência de salto de canal que é função do endereço do nó do mestre. Uma vez que as seqüências de salto de canal são pseudo-randômicas, a probabilidade de colisão entre *piconets* é pequena. Desse modo, *piconets* que compartilham a mesma área geográfica podem coexistir e operarem independentemente. Entretanto, quando o grau de compartilhamento é muito elevado, a performance de cada *piconet* começa a degradar.

Em alguns cenários de uso, entretanto, dispositivos de diferentes *piconets* podem necessitar comunicar-se uns com os outros. *Bluetooth* define a estrutura chamada de *scatternet* (redes de difusão) para facilitar a comunicação entre *piconets*. Como pode ser visto na figura 15, as conexões são formadas por *bridge nodes,* que são membros de duas ou mais *piconets*.

Um *bridge node* participa em cada *piconet* em um esquema de divisão de tempo. Depois de permanecer em uma *piconet* por algum tempo, um dispositivo *bridge* pode mudar para outra *piconet* comutando para a sua correspondente seqüência de salto.

Passando por todos os membros das *piconets*, o *bridge node* pode enviar e receber pacotes em cada *piconet* e também encaminhar pacotes de uma *piconet* para outra. Um *bridge node* pode ser um escravo em ambas *piconets*, ou ser um escravo em uma e um mestre em outra. Por exemplo, considere um salão cheio de pessoas, onde cada pessoa tem um telefone celular e um *headset* sem fio. Quando usuários falam com seus *headsets*, somente os telefones celulares correspondentes aos seus *headsets* deveriam receber o sinal. Neste exemplo, cada par de *headset* e telefone celular constituem uma *piconet* separada. Agora suponha que esses usuários também querem enviar mensagens de textos de seus telefones celulares uns aos outros. Isto será possível somente se todas as *piconets* são interconectadas para formar uma grande *scatternet* (rede de dispersão). As técnicas para a formação de *scatternets* estão ainda sob desenvolvimento.

2.2.4.13 *Inquiry e Paging*

Conforme Bragwat (2001), *Bluetooth* usa um procedimento conhecido como *inquiry* para descobrir outros dispositivos, e *paging* para subseqüentemente estabelecer conexões com eles. Tanto *inquiry* quanto *paging* são procedimentos assimétricos. Em outras palavras, isto significa que o dispositivo que faz o *inquiry* e o que recebe o *inquiry* (ou *paging* e *paged*) devem executar diferentes ações. Isto implica que quando dois nós estabelecem uma conexão, cada um deles necessita começar de um diferente estado inicial; caso contrário, eles nunca descobririam um ao outro. As especificações de perfis desempenham importante papel aqui, definindo o estado inicial requerido para cada dispositivo em todos os cenários de uso. Um

procedimento simétrico para estabelecimento de conexões é ainda um tópico de pesquisa.

Conceitualmente as operações *inquiry* e *paging* são simples, mas a característica de salto de freqüência na camada física torna os detalhes de baixo nível bastante complexos. Dois nós não podem trocar mensagens até que eles tenham concordado quanto à seqüência comum de salto de canal, bem como a correta fase dentro da seqüência escolhida. *Bluetooth* resolve esse problema simplesmente determinando o uso de uma seqüência de salto para *inquiry* bem conhecida de todos os dispositivos. Durante o *inquiry*, ambos os nós (um é o ouvinte e o outro o transmissor) saltam usando a mesma seqüência; mas o transmissor salta mais rápido do que o ouvinte, transmitindo um sinal em cada canal e ouvindo entre as transmissões por uma resposta. Quando mais do que um ouvinte está presente, suas respostas podem colidir. Para evitar colisão, ouvintes retardam suas respostas até expirar um tempo de espera aleatória. Finalmente o dispositivo transmissor coleta algumas informações básicas de seus ouvintes, tais como endereço de dispositivo e compensação de *clock*. Estas informações serão subseqüentemente usadas para a operação de *paging* com o dispositivo ouvinte selecionado.

Os passos de comunicação durante o procedimento de *paging* são similares, exceto que a mensagem de *paging* é do tipo *unicast* para um ouvinte selecionado, portanto o ouvinte não necessita esperar um tempo aleatório para responder. O transmissor também tem uma melhor estimativa do *clock* do ouvinte, o que o possibilita se comunicar com o ouvinte quase que instantaneamente. Tão logo receba um *ACK* para a mensagem *paging*, o transmissor torna-se o mestre e o ouvinte torna-se o escravo da *piconet* que acaba de se formar, e ambos os nós mudam para a

seqüência de salto de canal da *piconet*. Mais tarde, se necessário, os papéis de mestre e escravo podem ser trocados.

Os passos para admitir um novo escravo em uma *piconet* existente são ligeiramente mais complexos. O mestre pode tanto começar descobrindo novos nós em sua vizinhança e convidá-los para juntar-se à *piconet* ou, ao invés disso, esperar em estado de *scan* (ouvindo) para ser descoberto por outros nós. Com ambas as opções, a comunicação na *piconet* original deve ser suspensa pela duração do processo de *inquiry* e *paging*. A latência para admitir o novo nó em uma *piconet* pode ser grande se o mestre não muda para o modo *inquiry* ou *scan* com freqüência. Esta latência pode ser reduzida somente ao custo de alguma perda de capacidade para a *piconet*. O estudo desse custo-benefício é outro tópico que está em pesquisa.

2.2.4.14 *Segurança e autenticação*

Segurança é uma questão importante na especificação *Bluetooth*. Afinal, certamente ninguém gostaria de usar um dispositivo *Bluetooth* sabendo que outros dispositivos do mesmo tipo eventualmente existentes nas proximidades poderiam estar obtendo seus dados sem permissão, ou ainda que uma terceira pessoa poderia estar ouvindo sua conversação e vendo suas mensagens. *Bluetooth* inclui suporte para autenticação e criptografia. Essas duas funções de segurança são baseadas no uso de uma chave de *link* secreta que é compartilhada por um par de dispositivos (HELD, 2001). Esta chave secreta de *link* é gerada por um procedimento duplo quando dois dispositivos compatíveis com *Bluetooth* se comunicam pela primeira vez.

Existem três modos possíveis de segurança para um dispositivo *Bluetooth* (MILLER, 2001):

- Modo de segurança 1: Nesse modo, nenhuma medida de segurança é implementada. O dispositivo é efetivamente inseguro.
- Modo de segurança 2: Referido como segurança estabelecida no nível de serviço. Quando em modo de segurança 2, o dispositivo iniciará procedimentos de segurança após o estabelecimento da conexão. Este modo de segurança fornece a possibilidade de as aplicações terem diferentes políticas de acesso e permite suporte para rodar aplicações com diferentes requerimentos de segurança em paralelo (HELD, 2001).
- Modo de segurança 3: Referido como segurança estabelecida no nível do *link*. Quando neste modo de segurança, o dispositivo inicia os procedimentos de segurança antes de completar os procedimentos de estabelecimento de conexão.

A arquitetura de segurança usada pelo *Bluetooth* é bastante flexível, definindo a metodologia para autenticação e criptografia para o protocolo específico do *Bluetooth*, enquanto permite que protocolos não-específicos do *Bluetooth* que são transportados via *Bluetooth* possam ter suas próprias funções de segurança. Deve-se observar que a autenticação é fornecida usando um sistema de desafio-resposta, com uma conexão privada especificada para solicitar tanto uma via,via dupla, ou nenhuma autenticação. Considerando o método de autenticação construído no *Bluetooth*, ele somente autentica o dispositivo, e não seu usuário. Isso significa que os usuários de equipamento devem considerar o uso de um programa de segurança na camada de aplicação, ou tomarem cuidado para que equipamentos compatíveis com *Bluetooth* não caiam nas mãos de terceiros (HELD, 2001).

Embora autenticação seja suportada com base em uma chave de *link* armazenada, a arquitetura de segurança também permite que a autenticação ocorra pela formação de pares, através da entrada de um número pessoal de identificação (PIN). Autenticação é baseada no relacionamento confiável entre dois dispositivos. Isto é, um dispositivo confiável é aquele que foi autenticado previamente, ao passo que o dispositivo não confiável é aquele que não foi autenticado previamente ou para o qual nenhuma informação de segurança está disponível. Durante a preparação inicial de comunicação entre dispositivos *Bluetooth*, um relacionamento confiável será estabelecido. Também durante a preparação inicial, o nível de segurança é determinado baseado na necessidade para autenticação, autorização, e criptografia. Se uma necessidade para autenticação é determinada, o processo de autenticação entre dispositivos deverá ocorrer. Isso será seguido pelos processos de autorização e criptografia, se cada um deles é requerido. Uma vez que o módulo de segurança complete seu trabalho, o acesso é garantido entre os dispositivos. Observe que sob o módulo de segurança *Bluetooth*, um usuário de dispositivo pode configurar seu PDA, telefone celular, ou dispositivo similar compatível com *Bluetooth* para restringir comunicação para certos dispositivos, tal como, somente entre os produtos que o usuário tem, formando assim um grupo de usuário fechado (HELD, 2001).

2.2.4.15 *Gerenciamento de Chaves*

Conforme Miller (2001), o gerenciamento de chaves do *Bluetooth* funciona utilizando três tipos básicos de chaves:

- **Código PIN (*PIN Code*):** o número de identificação pessoal (PIN) é selecionado pelo usuário e deve ser um número de 48 bits.

- **Chave de link privada (*Private Link Key*):** um dispositivo *Bluetooth* pode utilizar um dos quatros tipos diferentes de chave de *link*, também denominada chaves de autenticação. Todas essas chaves são números aleatórios de 128 bits temporários ou semipermanentes, gerados na hora de cada transmissão. Uma chave de unidade é derivada por um único dispositivo *Bluetooth*. Uma chave de combinação é derivada de um par de dispositivos e é mais segura do que uma chave de unidade. Uma chave mestra é usada quando um dispositivo mestre em uma *piconet* deseja transmitir para vários dispositivos ao mesmo tempo e substitui a chave de *link* atual por uma de sessão. Finalmente, uma chave de inicialização é utilizada no processo de inicialização do dispositivo e protege os parâmetros de inicialização quando eles são transmitidos.
- **Chave de criptografia privada (*Private Encryption Key*):** a chave de criptografia é derivada da chave de *link* atualmente em uso. Toda vez que a criptografia é necessária, a chave de criptografia é alterada. A chave de criptografia pode variar entre 8 e 128 bits de extensão. Essa variação em extensão é necessária para se adaptar a legislação de exportação de vários países. Além disso, cada dispositivo *Bluetooth* individual tem seu próprio endereço de dispositivo exclusivo de 48 bits, atribuído pelo IEEE (*Institute of Electrical and Electronics Engineers*). As várias chaves, juntamente com o endereço do dispositivo, são usadas para gerar outras chaves “secretas" para cada *link* na conexão, para garantir que outros dispositivos *Bluetooth* (dentro ou fora da *piconet* atual do dispositivo) não possam monitorar secretamente uma conexão *Bluetooth*.

O Gerenciamento de Chave *Bluetooth* utiliza esses três tipos de chaves no processo de controle de chave. O processo funciona mais ou menos assim:

- O usuário (ou o dispositivo) digita uma senha numérica.
- O dispositivo gera uma chave de *link* privada e autentica com o segundo dispositivo.
- O dispositivo deriva uma chave de criptografia privada a partir da chave de *link* e, em seguida, autêntica com o segundo dispositivo. Como ilustrado na figura 16, se todas as chaves se corresponderem, os dois dispositivos serão conectados. Senão, a conexão será abortada.

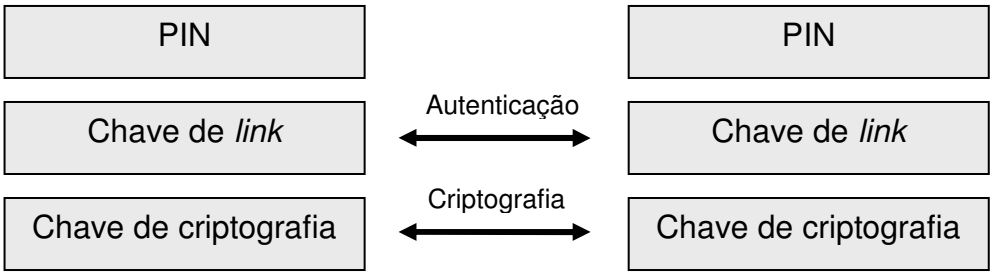

Figura 16 - A segurança de controle de chave entre dois dispositivos

2.2.4.16 *Autenticação de dispositivo*

Além do esquema de segurança de controle de dispositivo, a especificação *Bluetooth* também define um esquema de autenticação. Isso é essencialmente um esquema de "desafio e resposta", onde um protocolo de segurança especial é usado para verificar se o outro dispositivo reconhece uma chave secreta compartilhada (denominada “chave simétrica”). Se os dois dispositivos reconhecerem a mesma chave, a autenticação será bem sucedida; se qualquer um dos dispositivos não reconhecer a chave, a conexão será abortada.

Qual o dispositivo que enviará a mensagem "desafio" será determinado pela aplicação *Bluetooth* específica; não é necessariamente o dispositivo mestre que começa o processo. Algumas aplicações exigem autenticação apenas em um sentido, enquanto outras podem exigir autenticação mútua. Se autenticação falhar haverá um tempo de espera a ser guardado antes de uma nova tentativa de autenticação ser feita.

2.2.4.17 *Criptografia de pacotes*

A especificação *Bluetooth* requer uma criptografia sistemática de cada pacote que é transmitido. Há três modos de criptografia definidos:

- Modo de criptografia 1 - nesse modo, nenhum pacote é criptografado.
- Modo de criptografia 2 - nesse modo, o tráfego ponto-a-ponto é criptografado, mas o tráfego ponto-a-multiponto não é.
- Modo de criptografia 3 - nesse modo todo o tráfego é criptografado.

Apesar de todas essas medidas de segurança previstas na especificação *Bluetooth*, ainda assim há quem sustente que a tecnologia *Bluetooth* não é segura. Entretanto, isso não impede que a mesma seja utilizada em muitas aplicações úteis. E recursos adicionais de segurança podem ser incluídos na camada de aplicação.

2.2.4.18 *Os Modelos de Uso e Perfis*

Conforme Miller (2001), um *modelo de uso Bluetooth* é simplesmente uma descrição de uma possível aplicação da tecnologia sem fio *Bluetooth*. Os modelos de uso *Bluetooth* não são técnicos, são na verdade uma descrição de tarefas específicas

que os consumidores poderiam realizar com dispositivos *Bluetooth* em vários cenários potenciais de aplicação.

O *perfil Bluetooth* é a descrição técnica de como implementar um determinado modelo de uso. Ou seja, um perfil define em detalhes os procedimentos, parâmetros, características e protocolos necessários para implementar uma aplicação específica do modelo de uso.

Para garantir a interoperabilidade entre aplicações *Bluetooth* construídas por diferentes fornecedores, o *Bluetooth* SIG definiu previamente diversos modelos de uso e seus respectivos perfis. A expectativa é de que as implementações feitas com base nesses perfis sejam interoperáveis. A autoridade de certificação do *Bluetooth* usa esses perfis para testar e certificar a aderência, e permite o uso da marca *Bluetooth* somente para produtos que estão em conformidade com os métodos e procedimentos definidos nos perfis (BHAGWAT, 2001).

Todos os perfis são criados com base em uma camada principal de protocolos comuns, sobre a qual são colocadas camadas de protocolos específicos necessários para uma determinada aplicação ou modelo de uso.

Além de identificar protocolos, cada perfil define também procedimentos e mensagens que devem (ou podem) ser usadas para implementar uma aplicação específica. Assim, cada procedimento é identificado como sendo obrigatório, opcional, condicional, excludente ou não aplicável ao perfil atual.

Atualmente estão definidos 13 diferentes modelos de uso e perfis, conforme ilustrado na figura 17, mas serão citados os perfis, os quais estão relacionados diretamente e indiretamente com a aplicação sugerida neste trabalho (Perfis Genéricos e o perfil Serial Port).

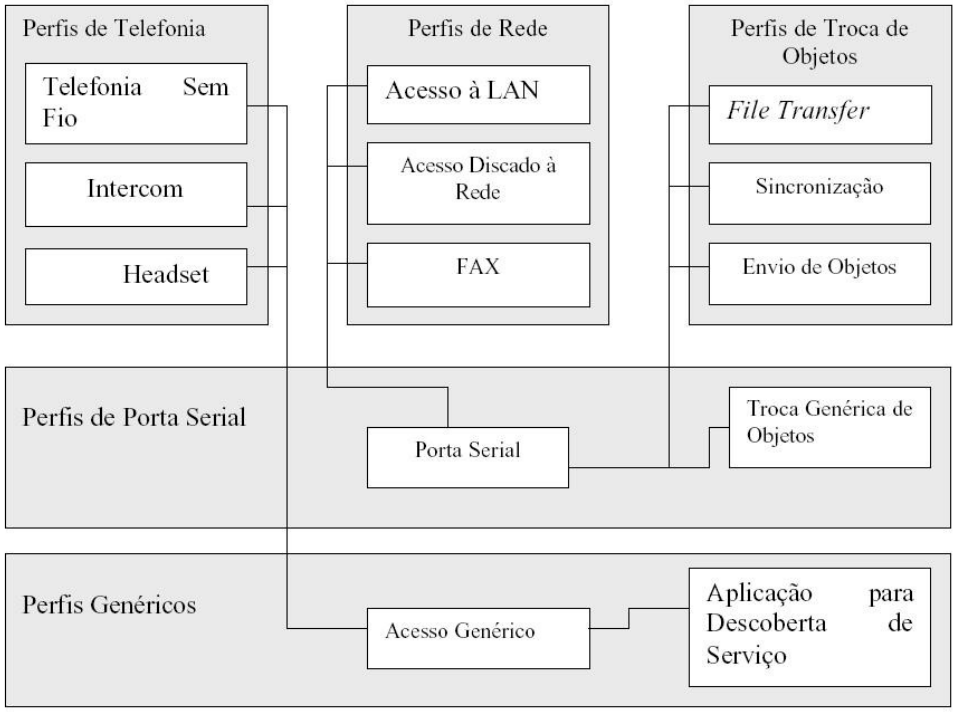

Figura 17 - A árvore da família do perfil *Bluetooth* (MILLER, 2001)

2.2.4.19 *Os Perfis Genéricos*

De acordo com Miller (2001), os dois primeiros perfis *Bluetooth* são denominados perfis genéricos, porque eles são essenciais para todas as formas de comunicação da tecnologia *Bluetooth*. Eles provavelmente serão implementados em todos os dispositivos compatíveis com a tecnologia *Bluetooth*.

2.2.4.19.1 *Perfil de Acesso Genérico*

O Perfil de Acesso Genérico (*Generic Access Profile-GAP*) define como duas unidades *Bluetooth* descobrem uma à outra e estabelecem uma conexão entre si. O *GAP* define modos e procedimentos que são genéricos e podem ser usados por outros perfis. Essencialmente, o *GAP* é a base sobre a qual todos os outros perfis

são criados. Ele abrange principalmente três tipos de itens: dicionário (uma coleção de termos e suas definições, de maneira que todos os fabricantes utilizem a mesma terminologia), conectividade (operações que permitem a um dispositivo *Bluetooth* se conectar a outros dispositivos e se autenticar com eles) e personalização (elementos que identificam e personalizam dispositivos individuais *Bluetooth*).

As funções a seguir estão definidas no *GAP*:

- **Classe do dispositivo** (*Device Class*). O *GAP* define o tipo de dispositivo e os tipos de serviços suportados por aquele tipo de dispositivo.
- **Nome do dispositivo** (*Device Name*). O *GAP* permite que os dispositivos tenham nomes amigáveis para os usuários, com até 248 bytes de extensão – embora alguns dispositivos *Bluetooth*, por causa de limitações de exibição, possam não ser capazes de exibir o nome todo.
- **Código secreto PIN Bluetooth** (*Bluetooth PIN*). O *GAP* estabelece que um número de identificação pessoal (PIN) pode ser digitado pelo usuário do dispositivo, para ser usado em processos de autenticação.
- **Modos de descoberta** (*Discovery Modes*). O *GAP* define três diferentes modos para detectar dispositivo: detectável genérico (permanentemente disponível para outros dispositivos), detectável restrito (disponível apenas por um período limitado de tempo ou sob condições específicas) e não detectável (não disponível para outros dispositivos).
- **Modos de paridade** (*Pairing Modes*). A paridade é um procedimento de inicialização onde dois dispositivos estabelecem uma chave de *link* comum para autenticação subseqüente. Existem dois modos diferentes de paridade, *com paridade* (aceita conexão) ou *sem paridade* (não aceita conexão).

- **Modos de segurança** (*Security Modes*). O *GAP* define três modos diferentes de segurança para dispositivos *Bluetooth*, o modo *de segurança* 1 (não restringe a segurança), o modo de segurança 2 (restrição de segurança na camada L2CAP) e o modo de segurança 3 (restrição de segurança na camada *Link*).
- **Procedimentos do modo ocioso** (*Idle mode procedures*). O *GAP* define vários procedimentos de modo ocioso para dispositivos *Bluetooth*, incluindo o consulta genérica (fornece endereço do dispositivo, relógio, classe e o modo digitalizar página de dispositivos detectáveis genéricos), consulta limitada (fornece as mesmas informações de dispositivos detectáveis restritos), *detecção de nome* (fornece apenas o nome do dispositivo) e detecção de dispositivo (fornece o endereço, relógio, classe, modo digitalizar página e o nome do dispositivo).
- **Vinculação** (*Bonding*). O *GAP* fornece dois tipos de relacionamento entre dispositivos *Bluetooth*, baseados em uma chave de *link* comum (denominada *vínculo*) que são: vinculação dedicada (os dispositivos criam e trocam uma chave de *link* comum e nenhuma outra informação) e vinculação genérica (os dispositivos examinam todos os canais e procedimentos de estabelecimento de conexão).
- **Procedimentos de estabelecimento** (*Establishment Procedures*). O *GAP* define os procedimentos necessários para estabelecer *links*, canais e conexões entre dois dispositivos *Bluetooth*. A conformidade com este perfil garante que qualquer par de dispositivos *Bluetooth*, independente de qual sejam os seus fabricantes, possam trocar informações via *Bluetooth* para descobrir que tipo de aplicações os dispositivos suportam.

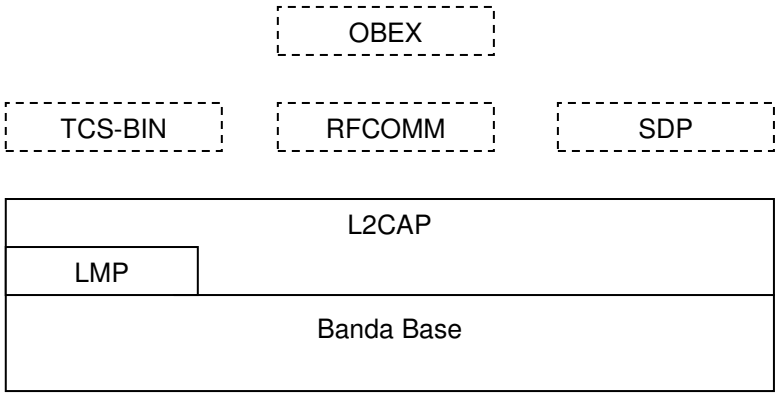

Figura 18 - A pilha de protocolos para o Perfil de Acesso Genérico

2.2.4.19.2 *Perfil de Aplicação para Descoberta de Serviço*

Esse perfil (*Service Discovery Application* Profile - SDAP) define um método padrão para a detecção de serviços disponíveis em uma outra unidade *Bluetooth*. A detecção de serviço é o processo pelo qual os dispositivos *Bluetooth* podem localizar, recuperar determinadas informações e fazer uso de serviços registrados em outros dispositivos *Bluetooth*. O perfil trata da pesquisa por serviços específicos bem conhecidos e também da pesquisa por serviços genéricos (AU-SYSTEM, 2000). O SDAP envolve a utilização de uma aplicação de usuário, a aplicação de detecção de serviço, que é requerida na unidade *Bluetooth* que deseja localizar serviços. Essa aplicação faz uso do Protocolo de Descoberta de Serviço (SDP, *Service Discovery Protocol*) para enviar e receber requisições de serviço de outras unidades *Bluetooth*. Portanto, o SDAP descreve como o protocolo SDP deve ser usado dentro de uma aplicação e como essa aplicação deve se comportar durante o processo de detecção de serviço.

Na terminologia *Bluetooth*, um dispositivo pode ser local ou remoto. O dispositivo que inicia o processo de detecção de serviço é rotulado como o dispositivo local; o

dispositivo que responde às consultas de serviço é o dispositivo remoto. Esse processo coloca o dispositivo local no papel de “cliente”, considerando-se um processo cliente/servidor, enquanto que o dispositivo remoto (carregado com as informações de "serviços" relativas às consultas do dispositivo iniciante), executa o papel de servidor. O resultado é que a pilha de protocolos do dispositivo local é ligeiramente diferente da pilha de protocolos do dispositivo remoto. Apenas o dispositivo local ou cliente, utiliza a aplicação SDAP (MILLER, 2001).

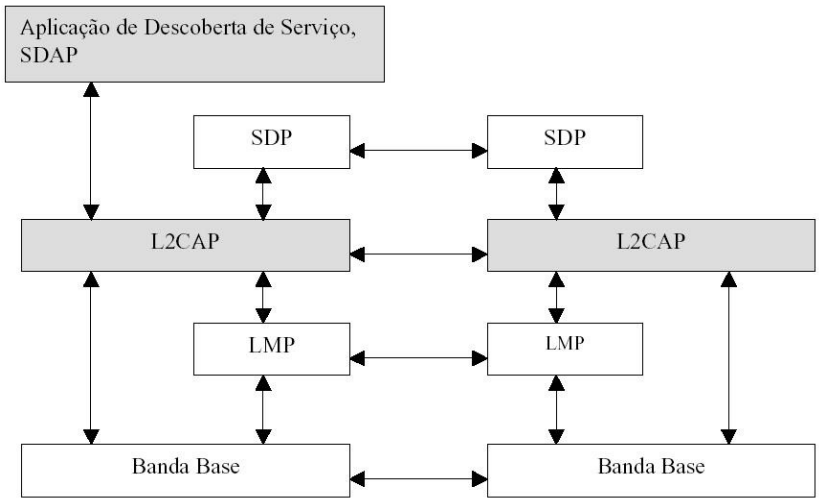

Figura 19 - Pilhas de protocolos local e remoto para o perfil SDAP

2.2.4.20 *Os Perfis de Porta Serial*

Os perfis de porta serial – às vezes denominado perfis de transporte – definem especificações para aplicações que têm de transferir dados de um dispositivo para outro, utilizando tipicamente os protocolos seriais RS-232 historicamente usados por dispositivos de computação.

2.2.4.20.1 *O perfil Serial Port (SPP, Serial Port Profile)*

O Perfil de Porta Serial (*Serial Port Profile*) define como estabelecer portas seriais virtuais em dois dispositivos e conectá-los usando *Bluetooth*. Usando esse perfil pode-se prover um dispositivo *Bluetooth* com a capacidade de emulação de um cabo serial com sinalização de controle RS-232, que é uma interface padrão comum para equipamentos de comunicação de dados. Esse perfil garante que taxas de transmissão de dados até 128 Kbits/s possam ser utilizadas [AU-SYSTEM, 2000).

Esse perfil utiliza os protocolos RFCOMM, SDP, L2CAP, LMP e Banda Base. Um componente importante do SPP é o protocolo RFCOMM, que é um protocolo de transporte simples que emula a porta serial RS-232, sendo utilizado para transportar dados do usuário, sinais de controle do modem e comandos de configuração em geral. A figura 20 mostra as pilhas de protocolos de dois dispositivos emulando uma conexão com cabo serial (MILLER, 2001).

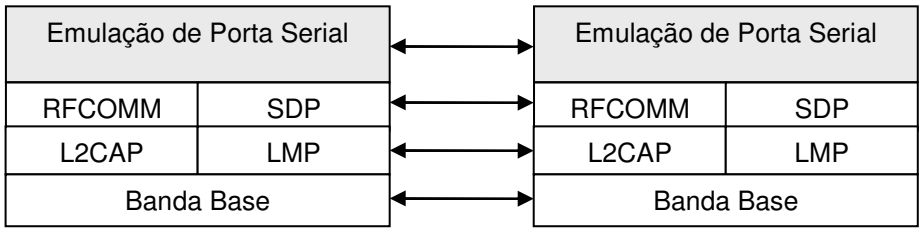

Figura 20 - As pilhas de protocolos do Perfil de Porta Serial

Estabelecer uma conexão serial por meio da tecnologia *Bluetooth* é uma operação relativamente simples. Em muitos casos, seria algo desse tipo:

1. O SDP do dispositivo local pergunta ao dispositivo remoto por seu número de canal do servidor.
2. O dispositivo remoto responde com seu número de canal do servidor.

3. Qualquer autenticação necessária acontece.
4. Uma conexão L2CAP é estabelecida.
5. Uma conexão RFCOMM é estabelecida no canal do servidor.

2.2.4.20.2 *Perfil de Troca Genérica de Objetos, GOEP*

O Perfil de Troca Genérica de Objetos (*Generic Object Exchange Profile*) define o conjunto de protocolos e procedimentos a serem usados por aplicações que tratam a troca de objetos. Diversos modelos de uso são baseados nesse perfil, como por exemplo, Transferência de Arquivos e Sincronização. Típicas unidades *Bluetooth* usando esse perfil são notebooks, PDAs, telefones celulares e telefones inteligentes [AU-SYSTEM, 2000).

Aplicações usando o perfil *GOEP* assumem que os *link*s e canais estão estabelecidos, conforme definido pelo *GAP*. O perfil descreve os procedimentos para o envio de dados de um dispositivo *Bluetooth* para outro, e também como extrair dados de outros dispositivos. O *GOEP* é dependente do perfil de Porta Serial, pois a maioria das transferências de dados envolvendo dispositivos de computação normalmente utiliza uma conexão serial RS-232.

Conforme Miller (2001) o GOEP utiliza os seguintes protocolos: OBEX, RFCOMM, SDP, L2CAP, LMP e Banda Base. Esse perfil, a exemplo do *SDAP*, opera em um modelo cliente/servidor. O dispositivo que inicia a conexão é definido como cliente, enquanto que o outro dispositivo é identificado como servidor. Após essa relação ter sido estabelecida, o GOEP fornece três recursos operacionais principais:

- **Estabelecer uma sessão OBEX** - utilizado para estabelecer uma sessão *Object Exchange* entre um dispositivo servidor e um dispositivo cliente.
- **Enviar um objeto de dados** - utilizado quando os dados necessitam serem transferidos do dispositivo servidor para o dispositivo cliente.
- **Recuperar um objeto de dados** - utilizado quando os dados necessitam fluir em outra direção, ou seja, serem transferidos do dispositivo cliente para o dispositivo servidor.

3 FERRAMENTAS UTILIZADAS

Neste capítulo serão apresentados todos os recursos utilizados para o desenvolvimento deste trabalho com o objetivo de descrever suas principais características de funcionamento.

Para o desenvolvimento e experimento do sistema proposto foram necessárias a utilização, configuração, aquisição e criação dos seguintes elementos: escada rolante para realização de testes do sistema, sistema de comunicação de dados baseado no padrão *Bluetooth*, composto de adaptador *RS232xBluetooth*, adaptador *USBxBluetooth*, adaptador controle de fluxo, sistema de inspeção, *Software BlueSoleil*, *Software Docklight* (analisador de protocolos) e um microcomputador.

3.1 Adaptador RS232 x Bluetooth

De acordo com a especificação *Bluetooth*, o protocolo RFCOMM, discutido no capítulo dois, permite a emulação de uma conexão serial RS-232 entre dois dispositivos tornando-se possível uma comunicação sem fio entre um equipamento e um PC ou até mesmo entre equipamentos. Como as placas de comando das escadas rolantes possuem uma saída padrão RS232 a qual é utilizada como via de comunicação com os respectivos *software*s proprietários a idéia foi pesquisar a existência de adaptadores RS232 x *Bluetooth* no mercado com o objetivo de emular esta conexão serial. Após pesquisa, foi constatada a existência de poucos adaptadores RS232 x *Bluetooth* certificados no mercado (listados na tabela 2), sendo todos importados sinalizando inclusive a oportunidade para o desenvolvimento de um produto nacional.

Tabela 2 - Adaptadores RS232 x *Bluetooth*

Modelo	Fabricante / Representante	Valor (R$)
Bluetooth RS232 adaptor Kit	LM Technologies	370,00
F2M01 Bluetooth RS-232 Plug	Free2Move	325,00
Socket Bluetooth RS-232 Adaptor	Socket	320,00
Bluecom	Naxos Tecnologia	290,00

Levando em consideração a relação custo / benefício foi adquirido um kit vendido pela Naxos Tecnologia modelo BlueCom para a realização dos testes cujas características serão descritas nas seções seguintes.

3.1.1 O conteúdo do kit do adaptador RS-232xBluetooth

- Adaptador RS232 x *Bluetooth* – modelo BlueCom da Naxos. A figura 21 ilustra o adaptador utilizado no projeto proposto.

Figura 21 - Adaptador RS232 x *Bluetooth* modelo BlueCom da Naxos

- Fonte de alimentação de 7,5V
- Cabo extensor RS232
- Cabo adaptador para periféricos

- CD utilitário

3.1.2 Conhecendo o adaptador RS232 x Bluetooth modelo BlueCom

Na figura 22 é ilustrada a interface do adaptador modelo Bluecom da Naxos e suas funcionalidades.

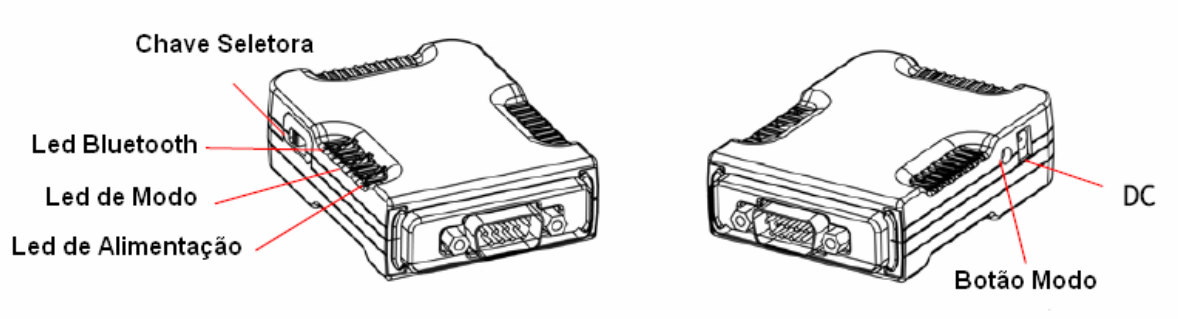

Figura 22 - Interface do adaptador Bluecom

- Botão Modo: Alterna entre o modo de configuração e o modo de dados
- Chave seletora: O adaptador pode ser alimentado por meio da fonte de alimentação (7.5Vcc) ou por meio do pino 9 (5Vcc).
- Led de alimentação Power (verde): Sinaliza quando o adaptador está energizado.
- Led *Bluetooth* (azul): Quando o led está aceso indica que uma conexão *Bluetooth* foi estabelecida. Caso esteja piscando sinaliza a transmissão ou recebimento de dados entre dois dispositivos sincronizados e conectados.
- Led de Modo (laranja): O Led Modo estará acesso quando o adaptador estiver no modo de configuração e apagado quando estiver no modo de dados.

- DC: Entrada para fonte de alimentação (7.5Vcc). Para utilização desta entrada a chave seletora deverá estar posicionada na opção de alimentação externa.

3.1.3 Características do adaptador

- Compatível com a versão padrão 1.1
- Suportado por Microsoft Windows 98SE,Me,2000 e XP
- Opera a distância de até 100 metros em espaço aberto
- *Bluetooth* classe 1 (20dbm)
- *Perfil Series Port Profile* (SPP) suportado
- Ambas interfaces DTE e DCE
- Freqüência de operação de aproximadamente 2.4835Ghz Banda livre ISM
- Operação *Frequency Hopping Spread Spectrum* (FHSS)
- Taxa de comunicação de 9600 a 460800 bps
- Antena interna ao circuito
- Tensão de entrada de 7,5Vcc

3.1.4 Software Utilitário

O *software* utilitário BT232Config.exe, que acompanha o kit, permite configurar os parâmetros de comunicação no adaptador. O *software* é utilizado para preparar os parâmetros de *hardware* de sincronismo, que são:

- *Role* (Função Mestre ou Escravo)

- *Security* (Segurança) – *Personal Identification Number* (PIN)
- *UART Configure* (Configuração UART) – parâmetros de *hardware* da porta COM.
- *Adapter Name* (Nome do adaptador)

3.1.5 Pinagem da saída RS232 do adaptador (DTE e DCE)

Na figura 23 é demonstrado a pinagem dos adaptadores DTE e dos adaptadores DCE.

BlueCOM DTE (macho)		
Pino	DTE	
	Nome	I/O
1	DCD	Input
2	RX	Input
3	TX	Output
4	DTR	Output
5	GND	Terra
6	DSR	Input
7	RTS	Output
8	CTS	Input
9	RI	Input

BlueCOM DCE (fêmea)		
Pino	DTE	
	Nome	I/O
1	DCD	Output
2	TX	Output
3	RX	Input
4	DTR	Input
5	GND	Terra
6	DSR	Output
7	RTS	Input
8	CTS	Output
9	RI	Output

Figura 23 - Pinagem dos adaptadores DTE e dos adaptadores DCE

3.2 Dispositivo USB Bluetooth

A função do dispositivo USB *Bluetooth* é permitir que outros dispositivos *Bluetooth* possam comunicar com um microcomputador (*desktop* ou *laptop*) que não possui a tecnologia *Bluetooth* integrada.

Figura 24 - Dispositivo USB *Bluetooth* – Classe 1

3.3 Microcomputador

O microcomputador utilizado no projeto tem as seguintes configurações:

- LapTop IBM ThinkPad
- Processador : Intel Pentium 1,70Ghz
- 512Mb de memória RAM

O sistema operacional utilizado no microcomputador é o Microsoft Windows XP Professional Versão 2002.

3.4 Escada Rolante

A escada rolante utilizada no experimento foi cedida por uma fabricante de escadas rolantes, a qual é utilizada como simulador no Centro de Treinamento situado na sua matriz, o que possibilitou verificar o comportamento do sistema proposto em um cenário próximo da realidade. Essa escada rolante possui as características citadas na introdução deste trabalho.

3.5 Adaptador Controle de Fluxo

Após inúmeros testes, tentando realizar uma comunicação sem fio com a escada rolante, foi constatada a necessidade do desenvolvimento de um adaptador com o objetivo de controlar o fluxo de dados entre o adaptador RS232 x *Bluetooth* instalado na placa de comando da escada rolante e o receptor USB x *Bluetooth* instalado no laptop. Analisando a estrutura de pinos da saída RS-232 da escada rolante, constatou-se somente a utilização dos pinos TX, RX e GND, ou seja, verificou-se que o controle de fluxo deve ser gerenciado por software.

Para que a comunicação fosse realizada, foi necessário então trabalhar a conexão dos pinos 7 e 8 do adaptador Bluecom, com o auxílio de um adaptador o qual foi denominado adaptador de controle de fluxo. Na figura 25 é ilustrado o esquema de ligação do adaptador.

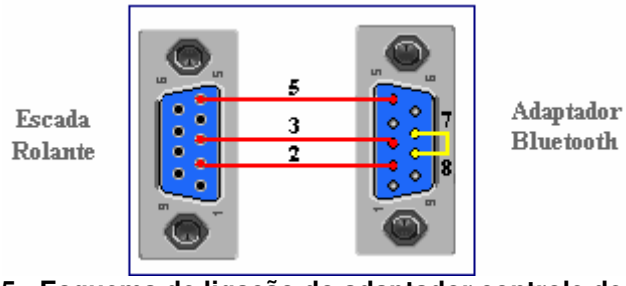

Figura 25 - Esquema de ligação do adaptador controle de fluxo

3.6 Componente de comunicação serial Delphi

Para o desenvolvimento do aplicativo de comunicação com a escada rolante foi utilizado a componente TSerialNG V2.0.15 desenvolvida por Ekkehard Domning (DOMIS, 2007) para ser utilizada na plataforma Delphi que permite que uma aplicação acesse a porta serial RS232.

Além da facilidade da integração da componente no aplicativo proposto neste trabalho existe a vantagem de ser uma componente pública.

Usa somente a WinAPI e as funções do Delphi (BORLAND, 2007), não necessitando de nenhum *software* auxiliar.

Esta componente permite configurar a porta de comunicação. Possibilita detectar o recebimento de dados por meio de evento. Possui eventos adicionais para sinalizar sinais e estados da porta de comunicação.

Características básicas da componente:

- Plataformas: Windows NT 4.0, Windows 2000, Windows 95, Windows 98
- Linguagens: Delphi 3,4,5,6 e 7
- Assíncrono ou síncrono (operação de leitura e escrita)
- Configuração do controle de fluxo
- Configuração do *timeout*
- Monitoração de eventos da porta de comunicação
- Terminal de comunicação

Na figura 26 é ilustrada as propriedades e os eventos disponíveis na componente TSerialNG.

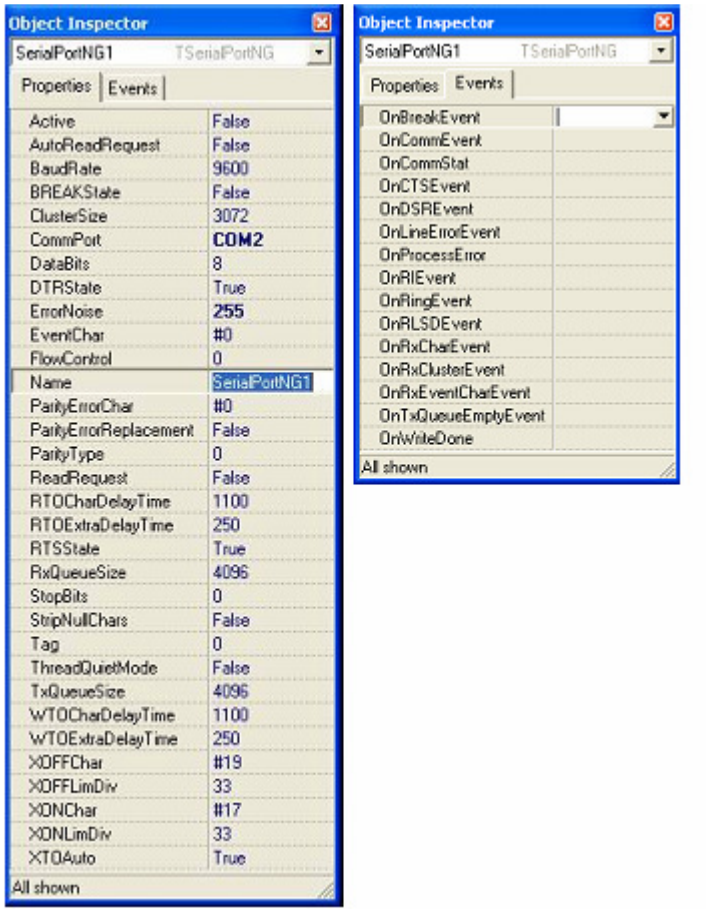

Figura 26 - Componente TSerialNG

3.7 Software Docklight (Analisador de Protocolos)

O *Software* Docklight (DOCKLIGHT, 2007) é uma ferramenta de simulação para análise de protocolo de comunicação serial (RS232, RS485/422 e outros). Permite monitorar a comunicação entre dois dispositivos consecutivos ou testar a comunicação serial de um único dispositivo. Pode ser executado nos sistemas operacionais Windows 98, Windows 2000, Windows NT ou Windows XP.

Tem como principal função simular um protocolo serial permitindo o envio e monitoração de mensagens de acordo com o padrão de comunicação gerado pelo usuário, possibilitando verificar o comportamento de um determinado dispositivo.

4 MÉTODO DE EXTRAÇÃO DO PROTOCOLO DE COMUNICAÇÃO

O pleno conhecimento do protocolo de comunicação da escada rolante, ou de qualquer outro equipamento que também possua um conjunto de regras para uma comunicação por meio da interface RS232, será imprescindível para o desenvolvimento de qualquer programa de monitoração alternativo ao programa original do fabricante.

Porém, normalmente o protocolo de comunicação de um equipamento não está disponível ou é desconhecido pelo desenvolvedor do programa alternativo. A dificuldade da extração do protocolo aumenta bastante quando o fabricante do equipamento não está envolvido no projeto ou quando também desconhece o protocolo.

Devido a variação de tecnologias entre os fabricantes de escadas rolantes onde cada um possui um protocolo de comunicação proprietário obedecendo a um conjunto de regras bem distintas, o desenvolvimento de uma metodologia de extração de protocolo é essencial para o mapeamento dos sinais de comunicação da placa de comando da escada rolante possibilitando desta forma o desenvolvimento de um aplicativo alternativo. Para aumentar a possibilidade de sucesso na aplicação desta metodologia é fundamental que além da placa de comando da escada rolante seja adquirido também o *software* de comunicação original do fabricante, pois em muitos casos é necessário que se conheça determinadas regras de comunicação as quais algumas são enviadas somente pelo emissor e outros somente pelo receptor funcionando como uma chave secreta.

Esta metodologia poderá também ser aplicada em outros equipamentos que possuam um protocolo de comunicação via interface RS232 e quando possível juntamente com o seu *software* de comunicação original.

A Figura 27 apresenta um fluxograma ilustrando as etapas essenciais para a extração do protocolo de comunicação de um equipamento.

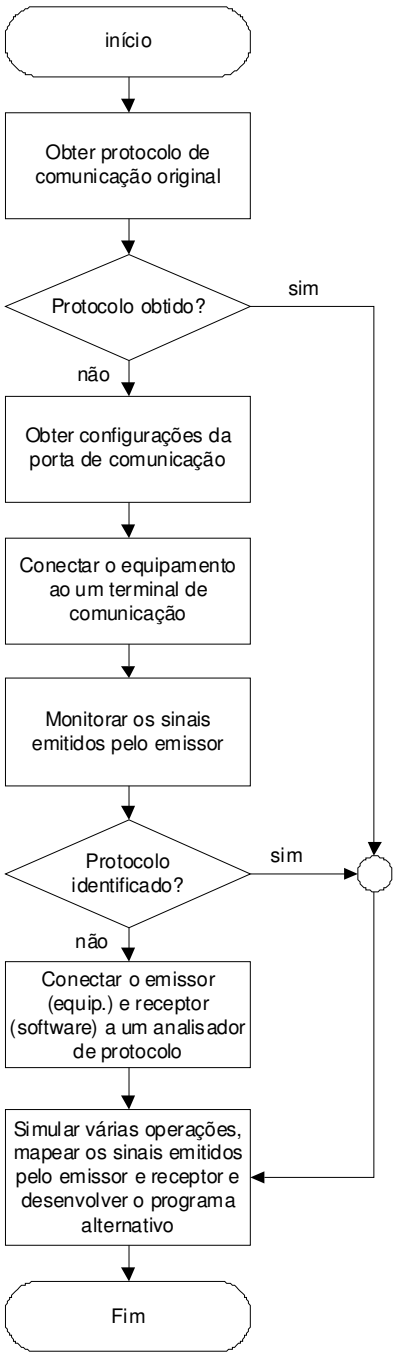

Figura 27 - Metodologia de Extração de Protocolo

A fim de verificar a viabilidade da metodologia de extração de protocolo apresentado neste trabalho, a metodologia foi aplicada na extração do protocolo de comunicação de uma escada rolante cujo protocolo é desconhecido, desta forma simulando a situação mais complexa de uma extração.

Para a monitoração do tráfego de dados da escada rolante foi utilizado um *software* analisador de protocolo que permite a verificação dos sinais emitidos ou recebidos pela escada rolante nos formatos caracteres, decimal, hexadecimal ou binário além de registrar o intervalo de tempo (em milisegundos) entre cada sinal. Com o objetivo de facilitar a visualização dos dados recebidos, o sistema de numeração adotado para a visualização dos dados foi o hexadecimal.

4.1 Software de comunicação original da Escada Rolante

A escada rolante estudada neste trabalho possui um *software* desenvolvido pelo fabricante que permite a comunicação com a sua placa de comando por meio da interface RS232 possibilitando a leitura e escrita de parâmetros de configuração e de códigos de falhas. Este programa é utilizado somente pelo departamento de engenharia do fabricante, ou seja, os técnicos de manutenção ou empresas de manutenção alternativas não têm acesso ao sistema.

O *software* e o protocolo de comunicação são proprietários impossibilitando qualquer adaptação no programa original.

Com a finalidade de conhecer o protocolo de comunicação da escada rolante a primeira busca foi o manual do fabricante o qual não fornecia nenhuma informação.

O *software* de comunicação do fabricante possui algumas limitações:

- Devido a falta de configuração do seu *timeout* torna-se impossível a integração da tecnologia *Bluetooth* no sistema.
- O programa mostra somente os códigos de falhas dificultando a interpretação pelos técnicos inexperientes.
- Não existe uma interface “amigável” para a identificação das falhas.
- *Software* desenvolvido por terceiros, impossibilitando autonomia para modificações no *software*.

4.2 Obtendo a configuração da porta de comunicação

Uma das etapas importantes para a extração do protocolo de comunicação é a configuração da porta de comunicação, ou seja, o terminal e o equipamento monitorado devem ter a mesma configuração a fim de permitir que as informações possam ser transmitidas ou recebidas.

Normalmente os *software*s de monitoração/controle permitem o ajuste dos parâmetros para a comunicação da porta de comunicação facilitando a obtenção destes dados. Porém, neste caso, o *software* da escada rolante possui uma configuração constante e pré-definida, que impossibilita a identificação desta configuração via *software*.

Como o fabricante também desconhecia esta informação, a solução encontrada para minimizar a busca da configuração da porta, foi pesquisar as possíveis configurações do microcontrolador da placa de comando da escada rolante. Como o

modelo do microcontrolador é do tipo convencional foi encontrado facilmente o seu *datasheet* permitindo a verificação das possíveis configurações da sua porta de comunicação (bps, bits de paridade, paridade, bits de parada e controle de fluxo).

Baseado nas características do microcontrolador foram constatados somente 5 possibilidades de configuração (MC68332 User´s Manual). Com o objetivo de encontrar os parâmetros de comunicação definido pelo fabricante, a escada rolante foi conectada a um terminal de comunicação convencional (hyperterminal) cujos parâmetros foram alternados até que a comunicação da escada rolante com o terminal fosse estabelecida.

Para a comunicação da placa de comando da escada rolante com o *software* analisador de protocolos foi confeccionado um cabo de comunicação utilizando dois terminais DB9 fêmea um em cada extremidade. Após a efetivação da comunicação, na primeira análise dos dados recebidos pelo terminal, foi observado que durante o *reset* da placa de comando da escada rolante é enviado uma palavra com uma largura de 7 caracteres por cinco vezes cessando a comunicação logo em seguida. A figura 28 ilustra este exato momento.

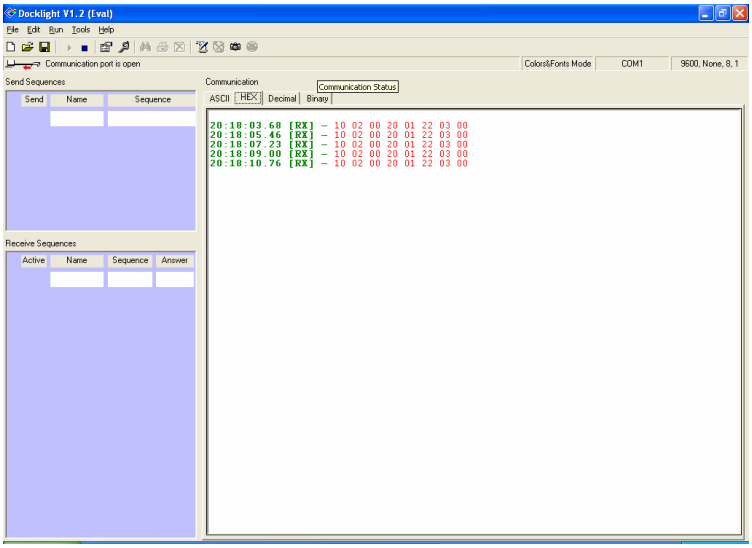

Figura 28 - Conexão entre a placa de comando da escada rolante e o software analisador de protocolos

Foram realizados vários testes, acreditando-se que a placa de comando ficaria constantemente enviando um sinal com o status atual da escada rolante, mas após inúmeros testes, inclusive gerando algumas falhas a fim de detectar alguma alteração no sinal de recepção, chegou-se a conclusão que a placa de comando da escada rolante aguarda um sinal (chave secreta) para que a comunicação seja estabelecida permitindo o acesso ao status da escada rolante.

A fim de verificar a existência de um sinal de retorno para que a placa de comando estabelecesse uma comunicação foi necessária a monitoração da comunicação entre a escada rolante e o *software* do fabricante. Para a realização desta monitoração, é necessária a confecção de uma derivação do cabo de comunicação possibilitando a visualização do tráfego de sinais entre o *software* do proprietário e a placa de comando da escada rolante, por meio de um *software* analisador de protocolo.

O analisador de protocolo escolhido para a monitoração dos sinais emitidos pela escada rolante, o Docklight, possibilita a visualização simultânea de dois

dispositivos. Porém, para que esta visualização seja possível o *software* analisador de protocolo precisa estar instalado em um PC que possua, pelo menos, duas portas seriais.

Caso o PC que irá monitorar a comunicação (PC Monitor) possua somente uma porta serial, é possível a aquisição e a instalação de uma placa multiserial, com duas portas seriais, permitindo a monitoração dos sinais emitidos tanto pelo *software* proprietário instalado em outro PC (PC de controle) quanto os sinais emitidos pela placa de comando da escada rolante. A figura 29 ilustra a arquitetura montada para extração do protocolo da escada rolante.

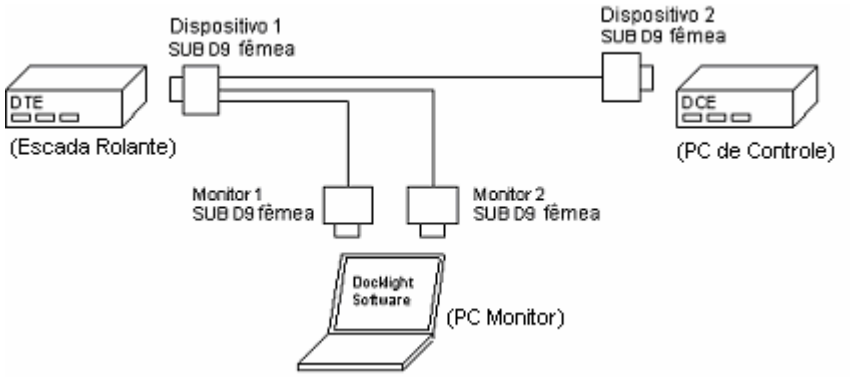

Figura 29 - Arquitetura para extração do protocolo da Escada Rolante

No conector ligado ao PC de controle o pino 2 representa o sinal RX ou recebimento de dados, o pino 3 o sinal TX ou transmissão de dados e o pino 5 o sinal de terra.

No conector ligado a placa de comando da escada rolante os pinos 2 e 3 tem uma representação invertida (condição original do cabo de comunicação do equipamento).

Com o intuito de monitorar os sinais enviados pelos dois dispositivos será necessário um terceiro PC chamado de PC monitor o qual irá monitorar os sinais emitidos pelos dispositivos.

Para que o PC monitor pudesse analisar os dois sinais em tempo real foi instalada uma placa multiserial com duas portas seriais reconhecidas como COM2 e COM3.

Na extensão do cabo temos mais dois conectores monitores: um recebendo sinal TX do PC de controle e o outro recebendo o TX da placa de comando da escada rolante.

Na figura 30 é ilustrado o esquema de pinagem para a montagem do cabo derivador.

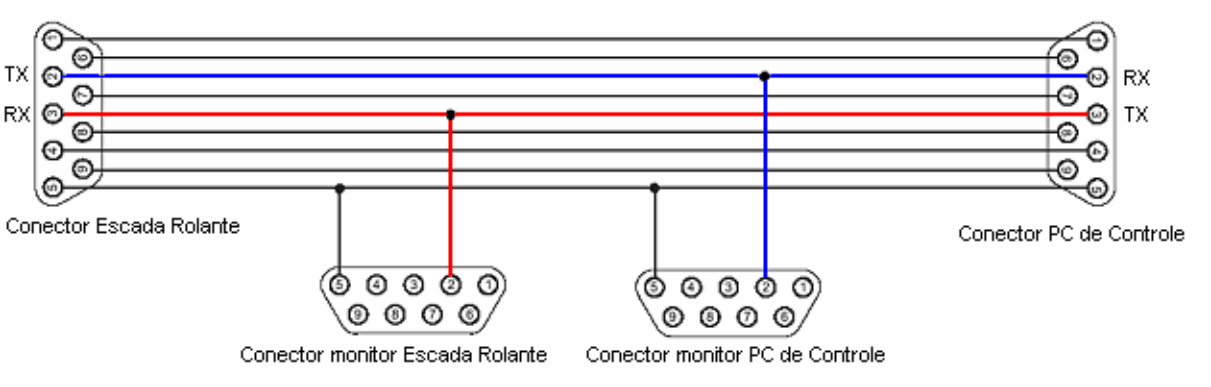

Figura 30 - Esquema de pinagem do cabo derivador

O primeiro conector, denominado conector PC de controle, foi ligado à saída RS232 da porta serial do PC de controle associado à porta COM1, ou seja, o PC que possui o *software* do fabricante da escada rolante instalado. O segundo conector, denominado de conector escada rolante, foi ligado na saída RS232 da placa de comando da escada rolante. O terceiro conector, chamado de conector monitor PC de controle, foi ligado na primeira saída RS232 da multiserial instalado no PC monitor associado à COM2. O quarto conector, chamado de conector monitor escada rolante, foi ligado na segunda porta serial da multiserial instalada no PC monitor e associado à porta COM3.

O PC monitor foi ligado e por meio de uma aplicativo analisador de protocolo as portas COM2 e COM3 foram configuradas de acordo com os parâmetros de comunicação entre a escada rolante e o *software*.

A escada rolante foi ligada e novamente o *frame*[9] de inicialização foi enviado, porém, desta vez, constatou-se que o PC de controle também enviou um frame de resposta efetuando a comunicação sem interrupção.

Desta forma, será possível gerar o mesmo frame de retorno por meio de um aplicativo alternativo mantendo a comunicação com a escada rolante.

4.3 Mapeamento dos Códigos de Falhas

Uma vez que a chave para a efetivação da comunicação entre a escada rolante e o *software* de monitoração e controle do fabricante foi encontrada, o foco é direcionado para o mapeamento das falhas. É uma etapa que exige bastante trabalho, pois, será necessário simular cada falha e observar a sua reação no protocolo de comunicação.

Como experimento foram geradas seis falhas relacionadas a segurança e coletados os códigos gerados pela placa de comando. Como comprovação, foram geradas as mesmas falhas em diversas situações e constatadas sempre a mesma codificação.

Para o mapeamento das falhas foi realizada a seguinte seqüência: Um computador, com o *software* analisador de protocolos, devidamente instalado e configurado de acordo com os parâmetros de comunicação da escada rolante foi conectado a placa de comando da escada rolante por meio de um cabo utilizando a porta serial. Em seguida, a escada rolante foi ligada no modo de subida e a chave de conexão, discutida na seção anterior, foi manualmente inserida no analisador de protocolos e enviada para a placa de comando fechando desta forma a conexão com a escada

[9] *Frame* – Pacote transmitido por uma linha serial. O tempo é derivado de um protocolo orientado a caractere que adiciona caracteres especiais de início e fim de frames na transmissão de pacotes. (CYCLADES BRASIL, 2000)

rolante. Na seqüência, foi acionado um contato de segurança manualmente gerando a paralisação da escada rolante e coletando o *frame* enviado para o analisador de protocolo o qual era anotado e associado à falha gerada. Como comprovação, a escada rolante era colocada em funcionamento novamente e o mesmo contato de segurança era acionado, constatando o mesmo frame enviado anteriormente. Foram realizados vários testes em seis contatos de segurança constatando que cada contato de segurança possui um *frame* único.

Com a chave que mantém a comunicação ativa e o conhecimento dos códigos de falhas gerados pela placa de comando da escada rolante torna-se possível o desenvolvimento de um sistema alternativo.

5. ANÁLISE COMPARATIVA ENTRE O PROCEDIMENTO ATUAL X SISTEMA PROPOSTO.

Neste capítulo será possível efetuar uma análise comparativa entre o procedimento atual para verificação do estado da escada rolante e o sistema proposto. Serão realizadas duas simulações citando todas as etapas necessárias para a realização de cada uma delas.

5.1 Simulação do procedimento atual

A idéia principal desta simulação é constatar as dificuldades, riscos e o tempo para a execução da manutenção no processo atual de manutenção de escadas rolantes. Como experimento, foi gerada na escada rolante uma falha referente ao contato de segurança na entrada do corrimão na parte inferior, o qual é bastante comum, principalmente por ser alvo de vandalismo. A mesma falha será utilizada na simulação utilizando o sistema proposto neste trabalho a fim de permitir uma análise comparativa. Na figura 31 temos todas as etapas para a solução desta falha utilizando o procedimento atual as quais serão comentadas logo a seguir.

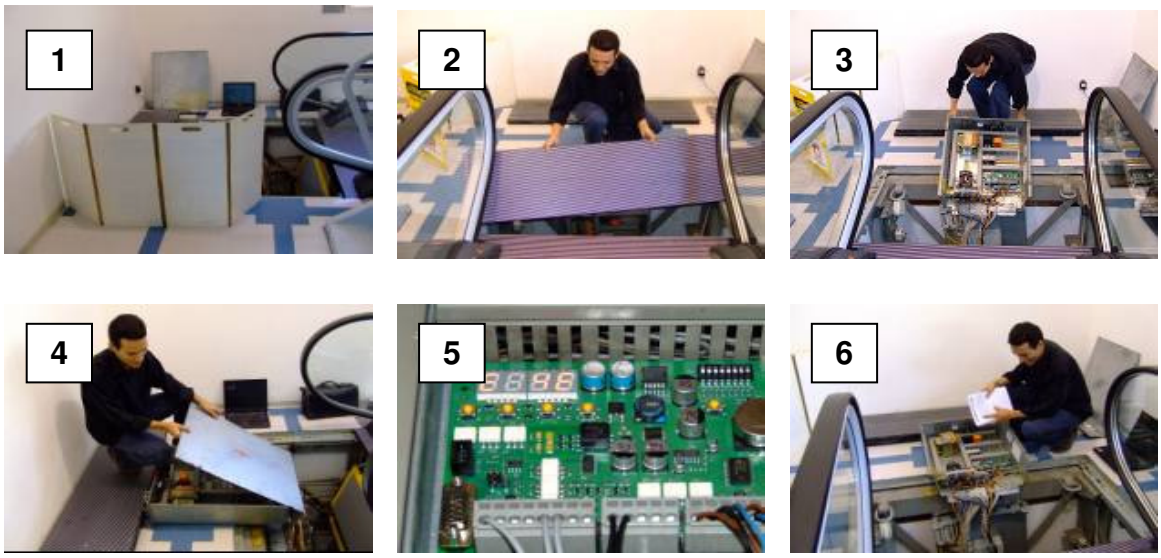

Figura 31 - Etapas para verificação do estado da escada rolante

Na etapa 1, a escada rolante, antes de qualquer ação, deve ser isolada tanto na parte inferior como na parte superior. Esta etapa toma muito tempo para a execução da manutenção, pois os bloqueios ficam em posse do estabelecimento e na maioria das vezes não estão próximos a escada rolante. Outra grande dificuldade é a paralisação da escada em ambientes de grande fluxo de pessoas.

Na etapa 2, somente após a execução da etapa 1, a tampa da plataforma inferior deve ser aberta com utilização de uma ferramenta especial. Devido ao peso desta tampa, além do desconforto para retirá-la, existe um grande risco de acidentes graves caso a tampa escape da mão do técnico.

Na etapa 3, o armário de comando é retirado da escada rolante pelo técnico. Além do desconforto devido ao peso do armário, também existe o risco de acidente.

Na etapa 4, utilizando uma chave de fenda a tampa do armário é retirada a fim de acessar a placa de comando. O armário possui 4 parafusos.

Na etapa 5, o técnico visualiza o *display* localizado na placa de comando para leitura do código da falha por meio de botões de controle. Nesta etapa, existe o risco da

placa ser danificada, pois o técnico tem contato com os dispositivos eletrônicos além de ficar exposto a componentes de alta tensão.

Na etapa 6, o técnico consulta um manual de instrução para decodificar a falha gerada pela placa de comando.

Em seguida a falha é solucionada e a escada rolante é colocada a disposição do público novamente.

Nesta simulação foram gastos 35 minutos para a realização de todas as etapas.

5.2 Simulação do sistema proposto

Antes da simulação da manutenção da escada rolante utilizando o sistema proposto neste trabalho será apresentada a configuração da escada rolante necessária para permitir a realização do ensaio. A figura 32 ilustra a arquitetura do sistema proposto neste trabalho a qual foi reproduzida nesta simulação.

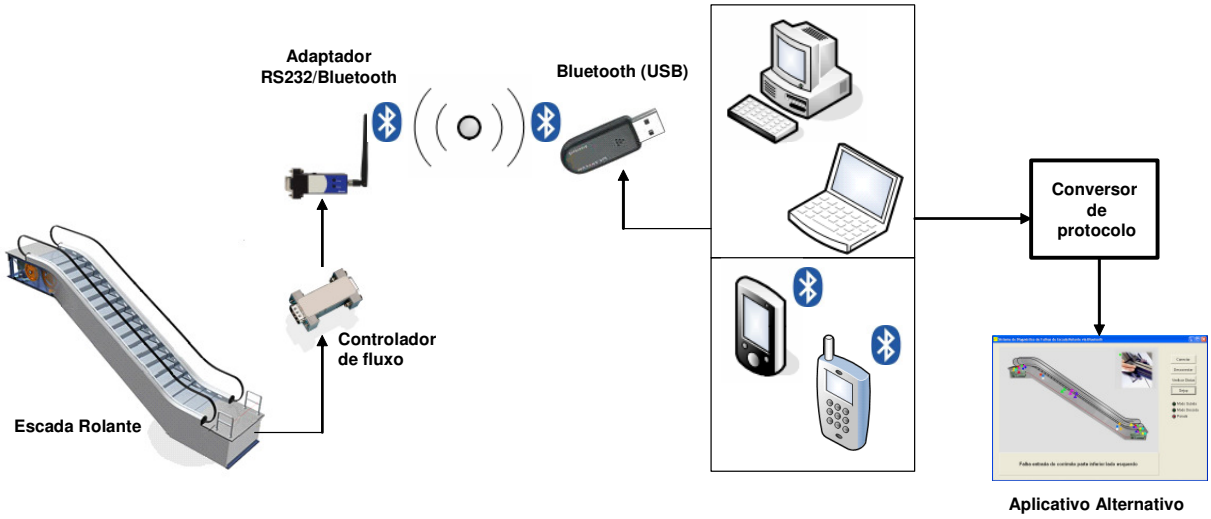

Figura 32 - Arquitetura do sistema proposto

5.2.1 Preparação da escada rolante

Para que a escada rolante possa enviar os sinais via sistema de comunicação *Bluetooth* quando solicitado pelo sistema de inspeção, primeiro será necessário a instalação do adaptador de controle de fluxo e posteriormente a instalação do adaptador Bluecom. O adaptador Bluecom foi previamente configurado de acordo com os parâmetros de comunicação da escada rolante (taxa de transferência, bit de paridade e bit de parada) coletados por meio do método de extração de protocolo comentado no capítulo 4. O adaptador Bluecom pode ser alimentado por meio de uma fonte externa ou por meio do pino 9, mas como o pino 9 da saída RS-232 da placa de comando da escada rolante não possui uma alimentação, será necessária a utilização de uma fonte externa que poderá, futuramente, ser ligada no próprio armário de comando. Como o adaptador Bluecom foi definido como *Slave* (escravo) a escada rolante ficará no aguardo de uma solicitação de conexão efetuado pelo dispositivo *Bluetooth* definido como *Máster* (Mestre), cujo endereço já está estabelecido e pré-configurado, não existindo o risco de outro dispositivo Bluetooth efetuar uma conexão com a escada rolante indevidamente.

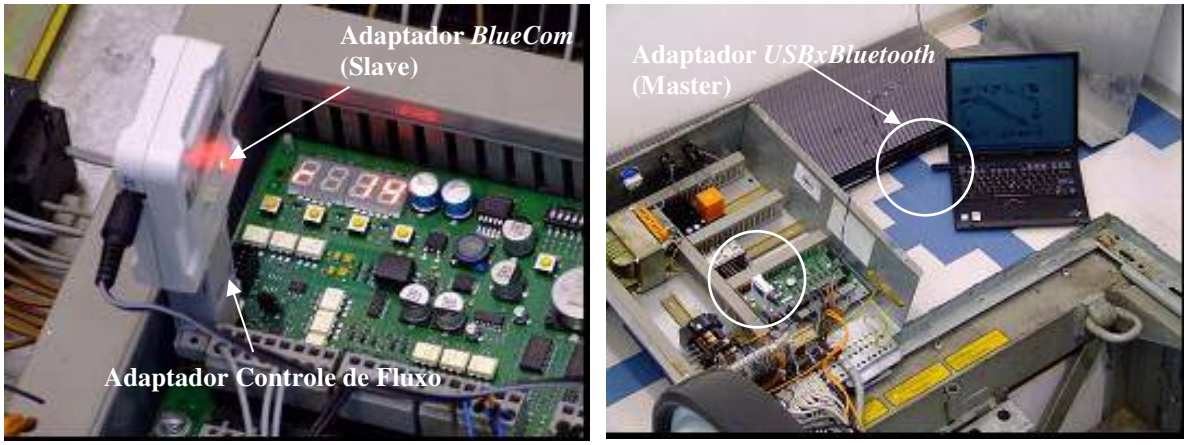

Figura 33 - Ligação dos adaptadores na placa de comando e no terminal

5.2.2 Preparação do terminal

Para efetuar uma conexão com a escada rolante, o adaptador *USBxBluetooth* deverá ser instalado no laptop em uma porta USB o qual deverá ser reconhecido automaticamente pelo software *BlueSoleil* instalado no laptop. Por meio do sistema de inspeção proposto a conexão com a escada rolante será efetuada imediatamente após a escolha da opção conectar (botão conectar), desde que os parâmetros de comunicação estejam coincidindo com os parâmetros de comunicação definidos na escada rolante. A partir desse momento a escada rolante estará pronta para enviar o seu estado atual a cada solicitação efetuada no laptop (botão solicitação de status). Na figura 34 podemos observar a conexão do sistema *BlueSoleil*, instalado no terminal (laptop), com a escada rolante.

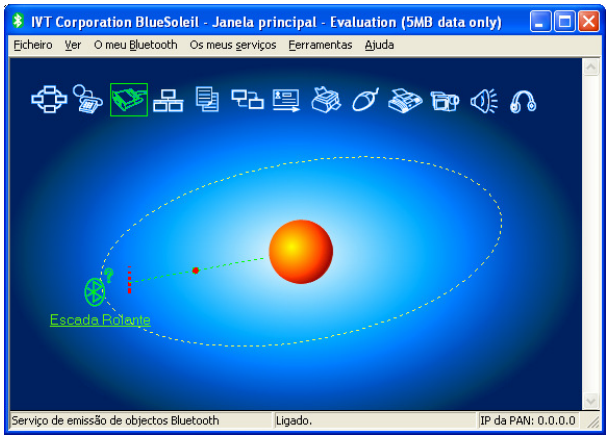

Figura 34 - Conexão *BlueSoleil* x Escada Rolante

5.2.3 Simulação de falha técnica na escada rolante

A fim de simular uma situação de uma falha técnica, a escada rolante foi colocada em operação no modo de subida, posteriormente, foi solicitado o estado da escada por meio do sistema de inspeção, o qual sinalizou a operação normal em modo de

subida. Simulando uma situação corriqueira, o contato do corrimão posicionado na parte inferior da escada rolante do lado esquerdo foi acionado, simulando um objeto estranho preso no corrimão. Após a paralisação da escada, devido ao contato de segurança de corrimão ter sido acionado, foi solicitado novamente o estado atual da escada, cujo sistema de inspeção detectou e interpretou a falha corretamente e inclusive ilustrando a localização do defeito. Nessa situação, o técnico simplesmente iria verificar se existe algo na entrada do corrimão e reinicializaria a escada rolante sem a necessidade de executar o seguinte procedimento: isolamento da escada rolante, abertura do patamar, retirada do armário de comando, abertura do armário de comando, verificação do IHM da placa de comando, decodificação da falha e execução do serviço. Na figura 35 podemos observar o comportamento do sistema de inspeção no momento da falha citada nesta seção.

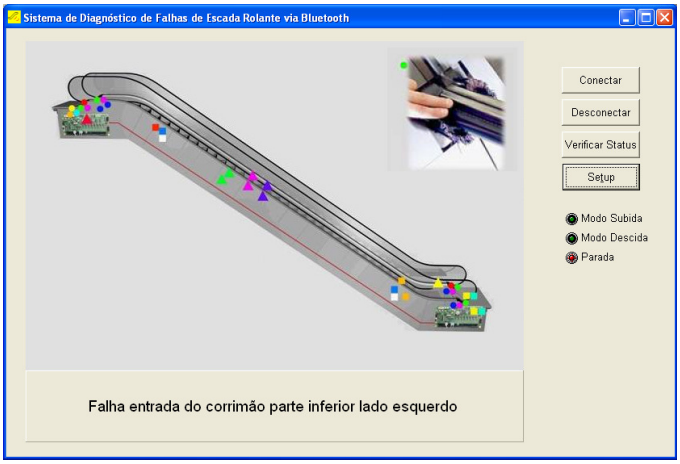

Figura 35 - Falha interpretada pelo sistema de inspeção

6 CONCLUSÃO

Após realizar os ensaios com o sistema ligado à escada rolante e simular diversas situações de não conformidade, verificou-se:

- Nos casos em que o sistema de inspeção não detectava problemas na escada rolante, o tempo de parada caiu para zero, uma vez que não houve a necessidade de isolamento para acesso ao comando eletrônico da escada rolante.
- Houve uma grande queda no tempo de reação dos técnicos às mensagens mostradas pelo sistema, que era de cerca de 35 minutos, considerando o isolamento da escada rolante, a abertura do painel de controle, leitura e identificação da não conformidade e o início da ação corretiva, para 15 minutos, com os técnicos necessitando somente realizar o isolamento da escada e o início da ação corretiva.
- Houve um aumento na segurança que os técnicos demonstraram ao interpretar as mensagens mostradas pelo sistema.

Não foi feito um levantamento do aumento do MTBC neste trabalho, uma vez que seria necessário que várias escadas tivessem o sistema acoplado e o tempo médio fosse levantado ao longo de um período considerável de tempo.

Na simulação efetuada com o sistema proposto neste trabalho, foi constatada uma grande oportunidade de ganho de produtividade dos técnicos bem como um aumento significativo de inspeções na escada rolante sem a necessidade de sua interrupção possibilitando um aumento no índice MTBC. Na simulação efetuada, o tempo que demandaria para a verificação da falha de contato do corrimão seria de aproximadamente de 30 a 40 minutos, pois o técnico teria que localizar e posicionar

os bloqueios de segurança, abrir a plataforma inferior, retirar o armário de comando da escada rolante, abrir a tampa, verificar e interpretar o código de falha sinalizado pelo IHM utilizando um manual de falhas. Porém, utilizando o sistema proposto a falha foi detectada em apenas 5 minutos. Nos casos em que não se detectou falhas, a escada continuou à disposição dos usuários.

Como etapas futuras no desenvolvimento do presente sistema podem ser citadas: o desenvolvimento de um sistema de controle que permita acionar o controle eletrônico da escada à distância, uma nova versão preparada para ser executada em dispositivos portáteis tipo *handhelds*, como os *palmtops* ou *Pocket PC*, os quais apresentam menores dimensões e maior portabilidade para os técnicos e realizar estudos efetivos para o levantamento dos índices de MTBC.

REFERÊNCIAS

ABOUT ESCALATORS. Disponível em: <http://www.otis.com/site/br/OT_DL_Documents/OT_DL_SiteDocuments/Bb4_About Escalators.pdf>. Acesso em: 4 .ago. 2008.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 8900: Projeto, fabricação e instalação de escadas rolantes. Rio de Janeiro, 1995. 34p.

AU-SYSTEM. *Bluetooth* Whitepaper 1.1. 2000.

BAATZ, SIMON. "*Bluetooth* Scatternets: An Enhanced Adaptive Scheduling Scheme". New York. Proc. 21. annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2002, 2002.

BAATZ, SIMON; FRANK, M.; KUHL, C.; MARTINI, P.; SCHOLZ, C. "Adaptive Scatternet Support for *Bluetooth* using Sniff Mode". Tampa, Proc. 26. Annual Conference on Local Computer Networks, 2001.

BHAGWAT, PRAVIN. "*Bluetooth*: Technology for Short-Range Wireless Apps". IEEE Internet Computing, p. 96-103, 2001.

BISDIKIAN, CHATSCHIK. "An Overview of the *Bluetooth* Wireless Techonology". IEEE Communications Magazine, p. 86-93, 2001.

BLUESOLEIL. BlueSoleil – Standard version. Disponível em: www.bluesoleil.com. Acesso em: 15.jan.2007.

BLUETOOTH SIG-a. Specification of *Bluetooth* System – Core. Specification Volume 1. Version 1.1. pp.1084, 2001. http://www.*Bluetooth*.com

BLUETOOTH SIG-b. Specification of *Bluetooth* System – Profire. Specification Volume 2. Version 1.1. pp.452, 2001. http://www.*Bluetooth*.com

BORLAND. Delphi 2007. Disponível em: www.borland.com. Acesso em: 10.mar.2007.

CYCLADES BRASIL. Guia Internet de Conectividade. SENAC, 2000.

DOCKLIGHT. RS232 Terminal / RS232 monitor. Disponível em: www.docklight.de. Acesso em: 27.jan.2007.

DOMIS. TSerialNG V2.0.1.5 desenvolvida por Ekkehard Domming. Disponível em: www.domis.de. Acesso em: 3.fev.2007.

ELEVADORES ATLAS SCHINDLER. Matriz São Paulo. Informação verbal obtida em jun. 2007.

HELD, GILBERT. Data Over Wireless Networks – *Bluetooth*. WAP & Wireless LANs. McGrawn-Hill, pp.344, 2001.

MC68332. User´s Manual. Disponível em: <http://www.freescale.com>. Acesso em: 4 .ago. 2008.

MILLER, BRENT A.; BISDIKIAN, C. Bluetooth Revealed – The Insider´s Guide to an Open Specification for Global Communications. Prentice-Hall, 2001.

MILLER, MICHAEL. Descobrindo Bluetooth. Editora Campus. 2001.

ROSS, J. Redes de Computadores. São Paulo: LVTEC, 2008.

WIKIPEDIA. Disponível em: <http://www.wikipedia.org>. Acesso em: 4 .ago. 2008.