



EMERSON PEREIRA DE CASTRO

**ESTUDO E DESENVOLVIMENTO DE UMA
PLATAFORMA DE COLETA, ANÁLISE E
VISUALIZAÇÃO DE DADOS
GEORREFERENCIAIS APLICADOS AO SETOR
DE TRANSPORTE PÚBLICO: MÓDULO DE
COLETA DE DADOS**

**LAVRAS - MG
2010**

EMERSON PEREIRA DE CASTRO

**ESTUDO E DESENVOLVIMENTO DE UMA PLATAFORMA DE
COLETA, ANÁLISE E VISUALIZAÇÃO DE DADOS
GEORREFERENCIAIS APLICADOS AO SETOR DE TRANSPORTE
PÚBLICO: MÓDULO DE COLETA DE DADOS**

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

Orientador:

Dr. Ahmed Ali Abdalla Esmín

**LAVRAS - MG
2010**

EMERSON PEREIRA DE CASTRO

**ESTUDO E DESENVOLVIMENTO DE UMA PLATAFORMA DE
COLETA, ANÁLISE E VISUALIZAÇÃO DE DADOS
GEORREFERENCIAIS APLICADOS AO SETOR DE TRANSPORTE
PÚBLICO: MÓDULO DE COLETA DE DADOS**

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

APROVADA em ____ de _____ de _____

Dr. João Carlos Giacomini UFLA

Dr. Raphael Winckler de Bettio UFLA

Dr. Ahmed Ali Abdalla Esmim

Orientador

**LAVRAS - MG
2010**

AGRADECIMENTOS

Em primeiro lugar agradeço a todos meus familiares, em especial meus pais e meu irmão, que sempre me apoiaram.

Aos meus colegas da fantástica turma de 2007/1, pois nunca me deixaram na mão, nem em dias de bebedeira e nem nas horas que antecedem as provas.

Aos professores da UFLA, que além de tudo me ensinaram o mais importante: como aprender por conta própria.

Aos funcionários da UFLA, pois sem eles isso não seria possível.

Aos colegas de trabalho, que nos últimos meses me proporcionaram grandes momentos de aprendizagem.

A todos meus amigos, que simplesmente fazem parte da minha vida.

RESUMO

O serviço de localização e monitoramento de veículos via satélites é bastante comum em empresas de transporte rodoviário, devido ao grande número de roubo de cargas que ocorrem durante as viagens, e em frotas de ônibus para uma melhor eficiência nas estimativas de custos e localização de veículos com problemas mecânicos. O projeto proposto é composto por uma plataforma que fará o monitoramento da posição geográfica em tempo real de uma frota de veículos do transporte público. Dessa forma usuários do transporte público poderão visualizar através da web e de seus celulares informações relevantes como previsão de embarque, duração da viagem e localização do veículo. O objetivo deste trabalho é projetar o módulo de coleta de dados. Este equipamento é composto por um receptor de GPS, um rádio GPRS, um microcontrolador e o software embutido, formando um sistema embarcado.

Palavras-chave: GPS, GSM, GPRS, sistemas embarcados, georreferenciamento

ABSTRACT

The tracking and monitoring service of vehicles via satellite is common because of the large number of cargo thefts that occur during trips, and bus fleets to improve efficiency in cost estimates and location of vehicles with mechanical problems. The proposed project consists of a platform that will real-time monitoring geographic location of a fleet of public transport vehicles. Public transport users can view the web through their mobile phones and relevant information such as estimates of shipment, the drive time and vehicle location. The objective is to design the module for data collection. This equipment involves of a GPS receiver, a GPRS radio, a microcontroller and embedded software, creating an embedded system.

Keywords: monitoring, tracking, satellites, GPS, GSM, GPRS, embedded systems, georeferencing

LISTA DE ILUSTRAÇÕES

Figura 1: Arquitetura Geral da Plataforma de Monitoramento.....	12
Figura 2: Segmentos do GPS.	17
Figura 3: Módulo GSM/GPRS.....	20
Figura 4: Google Maps.	25
Figura 5: Módulos da Plataforma de Monitoramento.....	29
Figura 6: Arquitetura do Colector Track.....	30
Figura 7: Fluxograma de Funcionamento do Módulo de Coleta.	31
Figura 8: Teste de Validação do Rastreamento.	33
Figura 9: Módulo Colector Track.....	34

LISTA DE ABREVIATURAS

GPS	<i>Global Positioning System</i>
GPRS	<i>General Packet Radio Service</i>
GSM	<i>Global System for Mobile Communications, ou Groupe Spécial Mobile</i>
API	<i>Application Programming Interface</i>
ITS	<i>Intelligent Transportation System</i>
DoD	<i>The United States Department of Defense</i>
DGPS	<i>Differential Global Positioning System</i>
PRN	<i>Pseudo Range Noise</i>
FM	<i>Frequency Modulation</i>
ISDN	<i>Integrated Services Digital Network</i>
CSD	<i>Circuit Switched Data</i>
SMS	<i>Short Message Service</i>
FTP	<i>File Transfer Protocol</i>
RISC	<i>Reduced Instruction Set Computing</i>
LED	<i>Light-Emitting Diode</i>
DSP	<i>Digital Signal Processing</i>
E2PROM	<i>Electrically Erasable Programmable Read-Only Memory</i>
AJAX	<i>Asynchronous JavaScript and XML</i>
SIG	<i>Sistemas de Informação Geográfica</i>
SFS	<i>Simple Features Specification</i>
OGC	<i>Open Geospatial Consortium</i>

SUMÁRIO

1	INTRODUÇÃO.....	9
1.1	Contextualização e Motivação	9
1.2	Objetivos Gerais	10
1.3	Objetivos Específicos	10
1.4	Estrutura do Trabalho	10
1.5	Arquitetura da Plataforma	11
2	REFERENCIAL TEÓRICO	14
2.1	Transporte Urbano e Sistema de Transporte Inteligente (ITS)	14
2.2	Sistema de Posicionamento Global (GPS).....	16
2.3	Tecnologia GSM / GPRS	19
2.4	Sistemas Embarcados	21
2.5	Google Maps API	24
2.6	Trabalhos Relacionados	26
3	METODOLOGIA.....	27
3.1	Tipo de Pesquisa.....	27
3.2	Procedimentos d Tecnologia	28
3.3	Arquitetura do Módulo de Coleta de Dados	29
4	RESULTADOS	33
4.1	Testes Realizados	33
4.2	Módulo Colector Track.....	34
5	CONCLUSÃO	36
5.1	Dificuldades Encontradas	36
5.2	Trabalhos Futuros.....	36
6	REFERÊNCIAS BIBLIOGRÁFICAS	38
7	APÊNDICE	40
7.1	<i>Firmware</i>	40
7.2	<i>Servidor de Testes</i>	47

1 INTRODUÇÃO

1.1 Contextualização e motivação

O serviço de localização e monitoramento de veículos via satélites é bastante comum em transportadoras, devido ao grande número de roubo de cargas que ocorrem durante as viagens, e em frotas de ônibus para uma melhor eficiência nas estimativas de custos e localização de veículos com problemas mecânicos. A tecnologia de comunicação vem evoluindo e com isso o desempenho de produtos, como o rastreador via satélites ou qualquer outro tipo serviço que demanda uma boa comunicação sem fio à distância. Hoje, é possível conectar-se à Internet por um custo aceitável com boas taxas de comunicação de qualquer lugar, que não seja excessivamente remoto, utilizando equipamentos móveis. Devido a isto, o rastreamento de veículos se torna cada vez mais popular.

Este serviço já é fornecido por muitas empresas que, usando a tecnologia do GPS (*Global Positioning System*) e um meio de comunicação, como a rede de telefonia celular, efetuam o monitoramento em tempo integral do caminho percorrido pelos veículos, fornecendo diversos relatórios para os clientes.

O setor de transporte público foi escolhido por se tratar de uma área que recebe investimentos modestos, que em sua grande maioria, são focados em tecnologias que evitem problemas de cobrança do serviço. Desta forma, torna-se necessário desenvolver tecnologias de baixo custo que tornem o transporte público mais eficiente e de melhor qualidade, trazendo ganhos tanto para o empreendedor quanto para o usuário. Outra motivação é que usando diversas técnicas computacionais, como Mineração de Dados e Realidade Aumentada, podemos diversificar o leque de serviços oferecidos em cima de uma plataforma de rastreamento.

1.2 Objetivos gerais

O objetivo deste trabalho é desenvolver um protótipo de um módulo de coleta de dados, incluindo hardware e software, que envie a posição geográfica de um veículo para um servidor web.

1.3 Objetivos específicos

Espera-se projetar e construir um equipamento eletrônico, usando um rádio GSM, um receptor de GPS e um microcontrolador, para receber sentenças de posicionamento geográfico e em seguida tratá-los devidamente. As informações extraídas devem então ser enviadas a um servidor específico na Internet.

Futuramente, espera-se integrar este módulo a uma plataforma de rastreamento, descrita no próximo tópico, que faz o monitoramento da posição geográfica em tempo real de uma frota de veículos do transporte público. Dessa forma, usuários do transporte público podem visualizar através da web e de seus celulares informações relevantes como previsão de embarque, duração da viagem e localização do veículo.

1.4 Estrutura do trabalho

O trabalho está estruturado nos seguintes tópicos:

- Transporte Coletivo Urbano e Sistemas de Transporte Inteligente – uma breve introdução aos conceitos de Transporte Urbano e como os ITS's funcionam;

- GPS: conceitos e funcionamento básicos do sistema;
- GSM/GPRS: apresentação da tecnologia GSM dando ênfase às características do GPRS;
- Sistemas Embarcados: alguns conceitos básicos sobre sistemas embarcados;
- Google Maps API: formas de utilização desta API da Google;
- Trabalhos Relacionados: apresentação de outros projetos semelhantes que foram estudados;
- Metodologia: como o trabalho foi desenvolvido, incluindo o hardware, firmware e softwares que foram desenvolvidos para atender os objetivos;
- Resultados e Conclusões: testes efetuados, dificuldades encontradas, as conclusões e os possíveis trabalhos futuros.

1.5 Arquitetura da Plataforma

Essa plataforma consiste em três módulos: Módulo de Coleta, Módulo Web e Módulo Móvel. A arquitetura geral pode ser vista na Figura 1.

O Módulo de Coleta será responsável por obter os dados GPS e enviá-los ao servidor com banco de dados para análise em tempo real. Esse módulo deve ser instalado no veículo que se deseja rastrear.

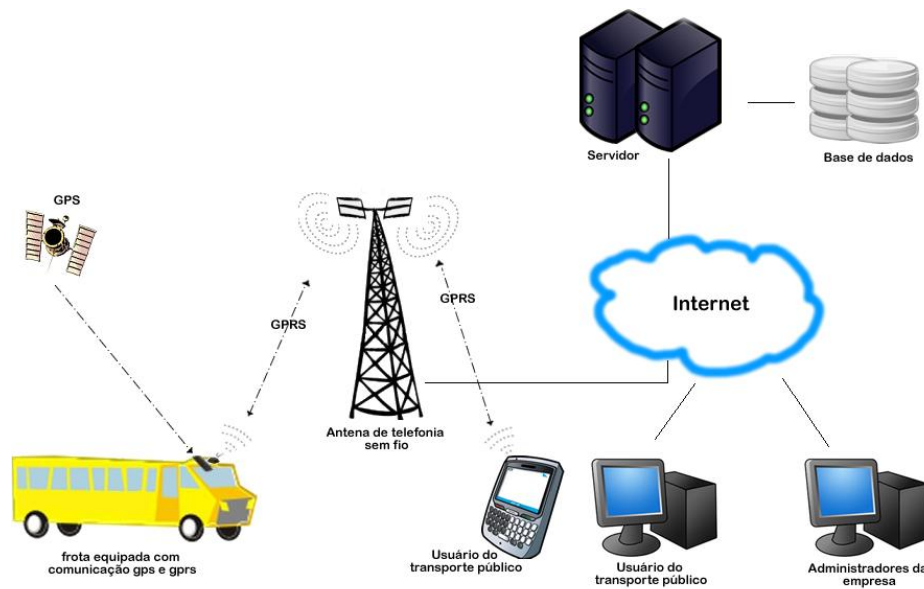


Figura 1: Arquitetura geral da plataforma de monitoramento.

O Módulo Web visa gerenciar os dados coletados no servidor para fins diversos, especialmente voltados para a administração e visualização dos dados obtidos em tempo real. Este módulo ainda contém uma aplicação para os administradores da empresa de ônibus e para os usuários. Usuários podem usufruir de funções relativas à visualização, como por exemplo, posição atual de cada ônibus da empresa, suas rotas, e tempo necessário para o ônibus chegar ao próximo ponto de parada. Os administradores podem visualizar tudo o que um usuário comum pode visualizar, e ainda contar com outras funções relativas ao gerenciamento da empresa, por exemplo, quantidade de quilômetros percorridos, dados do motorista e do ônibus, e velocidade atual. Ainda nesse módulo há uma interface para cadastro, alteração e exclusão de dados (ônibus, motoristas, rotas, etc.).

O Módulo Móvel está voltado para o usuário final com o objetivo de facilitar a integração e visualização dos dados em dispositivos móveis. É composto por uma aplicação para visualizar os dados em tempo real, como a posição atual do ônibus e tempo estimado para a chegada no ponto de parada, e outra aplicação utilizando realidade aumentada para informar a posição dos pontos de ônibus mais próximos ao usuário.

2 REFERENCIAL TEÓRICO

2.1 Transporte Urbano e Sistema de Transporte Inteligente (ITS)

A história do transporte coletivo urbano começa na França, entre o século 17 e 18. Nesta época, o rei vigente permitiu a exploração de várias rotas de viagens usando carruagens de tração animal. Esse serviço podia ser cobrado, e todas as carruagens tinham horários e rotas bem definidas. Mais tarde, S. Baudry passou a oferecer o mesmo serviço para seus clientes de uma casa de banhos. Entretanto, notando a grande utilidade, a população passou a utilizar os veículos para locomover sem a intenção de usar as casas de banho. Alguns anos depois, Baudry criou o *omnibus* juntamente com a Empresa Geral de Viaturas (STIEL, 2001).

Após a Revolução Industrial o Brasil teve um grande crescimento populacional, impactando diretamente o processo de urbanização. Com isso surgiu a necessidade de pensar no problema da locomoção da população. A primeira solução foi a importação de veículos de transporte. Esses veículos vinham da Europa apenas com o chassi e motor de caminhões. Toda a carroceria era montada localmente e encaixada sobre o chassi. Anos depois, os ônibus começaram a ser produzidos inteiramente no Brasil. A princípio, era um transporte quase exclusivo da classe média. Por esse motivo, foi necessário planejar formar de possibilitar o uso do sistema por todas as classes da população. (STIEL, 2001).

Com o enorme crescimento do uso do sistema surgiu a preocupação com a qualidade do serviço prestado. Motivo que fez as empresas optarem pela tecnologia como aliada. Foi criado então o conceito de Sistemas de Transporte Inteligente – ITS (*Intelligent Transportation System*). Estes sistemas utilizam informática, telecomunicações e automação para melhorar o desempenho e

produtividade dos meios de transporte (AQUINO, W.; AQUINO, N. B.; PEREIRA, 2001).

Atualmente sabe-se que a melhoria do sistema de transporte de um país não significa apenas a construção de novas estradas ou reparação de infra-estruturas envelhecidas. O futuro do transporte está cada vez mais no uso de TI (Tecnologia da Informação). A TI permite que os elementos dentro do sistema de transporte, veículos, estradas, semáforos, painéis de mensagem, etc, tornem-se inteligente ao integrá-las com microchips e sensores e capacitá-los para se comunicar uns com os outros através de tecnologias sem fio.

De acordo com MOON (2008), são sete as tecnologias chaves para o sucesso dos ITS:

- *Global Positioning System (GPS)*: Receptores de GPS incorporados em veículos para obtenção da sua posição geográfica.
- *Dedicated-Short-Range Communications*: Canais de comunicação sem fio de médio alcance operando no espectro de 5.8 ou 5.9GH, especificamente concebidos para aplicações automotivas.
- *Wireless Networks*: Essas redes possibilitam uma comunicação rápida entre veículos e estrada com um alcance de apenas algumas centenas de metros.
- *Mobile Telephony*: Aplicações ITS podem transmitir informações sobre o padrão 3G ou 4G de redes de telefonia móvel, ampliando a disponibilidade em cidades e ao longo das estradas principais.
- *Radiowave e Infrared*: Sistemas japoneses já utilizam sinais de microondas e infravermelho para comunicação em tempo real entre placas de informação.
- *Roadside Camera Recognition*: Câmeras com reconhecimento de padrões podem ser usadas na tarifação e outras aplicações nas estradas.

- *Probe Vehicles or Devices*: Os chamados “veículos de prova” espalhados pelo trânsito podem enviar informações de congestionamento ao ITS.

2.2 Sistema de Posicionamento Global (GPS)

O Sistema de Posicionamento Global (GPS) é um sistema de navegação baseado em satélite composto por uma rede de 24 satélites colocados em órbita. As respectivas estações terrestres são gerenciadas pelo Departamento de Defesa dos EUA (DoD). Foi criado em 1973 para uso militar, e a partir dos anos 80 teve seu uso por civis liberado, porém, com algumas restrições (UNSW, 2010).

Um receptor de GPS deve receber sinal de pelo menos três satélites para calcular uma posição 2D (latitude e longitude). Com quatro ou mais satélites em vista, o receptor pode determinar a posição 3D, obtendo a altitude além da latitude e longitude. Uma vez que a posição do veículo foi determinada, a unidade de GPS pode calcular outras informações, como velocidade, rota, distância de viagem, nascer e tempo de sol e muito mais. O GPS funciona sob quaisquer condições atmosféricas, em qualquer lugar do mundo, 24 horas por dia. Não existem comissões, *royalties*, ou qualquer restrição financeira para usar o GPS. Outros sistemas similares foram desenvolvidos para competir diretamente com o GPS. Porém, até a presente data, nem o europeu Galileo nem o chinês Compass está funcional.

O GPS é dividido basicamente em três seguimentos, como mostra a Figura 2¹ (UNSW, 2010):

- **Segmento Espacial**: Esta parte consiste em 24 satélites, fabricados pela Rockwell International. Eles têm tamanho aproximado de um carro. Cada satélite está em órbita a uma altitude de 20.200 Km e levam 12

¹ <http://www.overcar.com.br/como-funciona-o-gps/>

horas para uma volta completa ao redor do planeta. Existem seis planos orbitais cada um com 4 satélites. As órbitas são inclinadas ao equador da Terra em 55° para que haja cobertura das regiões polares. Os satélites orientam-se continuamente para garantir que os seus painéis solares fiquem apontados na direção do sol e suas antenas em direção à terra. Além disso, para manter a precisão, cada satélite transporta quatro relógios atômicos.



Figura 2: Segmentos do GPS.

- **Segmento de Controle:** Este seguimento é composto por estações autônomas que monitoram os satélites para rastrear sua posição exata no espaço e se certificar de que estão funcionando corretamente. As estações de monitoramento verificam constantemente as órbitas dos satélites e utilizam radares muito precisos para verificar a posição, altitude e velocidade. Os ajustes de relógio e correções de efeitos atmosféricos são transmitidos para os satélites que, por sua vez, usam essas atualizações nos sinais que eles para os receptores GPS.

- **O Segmento Usuário:** Seguimento constituído pela imensa gama de receptores de diversos tipos, desde portáteis, até os que podem ser instalados em um veículo. Os receptores GPS detectam, decodificam e processam os sinais recebidos dos satélites, calculando e armazenando então, a posição de cada satélite. O receptor é normalmente usado em conjunto com o software de computador para enviar as informações para o usuário na forma de um mapa. A precisão das coordenadas pode variar conforme a aplicação do receptor.

A mensagem de navegação GPS fornecidas aos usuários são obtidas por sinais transmitidos em duas frequências, L1 e L2, para possibilitar correções a serem feitas por atrasos devido à propagação do sinal na ionosfera e o tempo de resposta dos receptores.

A portadora L1 é modulada com um sinal de 10,23 MHz (chamado *P-code*) e um sinal de 1,023 MHz (chamado *C/A code*). Estes sinais são chamados *Pseudo Random Noise* (PRN). A portadora L2 é modulada apenas com P-code. Ambos os sinais L1 e L2 são também continuamente modulados num fluxo de 50 bits por segundo. O código P é para uso militar e está disponível apenas para usuários autorizados. O acesso ao GPS por usuários civis só pode ser obtido pelo *C/A code*. A mensagem de navegação consiste em um fluxo de 50 bits por segundo contendo informações que permitem o receptor realizar os cálculos necessários para a navegação. Cada satélite tem o seu próprio *C/A code* que o identifica possibilitando o rastreamento por parte do usuário (UNSW, 2010).

2.3 Tecnologia GSM / GPRS

Durante a década 80, os sistemas analógicos de telefonia celular estavam em uma fase de rápido crescimento na Europa, especialmente na Escandinávia, Reino Unido, França e na Alemanha. Cada país desenvolveu seu próprio sistema, que era incompatível com todos os outros em equipamentos e operação. Esta era uma situação um tanto quanto indesejável. Os equipamentos móveis eram limitados ao funcionamento dentro das fronteiras nacionais, e havia ainda um mercado muito limitado para cada tipo de equipamento.

Os europeus perceberam isso logo no início, e em 1982 formaram um grupo de estudo chamado de *Groupe Spécial Mobile* (GSM) para estudar e desenvolver um sistema móvel de comunicação. O sistema proposto tinha que cumprir certos critérios: boa qualidade de voz, baixo custo, suporte ao *roaming* internacional e suporte para uma gama de novos serviços.

Em 1989, a responsabilidade GSM foi transferida para o Instituto Europeu de Normalização das Telecomunicações (ETSI) e a fase I das especificações GSM foram publicados em 1990. O serviço comercial iniciou em meados de 1991, e em 1993 existiam 36 redes GSM em 22 países. Embora padronizada na Europa, esta tecnologia não é exclusivamente européia. No início de 1994, havia 1,3 milhão de assinantes em todo o mundo. Com a América do Norte fazendo uma entrada tardia no grupo GSM, e com diversos sistemas GSM existentes em cada continente, a sigla GSM passou a *Global System for Mobile Communications* (SILVENTOINEN, 1995).

Com investimento de muitos países, o GSM tornou-se um sistema tão completo que não se trata apenas de uma tecnologia de comunicação móvel. Diversos equipamentos foram criados, desde telefones celulares a pequenos módulos de comunicação (Figura 3). Além disso, toda uma gama de serviços foi

implementada sobre o padrão: troca de mensagens, tráfego de voz e vídeo, tráfego de dados, etc.



Figura 3: Módulo GSM/GPRS.

Entre os serviços implementados está o *General Packet Radio Service*, mais conhecido como GPRS: um serviço de tráfego de dados em velocidade relativamente alta, com tecnologia de comutação de pacotes, para o sistema GSM. Nele é possível enviar e receber tanto pequenos fluxos de dados, como email e navegação na web, como grandes volumes de dados através da uma rede de telefonia móvel.

O GPRS oferece uma conexão contínua à Internet para usuários de telefones móveis e computadores. Experiências mostraram que a maioria das aplicações de comunicação de dados não necessitam transferência contínua de dados. Usuários podem precisar estar conectados a uma rede de comunicação de dados (como uma LAN, WAN, a Internet ou uma intranet corporativa), mas isso não significa que eles estarão enviando e recebendo dados de todos a todo tempos. Foi deduzido que os usuários normalmente tendem a enviar mensagens

pequenas, mas recebem grandes downloads. Por isso, a grande maioria da transferência de dados está em uma única direção (SVERZUT, 2007).

De acordo com as especificações fornecidas pelo ETSI, a velocidade mais alta para uma única sessão de usuário é a codificação CS4, permitindo transferências a 21,4 Kbps por divisão. Teoricamente, uma conexão GPRS pode proporcionar uma velocidade de transmissão de dados de até 171,2 Kbps (cerca de três vezes maior do que uma linha fixa de 56K dial-up). O GPRS consegue assim, superar o HSCSD, que pode atingir até 57,6 Kbps (GSM SERVER, 2010).

Em resumo, podemos listar as principais vantagens do GPRS:

Cobertura: O GSM está disponível em todos os continentes. Apesar de que, em diferentes partes do mundo, o GSM opera em frequências de rádio diferentes, existem muitos dispositivos de múltiplas frequências.

Velocidade: Usando 4 divisões de tempo, a maioria dos aparelhos suporta uma taxa de 52 Kbits por segundo para *download* e *upload* de 13 Kbits por segundo.

Dispositivos: Atualmente a grande maioria dos dispositivos móveis, incluindo telefones celulares e *smartphones* suportam GPRS.

Tarifas: As operadoras podem optar pela cobrança por quantidade de dados trafegados, e oferecer planos com pacotes de dados variados, atingindo qualquer público.

2.4 Sistemas Embarcados

Neste século, a grande maioria dos processadores fabricados será usada em sistemas embarcados. Eles estão embutidos em muitos equipamentos comuns no dia-a-dia das pessoas. Um sistema embarcado pode ser definido basicamente

como um computador que criado para uma finalidade específica. Em contraste, um computador PC comum geralmente serve vários propósitos: verificar e-mails, navegar na internet, ouvir música, processar texto, etc. Sistemas embarcados normalmente têm apenas uma única tarefa, ou um número muito pequeno de tarefas relacionadas que eles são programados para executar (MARWEDEL, 2003).

Em casa temos vários exemplos de computadores embarcados. Qualquer aparelho que tenha um relógio digital, por exemplo, tem um microcontrolador pequeno embutido que executa uma tarefa diferente para mostrar o relógio. Os carros modernos têm computadores de bordo que controlam coisas como o ponto de ignição e freios ABS utilizando como entrada um grande número de diferentes sensores. Sistemas embarcados raramente têm interfaces comuns. Normalmente possuem um teclado e um monitor LCD, mas ainda assim eles raramente são capazes de usar muitos tipos diferentes de entrada ou saída. Um exemplo de um sistema embarcado com recursos de entrada e saída é um alarme de segurança com um *display* de LCD e um teclado para digitar uma senha.

Em geral, um sistema embarcado pode ser definido em três sentenças:

- É um sistema construído para realizar sua tarefa, total ou parcialmente independentes da intervenção humana.
- É especialmente projetado para executar uma tarefa da forma mais eficiente.
- Interage com elementos físicos em nosso meio: controla motores e lâmpadas, lê informações de sensores de luz, sensores de temperatura, etc.

O uso de sistemas embarcados é praticamente ilimitado. A cada dia novos produtos são introduzidos no mercado utilizando tecnologia embarcada de maneiras inovadoras. Nos últimos anos, hardware como microprocessadores,

microcontroladores e chips FPGA tornaram-se muito baratos. Assim, ao implementar uma nova forma de controle é mais interessante comprar esses chips e desenvolver sua aplicação para eles. Produzir um chip sob medida para lidar com uma determinada tarefa ou conjunto de tarefas custos muito mais tempo e dinheiro. Muitos sistemas embarcados ainda vêm com grandes bibliotecas, de modo que "escrever o seu próprio software" se torna uma tarefa muito trivial (GANSSLE, 2008).

Do ponto de vista de implementação, há uma grande diferença entre um computador e um sistema embarcado. Sistemas embarcados são muitas vezes obrigados a fornecer resposta em tempo real. Um sistema de tempo real é definido como um sistema que depende da prontidão da sua resposta. Exemplos de tais sistemas são os sistemas de controle de vôo de uma aeronave, sistemas de sensores em reatores nucleares e usinas de energia. Para estes sistemas, o atraso na resposta é um erro fatal. Tecnicamente, os *Real-Time Systems* podem ser classificados como:

- *Hard Real-Time Systems* - sistemas com restrições severas sobre o tempo de resposta.
- *Soft Real-Time Systems* - sistemas que toleram pequenas variações nos tempos de resposta.
- *Hybrid Real-Time Systems* - sistemas que apresentam restrições de hardware e software no quesito desempenho.

Sistemas embarcados podem ser econômicos, mas são muitas vezes sujeitos a alguns problemas muito específicos. Um computador PC pode ser fornecido com uma falha no software, e uma vez descoberto, uma correção de software muitas vezes pode ser enviada para corrigir o problema. Um sistema embarcado, no entanto, é programado uma vez, e o software normalmente não pode ser remendado. Outro problema com esses equipamentos é que eles são

geralmente instalados em sistemas para os quais insegurança não é uma opção. Por exemplo, o computador que controla os freios em seu carro não pode falhar sob qualquer condição. Para que haja segurança, muitas das técnicas de programação usadas em softwares para computadores pessoais não podem ser usadas em sistemas embarcados. A confiabilidade deve ser garantida antes do chip sair da fábrica. Isto significa que cada sistema embarcado precisa ser testada e analisada extensivamente.

Um sistema embarcado tem muito poucos recursos quando comparados aos computadores completos. A capacidade de memória, por exemplo, em um sistema embarcado é extremamente limitada. Por essas e outras razões é muito mais difícil desenvolver aplicações para um sistema embarcado do que sistemas para desktop.

2.5 Google Maps API

Os mapas se tornaram onipresentes em aplicações web. Eles estão sendo usados como uma maneira eficaz e não invasiva de coleta de dados georreferenciais do usuário ou para apresentar alguma informação específica. No entanto, alguns anos atrás, a incorporação de um mapa dentro de uma aplicação era uma tarefa especializada que exigia uma boa compreensão dos Sistemas de Informação Geográfica (SIG). Isso mudou com a chegada do Google Maps² e seu correspondente conjunto de APIs, conhecido como o Google Maps API. Qualquer desenvolvedor com um bom conhecimento de JavaScript pode, sem muito trabalho, incorporar um mapa interativo em um aplicativo Web usando o Google Maps API (GEO SPATIAL, 2010).

² <http://maps.google.com>

O Google Maps oferece não só o mapa ou imagem de satélite, mas também uma extensa gama de operações no mapa, como zoom, informação de pop-ups, sobreposições, etc. A API fornece uma interface para essas operações por meio de objetos JavaScript. As funções funcionam de forma não-intrusiva como o Google Maps original, sem o desenvolvedor precisar saber os detalhes do AJAX ou como ele é implementado no Google Maps. O desenvolvedor só precisa saber as classes e seus métodos para acessar os serviços do Google Maps. Entre as categorias de funções oferecidas pela API do Google Maps podemos citar: Configuração, Controles, Tipos de mapa, Mapa do Estado, Sobreposições, Janela de Informações, Mapa de navegação e Eventos. A Figura 4 mostra um exemplo de um mapa com alguns controles embutidos.



Figura 4: Google Maps.

2.6 Trabalhos relacionados

Existem soluções que se propõem a desempenhar papéis semelhantes ao proposto. Dentre estas, foram selecionadas: Sistema de Localização e Navegação Apoiado por GPS (HASEGAWA et al., 1999) e Implementação do Sistema de Mapeamento de uma Linha de Ônibus para um Sistema de Transporte Inteligente (WEIGANG et al., 2001).

Hasegawa et al. (1999, p. 1) dedicou especial interesse ao estudo de navegação, citando inclusive cidades dos Estados Unidos, Japão e outros países da Europa onde muitos carros de passeio contam com um sistema de localização, com o objetivo de auxiliar o motorista durante seu trajeto. Para que isso se transformasse em realidade também aqui no Brasil, Hasegawa (1999) desenvolveu um protótipo de um sistema de navegação, utilizando dados provenientes de um receptor GPS e integrando essas informações em uma base de dados geográficos em um computador portátil. As informações podem ser visualizadas na tela do computador, mostrando assim a posição imediata do veículo, sobre um mapa geográfico que caracteriza a região, possibilitando assim a identificação até mesmo da rua atual, que pode ser expandido inserindo maiores informações e detalhes de cada região específica.

Weigang et al. (2001) sugere que para auxiliar na programação e satisfação dos usuários de transporte coletivo público, desenvolva-se um sistema de informação sobre os horários individuais por ônibus. Para que isso ocorra, Weigang destaca a necessidade da utilização de tecnologias de telefonia celular, satélites do sistema GPS, internet.

3 METODOLOGIA

3.1 Tipo de pesquisa

Segundo CARVALHO (2001), em relação ao tipo de pesquisa esse trabalho pode ser classificado como uma pesquisa científica tecnológica aplicada e de caráter qualitativo, pelo fato de não utilizar dados estatísticos como o centro do processo de análise de um problema, não tendo, portanto, a prioridade de numerar ou medir unidades. Os dados coletados nessa pesquisa são descritivos, retratando o maior número possível de elementos existentes na realidade estudada.

As vantagens da pesquisa experimental são indiscutíveis, pois é a partir dela que obtemos maiores avanços nas ciências físicas e biológicas. Ela nos permite obter uma maior nitidez, exatidão, e objetividade nos resultados, porém, exige um controle extremo por envolver muitas variáveis que podem ser medidas e quantificadas (GIL, 1991). Dessa forma essa pesquisa ainda tem caráter experimental em relação aos procedimentos técnicos, ou meios de investigação.

Em relação ao usuário, MUNFORD (1988), define três modelos de participação em função do nível de diálogo e do tipo de participação no projeto: Consultivo, Representativo e Consensual. No desenvolvimento deste projeto optou-se pelo modelo consultivo, onde a tarefa de projetar é exclusividade do especialista em computação, sendo o usuário apenas um consultor, em momentos de escolha, e determinação de necessidades. Os usuários consultados foram pessoas comuns habitadas ao uso do transporte público.

3.2 Procedimentos e tecnologia

Para a realização deste trabalho foi feito um estudo bibliográfico a respeito do assunto a ser desenvolvido, neste caso, sobre o Sistema de Posicionamento Global, a tecnologia GSM e GPRS, os microcontroladores, banco de dados geográficos, a API do Google Maps e projetos e estudos já realizados, a fim de obter um embasamento teórico antes de partir para a modelagem em si. Foi feito ainda um breve estudo sobre a história do transporte coletivo urbano no Brasil com a finalidade de entender o seu funcionamento, assim como os Sistemas de Transporte Inteligente (ITS) com o objetivo de descobrir como este projeto se encaixa dentro dos ITS's.

Também foram estudados os erros que resultam na falta de precisão do GPS, e que por sua vez tornam a localização dos veículos monitorados imprecisa.

Em seguida foi projetado o módulo de coleta de dados georreferenciais baseado no microcontrolador ATmega328 da Atmel Products. Esse microcontrolador é encarregado de transmitir as coordenadas geográficas recebidas do módulo GPS ME-1000RW através do módulo GSM Benq M22 até o servidor com a aplicação de teste.

O servidor de teste é composto por um software que recebe as coordenadas através da URL e faz a plotagem do ponto usando a API do Google Maps. Nenhuma função adicional foi incluída no servidor por se tratar de um simples teste de validação.

Após a aferição dos dados, este módulo pode ser integrado ao módulo ManagerTrack, que em conjunto com os módulos MobileTrack e WebTrack formam uma plataforma única de monitoramento de veículos em tempo real (Figura 5).



Figura 5: Módulos da plataforma de monitoramento.

Quando o módulo entrar em funcionamento, este deve iniciar seu processo de boot, e em seguida iniciar o envio dos dados da posição geográfica através do rádio GSM/GPRS embutido no equipamento. O servidor deve fazer a exibição da posição geográfica diretamente num mapa do Google Maps, afirmando o correto funcionamento de todo o sistema.

3.3 Arquitetura do Módulo de Coleta de Dados

O módulo ColetorTrack foi desenvolvido a partir da plataforma de hardware livre Arduino, um GPS modelo ME-1000RW e um rádio GSM Benq

M22 alimentados por uma única fonte de 9 Volts. A arquitetura geral pode ser vista na figura abaixo:

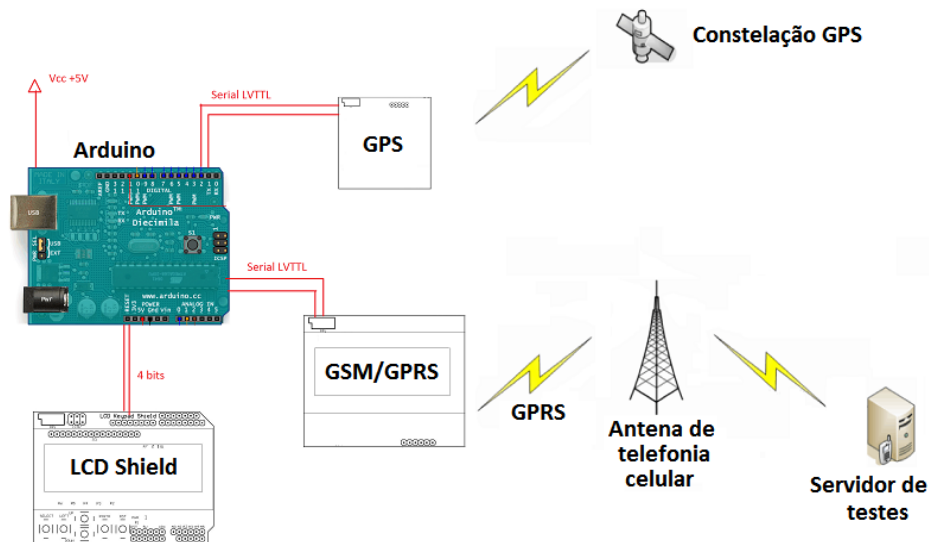


Figura 6: Arquitetura do Colector Track.

O modelo utilizado, Arduino Duemilanove, é composto por um microcontrolador ATmega328, um circuito integrado FT232RL para comunicação USB e alguns componentes adicionais. O ATmega328 é um microcontrolador com 16 MHz de velocidade de clock, memória Flash de 32 KB, memória SRAM de 2 KB e EEPROM de 1 KB. Para a visualização de algumas informações foi acrescentado ao Arduino um display de LCD de 16x2 caracteres, bem como cinco botões para controle de algumas funções que poderão ser acrescentadas no futuro. Este componente foi desenvolvido a partir da filosofia open-source hardware, tendo todo o seu projeto disponibilizado sob

a licença *Creative Commons Attribution Share-Alike*³, permitindo que qualquer modificação possa ser feita.

O módulo GPS ME-1000RW é um componente de baixo custo que possui 65 canais de aquisição, precisão de 5 metros, antena embutida, baixo consumo de energia, interface serial RS232 e LVTTTL. Esse componente envia através da porta serial, várias sentenças com informações diversas que estão descritas no protocolo NMEA 0183 versão 3.01⁴.

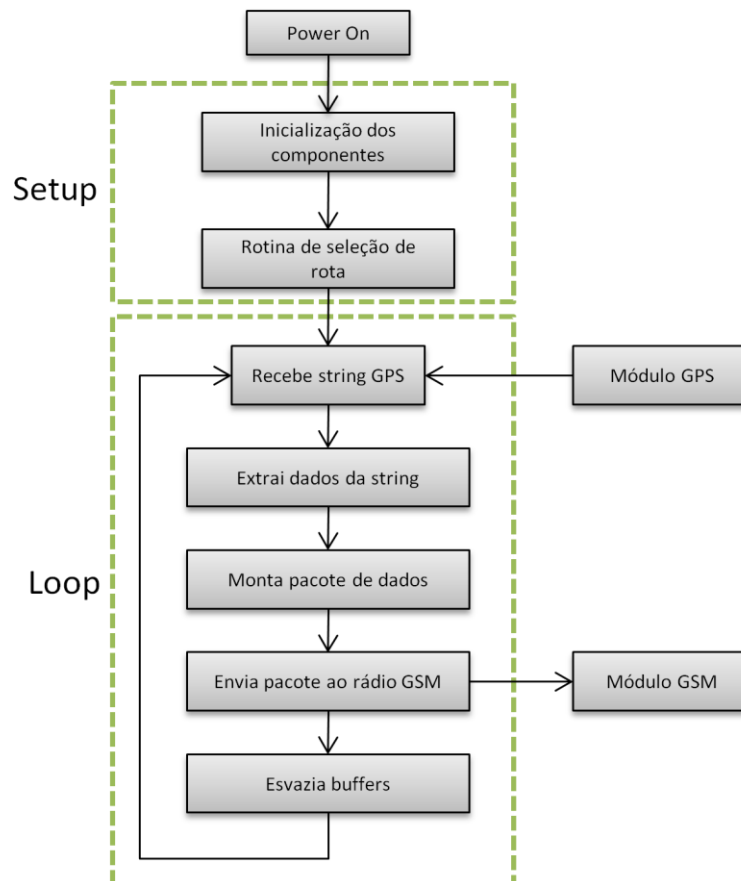


Figura 7: Fluxograma de funcionamento do módulo de coleta.

³ <http://creativecommons.org/licenses/by-sa/2.5>

⁴ <http://www.gpsinformation.org/dale/nmea.htm>

O rádio GSM/GPRS especificado foi o M22 da Benq. Este componente é basicamente um rádio capaz de fazer toda a comunicação com a rede GSM nas bandas de 900 MHz, 1800 MHz e 1900 MHz. Os principais fatores que justificaram a escolha são: possuir slot para o cartão SIM, conector para antena e interface serial RS232.

O firmware desenvolvido para este trabalho (Apêndice) foi escrito em C/C++ usando a IDE fornecida pelo próprio Arduino. A vantagem de usar esta IDE é a disponibilidade de várias bibliotecas compatíveis com o Arduino, como por exemplo a SoftwareSerial que permite tornar qualquer pino de I/O digital uma porta serial RS232. O código desenvolvido tem como finalidade receber as sentenças NMEA do GPS e extrair somente as informações úteis, enviando até o servidor através do rádio GSM/GPRS.

Quando em operação, o módulo conecta-se ao servidor de testes através da rede GSM usando uma URL definida, conforme o seguinte exemplo:

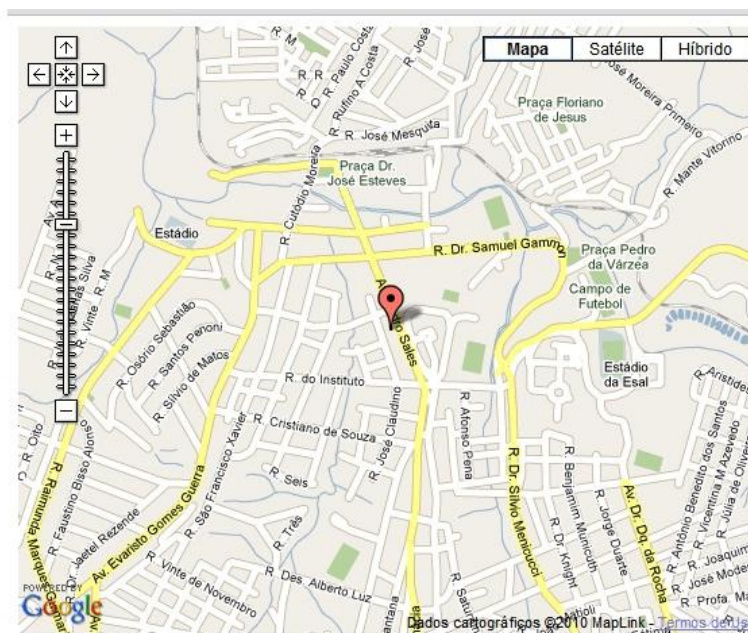
“<http://colectortrack.freehosting.com/save.php?long=77777&lat=88888&hora=5555&veloc=6666&idRota=9&idMod=3333>”

Essa URL contém dados de latitude e longitude, hora do rastreamento, velocidade do receptor, identificador de rota e identificador do módulo.

4 RESULTADOS

4.1 Testes Realizados

Para a validação dos dados foi desenvolvida uma pequena aplicação (Apêndice) em PHP e JavaScript que recebe as variáveis na própria URL e faz a exibição do mapa usando a Google Maps API JavaScript V3. Essa aplicação foi hospedada provisoriamente num servidor gratuito. Ao acessá-la é possível visualizar o último ponto rastreado. A Figura 8 mostra um dos diversos pontos coletados na cidade de Lavras-MG, onde foram efetuados os testes.



ID do modulo: 9
Rota: 4
Hora UTM: 172523.008
Latitude: 2113.8661S (-21.231101)
Longitude: 04459.8943W (-44.998238)
Velocidade: 000.0

Figura 8: Teste de validação do rastreamento.

4.2 Módulo Colector Track

Como resultado deste trabalho temos um protótipo de um módulo de coleta de dados do posicionamento geográfico (Figura 9). O módulo inicia seu funcionamento assim que a alimentação elétrica é ligada.

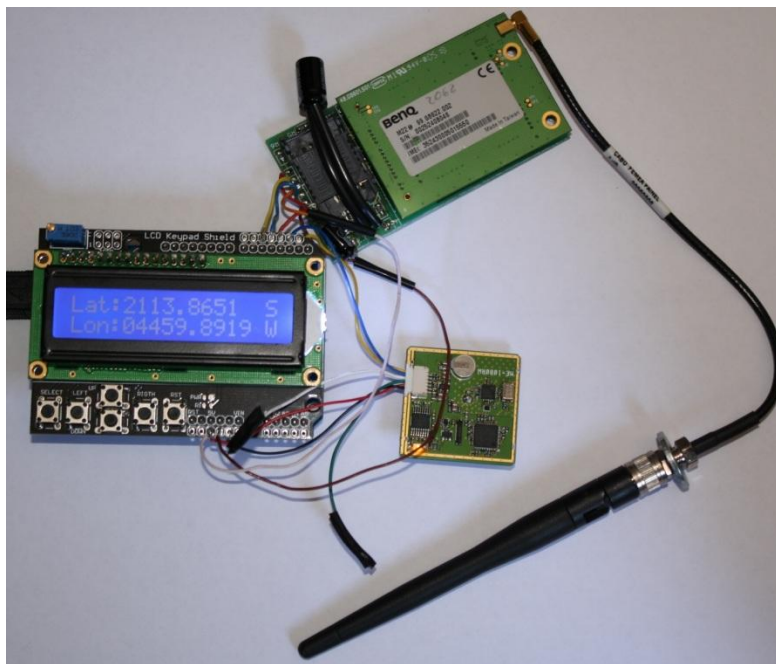


Figura 9: Módulo Colector Track.

Em poucos segundos a rotina de seleção da rota é iniciada e o usuário pode usar as teclas LEFT e RIGHT para mudar o número da rota e pressionar SELECT para confirmar a escolha. Para mudança de rota é necessário reiniciar o módulo através do botão RESET. Depois de selecionada a rota é exibida a mensagem “Aguardando sinal”, até que o GPS tenha dados disponíveis. Uma

vez obtidos, os dados de latitude e longitude são exibidos continuamente no LCD e enviados num intervalo de tempo pré-determinado ao servidor.

O custo aproximado de fabricação é de R\$450, incluindo o GPS, o rádio GSM/GPRS, o Arduino e o display de LCD. Entretanto, a maior vantagem deste equipamento em relação aos similares comerciais é a flexibilidade de poder adicionar ou remover funcionalidades. A modificação do hardware para o acréscimo de sensores, por exemplo, é facilitada pela utilização do modelo open-source do Arduino, com toda a documentação e diversos exemplos práticos disponíveis na Internet.

5 CONCLUSÃO

5.1 Dificuldades Encontradas

Durante o andamento do projeto, o módulo de coleta sofreu várias mudanças na sua arquitetura. A mais drástica foi a troca do microcontrolador PIC18F4550 pelo ATmega328. Essa medida foi tomada para contornar várias dificuldades na comunicação via porta serial, como por exemplo, a possibilidade de uso de duas portas seriais no mesmo microcontrolador.

Entretanto, a maior dificuldade encontrada foi a falta de documentação do módulo GSM/GPRS Benq M22. A lista de comandos AT disponibilizada pelo fabricante é confusa e não possui exemplos claros de uso do GPRS. Dessa forma, foi preciso recorrer aos fóruns de eletrônica e testar os comandos AT de outros módulos até encontrar uma combinação funcional. Essa medida gerou custos não previstos com a operadora de telefonia celular devido a grande quantidade de dados trafegados pela rede.

5.2 Trabalhos Futuros

Este projeto tem várias tarefas previstas. Algumas são puramente ergonômicas, como a instalação dos componentes numa caixa plástica com botões mais acessíveis e o acréscimo de um cabo de alimentação que possa ser facilmente conectado ao veículo.

Uma tarefa essencial é implementar a segurança no tráfego de dados. Se a URL for conhecida, qualquer pessoa, através da internet, pode enviar dados inconsistentes ao servidor prejudicando o funcionamento do sistema. Uma

possível solução é acrescentar ao pacote de dados um código de verificação, calculado no módulo de coleta e verificado no servidor.

Também está prevista uma rotina de detecção de movimento com a finalidade de minimizar o tráfego de dados na rede GSM. Essa rotina detecta se o veículo teve um deslocamento insignificante. Em caso positivo, o módulo envia uma *flag* indicando a parada e em seguida cessa o envio dos pacotes até que o deslocamento seja válido.

Para atender o objetivo do projeto ainda é necessário fazer a integração deste módulo ao Módulo Web, parte da plataforma proposta. Essa integração consiste em ajustar o pacote de dados de acordo com a necessidade do Módulo Web, acrescentando ou removendo itens, e o intervalo de envio dos dados de acordo com as necessidades do projeto.

6 REFERÊNCIAS BIBLIOGRÁFICAS

AQUINO, W.; AQUINO, N. B.; PEREIRA, W. F. **Considerações sobre o uso de ITS.** Revista dos Transportes Públicos, ano 23, n. 91, p. 33, 2º trim. 2001.

BARROS, E. S. **O ônibus e o turismo: posicionamento do turismo rodoviário no Brasil na visão dos transportadores turísticos por fretamento da grande São Paulo.** Dissertação (Mestrado) - Escola de Comunicações e Artes da Universidade de São Paulo, São Paulo, 2002.

CARVALHO, R. **Inovação Tecnológica e Tecnologias da Informação na Indústria Paulista.** In: FAPESP, Indicadores de Ciência, Tecnologia e Inovação em São Paulo, São Paulo, 2001.

GANSSELE, JACK G. **Embedded Systems.** Elsevier Inc, Oxford, 2008.

GEO SPATIAL. **Google Maps Courses: Introduction to the Google Maps API.** Disponível em <http://geospatialtraining.com>, acessado em: 23 de Novembro de 2010.

GIL, ANTÔNIO CARLOS, **Como elaborar projetos de pesquisa.** Editora Atlas, São Paulo, 1991.

GSM SERVER, INC. **What is GSM? GSM Overview.** Disponível em: <http://gsmserver.com/articles/>, acessado em 26 de maio de 2010.

HASEGAWA, J. K. et al. **Sistema de localização e navegação apoiado por GPS.** In: Congresso Brasileiro de Cartografia, Recife, 1999.

MARWEDEL P. **Embedded System Design.** Kluwer Academic Publishers, Dortmund, 2003.

MOON, Young-Jun, **Research Fellow at South Korea's Center for Advanced Transportation Technology,** 15th ITS World Congress, 2008.

MUNFORD, E. **El diseño participativo de sistemas: prática e teoria.** In: La automatization y el trabajo, Ministério de Trabajo y Seguridade Social, Madrid, Espanha, 1988.

SILVENTOINEN, Mark. **European Mobile Communications Business and Technology Report**, December 1995.

STIEL, W. C. **Ônibus: uma história do transporte coletivo e do desenvolvimento urbano no Brasil**. Comdesenho Estúdio e Editora, São Paulo, 2001.

SVERZUT, J. U. **Redes GSM, GPRS, EDGE e UMTS – Evolução a Caminho da Terceira Geração**. Editora Érica, 2007.

UNSW, UNIVERSITY OF NEW SOUTH WALES. **Notes on Basic GPS Positioning and Geodetic Concepts**. Disponível em: http://www.gmat.unsw.edu.au/snap/gps/gps_notes.htm, acessado em 26 de maio de 2010.

WEIGANG, L.; YAMASHITA, Y.; SILVA, O. Q.; XIJUN D.; PRAZERES, M. A. T OLIVEIRA, D. C. S. **Implementação do sistema de mapeamento de uma linha de ônibus para um sistema de transporte inteligente**. In: Anais do XXI Congresso da Sociedade Brasileira de Computação, Disponível em <http://www.cic.unb.br/docentes/weigang/>, acessado em: 8 de junho de 2010.

WEIGANG, L.. **Implementação do sistema de mapeamento de uma linha de ônibus para um sistema de transporte inteligente**. In: XXI Congresso da Sociedade Brasileira de Computação, Fortaleza, 2001.

7 APÊNDICE

7.1 *Firmware*

Segue abaixo o código fonte desenvolvido para o hardware, compilado com Arduino Alpha 0021.

ColectorTrack_code.pde

```
#include <LCD4Bit_mod.h>
#include <SoftwareSerial.h>
#define rxPin 2
#define txPin 3
#define DADOS_LEN 100
#define IDLEN 6
#define TEMPLEN 11
#define GPRMC 0
#define IDMODULO 000011

LCD4Bit_mod lcd = LCD4Bit_mod(2); //cria objeto para
controlar LCD (n de linhas)
SoftwareSerial gprsSerial = SoftwareSerial(rxPin, txPin);
char dados[DADOS_LEN]; //buffer do GPS
byte conta=0; //contador auxiliar
char* idnmea[] = {"$GPRMC"}; // IDs do NMEA que serão
analisados

byte verificador[]= {0,0}; //auxiliar para verificar ID
NMEA
byte indice[12]; //posição do caracter de inicio de cada
valor numa linha $GPRMC
byte contindice=0; //auxiliar de controle do indice[];
byte menu=9; //controle do menu
char tempmsg[TEMPLEN]; //valor de um dado extraído do GPS

int valorADC[5] ={30, 150, 360, 535, 760}; //valores do
divisor de tensão do teclado do LCD Shield
#define NUM_KEYS 5 //numero de teclas do teclado
int adc_valor_in; //valor da entrada analogica do teclado
byte key=-1; //tecla pressionada
```

```

byte oldkey=-1; //tecla pressionada anteriormente

#define NUMEROROTAS 15
int idRota = 0; //índice para aRotas[]
char* aRotas[] =
{"1","2","3","4","5","6","7","8","9","10","11","12","13","14","15"}; //rotas definidas
boolean fim = false; //fim da seleção de rota

/*   Rotina de setup, executada sempre que o módulo é
ligado ou resetado.
      Inicia os periféricos e define a rota do veículo.
*/
void setup(){
  lcd.init();
  lcd.clear();
  lcd.printIn("Colector Track");
  lcd.cursorTo(2, 0);
  lcd.printIn("aguarde...");
  delay(2000);
  limpaBuffer(); //limpa buffer do GPS (dados[])
  limpaTemp(); //limpa dado do GPS (tempmsg[])
  Serial.begin(9600); //inicia UART com 9600 bps

  // configura a porta serial para o modem GPRS
  pinMode(rxPin, INPUT);
  pinMode(txPin, OUTPUT);
  gprsSerial.begin(9600);

  lcd.clear();
  menu = 9;
  lcd.cursorTo(1,0);
  lcd.printIn("Selecao de rota");
  lcd.cursorTo(2,0);
  lcd.printIn(aRotas[idRota]);
  rotaSelect(); //rotina de seleção de rota

  lcd.cursorTo(1,0);
  lcd.printIn("Aguardando sinal");
  lcd.cursorTo(2,0);
  lcd.printIn("...");
}

/*   Rotina executada continuamente após o setup.
      Recebe os dados do GPS, monta o pacote de dados e
envia ao módulo GSM.
*/

```

```

void loop(){
    while(Serial.available()){
        dados[conta] = Serial.read(); //pega caracter
do buffer da Serial
        if(dados[conta]==13){ //Se o caracter = 13
então é final de linha -> interpretar
            verificador[GPRMC]=0; //verifica
idnmea[0] ($GPRMC)
                for(byte i=1;i<=IDLEN;i++){ //verifica o
ID NMEA da linha recebida
                    if(dados[i]==idnmea[GPRMC][i-1]){
//verifica se é $GPRMC
                        verificador[GPRMC]++;
                    }
                }
            if(verificador[GPRMC]==IDLEN){ //atualiza
os índices se GPRMC está correto
                contindice = 0;
                indice[contindice] = 1; //dados[]
inicia no caracter 1
                    contindice++;
                    for(byte i=1;
i<DADOS_LEN;i++){//percorre toda linha dados[]
identificando onde começa cada valor do GPS
                        if(dados[i]==',' ){
                            indice[contindice] =
i+1;
                                contindice++;
                        }
                    }
                // montagem do pacote de dados
                String
lon,lon2,lat,lat2,hora,veloc, pacote;
                lat = buscaDado(3); //requisita um
dado
                    lat2 = buscaDado(4);
                    lon = buscaDado(5);
                    lon2 = buscaDado(6);
                    hora = buscaDado(1);
                    veloc = buscaDado(7);

                //exibe latitude e longitude no LCD
                lcd.clear();
                lcd.cursorTo(1,0);
                lcd.printIn("Lat:");
                lcd.printIn(buscaDado(3));

```

```

        lcd.printIn(" ");
        lcd.printIn(buscaDado(4));
        lcd.cursorTo(2,0);
        lcd.printIn("Lon:");
        lcd.printIn(buscaDado(5));
        lcd.printIn(" ");
        lcd.printIn(buscaDado(6));

        //colectortrack.freehosting.com/save.php?long=77777&l
at=88888&hora=5555&veloc=6666&idRota=9&idMod=3333
        String URL =
"save.php?long="+lon+" "+lon2+"&lat="+lat+" "+lat2+"&hora="+h
ora;
        String servidor =
"http://colectortrack.freehosting.com/";
        String URL2 =
"&veloc="+veloc+"&idRota="+aRotas[idRota]+"&idMod="+IDMODUL
O;
        String URLF = servidor + URL +
URL2;

        //envio do pacote via GPRS
        gprsSerial.print("AT+CGDCONT=1," +
"IP" + ",tim.br"); //configura a APN (access point name)

        gprsSerial.print("AT%CGPCO=1,"+"PAP,tim,tim,"+"1");
//configura nome de usuário e senha
        gprsSerial.print("AT$TIMEOUT=500");
//timeout da conexão(100-5000 mSec)

        gprsSerial.print("AT$DESTINFO="+URLF+",1,80,1");
//requisição
        gprsSerial.print("AT$NOSLEEP=1");
//cancela modo Deep Sleep
        gprsSerial.print("ATD*97#");
//efetua a discagem abrindo uma conexão TCP
        gprsSerial.print("+++"); //muda
para o command mode
        gprsSerial.print("ATH"); //encerra
conexão TCP
        */
        delay(30000); //aguarda 30 segundos
para enviar outro pacote
    }
    conta = 0; //inicia proxima linha de dados[]
    limpaBuffer(); //limpa dados[]

```

```

        }else{
            conta++; //pula para próximo até chegar
ao final da linha
        }
    }

/*   Limpa variavel dados[]
*/
void limpaBuffer(){
    for (byte i=0;i<DADOS_LEN;i++){
        dados[i]=' ';
    }
}

/*   Limpa variavel tempmsg[]
*/
void limpaTemp(){
    for(byte i=0;i<TEMPLLEN;i++){
        tempmsg[i]=' ';
    }
}

/*   Funcao que busca na variavel dados[] o valor
procurado do GPS.
    2-timestamp (UTC), 3-latitude, 4-Norte/Sul, 5-
Longitude,
    6-Leste(E)/Oeste(W),7-velocidade em nós,9-date stamp
*/
char* buscaDado(byte inicio){
    limpaTemp();
    byte i;
    byte fim = indice[inicio+1]-2;
    inicio = indice[inicio];
    for(i=0;i<=(fim-inicio);i++){
        tempmsg[i] = dados[inicio+i];
    }
    tempmsg[i] = '\0';
    return tempmsg;
}

/*   Converte o valor lido no ADC no número correspondente
a tecla
*/
byte get_key(unsigned int entrada){
    int tecla=menu;
    for (byte i = 0; i < NUM_KEYS; i++){
        if (entrada < valorADC[i]){

```

```

        tecla=i;
        return tecla;
    }
}
return tecla;
}

/* Rotina de seleção de rota
*/
void rotaSelect(){
    while(!fim) {
        adc_valor_in = analogRead(0); //verifica
        entrada analógica do teclado
        key = get_key(adc_valor_in); //interpreta valor
        da entrada analógica
        if (key != oldkey){
            delay(50); //delay para o debounce
            adc_valor_in = analogRead(0); //le
            novamente a entrada analógica para confirmar
            key = get_key(adc_valor_in);
            if (key != oldkey){
                oldkey = key; //atualiza oldkey
                if (key >=0){
                    menu = key; //atualiza o menu
                }
            }
        }
    }

    switch(menu){ //0-botão RIGHT,3-botão LEFT, 4-
    botão SELECT
    case 3: //LEFT
        idRota -= 1; //decrementa rota
        if (idRota <= 0){
            idRota = 0;
        }
        lcd.clear();
        lcd.cursorTo(1,0);
        lcd.printIn("Selecao de rota");
        lcd.cursorTo(2,0);
        lcd.printIn(aRotas[idRota]);
        break;
    case 0: //RIGHT
        idRota += 1; //incrementa rota
        if (idRota >= NUMEROROTAS-1){
            idRota = NUMEROROTAS-1;
        }
        lcd.clear();

```

```
        lcd.cursorTo(1,0);
        lcd.printIn("Selecao de rota");
        lcd.cursorTo(2,0);
    lcd.printIn(aRotas[idRota]);
    break;
case 4: //SELECT
    lcd.clear();
    lcd.cursorTo(1,0);
    lcd.printIn("Rota selecionada");
    lcd.cursorTo(2,0);
    lcd.printIn(aRotas[idRota]);
    delay(2000);
    lcd.clear();
    fim = true;
break;
}
menu=9;
}
}
```

7.2 Servidor de Testes

Segue abaixo o código fonte desenvolvido para o servidor de testes. O arquivo “save.php” é responsável por salvar as informações recebidas em um arquivo de texto. Já o arquivo “index.php” exibe a última posição recebida no mapa.

save.php

```
<html>
<body>
<?php
/*formato da url esperada
save.php?long=77777&lat=88888&hora=5555&veloc=6666&idRota=9&idMod
=3333
*/

//abre o arquivo ou cria
$file = fopen("positions.txt","a+");

//extrai as variaveis da URL
$url =
"\r\n".$_GET["long"].",".$_GET["lat"].",".$_GET["hora"].",".$_GET["veloc"]."
,$_GET["idRota"].",".$_GET["idMod"];

//grava e fecha o arquivo
fwrite($file,$url);
fclose($file);
?>
</body>
</html>
```

index.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=UTF-8"/>
<title>Colector Track</title>
```

```

<script
src="http://maps.google.com/maps?file=api&v=2&sensor=false&key=INSIRA_AQUI_SUA_KEY_DO_GOOGLE_MAPS" type="text/javascript">
</script>

<?php
//abre o arquivo de posicoes
$file = fopen("positions.txt","r");

//retorna a ultima linha do arquivo
while (!feof($file)){
    $lastLine = fgets($file);
}
$point = explode(",",$lastLine);

$lon = $point[0];
$lat = $point[1];
$hora = $point[2];
$veloc = $point[3];
$SidRota = $point[4];
$SidMod = $point[5];

//extracao e conversao da posicao (minutos para graus)
$latDM = substr($lat,0,strlen($lat)-1);
$lonDM = substr($lon,1,strlen($lon)-1);

$latDG =
(substr($latDM,2,strlen($latDM))/60)+(substr($latDM,0,2));
if (strchr($lat,"S") == "S"){
    $latDG = "-".$latDG;
}
$latDG = substr($latDG,0,10);

$lonDG =
(substr($lonDM,2,strlen($lonDM))/60)+(substr($lonDM,0,2));
if (strchr($lon,"W") == "W"){
    $lonDG = "-".$lonDG;
}
$lonDG = substr($lonDG,0,10);

//fecha o arquivo
fclose($file);

```

```

?>
</head>
  <body>
    <b>Ultimo ponto enviado</b><br/><br/>

    <noscript>JavaScript must be enabled in order for you to use
Google Maps.</noscript>

    <div id="map" style="width: 550px; height: 450px"></div>

    <script type="text/javascript">
      //<![CDATA[
        if (GBrowserIsCompatible()) {
          //cria um marker com opcao de conteudo html
          function createMarker(point,html) {
            var marker = new GMarker(point);
            GEvent.addListener(marker, "click",
function() {
              marker.openInfoWindowHtml(html);
            });
            return marker;
          }

          //recupera a posicao extraida pelo codigo PHP
          var latDG = "<?php echo $latDG ?>";
          var lonDG = "<?php echo $lonDG ?>";

          //exibe o mapa e adiciona controles
          var map = new
GMap2(document.getElementById("map"));
          map.addControl(new GLargeMapControl());
          map.addControl(new GMapTypeControl());
          map.setCenter(new
GLatLng(latDG,lonDG),15);

          //exibe marker na posicao desejada
          var point = new GLatLng(latDG,lonDG);
          var marker = createMarker(point,'Veiculo
localizado!');

          map.addOverlay(marker);
        }
      ]>
    </script>
  </body>
</html>

```

```
                else {
                    alert("Sorry, the Google Maps API is not compatible
with this browser");
                }
            //]]>
        </script>

        <br/>
        <b>ID do modulo:</b> <?php echo $idMod ?> <br/>
        <b>Rota:</b> <?php echo $idRota ?> <br/>
        <b>Hora UTM:</b> <?php echo $hora ?> <br/>
        <b>Latitude:</b> <?php echo $lat . " (" . $latDG . ")" ?> <br/>
        <b>Longitude:</b> <?php echo $lon . " (" . $lonDG . ")" ?>
<br/>
        <b>Velocidade:</b> <?php echo $veloc ?> <br/>
    </body>
</html>
```