

**UNIVERSIDADE ESTADUAL PAULISTA (UNESP)
FACULDADE DE ENGENHARIA – CAMPUS DE ILHA SOLTEIRA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

Monitoramento de Dados via Internet baseado em Telefonia Celular

Autor: Getúlio Teruo Tateoki
Orientador: Jozué Vieira Filho

Dissertação submetida à Faculdade de Engenharia de Ilha Solteira – UNESP, como parte dos requisitos para obtenção do título de mestre em engenharia elétrica.

ILHA SOLTEIRA

Janeiro/2007

FICHA CATALOGRÁFICA

Elaborada pela Seção Técnica de Aquisição e Tratamento da Informação/Serviço Técnico de Biblioteca e Documentação da UNESP-Ilha Solteira

T217m	Tateoki, Getúlio Teruo. Monitoramento de dados via internet baseado em telefonia celular / Getúlio Teruo Tateoki. -- Ilha Solteira : [s.n.], 2007 123 p. : il., fotos (algumas color.) Dissertação (mestrado) - Universidade Estadual Paulista. Faculdade de Engenharia de Ilha Solteira, 2007 Orientador: Jozué Vieira Filho Bibliografia: p.86-90 1. Aquisição de dados. 2. Telefonia celular. 3. Sistemas de comunicação sem fio. 4. Internet sem fio.
-------	--



UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Campus de Ilha Solteira

CERTIFICADO DE APROVAÇÃO

TÍTULO: Monitoramento de Dados Via Internet Baseado em Telefonia Celular

AUTOR: GETULIO TERUO TATEOKI

ORIENTADOR: Prof. Dr. JOZUÉ VIEIRA FILHO

DATA DA REALIZAÇÃO: 13 DE ABRIL DE 2007.

Aprovada com parte das exigências para obtenção do Título de MESTRE em
ENGENHARIA ELÉTRICA pela Comissão Examinadora:

Prof. Dr. JOZUÉ VIEIRA FILHO

Departamento de Engenharia Elétrica / Faculdade de Engenharia de Ilha Solteira

Prof. Dr. SUELY CUNHA AMARO MANTOVANI

Departamento de Engenharia Elétrica / Faculdade de Engenharia de Ilha Solteira

Prof. Dr. JOSÉ RAIMUNDO DE OLIVEIRA

Departamento de Engenharia de Computação e Automação Industrial / Universidade
Estadual de Campinas

AGRADECIMENTOS

- À Deus por ter me dado força e a oportunidade de cumprir mais uma etapa na minha vida.
- À minha esposa Julia Lieko Tutumi e ao meu filho Victor Augusto Tateoki que sempre estiveram ao meu lado, apoiando e incentivando a realização deste trabalho.
- À minha mãe Maria Tateoki, que sempre deu apoio incondicional em todas as atividades da minha vida.
- Ao professor Dr. Jozué Vieira Filho pela competência, estímulo e a amizade que me foram fundamentais para o desenvolvimento deste trabalho.
- Aos professores Pe. Luis Fávero e André Luis Ornellas, Diretor e Vice-Diretor do Unisalesiano de Araçatuba/SP, que deram todo o apoio necessário para a concretização deste trabalho.
- Aos professores Dr. Marco Antonio Fernandes e Dr. Mário Jefferson Quirino Louzada que deram apoio fundamental para o início deste trabalho.
- Ao Sr. Luis Henrique Corrêa Bernardes e sua Equipe de Suporte Técnico da Duodigit que sempre nos atenderam com atenção e presteza.

SUMÁRIO

LISTA DE FIGURAS	8
LISTA DE TABELAS	10
LISTA DE SÍMBOLOS	11
LISTA DE ABREVIATURAS	12
RESUMO	14
ABSTRACT	15
CAPÍTULO 1	
INTRODUÇÃO.....	16
CAPÍTULO 2	
ASPECTOS BÁSICOS DE COMUNICAÇÃO SEM FIO E INTERNET	
2.1 Histórico.....	20
2.2 Sistema CDMA.....	21
2.2.1 Forma de transmissão do CDMA.....	22
2.3 Sistema 3G.....	23
2.4 Sistema GSM – Global System for Mobile.....	23
2.4.1 Frequências de Operação do GSM (MHz).....	24
2.4.2 Descrição do sistema GSM.....	24
2.4.3 A Estação Móvel.....	25
2.4.4 Subsistema Rádio Base.....	25
2.4.5 O Subsistema Rede.....	26

2.4.6 Codificação de canal e voz.....	27
2.5 GPRS - General Packet Radio Service.....	27
2.5.1 Arquitetura de Redes GPRS.....	28
2.5.2 Elementos Internos de uma Rede GPRS.....	30
2.5.3 Forma de Transmissão de uma Rede GPRS.....	31
2.5.4 Conexão de dados entre um terminal móvel e um servidor de uma rede IP externo.....	33
2.5.5. Forma de Identificação de assinante em uma Rede GPRS/GSM.....	33
2.5.6 Autenticação de um assinante em Redes GPRS/GSM.....	35
2.5.7. Terminais GPRS –Conexão.....	35
2.5.8. Classes de Terminais GPRS.....	36
2.5.9 Interface R.....	36
2.5.10 APN (Access Point Name).....	37
2.6 Internet - A Rede mundial de Computadores.....	37
2.6.1 Histórico.....	37
2.6.2 Serviços de Internet.....	38
2.6.3 Endereçamento na Rede Internet.....	39
2.6.4 DNS <i>Domain Name Server</i>	40

CAPÍTULO 3

ARQUITETURA PROPOSTA PARA O SISTEMA DE MONITORAMENTO DE DADOS VIA INTERNET USANDO TELEFONIA CELULAR

3.1 Contexto Tecnológico.....	43
3.2 Interface GPRS – INTERNET.....	45
3.3 Interface de AQUISIÇÃO DE DADOS – GPRS.....	48
3.4 Arquitetura final do sistema.....	50
3.4.1 Sub-Sistema 1: aquisição e controle dos dados.....	51

3.4.2 Sub-Sistema 2: comunicação via GPRS.....	52
3.4.3 Sub-Sistema 3: armazenamento e monitoramento dos dados via WEB.....	52

CAPÍTULO 4

IMPLEMENTAÇÃO DO SISTEMA PROPOSTO

4.1 O Modem GSM TC45.....	54
4.1.1 Modos de Operação.....	55
4.1.2 Controle Liga/Desliga.....	56
4.1.3 Utilização do RTC para Ligar o Modo Alarme.....	57
4.1.4 Ativando o Terminal (wake up).....	58
4.1.5 Interface SIM (<i>Subscriber Identification Module</i>).....	59
4.1.6 Led's de monitoramento.....	60
4.1.7 Programação do Modem TC45.....	60
4.1.8 Teste do Modem TC45 JAVA.....	61
4.1.9 Implementação do Programa no Terminal TC45 JAVA para o Sistema Proposto.....	64
4.2 O hardware para aquisição de dados.....	65
4.2.1 Sensor de Temperatura LM35.....	66
4.2.2 Programação do PIC.....	68
4.3 O Monitoramento via Internet.....	71
4.4 Aspectos técnicos e considerações do projeto.....	76

CAPÍTULO 5

TESTES E RESULTADOS OBTIDOS

5.1 O monitoramento da Temperatura via Internet.....	78
5.2 A comunicação via modem GSM.....	80
5.3 Custos do Projeto.....	81
5.3.1 Implementação.....	81
5.3.2 Serviço.....	82
5.4 O Sistema Obtido.....	83

CAPÍTULO 6

COMENTÁRIOS FINAIS.....	84
--------------------------------	-----------

REFERÊNCIAS BIBLIOGRÁFICAS.....	86
--	-----------

ANEXOS

Anexo 1 - Software Aplicativo do PIC.....	91
---	----

Anexo 2 - bscan – Software do Terminal Java TC45 JAVA.....	105
--	-----

Anexo 2.1 bscan.jar.....	105
--------------------------	-----

Anexo 2.2 main.....	106
---------------------	-----

Anexo 2.3 Serial ConnectionListener.....	108
--	-----

Anexo 2.4 Serial Connection Thread.....	118
---	-----

Anexo 2.5 HTTPUploadThread.....	112
---------------------------------	-----

Anexo 3 - Procedimento para configuração de Auto-Start do Terminal TC45 JAVA.....	117
--	-----

Anexo 4 - Script de captura da informação na Web e seu armazenamento em um banco de dados MySQL.....	120
---	-----

Anexo 5 - Página em PHP que efetua Consulta ao Banco de Dados.....	122
--	-----

LISTA DE FIGURAS

2.0 - Forma de Transmissão de um Canal de RF no sistema CDMA.....	22
2.1 - Estrutura de uma rede GSM.....	25
2.2 - Arquitetura básica do Sistema GPRS.....	29
2.3 - Elementos Internos de uma Rede GPRS.....	31
2.4 - Plano de Transmissão para TCP/IP.....	32
2.5 - Esquema de troca do TMSI no momento de atualização da localização geográfica feita pelo aparelho.....	34
2.6 - Uma Rede baseada no Protocolo TCP/IP.....	40
2.7 - Estrutura hierárquica do DNS.....	41
3.0 - Comparação entre Modem com e sem programação JAVA.....	46
3.1 - Diagrama de blocos do sistema.....	51
4.0 - Terminal Java TC45 observado externamente.....	55
4.1 - Vista Frontal e Posterior do Terminal TC45-JAVA.....	55
4.2 - Posição do Conector Interno do Terminal TC45 JAVA.....	55
4.3 - Modos de implementação de chave Liga/Desliga externa.....	57
4.4 - Disponibilização de acesso do Terminal TC45 efetuado por SMTK indicado como Module.....	62
4.5 - Indicação da quantidade de memória Flash do Terminal TC45 JAVA.....	62
4.6 - Programa rs232demo carregado no Terminal TC45 JAVA.....	63
4.7 - Conexão habilitada para realizar comunicação com a porta ASC0 e ASC1.....	63
4.8 - Fluxograma de operação do TC 45.....	65
4.9 - Circuito para o monitoramento de temperatura e comunicação com TC45.....	66
4.10 - Esquema Elétrico do Sensor de Temperatura.....	67
4.11 - Circuito Elétrico do Medidor de Temperatura.....	67
4.12 - Fluxograma do Programa SensorTemp VS21.....	69
4.13 - Gravador PIC de porta paralela.....	70
4.14 - Página web de visualização de dados de monitoração de temperatura.....	76
5.0 - Foto mostrando os valores Temperatura no Multímetro e Tensão do Sistema de	

Monitoração.....	79
5.1 - Visualização dos dados via página WEB.....	80
5.2 - Foto do Sistema de Telemetria montado e acondicionado em uma mochila para transporte.....	83

LISTA DE TABELAS

2.0 - Frequência de Operação do GSM.....	24
2.1 - Domínios de Níveis Mais Elevados.....	42
3.0 - Custo Operacional.....	48
4.0 - Modos de Operação do Terminal TC45 JAVA.....	56
4.1- Comandos AT's executado no Modo Alarme.....	58
4.2 - Descrição dos estados de transição.....	59
4.3 - Indicação do Led de Status de Operação.....	60
5.0 - Custos de Implementação.....	82
5.1 - Custo de Serviços.....	82

LISTA DE SÍMBOLOS

<i>Bit</i>	<i>Binary Digit</i> , em inglês, é a menor unidade de informação usada na Computação e na Teoria da Informação
<i>byte</i>	Conjunto de 8 bits
<i>GHz</i>	Giga Hertz - 1.000.000.000 Hz, Medida de frequência
<i>°C</i>	Graus Celcius - Medida de Temperatura
<i>KHz</i>	Kilo Hertz - 1.000 Hz, Medida de frequência
<i>keywords</i>	1024 endereçamentos de memórias diferentes
<i>MHz</i>	Mega Hertz - 1.000.000 Hz, Medida de frequência
<i>mV</i>	mili Volt - milésima parte de um Volt
<i>V</i>	Volts - Medida de Tensão Elétrica

LISTA DE ABREVIATURAS

APN	<i>Access Point Name</i>
ASCII	<i>American Standard Code II</i>
BG	<i>Border Gateway</i>
CR	<i>Carriage Return</i>
MSC	<i>Centro de Comutação Móvel</i>
CGF	<i>Charging Gateway Functionality</i>
CDMA	<i>Code Division Multiple Access</i>
CLP	<i>Controlador Lógico Programável</i>
DNS	<i>Domain Name Server</i>
EAGLE	<i>Easily Applicable Graphical</i>
EEPROM	<i>Electrically-Erasable Programmable Read-Only Memory</i>
BSC	<i>Estação de Radio Base de Controle</i>
GGSN	<i>Gateway GPRS Support Node</i>
GPRS	<i>General Packet Radio Service</i>
GSM	<i>Global System for Mobile</i>
GSN	<i>GPRS Support Node</i>
HTTP	<i>Hiper Text Transfer Protocol</i>
HLR	<i>Home Location Registrar</i>
PHP	<i>Hypertext Preprocessor</i>
IMEI	<i>International Mobile Equipment Identity</i>
IP	<i>Internet Protocol</i>
J2ME	<i>Java 2 Micro Edition</i>
JVM	<i>Java Virtual Machine</i>
LIG	<i>Lawful Interception Gateway</i>
LCD	<i>Liquid Crystal Display</i>
LR	<i>Location Register</i>

PDP	<i>Packet Data Protocol</i>
PIC	<i>Periférico Integrado de Controle</i>
PAN	<i>Personal Área Network</i>
PDA	<i>Personal Digital Assistants</i>
PIN	<i>Personal Identification Number</i>
PWM	<i>Pulse Width Modulation</i>
RAM	<i>Random Access Memory</i>
RISC	<i>Reduced Instruction Set Computing</i>
SMC	<i>Serviço Móvel Celular</i>
SGSN	<i>Serving GPRS Support Node</i>
SMS	<i>Short Message Service</i>
SMS-GMSC	<i>Short Message Service – Gate Mobile Services Switching Center</i>
BTS	<i>Sistema de Transmissão de Rádio Base</i>
TTL	<i>Transistor-Transistor-Logic</i>
URL	<i>Universal Resource Locator</i>
VPN	<i>Virtual Privative Network</i>
VLR	<i>Visitor Locantion Registrer</i>
WAP	<i>Wireless Aplication Protocol</i>
WLAN	<i>Wireless Local Area Network</i>

RESUMO

A aquisição de dados tem evoluído de forma significativa nos últimos anos e passa a ser indispensável em várias aplicações nas áreas de engenharia, agricultura, medicina, aeronáutica, etc. O estado da arte nessa área envolve o uso de sistemas de comunicação sem fio e da internet. Neste trabalho é apresentado um sistema moderno de aquisição de dados, que é baseado em uma comunicação sem fio via rede GSM (Global Mobile for System), com conexão via Internet através de um sistema de telefonia celular GPRS (General Packet Radio Service), monitoramento de dados via página Web (World Web Browser) e interface de aplicação com base em um microcontrolador. O sistema é bastante flexível e foi testado em operações reais no monitoramento de temperatura ambiente. Os resultados obtidos são altamente satisfatórios e mostra que o sistema pode servir de base para aplicações em diferentes áreas onde a telemetria seja indispensável.

ABSTRACT

Data acquisition has significantly been developed in the last years and became indispensable in many application area of engineering such as agriculture, medicine, aeronautic, etc. The state of art in this area is based on wireless communication system and Internet. In this work it is presented a modern data acquisition based on wireless communication through the Internet, which includes a Cellular Mobile GPRS (General Packet Radio Service) in a GSM (Global Mobile System) network, and remote data monitoring via WEB (World Web Browser) that can be used in a Telemetry Network construction. The system was tested in real conditions as a temperature monitoring system based on a microcontroller and it has presented good results. The high flexibility model proposed allows to use it in many distinct applications that require remote data acquisition.

INTRODUÇÃO

Monitoramento de dados e aquisição de dados são palavras que se confundem e trazem juntas uma idéia básica: digitalização de sinais analógicos. Nesse sentido, os primeiros sistemas de aquisição de dados (DAQ) eram basicamente sistemas de conversão analógico/digital (A/D) desenvolvidos em placas e acoplados a um microcomputador [1]. Esses sistemas evoluíram de simples placas de aquisição de dados para modernos sistemas independentes e controlados por microprocessador [2]. Com a evolução das telecomunicações, novos sistemas de aquisição de dados foram desenvolvidos e incorporaram novas tecnologias, como a transmissão de dados via rádio [3].

Nos sistemas de aquisição de dados modernos, portabilidade e mobilidade são fundamentais na maioria das aplicações, que são as mais variadas e incluem, dentre outras áreas: agricultura, medicina, geologia, oceanografia, aeronáutica, engenharia, etc. [4]. Nesse sentido, tornou-se indispensável agregar a rede mundial de computadores, ou internet, aos novos e modernos sistemas de aquisição de dados [5]. Atendendo a essa nova perspectiva, propõe-se neste trabalho um sistema de aquisição de dados baseado em tecnologia de ponta, que inclui a captação de dados usando microcontrolador, a transmissão dos dados via telefonia celular, mais especificamente usando a tecnologia GPRS /Internet.

O desenvolvimento de um sistema eficiente passa, necessariamente, por um estudo detalhado das diferentes tecnologias envolvidas. Considerando que o propósito do trabalho é obter um sistema que garanta flexibilidade e eficiência, optou-se de início pelo uso de uma comunicação via telefonia celular, descartando a aplicação de sistemas dedicados de comunicação via rádio. Essa decisão foi baseada nas tendências atuais na área de aquisição de dados, mas mantendo a flexibilidade que se busca no sistema [6]. Existem muitos estudos e alguns padrões para aplicação de sistemas de comunicação sem fio no monitoramento remoto de dados [7]. Porém, a maneira mais direta que se tem para uma

transmissão de dados usando comunicação sem fio é a tecnologia da telefonia celular. No Brasil, existem três padrões de tecnologia digital de comunicação sem fio que são CDMA (*Code Division Multiple-Access*), TDMA (*Time Division Multiple-Access*) e GSM (*Global System Mobile*). Com base em aspectos técnicos, acessibilidade ao sistema e disponibilidade de equipamentos no mercado brasileiro, optou-se pelo padrão GSM. Dentro do GSM pode-se pensar, inicialmente, em transmissão de dados via SMS (*Short Message Service*). O limitante, neste caso, é a baixa taxa de comunicação, que pode inviabilizar algumas aplicações. Uma outra possibilidade é o uso de uma conexão ponto a ponto via CSD (*Circuit Switched Data*). Neste caso, o sistema teria que ser planejado de acordo com a aplicação, para viabilizá-lo em termos de custos, além de oferecer também restrições em termos de taxa de transmissão. Assim, a opção foi o desenvolvimento de um sistema usando GPRS, que é uma tecnologia consolidada, possibilita uma taxa de transmissão bem superior ao que se consegue com SMS ou CSD e pode ser explorado em qualquer aplicação, mantendo-se um mesmo padrão de conectividade. Outra grande vantagem do GPRS é o seu baixo custo operacional em comparação com as alternativas de transmissão de dados usando SMS [8].

A integração do sistema de aquisição de dados com a Internet permite, sem dúvida, um monitoramento de dados mais eficiente e de fácil acesso porque utiliza o padrão World Wide Web (WWW) - qualquer dispositivo conectado à rede Internet poderá servir como interface de monitoramento.

A simples incorporação da tecnologia celular para a aquisição de dados para diversas aplicações seria por si só, um trabalho interessante para o uso desta tecnologia, visto tratar-se de uma solução inovadora e útil ainda que se constitua o uso mais simples e direto das características e recursos oferecidos pela tecnologia do SMC - Serviço Móvel Celular e Internet. O cenário de aplicação deste estudo inclui um ambiente complexo, pois se considera a busca de solução de uma ou várias aplicações de telemetria utilizando rede de comunicação de dados *wireless* de grande abrangência geográfica, alta capilaridade e de baixo custo. Isto exige uma exploração mais ampla e profunda da tecnologia celular, particularmente da rede GSM/GPRS, que é viável graças à integração de diferentes tecnologias em alguns equipamentos modernos. A implementação prática deste projeto envolve dispositivos e sistemas modernos, que vão desde a captação dos dados, que é feita

através de um microcontrolador, também usado na conexão sem fio, até a visualização dos dados em uma página WEB ([6] e [9]). A metodologia de trabalho utilizada para alcançar os objetivos estabelecidos considerou as atividades orientadas por dois eixos principais. O primeiro deles estava orientado para o desenvolvimento de atividades de estudo teórico e pesquisa baseada principalmente na literatura disponível sobre Redes de Telefonia Celulares em geral ([10] e [11]), Plataformas de Aplicação Web [12] e Dispositivos Microcontroladores [13]. O segundo eixo estava voltado para atividades de aplicação da tecnologia e conhecimento disponível para a solução de um problema específico, mas relevante no contexto atual do mundo em que vivemos [14].

A combinação destes dois focos, em que o conhecimento teórico alimenta as possíveis alternativas de solução de um problema, e, de outro lado, os problemas específicos de um caso concreto, fomenta e orienta a busca de mais conhecimento teórico, num processo contínuo de sinergia entre a prática e teoria, resultando numa dinâmica de trabalho de maior eficácia [15].

Para alcançar o objetivo proposto, foram estabelecidas e cumpridas as seguintes metas:

- Estudar os diversos sistemas de comunicação sem fio (telefonia celular) e identificar o mais adequado ao projeto proposto;
- Estudar os princípios de comunicação de dados via GPRS e Bluetooth [16], dois sistemas aparentemente semelhantes, mas que guardam muitas peculiaridades diferentes [9];
- Definir os contornos, requisitos básicos e funcionalidade de um sistema automatizado de monitoramento remoto de temperatura, pesquisando e estudando a operação de diversas placas/módulos GSM/GPRS existente no mercado;
- Desenvolver uma alternativa moderna para o projeto, estudando e pesquisando os microcontroladores PIC [17] e suas aplicações, com o objetivo de obter uma interface eficiente entre a aplicação e o ambiente WEB de monitoramento;

- Estudar os diversos tipos de dispositivos medidores de temperatura existentes no mercado e suas aplicações;
- Estudar tecnologias de plataformas WEB para captura e armazenamento de dados, bem como os mecanismos de disponibilização dos dados através de uma paginação de Internet [5];
- Considerar a possibilidade de aplicar os resultados obtidos em outros cenários similares que envolvam necessidades de comunicação com dispositivos dispersos em grandes áreas geográficas.

Para viabilizar o projeto proposto, o trabalho foi desenvolvido de acordo com as metas apresentadas. A estrutura da dissertação, após a apresentação deste capítulo, está organizada como segue:

No capítulo 2 discutem-se as tecnologias de comunicação sem fio usados em telefonia celular e os aspectos básicos que regem a rede mundial de computadores;

No capítulo 3 é apresentada a proposta completa do sistema desenvolvido, discutindo-se a visualização via internet e os aspectos básicos da tecnologia GPRS usada no projeto;

No capítulo 4 são apresentados os detalhes da implementação, com destaque para o sistema que faz a interface entre a rede GSM e a aplicação, que é baseado em microcontrolador;

No capítulo 5 são apresentados os resultados obtidos e no capítulo 6 são apresentadas as conclusões.

ASPECTOS BÁSICOS DE COMUNICAÇÃO SEM FIO E INTERNET

A transmissão de dados via telefonia celular se tornou uma solução inovadora de incorporação de novos valores em tecnologia, que permite somar a qualidade de serviços até então existente em operadoras de serviço de telefonia fixa com o fator mobilidade, o principal diferencial da telefonia celular. Aliado a isto tem-se hoje uma total integração entre a telefonia celular e a rede Web, o que forma um potencial tecnológico de vasta aplicação.

Em termos de comunicação sem fio, diversas tecnologias despontam no mercado apresentando soluções para Sistemas de Serviço Móvel Celular (SMC) [18]. Dentre várias existentes, podem ser citadas as tecnologias CDMA, GSM e 3G. Neste trabalho é dado um enfoque maior ao Sistema GSM - *Global System for Mobile*, que apresenta muitas vantagens em relação aos outros, que serão apresentadas ao longo do texto.

2.1 Histórico

Os Serviços Móveis Celulares (SMC) surgiram na década de 80 para atender uma necessidade latente de comunicação para uso pessoal. A primeira geração foi baseada em tecnologia analógica – os telefones eram, inicialmente, simples transmissores e receptores de rádio associados a um “número de telefone”. Tal sistema era bastante limitado, pois não garantia segurança nas comunicações e, principalmente, não atendia a demanda de usuários, que crescia vertiginosamente.

A limitação do sistema analógico fez com novas tecnologias fossem desenvolvidas, surgindo, então, os sistemas celulares de segunda geração, conhecidos como digitais, que permitem uma comunicação mais segura e, principalmente, que permitiram o desenvolvimento de terminais telefônicos compactos e de baixo consumo. As aplicações iniciais envolviam apenas sinais de voz, mas já era possível prever aplicações envolvendo a transmissão e recepção de dados, que veio logo em seguida. Porém, as conexões eram baseadas em alocação fixa de canal, tornando o sistema inadequado para aplicações em comunicação de dados de tráfego por demanda, como é o caso da Internet. O sistema CDMA-(*Code Division Multiple Access*) [19] baseou-se nesta tecnologia.

Visando atender ao aumento da demanda por tráfego de dados, principalmente na rede de *internet*, foi elaborado um projeto de terceira geração de telefonia móvel. Porém, antes foi criada uma família intermediária – a geração 2.5, agregando comunicação móvel de dados aos tradicionais serviços de voz. Com a evolução, o aspecto de segurança passou a ser um ponto importante, pois a comunicação de dados em sistemas móveis envolve troca de informação confidencial entre grandes blocos corporativos, além da privacidade exigida para as comunicações pessoais. As redes móveis agora têm como base uma tecnologia digital com transmissão por comutação de pacotes. Estes sistemas aceitam serviços clássicos de telefonia e comunicações de dados e são mais eficientes do que os sistemas de segunda geração, pois permitem que vários usuários e serviços utilizem simultaneamente os canais. Este sistema é conhecido como GSM - *Global System for Mobile* [20] e é baseado numa tecnologia TDMA (*Time Division Multiple Access*) [21].

A terceira geração dos SMC, conhecida como 3G, já implementada em vários países e em fase de implementação no Brasil, possibilita o acesso à internet em Banda Larga, o que irá viabilizar muitas aplicações específicas.

2.2 Sistema CDMA

O CDMA (*Code Division Multiple Access*) é um padrão digital de segunda geração da telefonia celular desenvolvido nos Estados Unidos [19]. Sua tecnologia utiliza espalhamento espectral (*spread spectrum*) de sinais como meio de acesso para permitir que vários usuários compartilhem uma mesma banda de frequências. Nessa tecnologia, os bits

provenientes da conversão da fala (conversão analógico-digital), em conjunto com outros bits de controle, são literalmente espalhados por todo o espectro de uma maneira "aleatória". As aspas são necessárias porque o aleatório do CDMA não é tão aleatório assim. O lado transmissor espalha os bits e o lado receptor reordena os bits. O CDMA permite uma melhor utilização do espectro possibilitando um aumento de capacidade dos sistemas celulares.

A família de normas da TIA (*Telecommunications Industry Association*) padronizou os sistemas celulares digitais de segunda geração conhecidos popularmente como CDMA, ou *cdmaOne*, que são baseados no padrão IS-95 [22]. Esta tecnologia foi, em grande, parte desenvolvida pela empresa americana *Qualcomm* [23].

2.2.1 Forma de transmissão do CDMA

A figura 2.0 mostra como é transmitido um canal de RF no sistema CDMA.

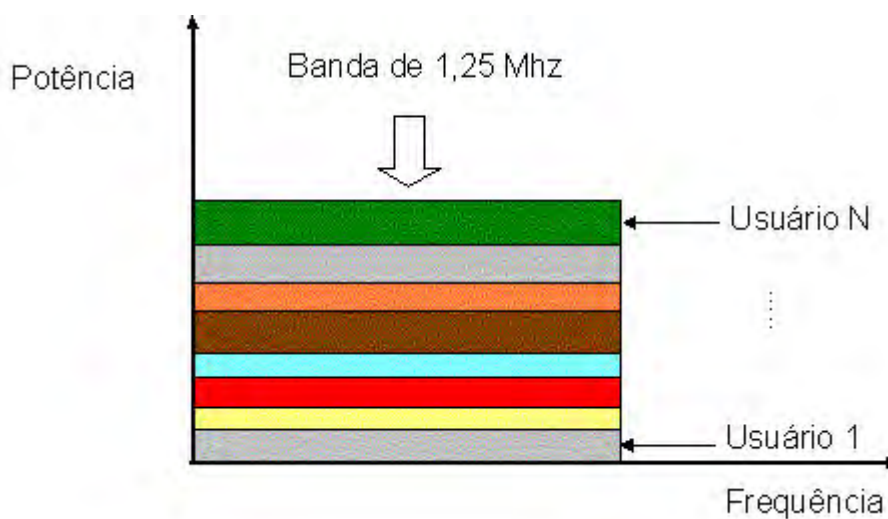


Figura 2.0 - Forma de Transmissão de um Canal de RF no sistema CDMA

Quanto maior o número de usuários, maior será o ruído e as interferências. Assim, existe um número máximo de canais por banda de frequência que permite um uso com qualidade aceitável. O problema se torna mais grave quando as potências de transmissão individuais aumentam.

2.3 Sistema 3G

A evolução do GSM para serviços de terceira geração, que permitem taxas de dados de até 2 Mbit/s, vem sendo padronizada pelo *3rd Generation Partnership Project* (3GPP). Esta evolução exigiu a definição de um novo padrão para a interface entre Estação Móvel e ERB com canais de RF de 5 MHz nos sistemas utilizados em CDMA. Este novo padrão WCDMA – *WideBand* CDMA [24] implica em mudanças na estrutura de canalização do GSM e exige uma banda adicional de frequências para implementação do serviço por parte das operadoras, mantendo, no entanto, a compatibilidade e demais interfaces da arquitetura GSM.

2.4 Sistema GSM – Global System for Mobile

Devido aos diversos sistemas que foram desenvolvidos com diferentes formas de envio de dados, protocolos e frequências de comunicação, em 1982 foi realizada a *Conference of European Posts and Telegraphs (CEPT)* onde se formou um grupo denominado *Group Special Mobile* (GSM), com o objetivo de estudar e desenvolver um sistema móvel baseado nas seguintes diretrizes básicas:

- Boa qualidade de voz;
- Eficiência espectral;
- Terminais pequenos e baixos custos;
- Suporte para "roaming" internacional;
- Capacidade para suportar "handheld" terminais;
- Suportar uma larga área de novos serviços e utilidades;
- Compatibilidade RDSI – Rede Digital de Serviços Integrados.

Em 1989 a responsabilidade passou para o *European Telecommunication Standards Institute* (ETSI) e em 1990 foram publicadas as especificações do GSM. Tal padrão generalizou-se então pelo resto do mundo. A sigla GSM passou a denominar então *Global System for Mobile Communication* e atualmente é considerado um padrão digital de

segunda geração do celular adotado na maior parte do mundo. Desenvolvido inicialmente para a faixa de 900 MHz, o GSM teve posteriormente uma versão adaptada para as faixas de 1800 e 1900 MHz [56].

2.4.1 Frequências de Operação do GSM (MHz)

Em um sistema de comunicação sem fio, existe as estações móveis (assinantes) e a estação fixa, conhecida como estação de Rádio Base (ERB). Como qualquer sistema de comunicação bi-direcional, os canais são pré-definidos por uma faixa de frequência. A tabela 2.0 ilustra a faixa de frequência de operação destes canais.

Tabela 2.0 – Frequência de Operação do GSM

	GSM 900 (MHz)	DCS 1800 (MHz)	PCS 1900 (MHz)
Estação Móvel -> ERB	880 - 915	1710 - 1785	1850 - 1910
ERB -> Estação Móvel	925 - 960	1805 - 1880	1930 - 1990
Espaçamento entre Frequências de Transmissão e Recepção	45	95	80

O GSM foi padronizado para operar nas faixas de frequências conforme apresentadas na Tabela 2.0, sendo o GSM 900 e o DCS 1800 adotados na Europa e o PCS 1900 nos Estados Unidos.

No Brasil, as Bandas C, D e E estão na faixa de frequências do DCS 1800, tendo sido licitados inicialmente 15 MHz por operadora em cada direção.

2.4.2 Descrição do sistema GSM

Uma rede GSM é composta por várias entidades com funções e interfaces específicas. A rede GSM pode ser dividida em três partes: a estação móvel, a estação de subsistema base, e o subsistema de rede, como apresentado na figura 2.1.

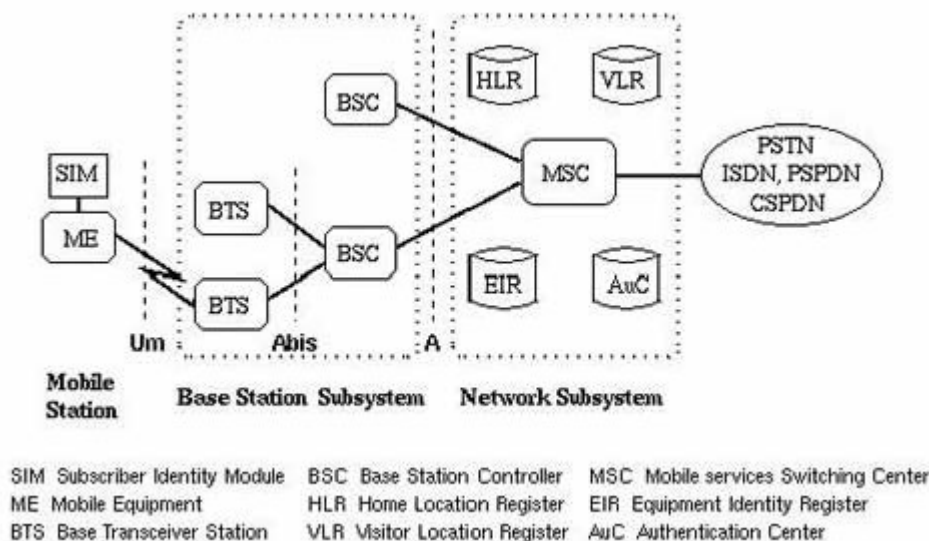


Figura 2.1. Estrutura de uma rede GSM.

2.4.3 A Estação Móvel

A estação móvel nada mais do que o aparelho celular (terminal) controlado por um cartão “inteligente” designado de SIM (*Subscriber Identity Module*). O cartão providencia mobilidade pessoal, de tal forma que o assinante consegue ter acesso aos serviços subscritos independentemente do terminal utilizado, isto é, ao inserir o cartão SIM num terminal diferente, o assinante pode usufruir os serviços a partir desse terminal. O cartão SIM tem uma identificação única mundial denominada de IMSI (*International Mobile Subscriber Identity*), o que garante maior mobilidade e segurança pessoal (uso não autorizado).

2.4.4 Subsistema Rádio Base

Este subsistema encarrega-se do controle de ligação rádio com a estação móvel. É dividido em duas partes: a estação rádio base de transmissão (BTS) e a estação rádio base de controle (BSC). A comunicação entre estas duas estações é realizada através da interface ABIS (*Air Base Interface Station*), permitindo (como no resto do sistema) a operação entre componentes realizada por diferentes fornecedores. A BTS é composta de receptores e transmissores de rádio, que definem a célula e suportam os protocolos de comunicação com

a estação móvel. O número de BTS's depende da área de cobertura do sistema. A BSC gerência os recursos para uma ou mais BTS's, tais como configuração dos canais de rádio, mudanças de frequência e transição entre células (*hand-off*). A BSC realiza a conexão entre as estações móveis celulares e o Centro de Comutação Móvel (MSC).

2.4.5 O Subsistema Rede

Este subsistema é composto basicamente pela MSC e é responsável pela comutação de chamadas entre estações móveis ou entre uma estação móvel e um terminal fixo. Comporta-se como um nó de comutação da PSTN (*Public Switched Telephone Network*) ou da ISDN (*Integrated Services Digital Network*) e, adicionalmente, providencia toda a funcionalidade necessária para o tratamento de um assinante móvel, tais como: registro, autenticação, atualização da localização, transição entre células (*Hand-off*), *roaming*, etc. Estes serviços são providenciados em conjunto com várias entidades funcionais e formam o subsistema rede, composto por MSC (*Mobile Services Switching Center*), HLR (*Home Location Register*), VLR (*Visitor Location Register*), EIR (*Equipment Identity Register*), AuC (*Authentication Center*). O HLR, o VLR e o MSC, em conjunto, providenciam as capacidades de *roaming* do GSM.

O HLR (*Home Location Register*) contém toda a informação administrativa de todos os assinantes registrados na correspondente rede GSM, juntamente com a localização da estação móvel. A localização da estação móvel está geralmente na forma do endereçamento do VLR (*Visitor Location Register*). As informações fornecidas pelo VLR são necessárias para controlar a chamada e providenciar os serviços de cada assinante, situada dentro de uma determinada área de controle. Outros dois registros são usados para segurança e autenticação. O EIR (*Equipment Identity Register*) é uma base de dados que contém listagens de todos os equipamentos móveis válidos na rede, com todas as estações móveis sendo identificadas pelo IMEI (*International Mobile Equipment Identity*). Um IMEI é considerado como inválido se declarado como roubado ou incompatível com a rede. O AuC (*Authentication Center*) é uma base de dados protegida que guarda uma cópia do

código de cada SIM (*Subscriber Identity Module*), que é usado para autenticar e criptografar dados do assinante através do canal de rádio.

2.4.6 Codificação de canal e voz

A voz em GSM é codificada digitalmente a uma taxa de 13Kbps (bits por segundo) e organizados em blocos de 260 bits a cada 20 ms. Com a adição posterior de código para a correção de erros, a taxa foi elevada para 22.8 Kbps e os dados agrupados em blocos de 456 bits a cada 20 ms. Os dados obtidos de 456 bits são organizados em 8 blocos de 57 bits e transmitidos em 8 *slots* de tempo sucessivos, o que protege contra erros de transmissão. Essa transmissão tem 156 bits e contém 2 blocos de 57 bits, além de uma seqüência de treinamento de 26 bits usada para equalização. Cada transmissão leva 0.577 ms para uma taxa total de 270.8 Kbps e é modulada usando GMSK (*Gaussian Minimum Shift Keying*) numa portadora de 200 kHz. O controle de erro e equalização contribuem para a robustez do sinal de rádio contra interferência e atenuação na transmissão. A natureza digital do sinal TDMA permite melhorar a qualidade de transmissão, aumentar o tempo de vida útil da bateria e a eficiência espectral.

O sistema GSM e os sistemas nele baseados, DCS1800 (operando a 1.8 GHz) e PCS1900 (operando a 1.9 GHz), são uma primeira aproximação para um sistema de comunicação verdadeiramente pessoal. O cartão SIM trouxe mobilidade pessoal e mobilidade para o terminal. Junto com o *roaming* internacional e o suporte a uma grande variedade de serviços tais como voz, transferência de dados, fax, SMS, e outros, o GSM chega próximo de uma satisfação total das necessidades de comunicação pessoal. Dessa forma, ele poderá ser baseado projeto UMTS (*Universal Mobile Telecommunication System*), que é uma das tecnologias que será utilizada da Terceira Geração – 3G. Deve-se salientar ainda que o GSM é compatível com a rede digital ISDN [57].

2.5 GPRS - General Packet Radio Service

Como já apresentado anteriormente, as redes GSM surgiram da tentativa do ETSI de padronizar os sistemas celulares privados que os países europeus vinham desenvolvendo

na década de 1980. Esses sistemas eram incompatíveis e permitiam apenas comunicação *roaming* nacional, limitando fortemente o mercado de celulares na Europa. O Sistema GSM foi desenvolvido para garantir boa qualidade de voz, baixo custo, compatibilidade com a rede ISDN e *roaming* por toda a Europa. No entanto, GSM cresceu muito mais do que o esperado e hoje é usado por mais de 200 países em todo o mundo [20].

Com a popularização do uso da Internet e de outros serviços de dados em meados da década de 1990, previu-se que as redes GSM não seriam capazes de comportar grandes quantidade de dados nos diferentes estágios do sistema. Para garantir os serviços futuros, em 1997 o ETSI publicou o modo de funcionamento do GPRS (*General Packet Radio Service*) na especificação da FASE 2+ do GSM.

As redes GPRS foram desenvolvidas para aceitar serviços de dados, pois foram criadas com base em transmissão por comutação de pacotes, diferentemente do GSM. Comutação de pacotes utiliza mais eficientemente a fonte de rádio para tráfego em rajadas, que é uma característica da maioria dos serviços de dados.

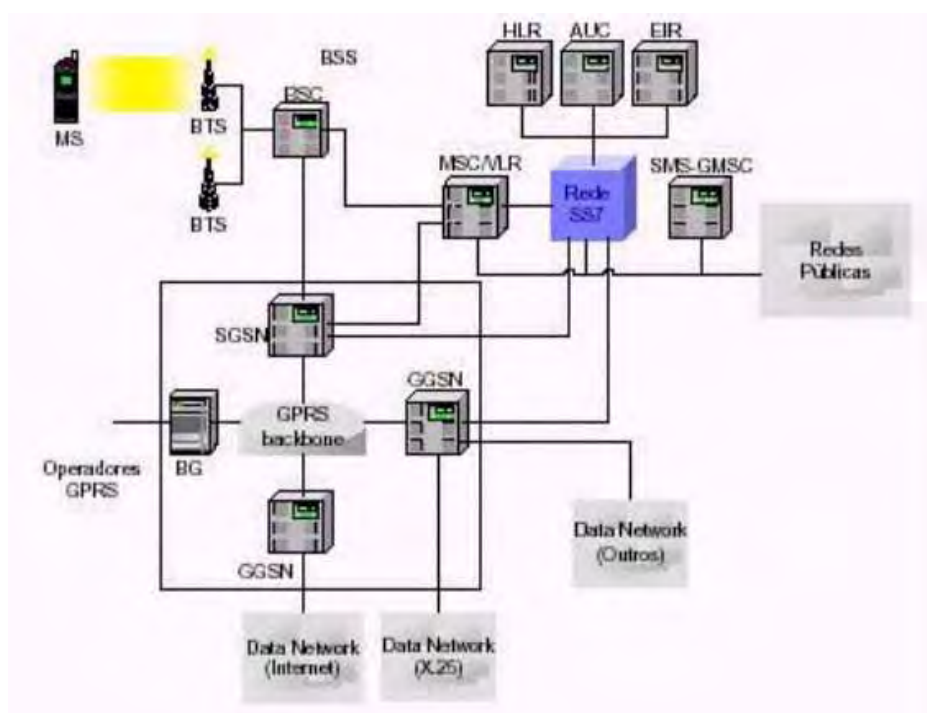
Para que as operadoras possam utilizar seus serviços GSM e os serviços de dados GPRS a partir de uma única base dinâmica e flexível, os dois sistemas compartilham várias características entre si, como bandas de frequência, estrutura de *frames* e técnicas de modulação. No entanto, a cobrança pelo uso do GPRS é feita por quantidade de dados transmitidos, enquanto que no GSM é feita por tempo de conexão.

A rede GPRS pode ser considerada como um revestimento à rede GSM, acrescentando tráfego orientado a pacotes mediante leves modificações da arquitetura. A rede GPRS pode ser analisada como sendo “GSM + dados”. Sua integração com a rede Internet permite o envio e o recebimento de dados de uma forma bem simples por qualquer aparelho que utiliza esta tecnologia.

2.5.1 Arquitetura de Redes GPRS

Para a adição de tráfego orientado a pacotes, alguns elementos foram substituídos e alguns adicionados à arquitetura GSM, viabilizando, assim, a rede GPRS. A principal diferença entre as arquiteturas é a substituição do MSC (*Mobile Services Switching Center*)

pelo SGSN (*Serving GPRS Support Node*) e a do GMSC (*Gate Mobile Services Switching Center*) pelo GGSN (*Gateway GPRS Support Node*). Entre os elementos que se diferenciam na rede GPRS estão: o BG (*Border Gateway*), o HLR (*Home Location Register*), o MSC/VLR (*Mobile Service Center/Visitor Location Register*), o SMS-GMSC (*Short Message Service – Gate Mobile Services Switching Center*) e o próprio *Backbone GPRS*.



O SMSC (*Short Message Service Center*) é responsável pelo tráfego de pacotes que tem as unidades móveis (MS) como origem ou destino e por funções como autenticação, roteamento, gerenciamento de mobilidade, controle de acesso e contabilidade de uso da rede de rádio.

parte dos MS's e por traduzir os formatos de pacotes de endereços externos para o formato de endereçamento GPRS e vice-versa.

Nesta arquitetura, tanto o SMSC quanto o GGSN possuem um LR (*Location Register*). O LR armazena a localização atual e as informações permanentes de todos os usuários que estão na área de serviço. É possível recuperar, a partir destas unidades, a célula e o endereço SMSC em que o MS está localizado, basicamente substituindo a funcionalidade do VLR nas redes GSM.

O *Backbone (Suporte Principal)* GPRS é essencialmente uma rede IP que interconecta os componentes internos às redes GPRS. O *roaming* entre diferentes redes é feito pelo BG. Algumas modificações foram feitas no HLR para comportar as informações GPRS pertinentes aos usuários. Este agora também armazena o endereço PDP (*Packet Data Protocol*) de cada usuário.

2.5.2 Elementos Internos de uma Rede GPRS

O GSN (*GPRS Support Node*) é composto pelos elementos que compõem a rede GPRS, os quais são interconectados pelo *backbone*. Os principais elementos são apresentados pela Figura 2.3.

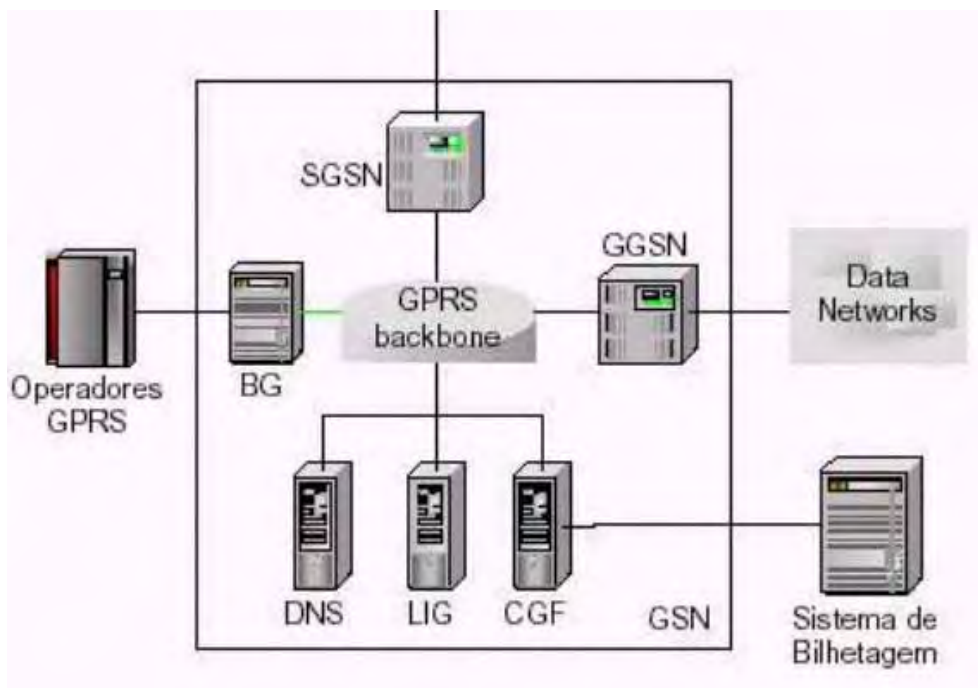


Figura 2.3 – Elementos Internos de uma Rede GPRS

O CGF (*Charging Gateway Functionality*) é responsável pela coleta e direcionamento ao sistema de bilhetagem dos dados de contabilidade; o LIG (*Lawful Interception Gateway*) permite interceptar e armazenar o tráfego de um determinado MS por questões legislativas; o DNS (*Domain Name Server*) é um servidor de nomes convencional que traduz nomes lógicos em endereços IP.

2.5.3 Forma de Transmissão de uma Rede GPRS

A transmissão de dados a partir do usuário é feita na forma de protocolos em camadas. Esta estrutura de protocolos é denominada pelas especificações do GPRS de Plano de Transmissão. As especificações do GPRS padronizaram o plano de transmissão para permitir serviços de dados IP e X.25. A figura 2.4 apresenta o plano de transmissão para o TCP/IP [18].

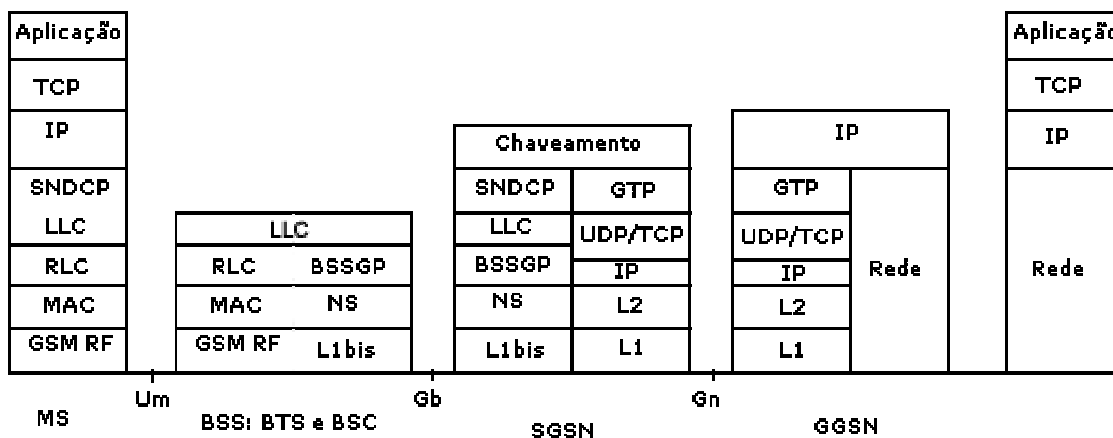


Figura 2.4 – Plano de Transmissão para TCP/IP

- **SNDCP (Subnetwork Dependent Convergence Protocol):** esta camada faz o mapeamento de características ao nível de rede em características das camadas inferiores da rede entre a Estação Móvel e o SGSN. Ela está especificada em GSM 04.65.
- **LLC (Logical Link Control):** esta camada prove um link lógico altamente confiável e criptografado. Ele é especificado em GSM 04.64.
- **BSSGP (Base Station System GPRS Protocol):** esta camada é responsável pelo roteamento e informação relativa a QoS entre o BSS e o SGSN. Ela não faz correção de erro. BSSGP esta especificada em GSM 08.18.
- **NS (Network Service):** esta camada transporta pacotes de dados do BSSGP. Ela é baseada no protocolo *Frame Relay* e especificada em GSM 08.16.
- **RLC/MAC (Radio Link Control/Media Access Control):** esta camada tem duas funções: O *Radio Link Control* fornece um *radio-solution-dependent reliable link*. A função de controle de acesso ao meio (MAC) controla a sinalização de acesso ao canal de radio, e o mapeamento dos *frames* LLC nos canais físicos RF gsm. RLC/MAC é definido em GSM 4.60

- GTP (*GPRS Tunnelling Protocol*): este protocolo provê um túnel para dados do usuário e sinalização entre os Nós de suporte do GPRS no *backbone* GPRS. GTP é especificado em GSM 09.60.
- L2: Camada 2 do modelo OSI (*Open Systems Interconnection*).
- L1: Camada 1 do Modelo OSI.

2.5.4 Conexão de dados entre um terminal móvel e um servidor de uma rede IP externo.

Este tipo de conexão é necessário em serviços de e-mail, acesso à internet ou a um servidor WAP (*Wireless Application Protocol*). Na operação, um pacote de dados TCP/IP é mapeado na camada LLC pelo SNDCP. A camada LLC garante um serviço confiável no enlace entre a estação móvel e o SGCN. Para chegar ao SGCN este pacote utiliza camadas de protocolo específicas das interfaces “Um” (composto de RLC/MAC e interface rádio) e “Gb” (BSSGP e NS baseado em *Frame Relay*) [25]. O chaveamento entre as camadas RLC e BSSGP no BSS é feito na camada LLC.

No SGSN, os pacotes são chaveados para o *backbone* GPRS (interface GN) onde são transportados através de um protocolo de tunelamento de dados (GTP) em uma rede IP utilizando TCP ou UDP como camada de transporte. Os dados trafegados no *backbone* GPRS acabam tendo o IP em dois níveis. Este procedimento não é o mais eficiente, mas torna a solução segura e fácil de implementar[18].

Finalmente, no *Gateway*, o pacote de dados é roteado através de uma rede IP externa até o servidor de aplicação

De modo análogo, as especificações do GPRS/GSM definem planos de sinalização para conexões entre os vários nós envolvidos na prestação do serviço GPRS. Estes protocolos são baseados no protocolo de sinalização SS7 [26].

2.5.5 Forma de Identificação de assinante em uma Rede GPRS/GSM

Cada usuário tem sua IMSI (*International Mobile Subscriber Identity*), única em todas as redes GSM ao redor do mundo, que não varia em função do tempo (a não ser em caso de perda ou troca do *SIM-Card* pelo assinante, por exemplo) [59]. O IMSI tem, nas

redes GSM, função análoga ao do ESN (*Eletronic Serial Number*) que é constituído por 11 números que indicam a combinação entre o número do telefone celular e o seu número serial existente nos celulares analógicos. Como esta identificação única, se fizer trafegar abertamente, irá permitir que o telefone possa ser clonado ou que o assinante do serviço possa ter sua localização traçada.

Para evitar que o IMSI seja interceptado, a rede GSM utiliza o recurso do TMSI (*Temporary Mobile Subscriber Identity*). Em geral, no momento em que o aparelho móvel é ligado, o IMSI é transmitido e a rede, no VLR, estabelece uma relação entre o IMSI e um TMSI gerado. A partir deste momento, somente o TMSI trafega. O IMSI só é reutilizado em caso de perda do TMSI por pane da MS ou da rede.

A substituição do TMSI ocorre a cada troca de VLR (havendo possível troca de operadora) e, eventualmente, mediante requisição de uma das partes. O esquema da figura 2.5 exemplifica a troca do TMSI no momento de atualização da localização geográfica feita pelo aparelho.

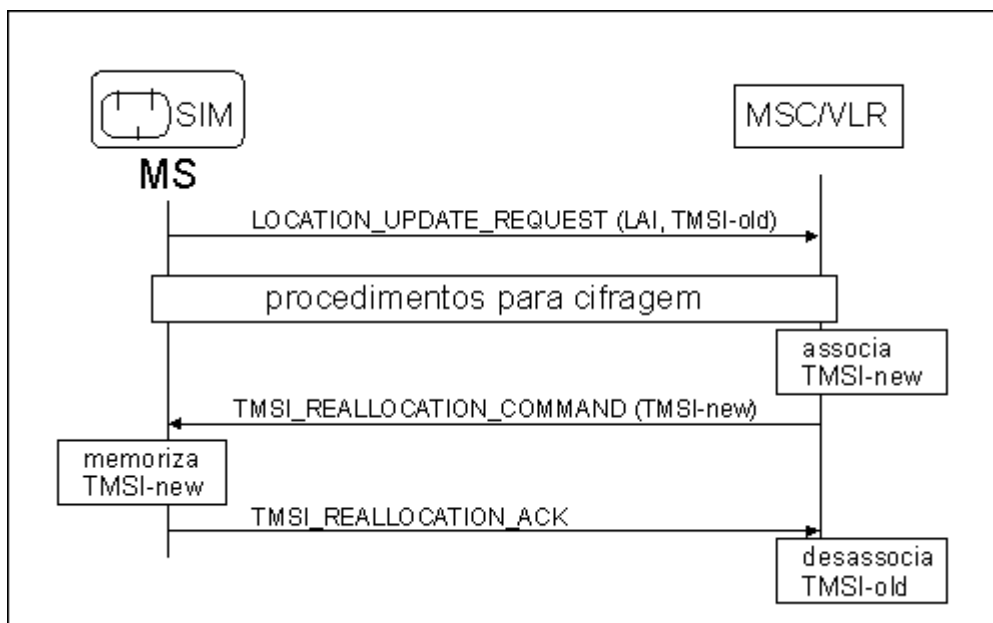


Figura 2.5 – Esquema de troca do TMSI no momento de atualização da localização geográfica feita pelo aparelho

2.5.6 Autenticação de um assinante em Redes GPRS/GSM

Autenticar a identidade do assinante (ou simplesmente identificá-lo) é assegurar que ele é realmente a entidade que diz ser. No caso dos telefones móveis, dá-se à entidade móvel, a partir de sua identificação, os direitos (de recepção e transmissão de informações) que o assinante teria sobre a rede. Se um usuário ‘L’ inapropriadamente se identificar como outro usuário (*impersonation*), por exemplo ‘A’, então ‘L’ poderá utilizar recursos como utilizar o sistema e fazer com que a cobrança decaia em ‘A’, se identificar como ‘A’ para receber mensagens de caixa postal, SMS, e-mail ou ainda utilizar o nome de ‘A’ para atividades ilegais (vide, por exemplo, episódio da utilização de telefones móveis em penitenciárias brasileiras).

As redes GSM-GPRS autenticam a identidade do assinante através de um mecanismo de desafio e resposta (*challenge-response mechanism*). A BTS envia para a MS um número aleatório de 128 bits (RAND). O SIM-Card então recebe o número RAND e computa, fornecendo RAND e uma chave Ki (compartilhada entre o SIM-card e a AUC) como parâmetro do algoritmo A3 – uma função de *Hash* dependente de chave – um número (SRES) de 32 bits. O número SRES é verificado pela AUC, que também conhece A3, Ki e RAND. O SRES calculado pelo SIM só será igual, portanto, caso ele tenha a mesmo Ki. Uma das questões polêmicas quanto a este procedimento é que a baixa capacidade de processamento do SIM-Card impede que Ki seja uma chave grande, o que torna o sistema relativamente frágil (vide item 6 sobre possibilidades de ataques). Como A3 é um algoritmo somente utilizando dentro do SIM-Card e da AUC, ele não é especificado pelos padrões GSM ou GPRS – cada operadora é livre para adotar seu próprio A3 [27].

2.5.7 Terminais GPRS –Conexão

A conexão de um terminal a uma rede GPRS é feita através dos seguintes passos:

1. Um terminal GPRS, ao ser energizado, será reconhecido pela rede de forma semelhante ao que ocorre com um terminal GSM para Voz. É então criado um enlace lógico entre o terminal e o SGSN. O Terminal é dito *attach*, o que significa que ele está registrado e autenticado na rede.

2. O próximo passo é conseguir um endereço IP estabelecendo uma conexão em GPRS, através da ativação do contexto do *Packet Data Protocol*. Este endereço IP é normalmente dinâmico sendo fornecido pela operadora móvel ou outro operador dependendo de como está configurada a rede.
3. O Terminal GPRS está então pronto para enviar e receber pacotes. Ele pode então assumir os seguintes estados de forma a economizar energia: *Idle* (ocioso), *Ready* (pronto) em que ele pode enviar e receber pacotes instantaneamente ou *stand-by* (Espera).

2.5.8 Classes de Terminais GPRS

As especificações GPRS definem três classes de terminais:

Classe A: Terminais que podem tratar voz e dados ao mesmo tempo.

Classe B: Terminais que podem tratar voz e dados, mas não ao mesmo tempo.

Classe C: Terminais que podem tratar apenas dados, como cartões GPRS PCM/CIA para computadores portáteis.

Devido ao alto custo dos terminais Classe “A”, a maior parte dos terminais lançados comercialmente é de classe B.

2.5.9 Interface R

O terminal GPRS pode ser utilizado diretamente para acesso de dados ou internet utilizando o WAP ou pode ser conectado a um outro equipamento, como por exemplo, um microcomputador. Um exemplo de conexão que pode ser utilizada neste caso é o *Bluetooth*[5].

As especificações do GSM definem uma interface de referência denominada de Interface R entre o terminal móvel e o equipamento terminal, quando estes estão fisicamente separados. Foram definidos comandos de atenção (AT), de acordo com a recomendação ITU V.25 ter (*Serial Asynchronous Dialing and Control*). A especificação GSM 07.07 descreve o conjunto de comandos AT para terminais GSM [28].

2.5.10 APN (Access Point Name)

A conexão entre o operador e uma rede IP externa é feita através de um APN (*Access Point Name*). O operador estabelece APNs para as várias redes, sendo um tipicamente definido para a rede pública WAP. O número de APNs de um terminal varia com o modelo e fabricante.

2.6 Internet - A Rede mundial de Computadores

Uma das inovações importantes deste trabalho é o uso da internet para monitorar os dados à distância. Como se sabe, essa rede está monopolizando todos os meios de comunicação. Atualmente, até pode-se utilizar na prática, serviços de comunicação de voz, processo inimaginável até pouco tempo atrás quando surgiu a “Grande Rede” ou a “Rede das Redes”. E os limites a serem alcançados simplesmente não se resumem somente a isso. Já se discutem meios para a transmissão digital de imagem em tempo real com alta qualidade, além da esperada convergência tecnológica, que prevê que todos os equipamentos eletrônicos poderão, no futuro, estarem conectados à Grande Rede.

2.6.1 Histórico

A Internet foi idealizada e criada em meados do ano de 1962 pelo governo norte americano com objetivos militares. O objetivo era manter o comando de mísseis e bombardeios depois de ataque nuclear, ou seja, uma rede de pesquisa nuclear que sobrevivesse a uma catástrofe nuclear de maneira que se qualquer cidade dos EUA fosse atacada, os militares teriam o controle de suas armas nucleares para um possível contra-ataque. Posteriormente, outros projetos surgiram como, por exemplo, a ARPANET, que colocou em rede os computadores do Departamento de Defesa dos EUA. Porém, a Internet tal qual se conhece, com uma interface gráfica denominada de WWW (*World Wide Web*) foi efetivamente implementada em 1993, época em que deixou de ser uma rede meramente acadêmica e passou a ser explorada comercialmente, passando a ter um *backbone* base para acesso à rede internet.

Atualmente, ela agrega diversas modalidades de serviços que permitem atender desde um simples usuário doméstico até grandes empresas e corporações, oferecendo

serviços complexos como a rede VPN (*Virtual Privative Network*). Para cada tipo de serviço utilizado existe um protocolo padrão da camada de Aplicação do Modelo OSI (*Open Systems Interconnection*) que é a linguagem de comunicação entre dois computadores. Além disso, toda conexão só é efetuada a partir da abertura de uma porta de comunicação.

2.6.2 Serviços de Internet

Entre várias modalidades de serviços existente na Grande Rede podem ser destacados:

Hospedagem de Páginas e Banco de Dados em um Servidor de Rede

O serviço de Hospedagem de Páginas, sem dúvida o mais utilizado, permite que qualquer usuário armazene e disponibilize informações para qualquer pessoa em qualquer parte do mundo em um provedor de uma forma simples e de fácil visualização. As informações podem estar em formato de texto, imagem, som e, atualmente, até em vídeo. Sua interface gráfica amigável, cujo arranjo e disposição foram previsto para a tela de um computador, permite atualmente a disponibilização em qualquer dispositivo que aceite o protocolo padrão HTTP (*Hiper Text Transfer Protocol*). Essas páginas podem possuir um caráter dinâmico, ou seja, interagir com quem estiver momentaneamente visualizando e trocar informações com o servidor em que estiver hospedado. Caso este servidor possua um Banco de Dados integrado com a interface Web, o mesmo pode oferecer as informações nele contidas.

Transferência de Arquivos

Para a Transferência de Arquivos de um servidor para outro se utiliza, na Internet, o protocolo FTP (*File Transfer Protocol*), que permite através de uma conexão virtual uma transferência rápida e segura.

E-mail

O serviço de entrega de mensagem eletrônica E-mail foi um dos primeiros a ser utilizados na Internet e, até hoje, ainda é um dos mais utilizados. Ele utiliza o protocolo POP3 (*Post Office Protocol*), que serve para o acesso remoto a uma caixa de correio eletrônico, e

também o SMTP (*Simple Mail Transfer Protocol*), que permite o envio e a postagem de uma mensagem eletrônica.

2.6.3 Endereçamento na Rede Internet

A Rede Internet utiliza o endereçamento IP (*Internet Protocol*) [55] para identificar uma máquina conectada a sua rede TCP/IP, que normalmente é um computador ou um roteador.

Tal endereço é um número composto por quatro Bytes ou Octetos, normalmente representados por pontos que tem os formatos x.y.z.w. Na Rede, não podem existir duas máquinas com o mesmo número IP. Se for configurada uma nova máquina com o mesmo endereço IP de um outro já existente, será gerado um conflito de Número IP para o segundo equipamento. O valor máximo para cada um dos números (x, y, z ou w) é 255 (2^8-1).

Uma parte do Número IP (1, 2 ou 3 dos 4 números) é a identificação da rede e a outra parte é a identificação da máquina dentro da rede. A definição dos números de identificação da rede e dos números de identificação da máquina é feita através de uma máscara de sub-rede (*subnet mask*). Na Figura 2.6 mostra-se uma rede com endereçamento baseado no protocolo TCP/IP. Se for tomado como exemplo de endereço do IP da máquina o número 10.200.150.1 e da sua Máscara de Sub-Rede o número 255.255.255.0, é verificado que as três primeiras partes da máscara de sub-rede (*subnet*) iguais a 255 indicam que os três primeiros números do endereço de rede representam a identificação da rede e o último número é a identificação do equipamento dentro da rede. Neste caso, a rede tem o endereço: 10.200.150 e, portanto, o endereço IP de todos os equipamentos desta rede começam com este número. Neste caso, tem-se um limite de 254 equipamentos que podem ser ligados nesta rede. O primeiro número (10.200.150.0) e o último número (10.200.250.255) não podem ser utilizados como números IP de equipamentos de rede. O primeiro é o próprio número da rede e o último é o endereço de Broadcast. Ao enviar uma mensagem para o endereço de Broadcast, todas as máquinas da rede receberão a mensagem.

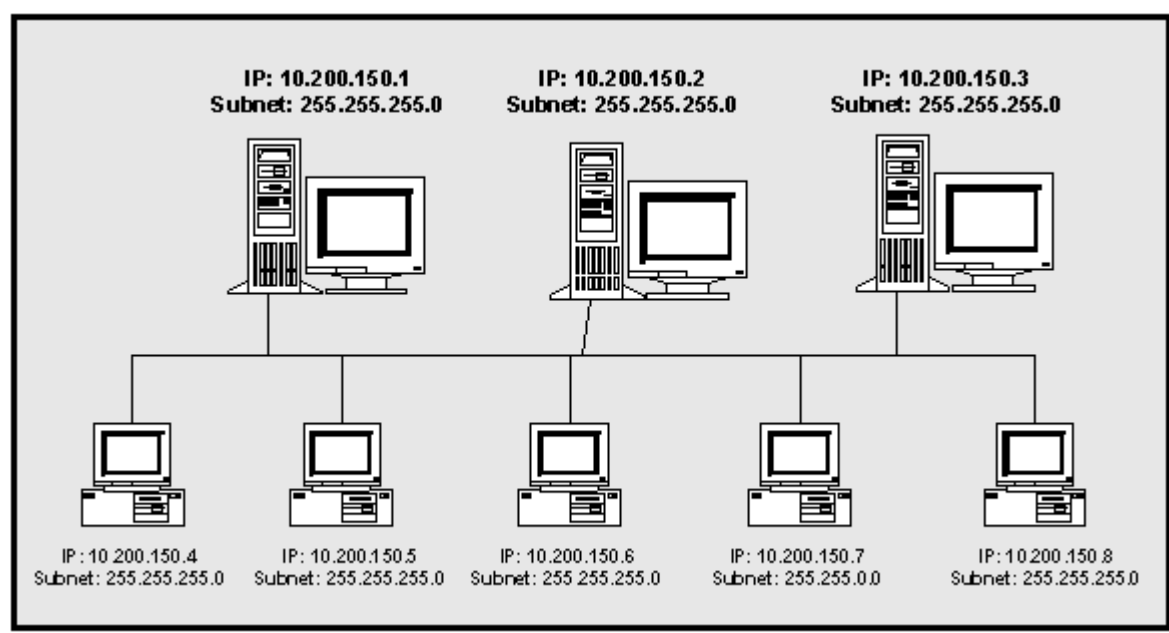


Figura 2.6 - Uma Rede baseada no Protocolo TCP/IP

2.6.4 DNS – Domain Name Server

O DNS é um serviço de resolução de nomes análogo a uma lista telefônica, que armazena todos os nomes e endereços, com os respectivos números de telefones, que tem como objetivo facilitar a consulta de um *site* através do navegador de Internet. Assim, um usuário da rede envia a busca através do nome do servidor ao invés do seu número IP, pois, para o usuário, é muito mais fácil decorar um nome do que consultar uma tabela de números IP. Por exemplo, pode-se digitar “<http://www.microsoft.com/brasil>” para acessar o site da Microsoft no Brasil, sem ter que se preocupar e nem saber qual o número IP do servidor onde está hospedado o site da Microsoft Brasil, pois o protocolo TCP/IP “descobre” (o termo técnico é resolver o nome) qual o número IP está associado a este endereço. O DNS pesquisa na sua base de dados ou envia a pesquisa para outros servidores DNS. Uma vez encontrado o número IP, o DNS retorna o número IP para a máquina que efetuou a consulta.

O DNS é baseado em conceitos de espaço de nomes e árvore de domínios. O espaço de nomes da Internet é hierárquico, baseado no DNS. Na figura 2.7 é representado este conceito.

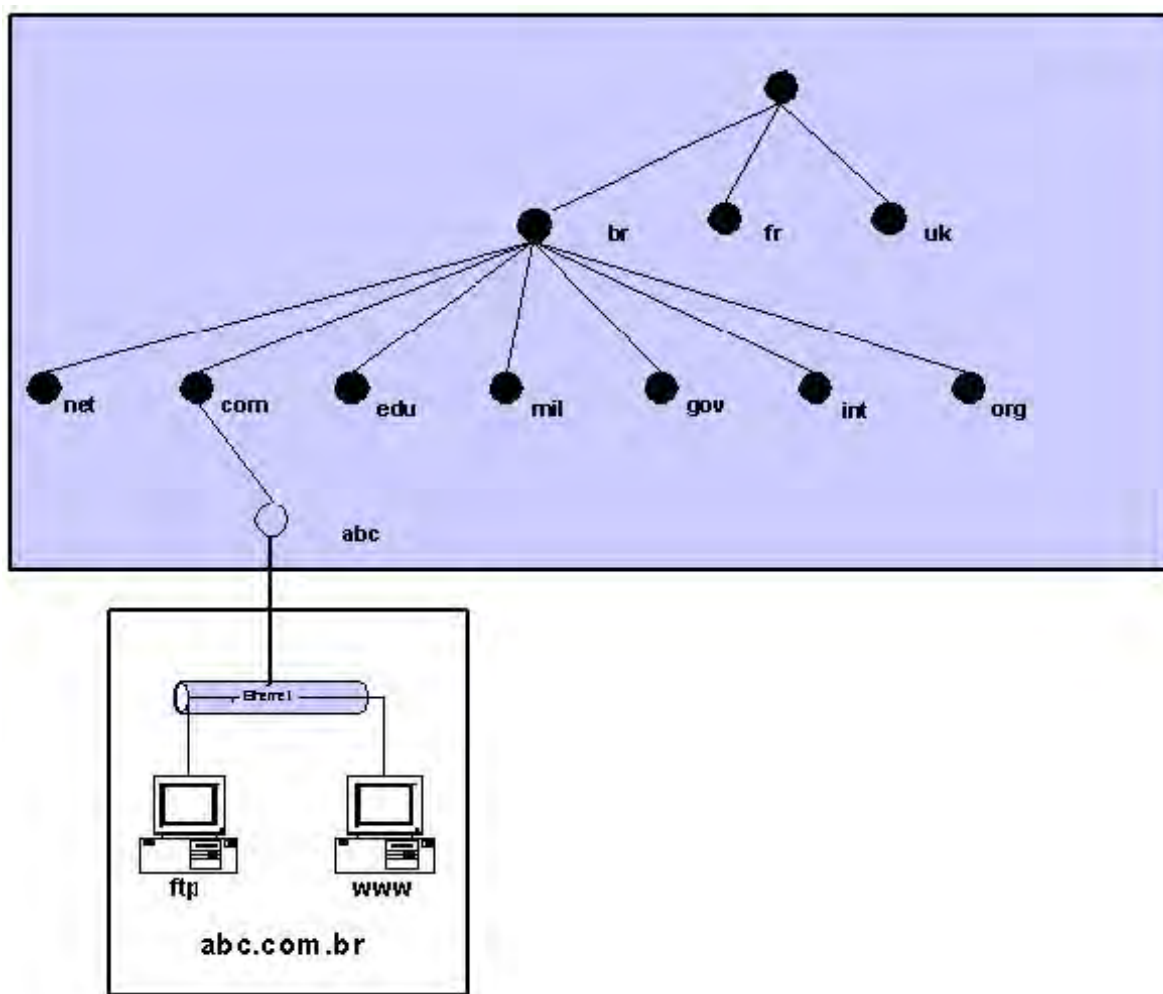


Figura 2.7 - Estrutura hierárquica do DNS

O principal domínio, o domínio *root*, é o de mais alto nível. No exemplo da figura 2.7, o mesmo está nomeado como sendo “abc” e separado por um ponto (.). No segundo nível estão definidos os domínios de níveis mais elevados (*Top-level-domains*). Estes domínios são bastante conhecidos, sendo os principais descritos na Tabela 11 [29].

Tabela 2.1 - Domínios de Níveis Mais Elevados

<i>Domínios de Níveis Mais Elevados</i>	Descrição
Com	Organizações comerciais
Gov	Organizações governamentais
Edu	Instituições educacionais
Org	Organizações não comerciais
Net	Diversos
Mil	Instituições militares

Na sequência da definição do endereço, a estrutura hierárquica continua aumentando, sendo então criados os subdomínios para cada país. Por exemplo: *br* para o Brasil (.com.br), *fr* para a França (.com.fr), *uk* para a Inglaterra (.com.uk) e assim por diante. Pode-se observar que o nome completo de um domínio é o nome do próprio domínio e mais os nomes dos domínios acima dele, no caminho até chegar ao domínio root. Neste exemplo, todos os equipamentos da rede da empresa abc.com.br, fazem parte do domínio.

ARQUITETURA PROPOSTA PARA O SISTEMA DE MONITORAMENTO DE DADOS VIA INTERNET USANDO TELEFONIA CELULAR

A tecnologia do Sistema de Rede Celular é bastante promissora como solução de comunicação pessoal à longa distância. Além disso, já existe a possibilidade de se aplicar esta tecnologia também nas redes de comunicação de dados de cobertura geográfica mundial, que podem dar suporte às redes de telemetria das mais variadas aplicações, como, por exemplo, o monitoramento de sistemas transmissão e distribuição de energia elétrica, de redes de telecomunicações, etc. Neste capítulo são discutidas e propostas soluções para um Sistema de Monitoramento de Dados via Internet usando Telefonia Celular através de uma arquitetura simples e custo bastante reduzido, tanto na implementação como também na utilização do serviço.

3.1 Contexto Tecnológico

A comunicação sem fio, ou simplesmente *wireless* [7] é, sem sombra de dúvida, um dos temas mais atuais na área de telecomunicações e de informática. Isto é devido, em grande parte, ao crescimento vertiginoso de celulares, PDAs (*Personal digital assistants*) e outros dispositivos móveis. No ano de 2005, foram vendidos mais 800 milhões de celulares no mundo[30], apenas para citar um exemplo. Além disso, aumentou-se a disponibilidade de serviços de Internet móvel, através de WAP, *I-Mode* (Japão) ou de redes locais sem fio do tipo Wi-Fi (IEEE802.11).

A tendência de demanda cada vez maior por equipamentos e serviços que possam ser usados em qualquer hora e em qualquer lugar parece não ter limites. Os exemplos são muitos, como por exemplo: serviços de mensagens de texto, telefonia, controle de

equipamentos à distância, monitoramento remoto de pacientes, serviços de transmissão de áudio e vídeo, etc [18].

Para atender essas expectativas, a indústria vem investindo pesado em diversas tecnologias, abrangendo desde soluções de redes de longa distância, como, por exemplo, a tecnologia celular de terceira geração (3G), passando por soluções de redes locais *wireless* (WLAN), onde o padrão IEEE802.11x ainda é o destaque, ou soluções *wireless* para redes de curta distância (*PAN* de *Personal Área Network*), onde diversas soluções brigam por espaço de mercado (por exemplo IrDA-*Infrared Data Association*, HomeRF e Bluetooth).

Neste contexto de comunicação sem fio, a Tecnologia GSM/GPRS, embora não seja a mais recente, é a que oferece melhor desempenho em termos de confiabilidade e qualidade de comunicação a um custo relativamente baixo. As aplicações com GSM/GPRS são convencionalmente denominadas de *WPAN* (*Wireless PAN* ou Rede Pessoal Sem Fio), mas com recursos para trabalhar com outros tipos de redes, tanto na faixa local como na faixa com alcance global. Trata-se de uma tecnologia bastante poderosa e versátil devido a sua eficácia na atuação de envio de informação, pois utiliza o formato de texto.

A tecnologia GSM/GPRS é uma integração das redes GSM (*Global Service for Mobile*) e GPRS (*General Packet Radio Service*), ou seja, é a incorporação de uma rede GPRS em cima da outra rede celular GSM, que possui abrangência global. A rede GPRS é uma rede de pacote de dados muito semelhante à rede de pacotes de dados X.25, ainda muito utilizada em redes de telefonia fixa. Diferentemente de outras modalidades de serviços de dados tais como acesso à Internet e Rede Virtual Privativa, que são tarifados de acordo com o tempo de utilização ou taxa fixa mensal, o serviço GSM/GPRS é oferecido e tarifado pelas operadoras de telefonia celular por quantidade de bytes de dados trafegados, o que reduz custos e o torna atrativa em muitas aplicações, como no caso do sistema proposto neste trabalho.

A tecnologia GSM/GPRS explora as redes de celulares e Internet já disponíveis, o que isenta o usuário de construir e manter a sua própria rede. A grande vantagem é a integração com a Internet, que permite conexão com qualquer ponto do mundo em diferentes equipamentos. Essa versatilidade é algo importante em qualquer sistema de monitoramento remoto de dados.

A tecnologia GSM/GPRS possui também alguns recursos de segurança, que podem ser usados para controlar e proteger conexões entre dispositivos. Estes recursos incluem chaves de autenticação, de criptografia, código de identificação pessoal (PIN, *Personal Identification Number*) e o uso de esquemas de segurança dos protocolos transportados nas camadas superiores.

3.2 Interface GPRS – INTERNET

A Interface GPRS-Internet é um dispositivo que permite realizar a conexão do equipamento do usuário com uma Rede GPRS-Internet. De modo análogo a um telefone celular, ele realiza basicamente, ao invés de voz, transferência de dados. Essa interface nada mais é do que um Modem (Modulação/Demodulação), que ficou bastante conhecido no início das comunicações de dados, já que era o dispositivo que permitia a conexão entre microcomputadores e a internet, ou mesmo efetuar uma conexão discada à um servidor *dial-up* (Computador que atende a uma ligação telefônica). Para isto, havia a necessidade inicial de executar todo um ritual ou procedimento que ficava a cargo do PC (*Personal Computer*). As aplicações do usuário ou informações a serem enviadas eram geradas por este mesmo PC ou agregadas através dele em uma das portas serial ou paralela.

Atualmente já existem interfaces microprocessadas que agregam todas estas funções, tanto de conexão como também de atividades antes executadas pelo PC, o que simplifica e facilita o desenvolvimento de projetos. Estes equipamentos são configurados através de programas muito compactos como por exemplo o javaME ou J2ME (*Java Micro Edition*) [31] que possibilita o desenvolvimento de software para sistemas e aplicações embarcadas. O esquema da figura 3.0 mostra as comparações entre uma interface (Modem) tradicional e outra com programação JAVA embarcada.

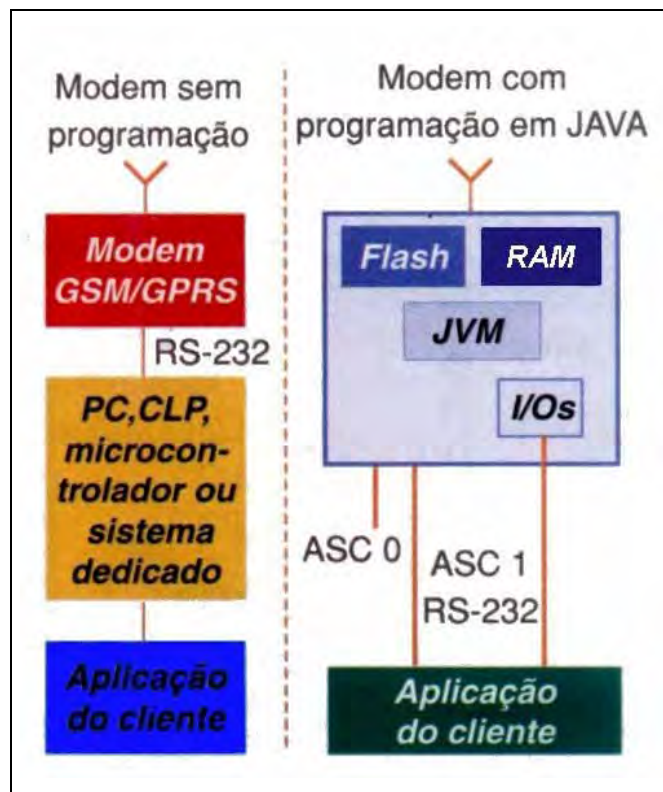


Figura 3.0 – Comparação entre Modem com e sem programação JAVA

Nesta figura comparativa, observa-se que, na aplicação com o Modem sem programação, é necessário um sistema de processamento externo como PC, CLP (Controlador Lógico Programável) ou um outro sistema dedicado para gerenciar a comunicação. Já no sistema embarcado este gerenciamento é feito internamente, tornando-se mais eficiente ainda em sistemas que necessitam de comunicação via GPRS, pois o gerenciamento da conexão GPRS e da Pilha TCP/IP é realizado internamente. Ele permite também efetuar conexões como CSD (*Circuit Switched Data*) ou SMS (*Short Message Service*). Além disso, alguns equipamentos permitem realizar a atualização do software JAVA remotamente através da função OTAP (*Over The Air Provisioning of Java Applications*)[32].

Atualmente, existem várias soluções no mercado. Dentre elas podem ser destacadas os Módulos da Motorola *g18*, *c18*, *g18 Data Modem* [33], bem como o Terminal Java TC45 da DuoDigit [34]. Estes equipamentos possuem características funcionais muito semelhantes entre si, porém as informações prestadas pelo fabricante para seu estudo, bem

como para sua configuração, são bastante restritos para a linha Motorola. Já o Terminal Java TC45 da DuoDigit possui um suporte técnico eficiente e um ferramental que facilita o desenvolvimento de projetos autônomos.

Outros pontos que influenciaram a adoção de uma de Interface GPRS-Internet neste projeto são:

- **Facilidade de Conexão com a Rede Internet**

Ao contrário de uma conexão discada CSD (*Circuit Switched Data*), que precisa estabelecer uma conexão física com a Rede Internet através de um provedor de acesso à Internet, quando um terminal GPRS é ligado numa rede GSM, o mesmo é reconhecido de forma semelhante aos terminais de voz. A partir desse instante é estabelecido um enlace lógico. Na sequência, a rede gera um endereço IP (*Internet Protocol*) e estabelece a conexão via PDP (*Packet Data Protocol*). O endereço IP gerado é normalmente dinâmico e é baseado na configuração estabelecida pela operadora ou pelo fornecedor do serviço. Portanto, como a Interface GPRS-Internet, uma vez ativada, fica todo o tempo em compasso de espera para o envio de dados, ao receber estes dados na sua entrada serial ele estabelece automaticamente uma conexão virtual GPRS com a Internet.

- **Baixo custo operacional em comparação com as alternativas de transmissão de dados usando SMS**

Foi feita uma análise comparativa dos valores médios de mercado, para uma média de quatro transmissões por hora ou 2880 transmissões mensais. Estes pacotes têm poucos dados a serem transmitidos. Em CSD a tarifação mínima é de trinta segundos por ligação e em SMS o custo é por mensagem transmitida, podendo-se enviar até 160 caracteres na mesma mensagem. Já em GPRS deve-se contabilizar os *headers* (Cabeçalho) TCP ou UDP mais os bytes de conexões e retransmissões. Nesta avaliação considerou-se uma média de 50 bytes para transmitir a mensagem em UDP e 150 bytes para transmitir em TCP. A tabela 3.0 resume o resultado da comparação entre SMS e GPRS. Verifica-se que o GPRS é mais vantajoso nesse tipo de aplicação. Já numa comparação entre GPRS e CSD, a aplicação deve ser considerada para se chegar a uma melhor opção de custo.

Tabela 3.0 – Custo Operacional

Sistema	Assinatura	Custo Unitário	Total Mensal
SMS	R\$20,00	R\$0,14/ mens	R\$423,20
GPRS UDP	R\$10,00	R\$6,00 / Mbytes	R\$15,00
GPRS TCP	R\$10,00	R\$6,00 / Mbytes	R\$15,00

No desenvolvimento deste trabalho, o objetivo básico foi propor um sistema que integrasse todas as fases da aquisição de dados, mantendo a conversão A/D como algo secundário que depende da aplicação. Para avaliação dos resultados, simulou-se uma aplicação com baixa taxa de transmissão. A eficiência e viabilidade do sistema foram testadas e apresentam excelentes resultados.

- **Estrutura do sistema reduzido**

Como o Modem efetua todo o procedimento de conexão e envio de dados, sem necessidade de outros equipamentos, o sistema fica bastante compacto, ganhando mobilidade e portabilidade.

- **Existência da prestadora de serviço de telefonia celular na região em estudo;**

- **Facilidade na implementação**

A colocação do sistema em operação exige apenas alguns conhecimentos na configuração de Equipamentos e em programação JAVA (J2ME);

- **Facilidade na aquisição do equipamento**

3.3 Interface de AQUISIÇÃO DE DADOS – GPRS

A Interface entre a AQUISIÇÃO DE DADOS e o módulo GPRS tem por finalidade colher as informações, processá-las enviar à Rede GPRS, ou seja, as informações são devidamente mensuradas e tratadas em forma de valores discretos (números) e enviados para a Interface GPRS-INTERNET (Modem). Existem no mercado diversas soluções que podem ser utilizadas para esta finalidade. Algumas delas podem ser destacadas:

- **Interface GPS -Sistema de Posicionamento Global**

A Interface GPS [35] monitora a localização de pessoas ou veículo por satélite através do Método de Triangulação [36]. Uma vez obtidos estes valores, os mesmos são processados e enviados através da porta serial para o Modem.

- **Interface de Leitura de Código de Barras**

Esta interface permite efetuar a leitura de Código de Barras [37] e o envio dos dados para uma porta serial. Praticamente, qualquer leitor de Código de Barras que possua este tipo de interface com porta serial permite realizar este processo.

- **Interface com microcontrolador**

As soluções existentes no mercado normalmente são direcionadas para atender uma determinada finalidade específica. Quando se deseja mensurar uma informação, como por exemplo, um nível de sinal elétrico proveniente de um sensor, é necessário que haja um sistema para fazer a conversão do valor Analógico para Digital (Conversão A/D) e transmitir o valor obtido para a interface GPRS-Internet em intervalos de tempos pré-determinado, de acordo com as necessidades do projeto. Existem várias formas de implementá-la, sendo que uma delas seria utilizar um microcomputador para desempenhar todas ou partes desta função. Porém, o uso da mesma tornaria o sistema não portátil, inviabilizando assim um dos objetivos do projeto.

Uma outra alternativa seria o uso de um microcontrolador que agregasse todas estas funções, o que tornaria o equipamento bastante compacto também. Além disso, o uso de um microcontrolador torna o sistema bastante versátil para ser utilizado em outras aplicações semelhantes, bastando apenas algumas alterações na configuração que pode ser efetuado via *software* (programação) ou mesmo no *hardware* (componentes físicos do projeto) quando assim se fizer necessário.

A tecnologia dos microcontroladores surgiu no início de 1988, quando os primeiros equipamentos dotados de CPU e comunicação serial foram comercializados. Sua evolução se tornou exponencialmente crescente de tal forma que hoje existe no mercado uma vasta gama de modelos e de diferentes fabricantes que utilizam diversas tecnologias, tanto na confecção como também na sua programação [13].

Pode-se facilmente constatar que os equipamentos modernos que são utilizados para atuação e controle estão cada vez mais eficientes, possuindo até processamento que permitem atuar de forma automática e independente, ou seja, recebem e armazenam instruções pré-programadas e as executam sem a necessidade de ingerência manual durante os processos. Estas instruções podem ser alteradas ou re-configuradas de acordo com as necessidades da aplicação, o que permite uma grande versatilidade na implementação de um projeto. Somado a isto, tem-se toda uma estrutura de um microcomputador em um único componente (*chip*) que permite executar tarefas específicas pré-programadas. Desta forma, os microcontroladores podem atuar em um número infindável de aplicações de processos autômatos.

Existem muitos tipos no mercado, como o 8051 da INTEL [22], o AVR® da ATMEL [38] e o COP8 da NATIONAL [39], sendo os da família PIC (Periférico Integrado de Controle) do fabricante Microchip© atualmente um dos mais populares, com publicações para pesquisa encontradas com maior facilidade o que foi um dos agentes motivadores para utilização deste projeto.

Os micro-controladores da Microchip© são todos baseados na tecnologia RISC (*Reduced Instruction Set Computing*), projetados para aplicações que exigem alto desempenho e baixo custo. A Microchip© possui mais de 140 tipos de microcontroladores, com uma enorme variedade de configurações de memórias e variedade de periféricos internos.

A escolha de um microcontrolador para uma aplicação determinada e específica depende das características do projeto. Neste trabalho os requisitos básicos são: possuir memória *flash*, conversor A/D (Analógico/Digital) e comunicação Serial RS-232. Como base nestes requisitos decidiu-se usar o PIC16f877A que preencheu estes requisitos básicos.

3.4 Arquitetura final do sistema

A figura 3.1 apresenta, em diagrama de blocos, a arquitetura básica do sistema proposto, dividido em três subsistemas básico que são: aquisição e controle de dados

(APLICAÇÃO), comunicação via GPRS (MODEM GPRS) e armazenamento e monitoramento via WEB (SERVIDOR e CLIENTE).

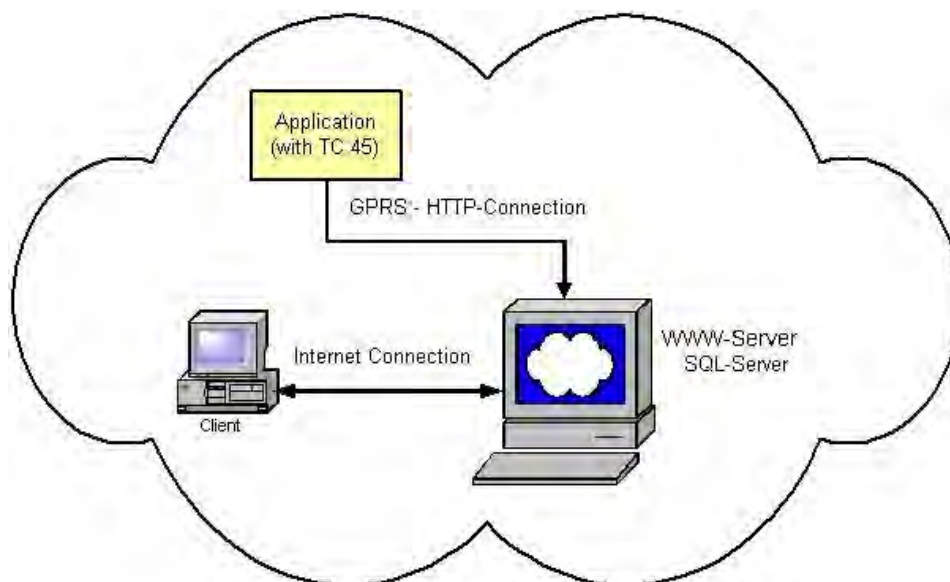


Figura 3.1 – Diagrama de blocos do sistema

3.4.1 Sub-Sistema 1: Aquisição e controle dos dados

Este subsistema é responsável pela amostragem do sinal a ser monitorado, aquisição dos dados e comunicação com o modem GPRS. Seu núcleo básico é o microcontrolador PIC16F877A da Microchip®, que incorpora características consideradas importantes para um sistema desse porte, tais como: conversão A/D de 10 bits, taxa de amostragem de até 50 kHz e comunicação serial padrão USART (*Universal Synchronous Asynchronous Receiver Transmitter*). O microcontrolador é necessário para estabelecer a comunicação entre a aplicação propriamente dita, que é basicamente a conversão A/D, e o modem GPRS. Porém, seu uso permite a elaboração de procedimentos sofisticados de aquisição de dados, tais como o monitoramento de vários pontos usando uma multiplexação adequada.

3.4.2 Sub-Sistema 2: Comunicação via GPRS

Para a transmitir dados para a Internet usando uma comunicação via GPRS é utilizado o modem TC45 JAVA desenvolvido pela Siemens (Siemens; Duodigit). A opção pelo TC45 foi baseada no uso da tecnologia GSM e na disponibilidade comercial do produto. O TC45 incorpora um sistema que permite sua programação em linguagem Java, uma configuração flexível dos parâmetros de operação e o gerenciamento de toda a tarefa de recepção e transmissão de informação via porta serial e protocolo HTTP (*Hyper Text Transfer Protocol*). Dentro da configuração adotada neste trabalho, o formato para a comunicação via internet é:

`http://endereço_site/get.php?IMEI=351277001542918&value=1,52`

onde:

- **endereço_site** é o endereço ou nome do *site* na internet onde se quer enviar os dados.
- **get.php** é a página em *php* que efetua a captura dos dados via conexão *http* e grava a informação no banco de dados.
- **IMEI** (*International Mobile Equipment Identity*) é a identificação do aparelho celular que está enviando os dados. No caso, o modem TC-45.
-

3.4.3 Sub-Sistema 3: Armazenamento e Monitoramento dos Dados via WEB

As informações recebidas na porta do Servidor de WEB são armazenadas em um banco de dados. A definição do banco de dados não é uma tarefa das mais difíceis, mas é importante que se pense na flexibilidade e facilidade. Nesse sentido, o banco de dados usado neste trabalho foi o MySQL, que possui as características desejadas e, além disso, é de uso aberto. Esta plataforma de banco de dados é hoje uma das mais populares na utilização em serviços de Internet, pois possui integração total com o sistema operacional linux, que também é de uso aberto, e o custo para sua implementação é praticamente nulo, sem considerar que ela não perde em nada em termos de qualidade técnica com outros sistemas comerciais. No banco de dados, ao se receber um determinado dado do sistema, armazenam-se, além do próprio dado, a data e hora da aquisição. Através de um aplicativo

simples, pode-se disponibilizar blocos de dados através de tabelas e gráficos. A visualização e/ou acesso por um determinado usuário é feita disponibilizando-se uma página WEB no servidor. Dependendo da taxa de aquisição de dados (ou taxa de amostragem) e da velocidade de conexão na internet, é possível obter um acompanhamento dos dados praticamente em tempo real. A plataforma do Sistema Operacional do servidor pode ser *Windows* ou *Linux*. Na prática, este servidor deverá ficar o tempo todo disponibilizado e conectado à internet.

IMPLEMENTAÇÃO DO SISTEMA PROPOSTO

A definição da arquitetura do sistema proposto foi discutida detalhadamente no capítulo anterior. Neste capítulo são apresentados os detalhes de implementação, com destaque para os vários subsistemas necessários para a viabilidade do projeto. O primeiro módulo a ser definido é o modem. Como discutido, o sistema GPRS/GSM foi a melhor opção para o projeto e sua viabilidade neste trabalho se deu através do modem TC 45 [40]. O segundo módulo é destinado a gerar o dado e enviá-lo ao modem. Esse módulo foi desenvolvido com base em um microcontrolador da família PIC [41], sendo que a geração dos dados foi feita a partir de um sensor de temperatura. Por último, foi implementado o módulo de visualização dos dados monitorados em uma página WEB.

4.1 O Modem GSM TC45

O Modem GSM TC45 da DuoDigit ([34] e [55]) (também denominado Terminal TC45 JAVA) é uma interface GSM/GPRS compacta que apresenta as seguintes características básicas: baixo consumo; incorpora um JVM (*Java Virtual Machine*); possui 300 Kbytes de memória Flash e 100 Kbytes de Memória RAM; é alimentado por 12 VDC; possui duas saídas seriais (ASC1 e ASC0); possui um conector de antena GSM; possui conectores para cartão SIM. Devido a sua facilidade de programação em JAVA J2ME, que é uma versão compacta do JAVA feita especialmente para sistemas embarcados, este modem substitui um PC. Internamente, tem incorporado o Módulo TC45 da Siemens [42] e utiliza todas as ferramentas e ambientes de programação desse módulo. Trabalha com banda dual GSM (Dual Band) de 900/1800MHz e GPRS classe 8.

O TC 45 permite a aquisição de dados via porta serial e o envio destes através do Sistema GSM/GPRS, com integração à Internet via protocolo TCP/IP. É utilizado em algumas aplicações comerciais como: Telemetria, Pagamento Eletrônico e Telemedicina. A

figuras 4.0 mostra o módulo externamente e a figura 4.1 são mostrados os conectores de interfaceamento como o Conector da Antena, DB15 e da Fonte de Alimentação, bem como os LED's da Fonte de Alimentação e Status e ainda a chave Liga/Desliga. Na figura 4.2 podem ser observados alguns detalhes internos como a localização do Módulo da Siemens TC45 e o SIM Card com o seu devido conector de encaixe.



Figura 4.0 – Terminal Java TC45 observado externamente

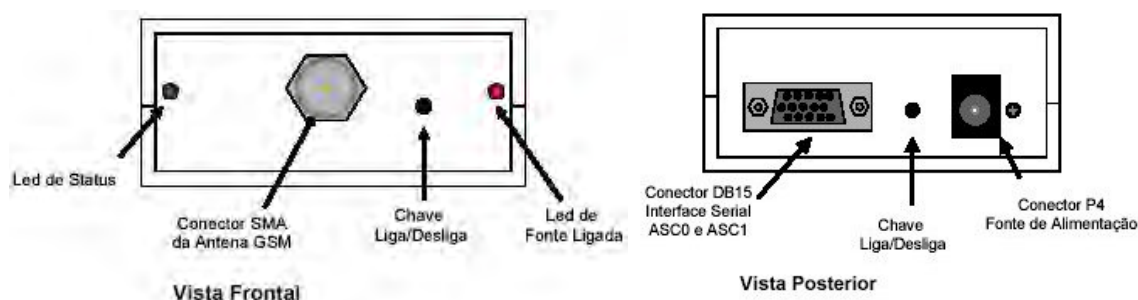


Figura 4.1- Vista Frontal e Posterior do Terminal TC45-JAVA

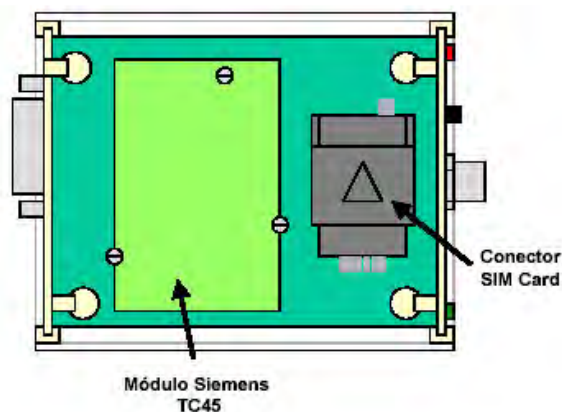


Figura 4.2- Posição do Conector Interno do Terminal TC45 JAVA

4.1.1 Modos de Operação

Na tabela 4.0 são apresentados os diferentes modos de operação do terminal TC 45, acompanhados de uma descrição sucinta.

Tabela 4.0 – Modos de Operação do Terminal TC45 JAVA

Modo	Função	
<i>Power Down</i>	O Terminal está desligado e o software do TC45 não está ativo. A tensão de entrada está conectada. Somente o regulador de tensão está ligado fornecendo energia ao RTC. A interface RS232 não está disponível.	
Operação Normal	<i>GSM/GPRS Sleep</i>	Permite a configuração de diferentes modos de economia de energia através de comandos do tipo AT+CFUN. O Terminal está com o mínimo de atividade.
	<i>GSM Idle</i> (Inativo)	O Software está ativo, registrado em uma rede GSM. O Terminal está pronto para enviar e receber dados.
	<i>GSM Talk</i>	Conexão entre dois assinantes ativa. O consumo de energia depende da área de cobertura da rede, ajustes individuais, como DTX <i>On/Off</i> , FR/EFR/HR, ganho da antena, etc.
	<i>GPRS Idle</i>	O Terminal está pronto para transferir dados em GPRS, mas nenhum dado está sendo transmitido ou recebido. O consumo de energia depende dos ajustes de rede e da configuração do GPRS (ex. Ajustes de <i>Multslots</i>).
	<i>GPRS Data</i>	Uma transferência de dados em GPRS está em progresso. O consumo de energia depende dos ajustes de rede de configuração do GPRS (ex. Ajustes de <i>Multslots</i>).
Modo Alarme	Operações restritas às execuções das funções de alerta do RTC quando o terminal está no modo <i>Power Down</i> . O Terminal não poderá ser registrado na rede GSM. Um número limitado de comandos AT's está disponível.	

4.1.2 Controle Liga/Desliga

Quando a tensão de alimentação é aplicada por um período mínimo de 1 segundo o Terminal JAVA é ligado e um processo de *Power On* é ativado. Uma vez inicializado, se a fonte de alimentação falhar por mais de 1ms o Terminal irá desligar. Se essa falha se prolongar por mais 6ms (num total de 7ms), o RTC (*Real Time Clock*) será reinicializado.

Para desligar o Terminal, deve-se utilizar o comando **AT^SMSO**, antes de desligar a fonte de alimentação, para garantir que todos os dados sejam devidamente armazenados.

Quando o Terminal é desligado via comando **AT^SMSO**, todas as linhas da interface serial RS-232 (ASC0 e ASC1) podem ficar ativadas por um período que varia de 50 ms a 3,5 segundos, o que pode gerar um envio de caracteres inválidos através das portas ASC0 e ASC1.

Uma vez desligado via comando **AT^SMSO**, o terminal pode novamente ser ligado acionando-se a chave de liga/desliga ou ativando-se a linha DTR (*Data Terminal Ready*) de 0 para 1 (-15V/-5V para +5V/+5V), com uma duração de 2 ms ou mais. Com o terminal desligado, mas com uma fonte conectada, pode-se também religá-lo usando-se a interrupção do RTC.

No painel do Terminal JAVA existe uma chave tipo *Push Button*, que tem a função de Ligar e Desligar o Terminal JAVA (função *Emergency Power OFF*) que é aplicável somente se a comunicação falhar, ou seja, quando os comandos AT's na serial ASC0 são rejeitados.

Cada vez que o módulo é desligado, os dados são gravados na memória *Flash*. O número máximo garantido de ciclos Liga/Desliga é de 100.000 ciclos.

O Terminal JAVA pode ser ligado e desligado também através de uma chave externa, conforme o circuito da Figura 4.3:

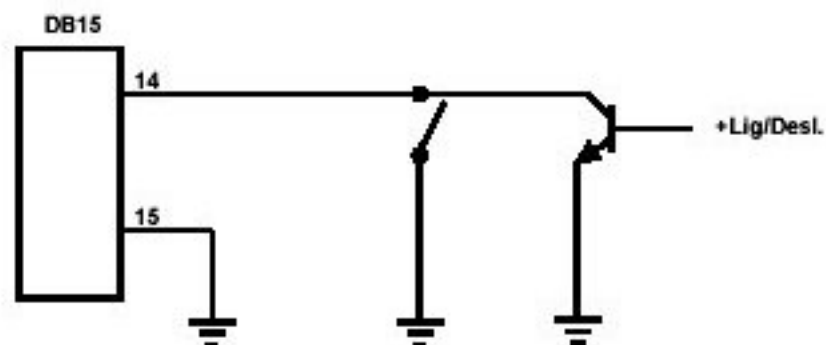


Figura 4.3- Modos de implementação de chave Liga/Desliga externa

4.1.3 Utilização do RTC para Ligar o Modo Alarme

Uma maneira de ligar o Terminal é utilizar a interrupção do RTC, que produz uma função de alerta que irá desligar o Terminal. Para prevenir que o terminal se registre em

rede GSM, esse procedimento é restrito a um número pequeno de comandos, em um conjunto denominado de Modo Alarme. O tempo do alarme é ajustado através de um comando específico **AT+CALA**. O RTC mantém o tempo do alarme se o Terminal for desligado pelo comando **AT^SMSO**. Uma vez que o alarme foi temporizado e executado, o Terminal entra no modo Alarme que é indicada pelo URC:

No Modo Alarme (**^SYSSTART ALARM MODE**), o Terminal executa um pequeno número de comandos AT's, como os apresentados na Tabela 4.1.

Tabela 4.1- Comandos AT's executado no Modo Alarme

Comando AT	Uso
AT+CALA	Ajusta o tempo de alarme
AT+CCLK	Ajusta data e hora do RTC
AT+CFUN=0,1	Muda para Modo de Operação Normal
AT^SBC	Verifica a tensão interna
AT^SCTM	Verifica temperatura, habilita/desabilita URC's para reportar faixas de temperaturas críticas.
AT^SMSO	Desliga o Terminal

Para assegurar a perfeita operação do Terminal nas mais variadas condições de temperatura e tensão, são efetuadas desligamentos automáticos, equivalentes ao comando de *Power-Down* **AT^SMSO**, exceto para o caso de sobre-tensão no módulo TC45 interno. As URC's são enviadas como mensagens de alerta e podem ser ativadas/desativadas através de comandos AT's.

Para se ter economia de energia é utilizado o comando *sleep*, que reduz as funcionalidades do Terminal ao mínimo. A definição da funcionalidade é feita através do comando AT+CFUN, onde <fun>= 0, 1, 5, 6, 7 ou 8. Para o nível de funcionalidade <fun>= 1, ativa-se apenas de economia "default", que é o modo ativado quando o terminal é ligado.

4.1.4 Ativando o Terminal (*wake up*)

Qualquer evento que desliga o Modo Sleep coloca o Terminal no Modo de operação normal como se estivesse configurado para AT+CFUN=1. Na Tabela 4.2, onde são

apresentados os estados de transição , pode-se perceber que o ‘SIM’ significa que o Terminal sai do modo Sleep e o ‘NÃO’ significa que o Terminal não sai do Modo Sleep.

Tabela 4.2 - Descrição dos estados de transição

Evento	Do Modo Sleep AT+CFUN=0 para AT+CFUN=1	Do Modo Sleep AT+CFUN=5 ou 6 para AT+CFUN=1	Do Modo Sleep AT+CFUN=7 ou 8 para AT+CFUN=1
Power On	Sem efeito	Sem efeito	Sem efeito
/RTS0 ou /RTS1 (borda de descida)	Sim	Sem efeito	Sem efeito
URC	SIM	SIM	NÃO
Qualquer comando AT (incluindo saída de voz,dados ou SMS)	No JAVA: SIM Fora do JAVA: NÃO	NÃO	NÃO
Recepção de SMS AT+CMNI=0,0 (Padrão)	NÃO	NÃO	NÃO
AT+CMNI=1,1 (Mostra URC)	SIM	SIM	NÃO
Transferência de dados GPRS	No JAVA: SIM Fora do JAVA: NÃO	NÃO	NÃO
Fim do programa JAVA	SIM	NÃO	NÃO
Alarme RTC	SIM	SIM	NÃO
AT+CFUN=1	No JAVA: SIM Fora do JAVA: NÃO	SIM	SIM

4.1.5 Interface SIM (*Subscriber Identification Module*)

A interface SIM foi desenvolvida para cartões SIM de 3V que seguem a norma GSM 11.12 Phase 2. Para a sua retirada ou colocação, o Terminal deve estar com a fonte de alimentação desconectada. Como ele fica situado na parte interna do Terminal, é necessário abrir a parte superior do Terminal desparafusando os 4 parafusos Philips.

4.1.6 Leds de monitoramento

São encontrados no painel de vista frontal, como apresentados na figura 4.2. Os dois Leds indicativos são:

- Led de Fonte Ligada: é o led vermelho do Terminal que, quando ligado, indica que o Terminal está com a Fonte de Alimentação ligada.
- Led de Status: é o led verde do Terminal, que é ativado de acordo com a opção do comando **AT+SSYNC=1**, apresentada na Tabela 4.3.

Tabela 4.3 - Indicação do Led de Status de Operação

Modo do Led	Status da Operação
Desligado	Terminal desligado ou em Modo Sleep
600ms Ligado / 600ms Desligado	Não tem SIM Card ou o PIN não foi ativado, ou em procura de rede, ou em processo de login na rede.
75ms Ligado / 3s Desligado	Registrado na Rede (monitorando os canais de controle e interações do usuário). Não tem chamada em processo.
75ms Ligado / 75ms Desligado / 75ms Ligado / 3s Desligado	Um ou mais “ <i>GPRS contexts</i> ” estão ativados
Piscando	Indica transferência de dados em GPRS. Quando a transferência está em processo, o Led fica aceso até 1 segundo após os pacotes de dados ter sido enviado/recebido.
Ligado	Depende do tipo da ligação: Chamada de Voz -> Conectado com sistema remoto Chamada de Dados -> Conectado com o sistema remoto ou durante a troca de parâmetros na conexão ou desconexão da chamada.

4.1.7 Programação do Modem TC45

O Terminal TC45 JAVA permite uma programação direta em linguagem JAVA. Internamente, são reservados 300 kbytes de memória Flash para o armazenamento de aplicativos. Para auxiliar no desenvolvimento de aplicativos existem três softwares básicos, fornecidos juntos com o equipamento, que são:

-RS232Demo, que é um software de testes do Terminal;

- TC45 *Software Development Kit*, que contém um conjunto de ferramentas para exploração do terminal e é baseado no SMTK (*Siemens Mobile Tools Kit*);
- SUNONE, que é a plataforma de desenvolvimento de aplicativos em J2ME.

4.1.8 Teste do Modem TC45 JAVA

Para o teste do Modem TC45 JAVA foi instalado um conjunto de softwares em um Microcomputador PC com Sistema Operacional WindowsXP, quais sejam:

- J2SDK - aplicativo que cria um ambiente JAVA em um microcomputador tipo PC;
- SUNONE – aplicativo para o ambiente de estações SUN [60].
- SMTK (*Siemens Mobile Tools Kit*) - conjunto de ferramentas para desenvolvimento de software de aparelhos móveis, desenvolvido pela SUN MICROSYSTEMS, que acompanha o *software TC45 SOFTWARE DEVELOPMENT KIT*.

Após a instalação destes softwares foi observado que, automaticamente é criado um driver no Windows, que é identificado no “Meu Computador” como “ *Module*” conforme a Figura 4.4

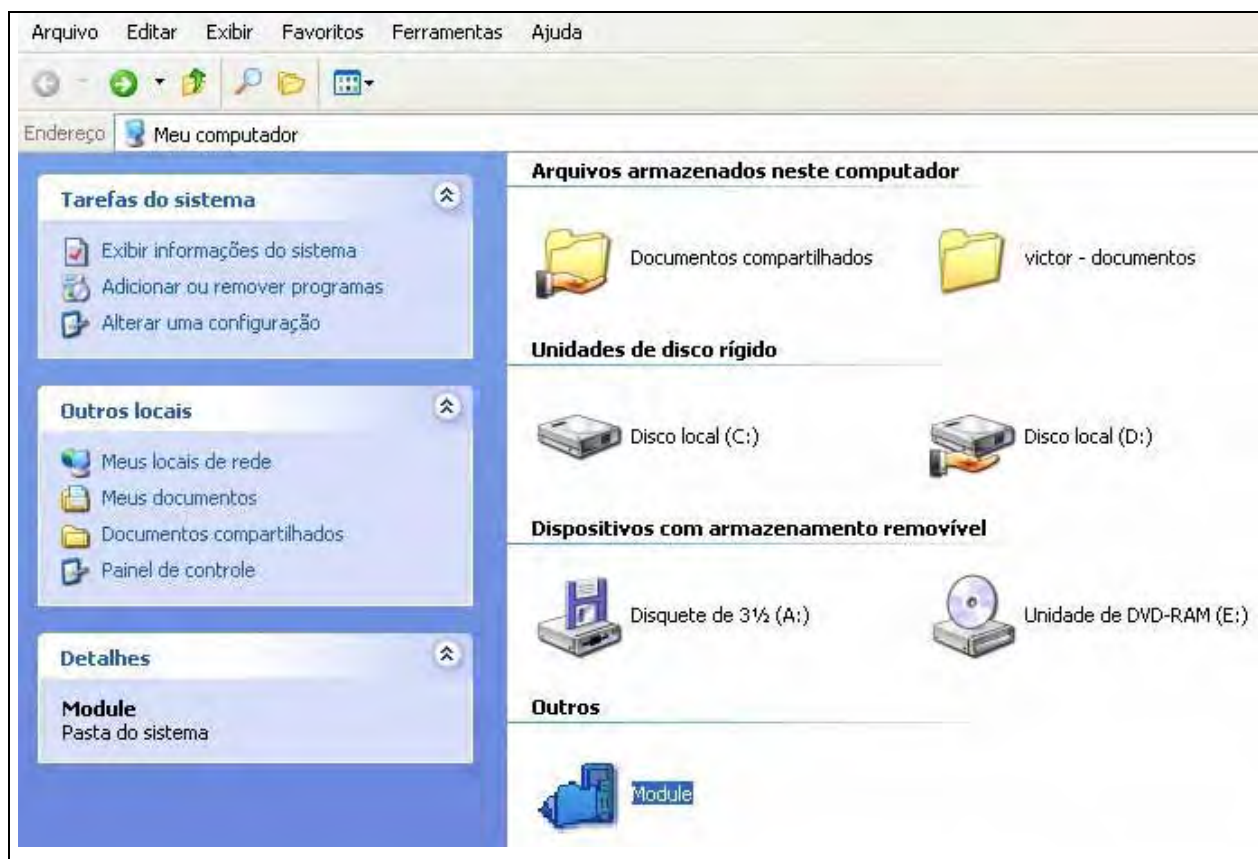


Figura 4.4 – Disponibilização de acesso do Terminal TC45 efetuado por SMTK indicado como Module

Quando o “*Module*” é ativado, observa-se que existe em seu interior 300KBytes de memória Flash indicado como “*Module Disk (A:)*”, conforme a Figura 4.5.

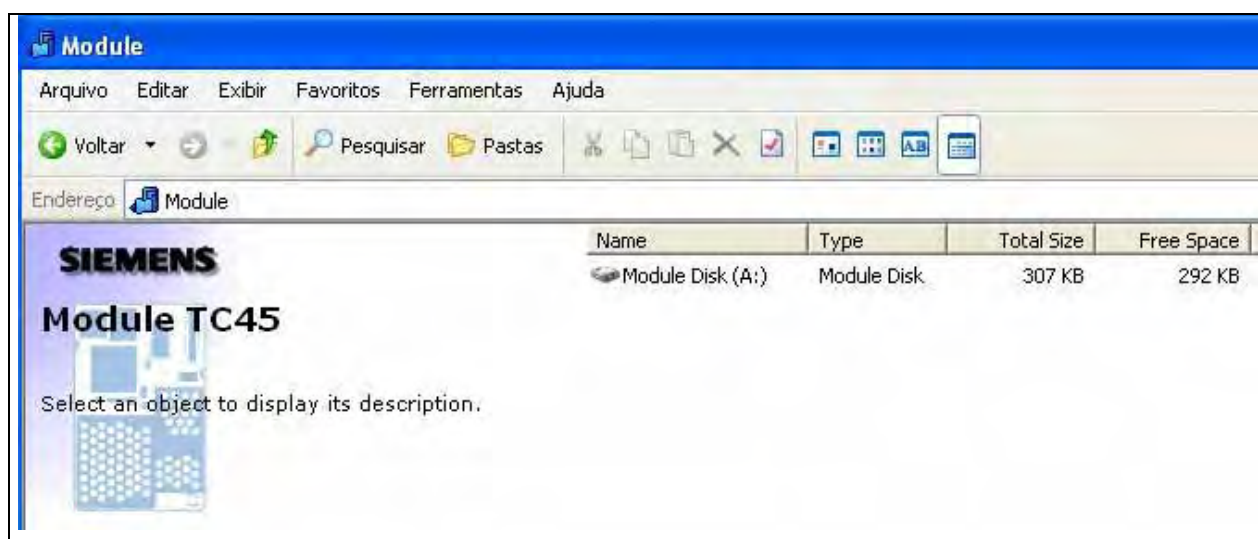


Figura 4.5– Indicação da quantidade de memória Flash do Terminal TC45 JAVA

Ainda nesta Figura 4.5, observa-se que o espaço livre efetivo é de 292Kbytes, já que foi carregado o programa o software RS232Demo, como indicado na Figura 4.6.

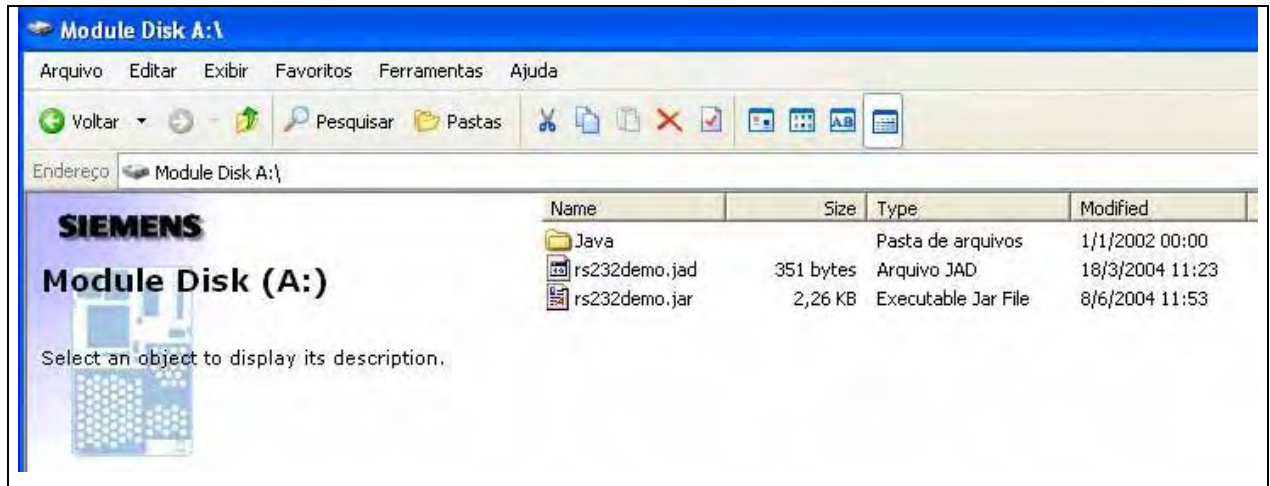


Figura 4.6 – Programa rs232demo carregado no Terminal TC45 JAVA

Na seqüência, é efetuado na tela do *HyperTerminal* o comando direcionado para ASC0, que é: **AT^SJRA=A:/rs232demo.jar**

Automaticamente o Terminal fica habilitado a receber informações na sua porta ASC1 conforme ilustra a figura 4.7.

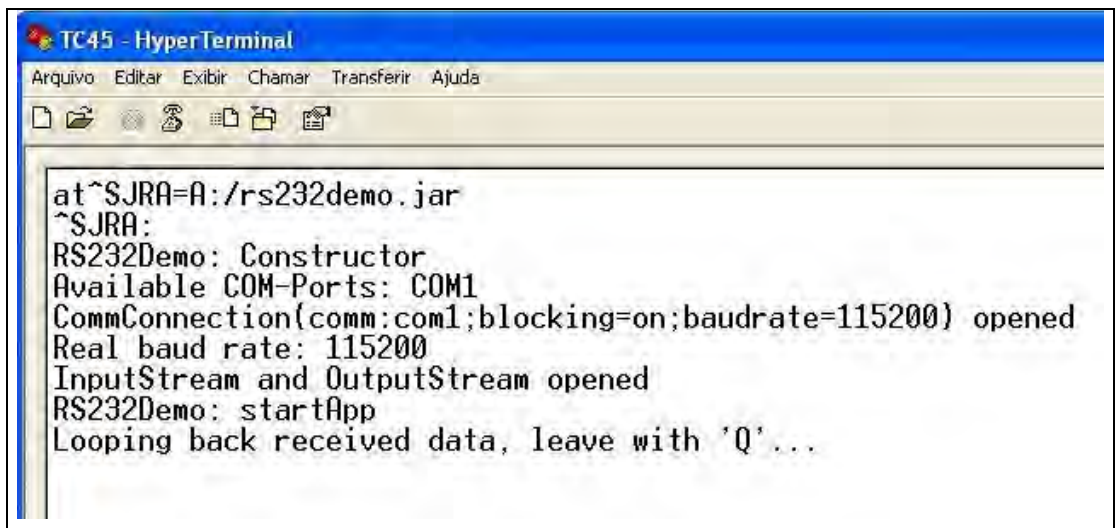


Figura 4.7 – Conexão habilitada para realizar comunicação com a porta ASC0 e ASC1

Foi conectada na ASC1 outro Microcomputador PC e enviado na sua porta serial através do aplicativo *HyperTerminal* algumas palavras. Automaticamente estas palavras aparecerem na tela do primeiro Microcomputador. Isto significou que qualquer informação enviada na ASC1 é devolvida na ASC1 e ASC0. Para serem finalizados os testes foi digitado o comando ‘Q’.

4.1.9 Implementação do Programa no Terminal TC45 JAVA para o Sistema Proposto

Para facilitar a sua configuração foi utilizado um programa demonstrativo em JAVA [40] fornecido pela *Siemens* ([42] e [58]). Este programa, denominado de **bcscan.jar** e detalhado no Anexo 2, monitora continuamente a porta serial do terminal. Quando um dado é disponibilizado na porta, o mesmo é enviado à rede de acordo com o endereço especificado (Internet). O fim da transmissão é baseado na sequência de bits 0000 1101, que representa o CR (*Carriage Return*). O dado é recebido na rede via conexão GPRS - HTTP e armazenado num banco de dados tipo MySQL. O conjunto dos dados é, por sua vez, disponibilizado através da Internet via paginação Web.

Para que o sistema funcione sem a necessidade de estar ligado a um terminal ou console, é preciso estar configurado com o que é denominado de *Auto – Start*, descrito no Anexo 3, ou seja, toda vez que o Terminal é ligado, automaticamente o programa de configuração **bcscan.jar** inicializa a sua execução, estando apto para funcionar.

A sequência de operação do programa implementado no TC 45 é apresentada no fluxograma da figura 4.8. A descrição da operação indica uma transferência de dados por demanda. Porém, pode-se fazer, sem qualquer restrição, transferência de blocos de dados. A definição de qual tipo de transferência deve ser usada dependerá da aplicação. É importante destacar que essa opção não afeta o custo operacional do sistema usando GPRS.

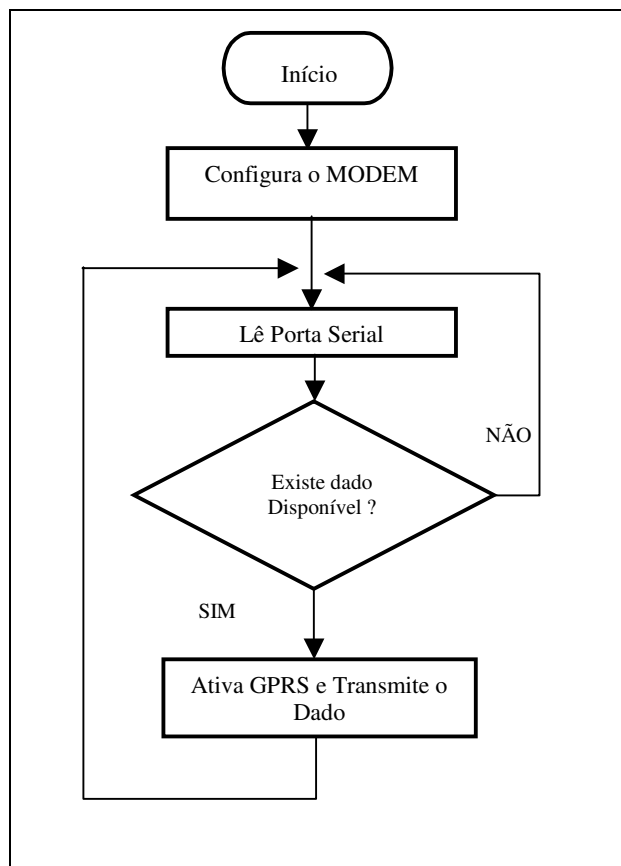


Figura 4.8 – Fluxograma de operação do TC 45

4.2 O hardware para aquisição de dados

Conforme a Arquitetura do Sistema Proposto descrito no Capítulo 3, o hardware para aquisição de dados teve como base o uso do microcontrolador PIC 16F877A da Microchip© ([33] e [54]).

Para a elaboração do projeto, bem como a confecção da placa do circuito impresso, foi utilizada o software EAGLE (*Easily Applicable Graphical , Layout Editor, Version 4.16 for Windows Light Edition*)disponibilizada para o seu uso livre pela empresa Cadsoft [68]. O diagrama elétrico do circuito é mostrado na figura 4.9.

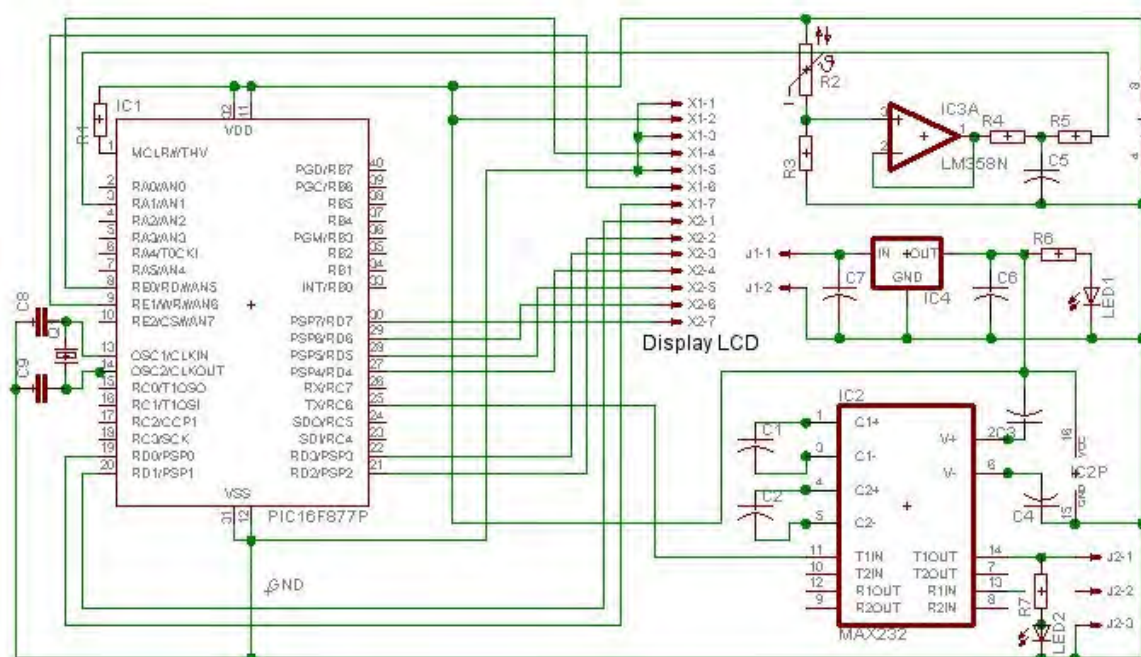


Figura 4.9 – Circuito para o monitoramento de temperatura e comunicação com o TC 45

Para o projeto, foi utilizado como sensor de temperatura o componente LM35 [44]. A tensão obtida na saída do LM35 é digitalizada pelo PIC 16F877A Conversão A/D (de 10 bits). A resolução é de uma casa decimal. O intervalo de amostragem é de 20 segundos e o valor obtido é enviado um Display de Cristal Líquido LCD, para visualização imediata, e também para uma porta serial através do MAX232 ([45], [46] e [47]).

4.2.1 Sensor de Temperatura LM35

O LM35 é um Sensor de Temperatura Linear que capta temperatura em níveis de tensão numa escala linear. Como resultado, tem-se um sistema que produz, na saída, tensões variáveis proporcionais às variações de temperatura. A calibração é feita através do potenciômetro R1. Cada grau de temperatura corresponde a 0,1Volts. Para o circuito implementado, uma temperatura ambiente de 23°C, por exemplo, equivale a uma tensão de 2,3 Volts. O circuito foi projetado para uma leitura entre 0° C e 43°C, que corresponde a uma faixa razoável de variação de temperatura ambiente das diversas regiões do Brasil. Isto

significa uma faixa de tensão entre 0 e 4,3 Volts, que é aplicada na entrada AD do Microcontrolador.

O circuito do sensor de temperatura LM35 é apresentado em detalhes na figura 4.10.

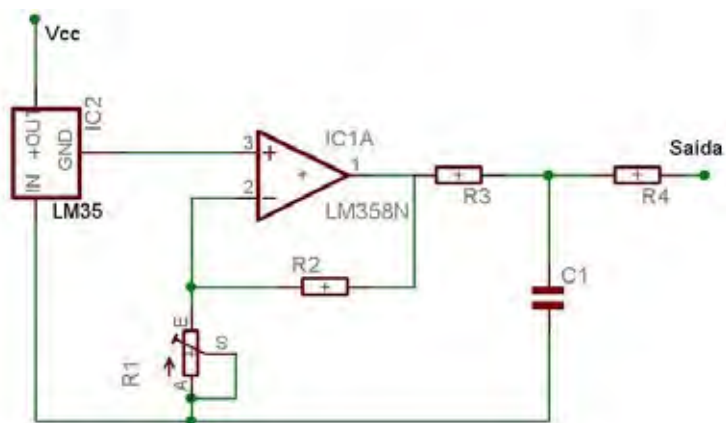


Figura 4.10 – Esquema Elétrico do Sensor de Temperatura.

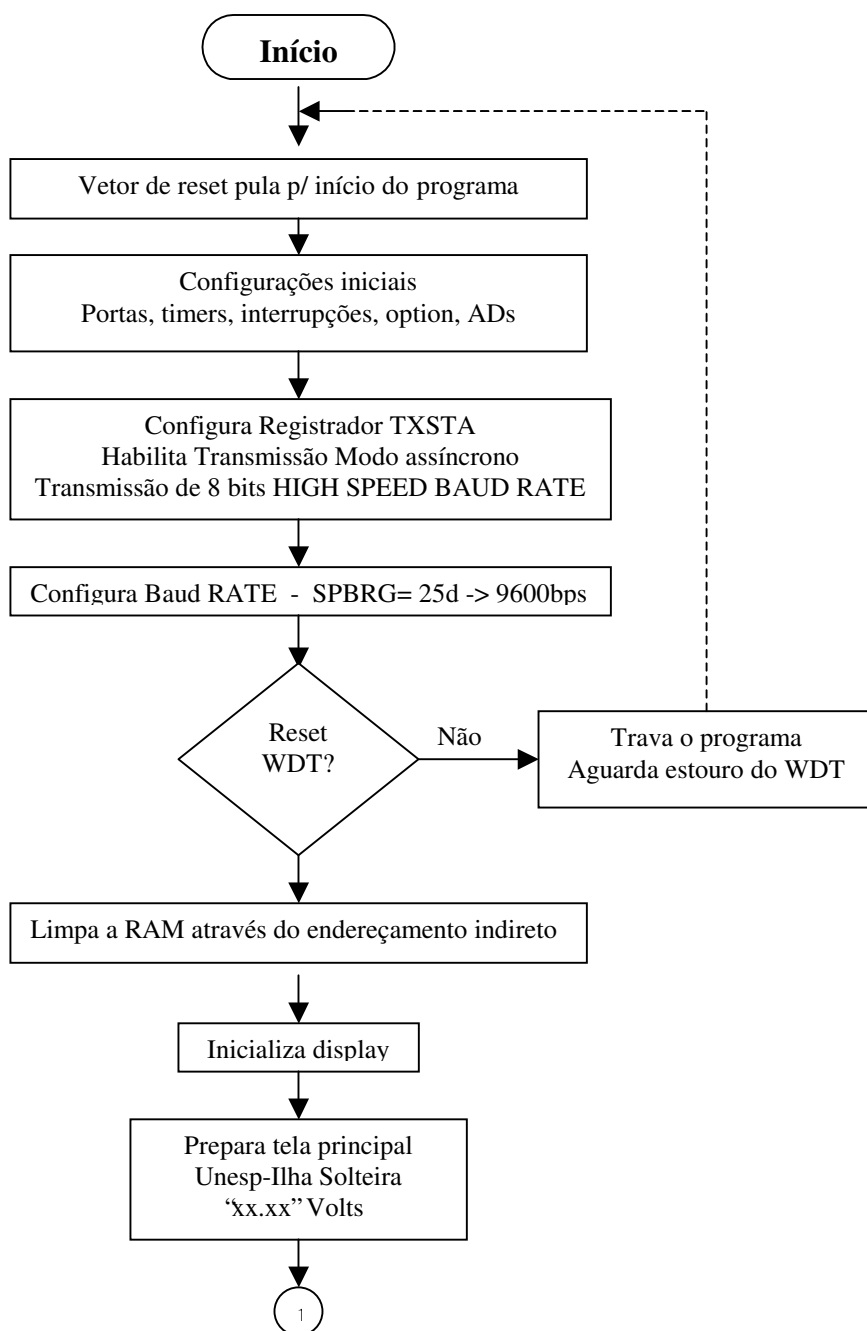
O foto do circuito completo montado em uma placa de circuito impresso é apresentado na figura 4.11.



Figura 4.11 –Circuito Elétrico do Medidor de Temperatura

4.2.2 Programação do PIC

Para o Microcontrolador PIC, foi elaborado um programa *SensorTempVS21*, em linguagem de máquina “*Assembly*”, apresentado em detalhes no Anexo 1. Os procedimentos efetuados pelo software também estão descritos no fluxograma da figura 4.12.



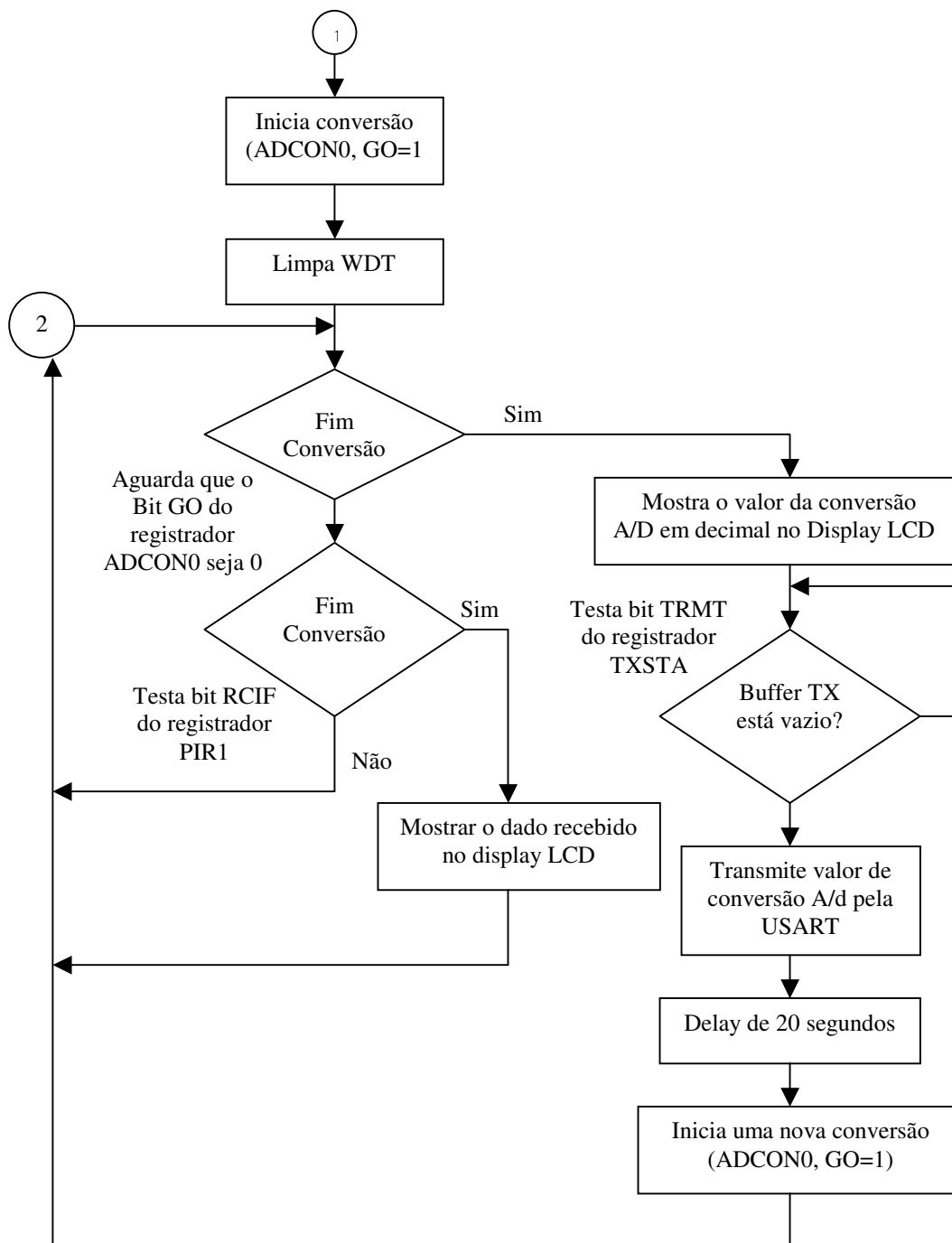


Figura 4.12 Fluxograma do Programa SensorTemp VS21

O programa necessita ser compilado previamente para ser gravado no micro-controlador. Para esta compilação, foi utilizado o programa MPASM v02.30 obtido do site do próprio fabricante [17].

Para a sua gravação, foi utilizado um gravador PIC que efetua um interfaceamento paralelo com um microcomputador PC. Este gravador foi elaborado de acordo com o artigo do Engº Márcio José Soares publicado na revista Mecatônica Fácil [32]. Este dispositivo é mostrado na foto da figura 4.13.



Figura 4.13 Gravador PIC de porta paralela

Para efetuar a gravação foi utilizado o Programa IC-PROG [49]. Uma vez gravado o programa, o PIC está pronto para ser operado no circuito.

Quando um dado é amostrado na entrada do conversor A/D do microcontrolador, o mesmo é enviado junto com um valor adicional 0x0D (código ASCII do *Carriage Return - CR*), que sinaliza o término do envio de dados. Na aplicação realizada, o processo dura, aproximadamente, 20 segundos. Porém, esse tempo pode ser facilmente alterado de acordo com a aplicação desejada.

4.3 O Monitoramento via Internet

Existem diversas tecnologias Web para o processamento de informação que realizam o recebimento, armazenamento e disponibilização de dados via Internet [55].

Para o projeto em questão, foi utilizado um sistema de paginação em PHP (*Hypertext Preprocessor*), que oferece dinamismo às páginas de visualização na Internet e permite a interação com um servidor de banco de dados. A plataforma de desenvolvimento e utilização do banco de dados foi o MySQL [52], por proporcionar soluções de praticidade e fácil integração com a plataforma Web, além de ser uma tecnologia na categoria de software livre.

O desenvolvimento do software de Monitoração via Internet foi efetuado em um microcomputador Pentium II, com *clock* 400 MHz, sobre a plataforma do sistema operacional **Windows Me da Microsoft**. O ambiente de desenvolvimento foi composto pelo Kit de Software **PHPTriad 2.2.1** [49], que instala o PHP, o Servidor de Banco de Dados MySQL e o Servidor de Páginas Web Apache [50]. Para o desenvolvimento de páginas Web foi utilizado o software **MPS PHP DESIGNER** [51] também de uso livre.

Para o servidor Web, o processo consiste no recebimento de dados do usuário, processamento e envio de uma resposta dinâmica. Uma vez enviada a resposta, é encerrado o contato entre o servidor e o cliente. Portanto, o primeiro passo é entender como fazer para receber os dados enviados por um navegador (*browser*) para o servidor (ou por outro dispositivo como o Modem TC45).

O protocolo HTTP (Protocolo de Transferência de Hiper-Texto) provê dois métodos para enviar informações para o servidor web, além da URL referente ao arquivo solicitado. Estes métodos são o “**POST**” e o “**GET**”.

O Método **POST**

O método **POST** permite o envio de informações através de um navegador (*Browser*). O Terminal TC 45 JAVA também utiliza este método para o envio da informação “IMEI” (*International Mobile Equipment Identity*), que é o número de identificador do Modem Celular TC45 JAVA, e o parâmetro “*value*”, através da seguinte linha de comando:

http://" + HOST + "?" + "IMEI=" + IMEI + "&" + "value=" + str.

Esta linha está contida no arquivo *HTTPUploadThread* (Anexo 2), que é usado em outras aplicações dentro do Terminal TC 45 JAVA

Ainda, em termos de linhas de comandos, no arquivo *HTTPUploadThread* são definidos os caminhos da rede e o seu endereço, como segue:

DEFAULT_HOST = "endereço_site/get.php"

O Método GET

É um método que permite fazer requisição através de um servidor Web conectado diretamente à porta 80 (servidor HTTP). Toda vez que chega no servidor de rede Web a informação correspondente ao valor “IMEI” e “value” que foi enviado do Modem TC 45 JAVA, ele realiza um processo de captura e os encaminha para o armazenamento em um banco de dados. O comando é executado dentro de um script PHP que está contido na página *get.php* (Anexo 4) implementado no Servidor Web:

```
if (isset($_GET["IMEI"]) && isset($_GET["value"]))
{
    $IMEI = $_GET["IMEI"];
    $value = $_GET["value"];
}
else
```

Em outras palavras, são armazenadas dentro de uma variável *\$IMEI* e *\$value* os valores correspondentes ao *IMEI* e *value* capturados na porta 80.

Informações de Data e Hora

Os dados recebidos necessitam de registros de data e hora da aquisição. Estes valores são armazenados numa variável string (variável temporária) com o formato *\$date* e

\$time capturados do relógio do servidor no instante em que as informações são armazenadas através dos comandos:

```
$date = date("d-m-Y") e $time = date("H:i:s");
```

Armazenamento no Banco de Dados MySQL

No MySQL [52] do servidor foi criado o banco de dados *nome_db* contendo a tabela *tabela_db* com os campos: *Index*, *IMEI*, *Data* e *Hora*.

O script de execução, que também está contido na página de captura *get.php*, é:

```
$link = mysql_connect('endereço_ip_db_server', 'usuario', 'senha');  
if (!$link) {  
    die('Not connected : ' . mysql_error());  
}  
  
// make nome_db the current db  
$db_selected = mysql_select_db('nome_db', $link);  
if (!$db_selected) {  
    die ('Can\'t use nome_db: ' . mysql_error());  
}  
  
$cadastra = "INSERT INTO tabela_db  
    (IMEI, value, data, hora)  
    VALUES  
    ('{$IMEI}', '{$value}', '{$date}', '{$time}');  
$resultado = mysql_query($cadastra)  
    or die("Falha na execução do cadastro");  
  
echo "Dados adicionados com sucesso";
```

Basicamente, este script abre uma conexão com o servidor de banco de dados com endereço IP nº *endereço_ip_db_server*, mediante nome do usuário (*usuário*) e senha (*senha*) e em seguida são inseridos os valores *IMEI*, *value*, *data* e *hora*.

Disponibilização de dados via Web

Os dados armazenados no banco são disponibilizados para o mundo Web através de uma página denominada *consulta.php*.(Anexo 5)

Nesta página, é feita inicialmente a conexão com o banco de dados através do *script*:

```
$link = mysql_connect("endereço_ip_db_Server", "usuario", "senha")  
or die("Não foi possível conectar");
```

Em seguida é selecionado o banco de dados *nome_db* e a tabela deste banco *tabela_db*:

```
mysql_select_db("nome_db", $link)  
or die("Não foi possível selecionar o banco de dados");
```

```
$consulta = "SELECT * FROM tabela_db ORDER BY numero DESC";  
$resultado = mysql_query($consulta)  
or die("Falha na execução da consulta");
```

Finalmente, é estruturada uma tabela com os dados e disponibilizada na página através do *script*:

```
echo "<table border=1 width=60%>";  
echo "<tr>";  
echo "<td width=10% align=center><b>Nº</b></td>";  
echo "<td width=10% align=center><b>IMEI</b></td>";  
echo "<td width=10% align=center><b>Valor</b></td>";
```

```

echo "<td width=10% align=center><b>Data</b></td>";
echo "<td width=10% align=center><b>Hora</b></td><p>";
echo "</tr>";
while ($linha = mysql_fetch_assoc($resultado))
{
    $numero = $linha["numero"];
    $IMEI = $linha["IMEI"];
    $value = $linha["value"];
    $data = $linha["data"];
    $hora = $linha["hora"];
    echo "<tr>";
    echo "<td width=10% align=center>$numero</td>";
    echo "<td width=10% align=center>$IMEI</td>";
    echo "<td width=10% align=center>$value</td>";
    echo "<td width=20% align=center>$data</td>";
    echo "<td width=20% align=center>$hora</td><p>";
    echo "</tr>";
}
echo "</table>"

```

A figura 4.14 mostra esta visualização dos dados. Os números da primeira coluna servem como indexador do vetor de dados obtido. O valor de tensão representa a conversão A/D. Para cada aplicação deverá ter-se uma configuração de visualização adequada, podendo-se adotar planilhas, tabelas, gráficos, etc., bem como fornecer mais informações sobre os sensores e os pontos monitorados. A taxa de amostragem usada foi baixa, mas o sistema permite taxas de mais de 10kHz.

UNESP-Ilha Solteira/SP

Monitoramento Remoto de Dados Via Internet usando Comunicação sem Fio
<http://www.getulio.eng.br/mestrado/consulta.php>

Resultado da Consulta

N°	IMEI	Valor	Data	Hora
345	351277001542918	2,4	24-08-2006	16:15:19
344	351277001542918	2,4	24-08-2006	16:14:59
343	351277001542918	2,4	24-08-2006	16:14:38
342	351277001542918	2,4	24-08-2006	16:14:19
341	351277001542918	2,4	24-08-2006	16:13:59
340	351277001542918	2,4	24-08-2006	16:13:38
339	351277001542918	2,4	24-08-2006	16:13:17
338	351277001542918	2,4	24-08-2006	16:12:58

Figura 4.14 - Página web de visualização de dados de monitoração de temperatura

Para um monitoramento mais efetivo, foi efetuada a inclusão do script no cabeçalho da página *consulta.php*, que propicia a atualização dinâmica no intervalo de 10 segundos, como segue:

```
<head>
<meta http-equiv="refresh" content="10;
URL=http://www.getulio.eng.br/mestrado/consulta.php">
</head>
```

Na prática, a partir do instante em que a página é carregada no navegador ou browser, o script realiza uma contagem de 10 segundos realizando então um “*refresh*” ou atualização da página “*consulta.php*”.

4.4 Aspectos técnicos e considerações do projeto

Na implementação deste projeto de Sistema de Monitoramento de Dados via Internet usando Telefonia Celular foram encontradas algumas dificuldades que foram solucionadas de acordo com as necessidades específicas da aplicação. As principais foram:

- Como o processo de aquisição de dados seria efetuado em tempo real em um intervalo de amostragem de 20 segundos, houve a necessidade de se utilizar sensor que oferecesse uma resposta razoável de medida dentro destes parâmetros e que pudesse ser aferida ou calibrada facilmente com um outro instrumento de medição. Dentro destas possibilidades, chegou-se a conclusão de que um Sensor de Temperatura seria o mais indicado, especificamente o LM35, devido à relação linear entre tensão, em Volts, e a temperatura em graus Celsius. Caso fosse utilizado qualquer outro tipo de sensor não linear, seria necessário o uso de mecanismos de correção para obtenção de dados fiéis às aquisições realizadas.
- Para a aquisição de dados e envio à Interface GPRS - Internet, optou-se por usar um microcontrolador, que deu ao projeto final algumas características importantes, como modularidade, fidelidade na aquisição de dados, simplicidade na implementação do circuito e flexibilidade ao sistema. Contudo, projetar o circuito e programar o microcontrolador foi uma tarefa árdua que não aparece no resultado final.
- A utilização do Modem GPRS-Internet TC45 demandou muito estudo e trabalho, pois houve a necessidade de se conhecer e aprender as configurações do equipamento e a plataforma de desenvolvimento e programação de linguagem JAVA, principalmente por se tratar de uma tecnologia bastante recente.
- Para o projeto tornar-se 100% funcional houve a necessidade de se registrar o nome de domínio ***getulio.eng.br*** na FAPESP - Fundação de Amparo a Pesquisa do Estado de São Paulo e utilizar o serviço de hospedagem de páginas Web e banco de dados no provedor HostMais [20]. Apesar do desenvolvimento do Software de Monitoramento de Dados ter sido efetuado com relativa facilidade, a sua implementação no servidor do Provedor foi uma tarefa muito difícil devido à falta de suporte técnico.

TESTES E RESULTADOS OBTIDOS

O sistema desenvolvido foi implementado gerando resultados satisfatórios. Como já mencionado, a aplicação básica para testes foi o monitoramento de temperatura, com a aquisição de dados em intervalos de vinte segundos e visualização via internet. O sistema apresenta desempenho que o habilita a ser usado em diferentes aplicações reais.

5.1 O monitoramento da Temperatura via Internet

Um dos requisitos básicos para operação do sistema desenvolvido é que a aplicação deve estar contida em uma área de abrangência de uma operadora de telefonia celular. Para o projeto em questão, o serviço GSM/GPRS foi contratado sob o formato pré-pago da empresa CLARO [53].

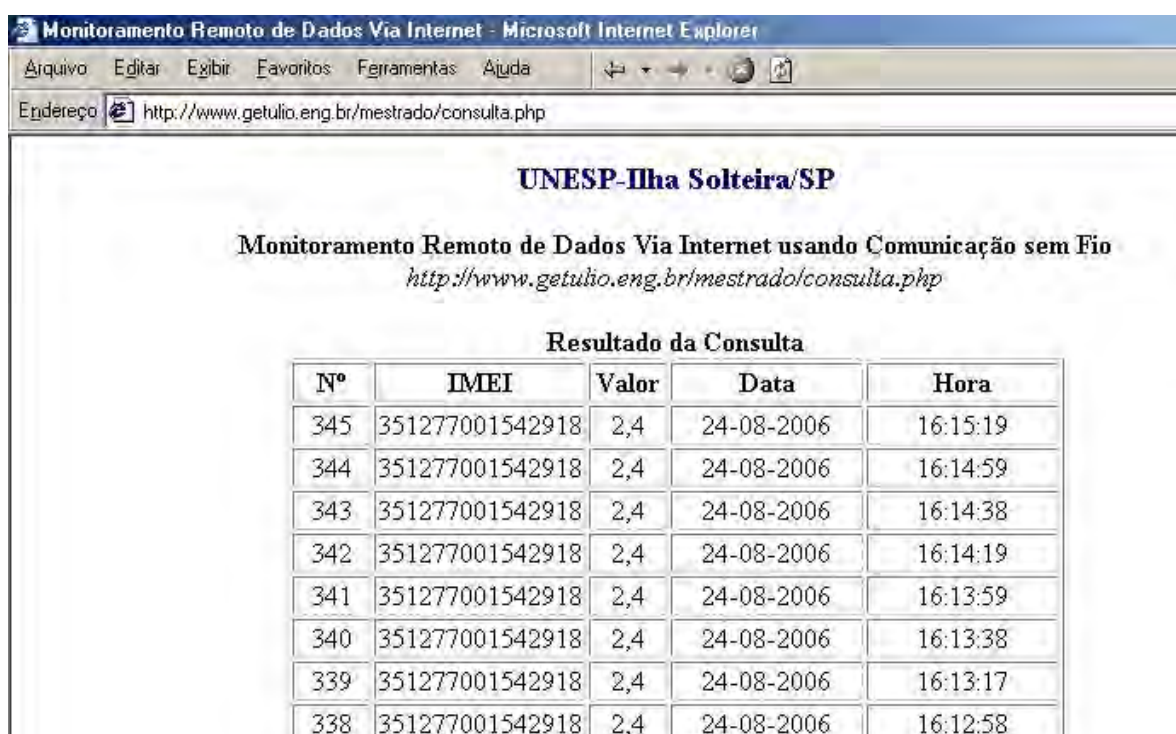
Para a realização dos testes, foi inicialmente verificada a temperatura ambiente através do sistema e de um multímetro digital com sensor de temperatura, constatando-se um valor de 23°C (2,3 volts), conforme foto apresentada na figura 5.0. Deve-se lembrar que o sistema permite uma visualização local dos dados obtidos através de um display de cristal líquido, como descrito no capítulo anterior. Na sequência, os sensores do multímetro e do sistema foram submersos em um recipiente contendo cubos de gelo. Após atingir o valor de 0° C do Multímetro e 0 Volts de tensão no sistema, os sensores foram retirados do recipiente e deixados em ambiente normal para que a mudança de temperatura pudesse ser acompanhada. Foi verificado que o valor da temperatura registrado no Multímetro Digital, que aumentou gradativamente até se chegar ao valor de temperatura ambiente de 23°C, correspondia ao valor de 2,3 Volts apresentado no Display do Sistema de Monitoramento, como esperado. A variação de temperatura foi relativamente rápida (menos de 15 segundos) e para evitar um acúmulo de dados desnecessários nos testes via internet, preferiu-se usar essa primeira etapa apenas como teste de calibração, sem envolver

qualquer tipo de comunicação de dados. Para testar os recursos de monitoramento dos dados via internet, os dois sensores ficaram expostos em ambientes que apresentavam pouca variação de temperatura, obtendo-se uma indicação correta dos dados na página WEB do sistema.



Figura 5.0 - Foto mostrando os valores Temperatura no Multímetro e Tensão do Sistema de Monitoração

Os resultados obtidos foram transmitidos e visualizados na Internet no endereço <http://www.getulio.eng.br/mestrado/consulta.php>. A figura 5.1 apresenta a disposição dos dados na página.



Monitoramento Remoto de Dados Via Internet - Microsoft Internet Explorer

Arquivo Editar Exibir Favoritos Ferramentas Ajuda

Endereço <http://www.getulio.eng.br/mestrado/consulta.php>

UNESP-Ilha Solteira/SP

Monitoramento Remoto de Dados Via Internet usando Comunicação sem Fio
<http://www.getulio.eng.br/mestrado/consulta.php>

Resultado da Consulta

Nº	IMEI	Valor	Data	Hora
345	351277001542918	2,4	24-08-2006	16:15:19
344	351277001542918	2,4	24-08-2006	16:14:59
343	351277001542918	2,4	24-08-2006	16:14:38
342	351277001542918	2,4	24-08-2006	16:14:19
341	351277001542918	2,4	24-08-2006	16:13:59
340	351277001542918	2,4	24-08-2006	16:13:38
339	351277001542918	2,4	24-08-2006	16:13:17
338	351277001542918	2,4	24-08-2006	16:12:58

Figura 5.1 - Visualização dos dados via página WEB

Opcionalmente, o resultado pode ser apresentado em forma de gráfico. Porém, dentro dos objetivos iniciais de testes do sistema, a apresentação na forma de tabela é suficiente para verificar a correta operação do sistema.

Em relação aos dados enviados à página, optou-se por utilizar uma interface WEB que se conecta em um banco de dados MySQL, armazenando a informação recebida e disponibilizando-a em tempo real. A vantagem que se tem em armazenar os dados obtidos é poder trabalhar com o histórico dos dados, que pode ser importante em diferentes aplicações.

5.2 A comunicação via modem GSM

No sistema implementado, a transferência de dados entre a aplicação e o servidor WEB é feito por demanda. Assim, sempre que um dado é obtido do sensor, faz-se uma conexão GPRS para enviá-lo ao servidor. Porém, o sistema permite que uma transferência seja realizada em blocos e essa opção depende da aplicação. É importante salientar que

existe um tempo necessário para que a conexão GPRS seja completamente estabelecida e também um tempo para que a página WEB seja completamente atualizada. Esses dois atrasos devem ser considerados em outras aplicações. Nos testes realizados, constatou-se um tempo médio de 12 segundos para uma conexão GPRS completa. Com relação à atualização da página, optou-se por fixar um tempo médio de 10 segundos entre cada atualização.

5.3 Custos do Projeto

Este projeto apresentou dois custos básicos: o de implementação e o do serviço de utilização da operadora de telefonia celular.

5.3.1 Implementação

Os custos de implementação foram divididos em dois blocos: o primeiro, foi baseado no módulo de aquisição de dados, que é constituído pelo microcontrolador, sensor de temperatura e periféricos. O custo básico foi de, aproximadamente, R\$ 100,00 (cem reais). O segundo bloco inclui o terminal TC45 Java e o a aquisição do SIM Card (*Subscriber Identity Module*). O modem adquirido teve o custo final de R\$ 857,50 (oitocentos e cinquenta e sete reais). Já o cartão SIM custou apenas R\$10,00 (dez reais).

Para que o fator portabilidade se tornasse evidente, adquiriu-se uma pequena bolsa transportável com bateria de chumbo ácido selado e com um circuito de proteção contra sub-tensão, com um custo adicional de R\$ 70,00 (setenta reais).

Computando-se todos os valores, o custo final da implementação foi de R\$ 1.037,00 (um mil e trinta e sete reais). A tabela 5.0 mostra os custos de implementação de projeto.

Tabela 5.0 - Custos de Implementação

Implementação do Projeto	Custos (R\$)
Módulo Micro-controlador e Temperatura	100,00
SIM Card	10,00
Terminal TC45 JAVA	857,50
Bolsa transportável com Bateria Chumbo-Acido	70,00
Custo Total	1.037,50

5.3.2 Serviço

Para a utilização do serviço, é necessária a inserção de créditos nas lojas de operadoras de telefonia celular. O valor é tarifado de acordo com o tráfego de bytes na rede, que atualmente custa R\$ 6,00 por Megabyte. No caso específico da aplicação, são enviados 3 Bytes de informação relativos ao valor de temperatura convertido em tensão, além de outros 15 Bytes relativos ao número do SIM Card, totalizando 18 Bytes. A idéia de se enviar o número do SIM Card é mostrar que o sistema poderia gerenciar vários pontos de aquisição com outros terminais TC 45 JAVA.

Contribui também para o custo dos serviços a manutenção do *site* onde são alocados os dados para armazenamento e consulta ao banco de dados. No caso do projeto, foi contratado um serviço de armazenamento de páginas e banco de dados ao provedor HostMais (<http://www.hostmais.com.br>) ao custo anual de R\$ 70,80. Houve também a necessidade do registro do nome de domínio *www.getulio.eng.br* junto à FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo) a um custo também anual de R\$ 30,00. A tabela 5.1 mostra os valores dos custos de serviço.

Tabela 5.1 – Custo de Serviços

Serviços	Custos (R\$)
Hospedagem do Site - Anual	70,80
Registro do Nome de Domínio - Anual	30,00
Envio de 56 Medidas – (Com o n° do PIN)	6,00
Envio de 341 Medidas – (Sem o n° do PIN)	6,00

5.4 O Sistema Obtido

O sistema obtido foi elaborado no sentido de ficar bastante compacto e de fácil transporte podendo ser acondicionado em uma bolsa de nylon portátil de dimensões 22 x 15 x 12 cm conforme mostra a foto da figura 5.2. Os resultados obtidos foram satisfatórios e muitas aplicações podem ser desenvolvidas com o sistema.



Figura 5.2 Foto do Sistema de Telemetria montado e acondicionado em uma mochila para transporte

COMENTÁRIOS FINAIS

Neste trabalho foi apresentado um sistema de aquisição e monitoramento de dados usando o que há de mais moderno em termos de tecnologia. O resultado foi um sistema que pode ser usado em diferentes áreas do conhecimento onde mobilidade e portabilidade sejam essenciais. O uso da tecnologia GPRS/Internet se mostrou mais eficaz e o uso de um microcontrolador para a interface com a aplicação tornaram o sistema genérico e altamente flexível. Existem, porém, algumas limitações que podem ser gerenciadas de acordo com a aplicação. Uma das principais é o tempo de conexão GPRS na média em torno de 8 segundos. Porém, é fácil alterar o sistema para que o módulo de aquisição de dados possa armazenar blocos de dados por um determinado período e enviá-los posteriormente ao servidor.

A aplicação desenvolvida e usada para testar o sistema é apenas um exemplo do potencial do projeto proposto. Assim, com a flexibilidade e eficiência verificadas neste trabalho, várias outras aplicações são possíveis, dentre as quais podem ser citadas, como exemplos:

- **Monitoração de Consumo de Energia Elétrica**

Para os consumidores de energia elétrica, notadamente para clientes específicos que desejam ter um controle mais rígido de consumo, o sistema pode ser usado para monitorar o consumo e detectar os períodos críticos, servindo de base para um melhor gerenciamento e, conseqüentemente, redução de consumo e de custos.

- **Monitoração de Pacientes**

Os pacientes que necessitam de cuidados especiais, que precisam ter sob controle diversas funções do organismo como, por exemplo, pressão arterial, nível de glicemia,

batimento cardíaco, etc., podem ser monitorados diretamente de suas residências, o que significa mais conforto e segurança.

- **Monitoração de Veículos**

O monitoramento de veículos via satélite é algo consolidado. Porém, a tecnologia sem fio, dependendo da área de cobertura, pode ser explorada com eficiência e reduzir significativamente os custos do serviço.

- **Geo-Processamento**

Para o mapeamento de recursos florestais, ambientais e agrícolas com precisão, rapidez e baixo custo que são efetuados pelo Geo-Processamento, sistemas simples baseados neste projeto podem ser implementados e gerar resultados eficientes.

Finalizando, este trabalho, além de gerar conhecimentos teóricos e práticos, abre um leque de possibilidade de aplicações em diferentes áreas do conhecimento. A principal contribuição se refere à aquisição de dados via telefonia celular, que é uma proposta de aplicação simples e de baixo custo, independentemente da aplicação. O sistema foi testado em um exemplo simples de tomada de medida de temperatura em um ambiente aberto, mas pode ser explorado em processos complexos de telemetria onde haja necessidade de redes *wireless* para interligar diversos dispositivos eletrônicos dispostos em ampla área geográfica, como por exemplo sensores de segurança, medidores de gás, energia e ainda equipamentos de monitoração médica. Os equipamentos comerciais similares encontrados no mercado são de tecnologia fechada e de uso restrito à determinadas aplicações, o que impossibilite seu uso em outras aplicações, ao contrário do projeto proposto neste trabalho.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] TOMPKINS, W.J. and WEBSTER, J. G. *Interfacing Sensors to the IBM PC*, Ed. Prentice Hall, USA, 1988.
- [2] NASCIMENTO, R.L. e JACCOUD, C.F.B. “Sistema de Conversão A/D e D/A linear de 12 bits para Sinais de Voz e Áudio”, Anais do 6^o. Simpósio Brasileiro de Telecomunicações, Campina Grande, PB, 1988.
- [3] VIEIRA FILHO, J. *Sistema de Aquisição de Dados baseado em Microprocessador para Sinais na Faixa de Frequência de 0 a 50 kHz*, Dissertação de Mestrado, UNICAMP, fevereiro/1991.
- [4] NIEVA, T.M. and WEGMANN, A. “A conceptual Model for Remote Data Acquisition Systems”, *Computers in Industry*, vol. 47, No. 2, February, 2002.
- [5] CHOI, J. et al. “An Intelligent Remote Monitoring System for Artificial Heart”, *IEEE Transactions on Information Technology in Biomedicine*, Vol.9, No.4, pp.564-573, December, 2005.
- [6] HAITAO, J. and CAO, L. “A remote data acquisition system based on SMs”, *IEEE Int. Conference on Systems, Man and Cybernetics*, pp.6155-6159, 2004.
- [7] WANG, N. et al. “Wireless sensors in agriculture and food industry – Recent development and future perspective, *Computers and electronics in agriculture*”, Elsevier, Vol. 50 Issue 1, pp.01-14, 2006.
- [8] “Conectividade Móvel”. Disponível em
< http://www.mobilezone.com.br/conect_wwan.htm > Acesso em 23.04.2007.
- [9] LOKER, D.R. et al. “Remote Data Acquisition using Bluetooth”, *Proceedings of the American Soc. For Eng. Education (ASEE)*, 2005.
- [10] SUBRAMANIAN, C. S. et al. “A wireless Multipoint Pressure Sensing System: Design and Operation”, *IEEE Sensors Journal*, Vol.5, No.5, pp.1066-1074, 2005.
- [11] TALLEDO, J. P. and MIRANDA, J.C. “Design of a Multi-Point Wireless Remote Data Acquisition System for Mobile Application, *Proceedings of the IEEE*” (CONIELECOMP), 2005.
- [12] ZAVALIK, C., LACERDA, G. e OLIVEIRA, J. P. M., “Implementando Web Services com Software Livre”. Disponível em
< <http://www.inf.ufrgs.br/%7Epalazzo/OAI/04%20Software%20Livre%20-%20Web%20Serv.pdf> > Acesso em 17.Fev.2007.

- [13] ZELENOVSKY, R. e MENDONÇA, A. “Arquitetura de Microcontroladores Modernos”.Disponível em
< http://www.mzeditora.com.br/artigos/mic_modernos.htm >Acesso em 17.Fev.2007.
- [14] Software Eagle da CadSof. Disponível em < <http://www.cadsoft.de> > Acesso em 21.04.2007
- [15] TATEOKI, G.T. e VIEIRA FILHO, J. “Monitoramento Remoto de Dados Usando Comunicação Sem Fio”, Congresso Brasileiro de Automática, CBA2006, 2006.
- [16] STEINN, P. “BlueTooth”. Disponível em
< http://www.gta.ufrj.br/seminarios/semin2003_1/stein/index.html > Acesso em 07.Dez.2006.
- [17] Microchip©, Inc., Disponível em < <http://www.microchip.com> > Acesso em 12.Nov.2006.
- [18] SVERZUT, J. U. “Redes GSM, GPRS, EDGE e UMTS – Evolução a Caminho da Terceira Geração”, Editora Érica, 2007.
- [19] “CDMA”.Disponível em < <http://www.planetacelular.com.br/cdma.htm> > Acesso em 17.Fev.2007.
- [20] “GSM World from the GSM Association”. Disponível em
< <http://www.gsmworld.com/index.shtml>> Acesso em 20.Jul.2006.
- [21] “TDMA to CDMA2000 White Paper”, CDMA Development Group. Disponível em
< http://www.cdg.org/technology/cdma_technology/shosteck/portuguese/comparing.asp > Acesso em 13 Julho.2006.
- [22] JUNQUEIRA, F.P, “IS 95 – padrão CDMA do EIA/TIA”. Disponível em
< <http://www.gta.ufrj.br/~flavio/commovel/IS-95.htm> > Acesso em 17.Fev.2007.
- [23] “Viabilizando a Indústria sem Fio”. Disponível em < <http://www.qualcomm.com.br> > Acesso em 17.Fev.2007.
- [24] “WCDMA”. Disponível em
< <http://www.qualcomm.com.br/technology/wcdma.html> >Acesso em 17.Fev.2007.
- [25] KOCHEM,C., PEREIRA, B. e COUTINHO, M., “Interfaces Um e Gb”. Disponível
< http://www.ipg.pt/user/~sduarte/rc/Trabalhos2005/Rede_GPRS/Protocolo.htm > Acesso em 17.Fev.2007.
- [26] MORETO, M., “Signaling System Number 7 (SS7)”. Disponível em
< <http://www.penta.ufrgs.br/tp951/rdsi/SS7.html> > Acesso em 17.Fev.2007.

- [27] ANSELMO, F., “Tudo Sobre JSP”. Editora Visual Books, 2006.
- [28] “GSM 07.07”. Disponível em
< <http://www.anatel.gov.br/Tools/frame.asp?link=/certificacao/crl0187.pdf> >
Acesso em 17.Fev.2007.
- [29] SAMPAIO, C., “TCP/IP e Intranets”. Editora Brasoft –2ª Edição, 2002.
- [30] “Venda de Celular em 2006”. Gazeta mercantil. Disponível em
< <http://www.gazeta.com.br> > Acesso em 17.Fev.2007.
- [31] “J2ME”. Disponível em < <http://java.sun.com/javame/index.jsp> > Acesso em
17.Fev.2007.
- [32] BERNARDES, L.H.C., “Modem GSM Programável em Java para Comunicação Wireless”,Saber Eletrônica nº 388, pág. 10,11,12,Maio.2005.
- [33] “Módulos da Motorola g18, c18, g18 Data Modem”. Disponível em
< <http://www.informattechnology.com.br> > Acesso em 17.Fev.2007.
- [34] Duodigit Soluções em Tecnologia. Disponível em < <http://www.duodigit.com.br> >
Acesso em 20. Nov.2006.
- [35] GPS. Disponível em < <http://pt.wikipedia.org/wiki/GPS> > Acesso em 17.Fev.2007.
- [36] “Método de Triangulação. Princípio de Funcionamento do Navstar-GPS”.
Disponível em < <http://www.turma-aguia.com/davi/gps/principio.htm> > Acesso em
17.Fev.2007.
- [37] MACORATTI, J.C. “Leitor de Código de Barras. Conceitos Básicos e Utilização no VB”. Disponível em < http://www.macoratti.net/cod_bar.htm > Acesso em
17.Fev.2007.
- [38] “AVR 8 -Bit RISC”. Disponível em < <http://www.atmel.com/products/avr/> > Acesso
em 17.Fev.2007.
- [39] National Semic., Inc. Disponível em < <http://www.national.com> > Acesso em
10.Dez.2006.
- [40] TC45 Java User Guide – Manual do Usuário do TC45 fornecido pelo fabricante
(Duodigit), 2005.
- [41] SOUZA,D.J. e LAVINIA, N. C., “Conectando o PIC - Recursos Avançados”.
2ª Edição–Ed. Érica, 2007.
- [42] Siemens, Inc. Disponível em < <http://siemens.com> > Acesso em 14.Nov.2006.

- [43] “3GSM Plataform”. Disponível em
< <http://www.gsmworld.com/technology/3g/index.shtml> > Acesso em 12.Fev.2006.
- [44] PIM, V. B. “Sensor de Temperatura, Técnicas de Medição de Temperatura”. Disponível em
<<http://vinicius.brasil.vilabol.uol.com.br/electronica/Sensortemp/SENSORTEMP.htm>> Acesso em 23. Abril.2007.
- [45] Maxim/Dallas, Inc. Disponível em < <http://www.maxin-ic.com> > Acesso em 09.Nov.2006.
- [46] TOMPKINS, W.J. and Webster, J. G. “Interfacing Sensors to the IBM PC”, Ed. Prentice Hall, USA.
- [47] BRAGA, N.C. “Módulos inteligentes LCD Multi-Matrix”. Saber Eletrônica, nº 201, pp. 11-28, 1989.
- [48] SOARES, M.J., “Gravador de PIC – PROG PIC II”, Mecatrônica Fácil, nº 13, Editora Saber, Nov/Dez.2003.
- [49] ROMÃO, B., “PhPTriad 2.2.1: Pacote de Desenvolvimento de PHP”. Disponível em
< <http://www.portaldaprogramacao.com/artigos2.asp?n=868> > e
< <http://www.pontophp.com.br/downloads.php> > Acesso em 17.Fev.2007.
- [50] Servidor Web Apache. Disponível em < <http://www.apache.org> > Acesso em 17.Fev.2007.
- [51] MPS PHP DESIGNER. Disponível em < <http://www.mpssoftware.dk/> > Acesso em 17.Fev.2007.
- [52] ANSELMO, F. “Tudo o Que Você Queria Saber Sobre o JSP Quando Utiliza o Servidor TomCat com o Banco MySQL”, Editora Visual Books, 2006.
- [53] Operadora de Telefonia Celular CLARO. Disponível em <http://www.claro.com.br>
Acesso em 25.Fev.2007.
- [54] SOUZA, D.J., *Desbravando o PIC*. Editora Érica, 2006.
- [55] ZAVALIK, C., LACERDA, G. e Oliveira, J.P.M., *Implementado Web Server com Software Livre*. Disponível em < <http://www.inf.ufrgs.br/~palazzo/OAI> >
Acesso em 17.Fev.2007.
- [56] SHOSTECK, G., CDMA, D.G., “GSM ou CDMA: Os desafios comerciais e tecnológicos para operadoras de TMA”. Disponível em
<http://arcari.br.tripod.com/Artigos/GSM_ou_CDMA.html>. Acesso em 30.Jan..2006.

- [57] ‘G SM - Comparação para vias de Migração 3G’. Disponível em < http://www.cdg.org/technology/cdma_technology/shosteck/portuguese/comparing.asp > Acesso em 12.jul.2006.
- [58] MATOS, E. T., ‘Programação JAVA para Wireless’ –Editora Digerati, 2005.
- [59] ‘Uma Visão sobre Segurança nas redes GSM, GPRS e UMTS’, Disponível em < <http://www.dcc.unicamp.br/~980649/sec.html> > Acesso em 10.Out.2006.
- [60] SunOne. Plataforma de Desenvolvimento da Sun Microsystems. Disponível em < <http://www.sun.com> > Acesso em 23.04.2007.

Anexo 1 Software aplicativo do PIC

```

;*****
;*      UNESP-ILHA SOLTEIRA - TRABALHO DE MESTRADO      *
;*      AMOSTRADOR DE TEMPERATURA/TENSÃO DE 0 - 5 Volts  *
;*      TAXA TRANSMISSÃO DE 20 SEGUNDOS                  *
;*****
;*  VERSÃO : 2.1                                         *
;*  DATA : 29/07/2006                                   *
;*****

;*****
;*      DESCRIÇÃO GERAL                                  *
;*****
; ESTE PROJETO FOI ELABORADO PARA AMOSTRAR UMA TEMPERATURA/TENSÃO
; VARIÁVEL
; DE 0 A 40 GRAUS CELSIUS.O SOFTWARE CONVERTE O CANAL 1 DO CONVERSOR A/D
; (POTENCIÔMETRO) E MOSTRA
; NO DISPLAY O VALOR CONVERTIDO EM DECIMAL O SEU VALOR DE TEMPERATURA
; CORRESPONDENTE EM VOLTS.
; ALÉM DE MOSTRAR O VALOR NO DISPLAY, O SOFTWARE TRANSMITE PELA USART O
; VALOR
; DA CONVERSÃO. OS VALORES RECEBIDOS PELA USART TAMBÉM SÃO MOSTRADOS NO
; LCD
; COMO CARACTERES ASCII.
;
;*****
;*      CONFIGURAÇÕES PARA GRAVAÇÃO                    *
;*****

__CONFIG _CP_OFF & _CPD_OFF & _DEBUG_OFF & _LVP_OFF & _WRT_OFF & _BODEN_OFF &
_PWRTE_ON & _WDT_ON & _XT_OSC

;*****
;*      DEFINIÇÃO DAS VARIÁVEIS                          *
;*****
; ESTE BLOCO DE VARIÁVEIS ESTÁ LOCALIZADO LOGO NO INÍCIO DO BANCO 0

CBLOCK      0X20      ; POSIÇÃO INICIAL DA RAM

ACCaHI      ; ACUMULADOR a DE 16 BITS UTILIZADO
ACCaLO      ; NA ROTINA DE DIVISÃO

ACCbHI      ; ACUMULADOR b DE 16 BITS UTILIZADO
ACCbLO      ; NA ROTINA DE DIVISÃO

ACCcHI      ; ACUMULADOR c DE 16 BITS UTILIZADO
ACCcLO      ; NA ROTINA DE DIVISÃO

```

```

ACCdHI          ; ACUMULADOR d DE 16 BITS UTILIZADO
ACCdLO          ; NA ROTINA DE DIVISÃO

temp            ; CONTADOR TEMPORÁRIO UTILIZADO
                ; NA ROTINA DE DIVISÃO

H_byte          ; ACUMULADOR DE 16 BITS UTILIZADO
L_byte          ; P/ RETORNAR O VALOR DA ROTINA
                ; DE MULTIPLICAÇÃO

mulplr          ; OPERADOR P/ ROTINA DE MUTIPLICAÇÃO
mulcnd          ; OPERADOR P/ ROTINA DE MUTIPLICAÇÃO

TEMPO0
TEMPO1          ; TEMPORIZADORES P/ ROTINA DE DELAY

AUX             ; REGISTRADOR AUXILIAR DE USO GERAL

UNIDADE         ; ARMAZENA VALOR DA UNIDADE DA TENSÃO
DEZENA          ; ARMAZENA VALOR DA DEZENA DA TENSÃO
CENTENA         ; ARMAZENA VALOR DA CENTENA

```

ENDC

```

; *****
; *      DEFINIÇÃO DAS VARIÁVEIS INTERNAS DO PIC      *
; *****
; O ARQUIVO DE DEFINIÇÕES DO PIC UTILIZADO DEVE SER REFERENCIADO PARA QUE
; OS NOMES DEFINIDOS PELA MICROCHIP POSSAM SER UTILIZADOS, SEM A NECESSIDADE
; DE REDIGITAÇÃO.

```

```

#include <P16F877A.INC>          ; MICROCONTROLADOR UTILIZADO

```

```

; *****
; *      DEFINIÇÃO DOS BANCOS DE RAM      *
; *****
; OS PSEUDOS-COMANDOS "BANK0" E "BANK1", AQUI DEFINIDOS, AJUDAM A COMUTAR
; ENTRE OS BANCOS DE MEMÓRIA.

```

```

#define BANK1 BSF    STATUS,RP0  ; SELECIONA BANK1 DA MEMORIA RAM
#define BANK0 BCF    STATUS,RP0  ; SELECIONA BANK0 DA MEMORIA RAM

```

```

; *****
; *      CONSTANTES INTERNAS      *
; *****
; A DEFINIÇÃO DE CONSTANTES FACILITA A PROGRAMAÇÃO E A MANUTENÇÃO.

```

```

; ESTE PROGRAMA NÃO UTILIZA NENHUMA CONSTANTE.

```

```

; *****
; *      DECLARAÇÃO DOS FLAGs DE SOFTWARE      *
; *****
; A DEFINIÇÃO DE FLAGs AJUDA NA PROGRAMAÇÃO E ECONOMIZA MEMÓRIA RAM.

```

```

; ESTE PROGRAMA NÃO UTILIZA NENHUM FLAG DE USUÁRIO

```

```

; *****
; *                               *
; *****
; AS ENTRADAS DEVEM SER ASSOCIADAS A NOMES PARA FACILITAR A PROGRAMAÇÃO E
; FUTURAS ALTERAÇÕES DO HARDWARE.

; ESTE PROGRAMA UTILIZA UMA ENTRADA P/ O CONVERSOR A/D.
; ESTA ENTRADA NÃO PRECISA SER DECLARADA, POIS O SOFTWARE NUNCA FAZ
; REFERÊNCIA A ELA DE FORMA DIRETA, POIS O CANAL A/D A SER CONVERTIDO É
; SELECIONADO NO REGISTRADOS ADCON0 DE FORMA BINÁRIA E NÃO ATRAVÉS DE
; DEFINES. PORÉM PARA FACILITAR O ENTENDIMENTO DO HARDWARE VAMOS DECLARAR
; ESTA ENTRADA NORMALMENTE.

#define      CAD_P2      PORTA,1      ; ENTRADA A/D P/ O
POTENCIÔMETRO P2

; ALÉM DA ENTRADA DO CONVERSOR A/D, TEMOS A ENTRADA DA USART (RECEPÇÃO).
; NOVAMENTE ESTA ENTRADA NÃO NECESSITA SER DECLARADA, PORÉM, PARA
; FACILITAR O ENTENDIMENTO DO HARDWARE VAMOS DECLARAR ESTA ENTRADA
; NORMALMENTE.

#define      RXUSART     PORTC,7      ; ENTRADA DE RX DA USART

; *****
; *                               *
; *****
; AS SAÍDAS DEVEM SER ASSOCIADAS A NOMES PARA FACILITAR A PROGRAMAÇÃO E
; FUTURAS ALTERAÇÕES DO HARDWARE.

#define      DISPLAY     PORTD      ; BARRAMENTO DE DADOS DO DISPLAY

#define      RS           PORTE,0     ; INDICA P/ O DISPLAY UM DADO OU
COMANDO
; 1 -> DADO
; 0 -> COMANDO

#define      ENABLE      PORTE,1     ; SINAL DE ENABLE P/ DISPLAY
; ATIVO NA BORDA DE DESCIDA

; TEMOS TAMBÉM A SAÍDA DE TX DA USART.
; NOVAMENTE ESTA SAÍDA NÃO NECESSITA SER DECLARADA, PORÉM, PARA FACILITAR O
; ENTENDIMENTO DO HARDWARE VAMOS DECLARAR ESTA SAÍDA NORMALMENTE.

#define      TXUSART     PORTC,6     ; SAÍDA DE TX DA USART

; *****
; *                               *
; *****
; VETOR DE RESET DO MICROCONTROLADOR
; *****
; POSIÇÃO INICIAL PARA EXECUÇÃO DO PROGRAMA

      ORG    0X0000      ; ENDEREÇO DO VETOR DE RESET
      GOTO   CONFIG     ; PULA PARA CONFIG DEVIDO A REGIÃO
                        ; DESTINADA AS ROTINAS SEGUINTEs

; *****

```

```
; *          ROTINA DE DELAY (DE 1MS ATÉ 256MS)          *
; *****
; ESTA É UMA ROTINA DE DELAY VARIÁVEL, COM DURAÇÃO DE 1MS X O VALOR PASSADO
; EM WORK (W).
```

```
DELAY_MS
    MOVWF    TEMPO1          ; CARREGA TEMPO1 (UNIDADES DE MS)
    MOVLW    .250
    MOVWF    TEMPO0          ; CARREGA TEMPO0 (P/ CONTAR 1MS)

    CLRWDT          ; LIMPA WDT (PERDE TEMPO)
    DECFSZ    TEMPO0,F      ; FIM DE TEMPO0 ?
    GOTO    $-2          ; NÃO - VOLTA 2 INSTRUÇÕES
                        ; SIM - PASSOU-SE 1MS
    DECFSZ    TEMPO1,F      ; FIM DE TEMPO1 ?
    GOTO    $-6          ; NÃO - VOLTA 6 INSTRUÇÕES
                        ; SIM
    RETURN          ; RETORNA
```

```
; *****
; *          ROTINA DE ESCRITA DE UM CARACTER NO DISPLAY          *
; *****
; ESTA ROTINA ENVIA UM CARACTER PARA O MÓDULO DE LCD. O CARACTER A SER
; ESCRITO DEVE SER COLOCADO EM WORK (W) ANTES DE CHAMAR A ROTINA.
```

```
ESCREVE
    MOVWF    DISPLAY          ; ATUALIZA DISPLAY (PORTD)
    NOP          ; PERDE 1US PARA ESTABILIZAÇÃO
    BSF     ENABLE          ; ENVIA UM PULSO DE ENABLE AO DISPLAY
    GOTO    $+1          ; .
    BCF     ENABLE          ; .

    MOVLW    .1
    CALL    DELAY_MS      ; DELAY DE 1MS
    RETURN          ; RETORNA
```

```
; *****
; *          AJUSTE DECIMAL          *
; *          W [HEX] = DEZENA [DEC] ; UNIDADE [DEC]          *
; *****
; ESTA ROTINA RECEBE UM ARGUMENTO PASSADO PELO WORK E RETORNA NAS
; VARIÁVEIS
; CENTENA, DEZENA E UNIDADE O NÚMERO BCD CORRESPONDENTE AO PARÂMETRO
; PASSADO.
```

```
AJUSTE_DECIMAL
    MOVWF    AUX          ; SALVA VALOR A CONVERTER EM AUX

    CLRF    UNIDADE
    CLRF    DEZENA      ; RESETA REGISTRADORES

    MOVF    AUX,F
    BTFSC   STATUS,Z      ; VALOR A CONVERTER = 0 ?
    RETURN          ; SIM - RETORNA
                        ; NÃO
    INCF    UNIDADE,F      ; INCREMENTA UNIDADE
```

```

MOVF UNIDADE,W
XORLW      0X0A
BTFSS STATUS,Z          ; UNIDADE = 10d ?
GOTO  $+3                ; NÃO
                        ; SIM

CLRF  UNIDADE            ; RESETA UNIDADE
INCF  DEZENA,F           ; INCREMENTA DEZENA
DECFSZ      AUX,F        ; FIM DA CONVERSÃO ?
GOTO  $-.8               ; NÃO

RETURN                        ; SIM

```

```

; * * * * *
; *                               *
; *          ROTINA DE DIVISÃO          *
; * * * * *
; * * * * *
; *****
; Double Precision Division
; *****
; Division : ACCb(16 bits) / ACCa(16 bits) -> ACCb(16 bits) with
; Remainder in ACCc (16 bits)
; (a) Load the Denominator in location ACCaHI & ACCaLO ( 16 bits )
; (b) Load the Numerator in location ACCbHI & ACCbLO ( 16 bits )
; (c) CALL D_divF
; (d) The 16 bit result is in location ACCbHI & ACCbLO
; (e) The 16 bit Remainder is in locations ACCcHI & ACCcLO
; *****

```

```

D_divF
    MOVLW      .16
    MOVWF      temp                ; CARREGA CONTADOR PARA DIVISÃO

```

```

    MOVF  ACCbHI,W
    MOVWF ACCdHI
    MOVF  ACCbLO,W
    MOVWF ACCdLO                ; SALVA ACCb EM ACCd

```

```

    CLRF  ACCbHI
    CLRF  ACCbLO                ; LIMPA ACCb

```

```

    CLRF  ACCcHI
    CLRF  ACCcLO                ; LIMPA ACCc

```

DIV

```

    BCF  STATUS,C
    RLF  ACCdLO,F
    RLF  ACCdHI,F
    RLF  ACCcLO,F
    RLF  ACCcHI,F
    MOVF  ACCaHI,W
    SUBWF ACCcHI,W              ;check if a>c
    BTFSS STATUS,Z
    GOTO  NOCHK
    MOVF  ACCaLO,W
    SUBWF ACCcLO,W              ;if msb equal then check lsb

```



```

NOCHK
    BTFSS STATUS,C           ;carry set if c>a
    GOTO NOGO
    MOVF ACCaLO,W           ;c-a into c
    SUBWF ACCcLO,F
    BTFSS STATUS,C
    DECF ACCcHI,F
    MOVF ACCaHI,W
    SUBWF ACCcHI,F
    BSF STATUS,C            ;shift a 1 into b (result)
NOGO
    RLF ACCbLO,F
    RLF ACCbHI,F

    DECFSZ temp,F           ; FIM DA DIVISÃO ?
    GOTO DIV                ; NÃO - VOLTA P/ DIV
                                ; SIM
    RETURN                  ; RETORNA

```

```

; *****
; *                               *
; *      ROTINA DE MULTIPLICAÇÃO      *
; *                               *
; *****
; 8x8 Software Multiplier
; ( Fast Version : Straight Line Code )
; *****
;
; The 16 bit result is stored in 2 bytes
; Before calling the subroutine " mpy ", the multiplier should
; be loaded in location " mulplr ", and the multiplicand in
; " mulcnd ". The 16 bit result is stored in locations
; H_byte & L_byte.
; Performance :
;     Program Memory : 37 locations
;     # of cycles    : 38
;     Scratch RAM    : 0 locations
; *****

```

```

; *****
; Define a macro for adding & right shifting
; *****

```

```

mult MACRO bit                ; Begin macro

    BTFSC mulplr,bit
    ADDWF H_byte,F
    RRF H_byte,F
    RRF L_byte,F

    ENDM                      ; End of macro

```

```

; *****
; Begin Multiplier Routine
; *****

```

mpy_F

```
CLRF  H_byte
CLRF  L_byte
MOVF  mulcnd,W      ; move the multiplicand to W reg.
BCF   STATUS,C      ; Clear carry bit in the status Reg.
```

```
mult  0
mult  1
mult  2
mult  3
mult  4
mult  5
mult  6
mult  7
```

```
RETURN          ; RETORNA
```

```
; *****
; *          CONFIGURAÇÕES INICIAIS DE HARDWARE E SOFTWARE          *
; *****
; NESTA ROTINA SÃO INICIALIZADAS AS PORTAS DE I/O DO MICROCONTROLADOR E AS
; CONFIGURAÇÕES DOS REGISTRADORES ESPECIAIS (SFR). A ROTINA INICIALIZA A
; MÁQUINA E AGUARDA O ESTOURO DO WDT.
```

CONFIG

```
CLRF  PORTA      ; GARANTE TODAS AS SAÍDAS EM ZERO
CLRF  PORTB
CLRF  PORTC
CLRF  PORTD
CLRF  PORTE
```

```
BANK1          ; SELECIONA BANCO 1 DA RAM
```

```
MOVLW  B' 11111111'
MOVWF  TRISA      ; CONFIGURA I/O DO PORTA
```

```
MOVLW  B' 11111111'
MOVWF  TRISB      ; CONFIGURA I/O DO PORTB
```

```
MOVLW  B' 10111111'
MOVWF  TRISC      ; CONFIGURA I/O DO PORTC
```

```
MOVLW  B' 00000000'
MOVWF  TRISD      ; CONFIGURA I/O DO PORTD
```

```
MOVLW  B' 00000100'
MOVWF  TRISE      ; CONFIGURA I/O DO PORTE
```

```
MOVLW  B' 11011011'
MOVWF  OPTION_REG      ; CONFIGURA OPTIONS
                        ; PULL-UPs DESABILITADOS
                        ; INTER. NA BORDA DE SUBIDA DO RB0
                        ; TIMER0 INCREM. PELO CICLO DE MÁQUINA
                        ; WDT - 1:8
                        ; TIMER - 1:1
```

```

MOVLW    B' 00000000'
MOVWF    INTCON                      ; CONFIGURA INTERRUPTÕES
                                           ; DESABILITA TODAS AS INTERRUPTÕES

MOVLW    B' 00000100'
MOVWF    ADCON1                      ; CONFIGURA CONVERSOR A/D
                                           ; RA0, RA1 E RA3 COMO ANALÓGICO
                                           ; RA2, RA4 E RA5 COMO I/O DIGITAL
                                           ; PORTE COMO I/O DIGITAL
                                           ; JUSTIFICADO À ESQUERDA
                                           ; 8 BITS EM ADRESH E 2 BITS EM ADRESL
                                           ; Vref+ = VDD (+5V)
                                           ; Vref- = GND ( 0V)

MOVLW    B' 00100100'
MOVWF    TXSTA                      ; CONFIGURA USART
                                           ; HABILITA TX
                                           ; MODO ASSINCRONO
                                           ; TRANSMISSÃO DE 8 BITS
                                           ; HIGH SPEED BAUD RATE

MOVLW    .25
MOVWF    SPBRG                      ; ACERTA BAUD RATE -> 9600bps

BANK0                                ; SELECIONA BANCO 0 DA RAM

MOVLW    B' 10010000'
MOVWF    RCSTA                      ; CONFIGURA USART
                                           ; HABILITA RX
                                           ; RECEPÇÃO DE 8 BITS
                                           ; RECEPÇÃO CONTÍNUA
                                           ; DESABILITA ADDRESS DETECT

MOVLW    B' 01001001'
MOVWF    ADCON0                      ; CONFIGURA CONVERSOR A/D
                                           ; VELOCIDADE -> Fosc/8
                                           ; CANAL 1
                                           ; MÓDULO LIGADO

```

; AS INSTRUÇÕES A SEGUIR FAZEM COM QUE O PROGRAMA TRAVE QUANDO HOUVER UM
; RESET OU POWER-UP, MAS PASSE DIRETO SE O RESET FOR POR WDT. DESTA FORMA,
; SEMPRE QUE O PIC É LIGADO, O PROGRAMA TRAVA, AGUARDA UM ESTOURO DE WDT
; E COMEÇA NOVAMENTE. ISTO EVITA PROBLEMAS NO START-UP DO PIC.

```

BTFSC    STATUS,NOT_TO              ; RESET POR ESTOURO DE WATCHDOG TIMER ?
GOTO     $                          ; NÃO - AGUARDA ESTOURO DO WDT
                                           ; SIM

```

```

; * * * * *
; *          INICIALIZAÇÃO DA RAM          *
; * * * * *

```

; ESTA ROTINA IRÁ LIMPAR TODA A RAM DO BANCO 0, INDO DE 0X20 A 0X7F

```

MOVLW    0X20
MOVWF    FSR                          ; APONTA O ENDEREÇAMENTO INDIRETO PARA
                                           ; A PRIMEIRA POSIÇÃO DA RAM

```

```

LIMPA_RAM
    CLRf   INDF                ; LIMPA A POSIÇÃO
    INCf   FSR,F              ; INCREMENTA O PONTEIRO P/ A PRÓX. POS.
    MOVf   FSR,W
    XORLW   0X80                ; COMPARA O PONTEIRO COM A ÚLT. POS. +1
    BTFSS  STATUS,Z            ; JÁ LIMPOU TODAS AS POSIÇÕES?
    GOTO   LIMPA_RAM           ; NÃO - LIMPA A PRÓXIMA POSIÇÃO
                                ; SIM

```

```

; *****
; *          CONFIGURAÇÕES INICIAIS DO DISPLAY          *
; *****
; ESTA ROTINA INICIALIZA O DISPLAY P/ COMUNICAÇÃO DE 8 VIAS, DISPLAY PARA 2
; LINHAS, CURSOR APAGADO E DESLOCAMENTO DO CURSOR À DIREITA.

```

```

INICIALIZACAO_DISPLAY
    BCF    RS                  ; SELECIONA O DISPLAY P/ COMANDOS

    MOVLW   0X30                ; ESCRIVE COMANDO 0X30 PARA
    CALL    ESCRIVE            ; INICIALIZAÇÃO

    MOVLW   .3
    CALL    DELAY_MS           ; DELAY DE 3MS (EXIGIDO PELO DISPLAY)

    MOVLW   0X30                ; ESCRIVE COMANDO 0X30 PARA
    CALL    ESCRIVE            ; INICIALIZAÇÃO

    MOVLW   0X30                ; ESCRIVE COMANDO 0X30 PARA
    CALL    ESCRIVE            ; INICIALIZAÇÃO

    MOVLW   B' 00111000'        ; ESCRIVE COMANDO PARA
    CALL    ESCRIVE            ; INTERFACE DE 8 VIAS DE DADOS

    MOVLW   B' 00000001'        ; ESCRIVE COMANDO PARA
    CALL    ESCRIVE            ; LIMPAR TODO O DISPLAY

    MOVLW   .1
    CALL    DELAY_MS           ; DELAY DE 1MS

    MOVLW   B' 00001100'        ; ESCRIVE COMANDO PARA
    CALL    ESCRIVE            ; LIGAR O DISPLAY SEM CURSOR

    MOVLW   B' 00000110'        ; ESCRIVE COMANDO PARA INCREM.
    CALL    ESCRIVE            ; AUTOMÁTICO À DIREITA

```

```

; *****
; *          ROTINA DE ESCRITA DA TELA PRINCIPAL          *
; *****
; ESTA ROTINA ESCRIVE A TELA PRINCIPAL DO PROGRAMA, COM AS FRASES:
; LINHA 1 - "UNESP-Ilha Solteira"
; LINHA 2 - "V:  Volts"

```

```

    MOVLW   0X81                ; COMANDO PARA POSICIONAR O CURSOR
    CALL    ESCRIVE            ; LINHA 0 / COLUNA 0
    BSF     RS                  ; SELECIONA O DISPLAY P/ DADOS

```

```

MOVLW      ' U'
CALL  ESCRVE
MOVLW      ' N'
CALL  ESCRVE
MOVLW      ' E'
CALL  ESCRVE
MOVLW      ' S'
CALL  ESCRVE
MOVLW      ' P'
CALL  ESCRVE
MOVLW      '.'
CALL  ESCRVE
MOVLW      ' I'
CALL  ESCRVE
MOVLW      ' I'
CALL  ESCRVE
MOVLW      ' h'
CALL  ESCRVE
MOVLW      ' a'
CALL  ESCRVE
MOVLW      ' '
CALL  ESCRVE
MOVLW      ' S'
CALL  ESCRVE
MOVLW      ' o'
CALL  ESCRVE
MOVLW      ' I'
CALL  ESCRVE
MOVLW      ' t'
CALL  ESCRVE
MOVLW      ' e'
CALL  ESCRVE
MOVLW      ' i'
CALL  ESCRVE
MOVLW      ' r'
CALL  ESCRVE
MOVLW      ' a'
CALL  ESCRVE

```

```

BCF  RS
MOVLW      0XC7
CALL  ESCRVE
BSF  RS

```

```

; SELECIONA O DISPLAY P/ COMANDOS
; COMANDO PARA POSICIONAR O CURSOR
; LINHA 1 / COLUNA 7
; SELECIONA O DISPLAY P/ DADOS
; COMANDOS PARA ESCREVER AS
; LETRAS DE "Volts"

```

```

MOVLW      ' V'
CALL  ESCRVE
MOVLW      ' o'
CALL  ESCRVE
MOVLW      ' l'
CALL  ESCRVE
MOVLW      ' t'
CALL  ESCRVE
MOVLW      ' s'
CALL  ESCRVE

```

```

; *****
; *                                *
; *****
; A ROTINA PRINCIPAL FICA AGUARDANDO O FINAL DA CONVERSÃO A/D E VERIFICANDO
; SE ALGUM DADO FOI RECEBIDO PELA USART

```

```

BSF    ADCON0,GO        ; INICIA CONVERSÃO A/D
                        ; EXECUTADA APENAS UMA VEZ

```

LOOP

```

CLRWDI                                ; LIMPA WATCHDOG TIMER

```

```

BTFSS  ADCON0,GO        ; FIM DA CONVERSÃO ?
GOTO   FIM_CONVERSAO_AD ; SIM
                        ; NÃO
BTFSC  PIR1,RCIF        ; RECEBEU ALGUM DADO NA SERIAL ?
GOTO   DADO_RECEBIDO    ; SIM
                        ; NÃO
GOTO   LOOP             ; VOLTA P/ LOOP

```

```

; *****
; *                                *
; *****
; ESTA ROTINA MOSTRA O VALOR DA CONVERSÃO A/D NO DISPLAY LCD TANTO EM
; DECIMAL
; COMO EM HEXADECIMAL. O VALOR DA CONVERSÃO TAMBÉM É TRANSMITIDO PELA
; USART.
; AO FINAL, A ROTINA REQUISITA UMA NOVA CONVERSÃO A/D.

```

FIM_CONVERSAO_AD

```

; ***** MOSTRA VALOR DA CONVERSÃO A/D EM DECIMAL *****

```

```

MOVF   ADRESH,W          ; SALVA VALOR DA CONVERSÃO NO WORK
MOVWF  mulplr            ; CARREGA WORK EM mulplr

MOVLW  .50
MOVWF  mulcnd            ; CARREGA 50d EM mulcnd

CALL   mpy_F             ; CHAMA ROTINA DE MULTIPLICAÇÃO

MOVF   H_byte,W
MOVWF  ACCbHI            ; SALVA VALOR DA MULTIPLICAÇÃO
MOVF   L_byte,W          ; EM ACCb PARA SER UTILIZADO NA
MOVWF  ACCbLO            ; ROTINA DE DIVISÃO

CLRF   ACCaHI            ; CARREGA ACCa COM 255d (FUNDO DE
MOVLW  .255              ; ESCALA DO CONVERSOR A/D)
MOVWF  ACCaLO            ; (ESTÃO SENDO UTILIZADOS 8 BITS)

CALL   D_divF            ; CHAMA ROTINA DE DIVISÃO

MOVF   ACCbLO,W          ; FAZ O AJUSTE DECIMAL PARA

```

CALL AJUSTE_DECIMAL ; MOSTRAR NO DISPLAY (LCD)

BCF RS ; SELECIONA O DISPLAY P/ COMANDO
MOVLW 0XC3 ; COMANDO PARA POSICIONAR O CURSOR
CALL ESCRIVE ; LINHA 1 / COLUNA 3
BSF RS ; SELECIONA O DISPLAY P/ DADOS

MOVF DEZENA,W
ADDLW 0X30 ; CONVERTE BCD DA DEZENA EM ASCII
CALL ESCRIVE ; ENVIA AO LCD

MOVLW ','
CALL ESCRIVE ; ESCRIVE UMA VIRGULA NO LCD

MOVF UNIDADE,W
ADDLW 0X30 ; CONVERTE BCD DA UNIDADE EM ASCII
CALL ESCRIVE ; ENVIA AO LCD

; ***** MOSTRA VALOR DA CONVERSÃO A/D EM HEXADECIMAL *****

SWAPF ADRESH,W ; INVERTE NIBLE DO ADRESH
ANDLW B' 00001111' ; MASCARA BITS MAIS SIGNIFICATIVOS
MOVWF AUX ; SALVA EM AUXILIAR

MOVLW 0X0A
SUBWFAUX,W ; AUX - 10d (ATUALIZA FLAG DE CARRY)
MOVLW 0X30 ; CARREGA WORK COM 30h
BTFSK STATUS,C ; RESULTADO É POSITIVO? (É UMA LETRA?)
MOVLW 0X37 ; SIM - CARREGA WORK COM 37h
; NÃO - WORK FICA COM 30h (NÚMERO)
ADDWF AUX,W ; SOMA O WORK AO AUXILIAR
; (CONVERSÃO ASCII)

MOVF ADRESH,W ; CARREGA WORK COM ADRESH
ANDLW B' 00001111' ; MASCARA BITS MAIS SIGNIFICATIVOS
MOVWF AUX ; SALVA EM AUXILIAR

MOVLW 0X0A
SUBWFAUX,W ; AUX - 10d (ATUALIZA FLAG DE CARRY)
MOVLW 0X30 ; CARREGA WORK COM 30h
BTFSK STATUS,C ; RESULTADO É POSITIVO? (É UMA LETRA?)
MOVLW 0X37 ; SIM - CARREGA WORK COM 37h
; NÃO - WORK FICA COM 30h (NÚMERO)
ADDWF AUX,W ; SOMA O WORK AO AUXILIAR
; (CONVERSÃO ASCII)

; ***** ROTINA DE DELAY DE 20 SEGUNDOS *****

```

    MOVLW .80
    MOVWF temp
    MOVLW .250
    CALL DELAY_MS
    DECFSZ temp,F
    GOTO $-3

; ***** TRANSMITE VALOR DA CONVERSÃO A/D PELA USART *****

    MOVF ADRESH,W          ; CARREGA WORK COM O VALOR DO A/D

    BANK1                  ; ALTERA P/ BANCO 1 DA RAM
    BTFSS TXSTA,TRMT       ; O BUFFER DE TX ESTÁ VAZIO ?
    GOTO $-1               ; NÃO - AGUARDA ESVAZIAR
    BANK0                  ; SIM - VOLTA P/ BANCO 0 DA RAM

    MOVF DEZENA,W
    ADDLW 0X30              ; CONVERTE BCD DA DEZENA EM ASCII
    MOVWF TXREG            ; SALVA WORK EM TXREG (INICIA TX)
    CALL DELAY_MS

    MOVLW ' ',
    MOVWF TXREG            ; SALVA WORK EM TXREG (INICIA TX)
    CALL DELAY_MS

    MOVF UNIDADE,W
    ADDLW 0X30              ; CONVERTE BCD DA UNIDADE EM ASCII
    MOVWF TXREG            ; SALVA WORK EM TXREG (INICIA TX)
    CALL DELAY_MS

    MOVLW 0x0D             ; CARREGA (0D) NO WORK (CR)
    MOVWF TXREG            ; SALVA WORK EM TXREG (INICIA TX)
    CALL DELAY_MS

; ***** INICIA UMA NOVA CONVERSÃO *****

    BSF ADCON0,GO          ; PEDE UMA NOVA CONVERSÃO A/D

    GOTO LOOP              ; VOLTA PARA LOOP

; *****
; *          ROTINA DE RECEPÇÃO DE DADOS NA USART          *
; *****
; ESTA ROTINA É EXECUTADA TODA VEZ QUE UM NOVO DADO É RECEBIDO PELA USART.
; O DADO RECEBIDO É MOSTRADO NO LCD (EM ASCII).

DADO_RECEBIDO
    BCF RS                ; SELECIONA O DISPLAY P/ COMANDO
    MOVLW 0XCF            ; COMANDO PARA POSICIONAR O CURSOR
    CALL ESCRIVE          ; LINHA 1 / COLUNA 15

```



```

BSF    RS                                ; SELECIONA O DISPLAY P/ DADOS

MOVF   RCREG,W                          ; CARREGA DADO RECEBIDO NO WORK
CALL   ESCRVE                           ; ENVIA AO LCD
                                           ; AO LER O REGISTRADOR RCREG O BIT
                                           ; RCIF DA INTERRUPÇÃO É LIMPO
                                           ; AUTOMATICAMENTE.

GOTO   LOOP                             ; VOLTA P/ LOOP PRINCIPAL

;*****
;*                                     *
;*          FIM DO PROGRAMA          *
;*****

END                                     ; FIM DO PROGRAMA

```

Anexo 2 bcscan - Software do Terminal TC45 JAVA

Basicamente este software é composto de quatro aplicativos executáveis além do *bcscan.jad* que contém alguns parâmetros de referência de configuração que são “empacotados” em um único denominado de *bcscan.jar*.

-Main

-SerialConnectionListener

-SerialConnectionThread

-HTTPUploadThread

Anexo 2.1 bcscan.jar

debug: true

MIDlet-Jar-Size:

MIDlet-1: barCodeScanner, , com.siemens.icm.ws.bcscanner.Main

http_host: www.getulio.eng.br/mestrado/get.php

MIDlet-Jar-URL: a:/bcScan.jar

MicroEdition-Configuration: CLDC-1.0

serial_connection_descr:

comm:com1;baudrate=9600;bitsperchar=8;stopbits=1;parity=none;blocking=on

MIDlet-Version: 0.1

MIDlet-Name: barCodeScanner

MIDlet-Description: bar code scanner

MIDlet-Vendor: Siemens AG

MicroEdition-Profile: MIDP-1.0

gprs_atc: at^sjnet=gprs,claro.com.br,claro,claro,claro.com.br

Anexo 2.2 Main

```
/*
 * Created on 18.07.2003
 */
package com.siemens.icm.ws.bcscanner;

import javax.microedition.midlet.MIDlet;
import javax.microedition.midlet.MIDletStateChangeException;

import com.siemens.icm.io.ATCommandFailedException;

/**
 * @author Anton Senger ICM WM AE CS
 * The main class of the application
 */
public class Main extends MIDlet implements SerialConnectionListener {
    private SerialConnectionThread serialConnectionThread;
    private HTTPUploadThread httpUploadThread;
    private final static String VERSION = "0.1.1";
    private final static String BUILD_NR = "2003081205";
    public final boolean DEBUG;

    public Main () {
        super ();
        System.out.println (Runtime.getRuntime().freeMemory()+" bytes of "+
            Runtime.getRuntime().totalMemory()+" bytes available");
        String debugProperty = getAppProperty("debug");
        if (debugProperty != null) debugProperty = debugProperty.toLowerCase();
        DEBUG = ! ("false".equals(debugProperty) || "no".equals(debugProperty));
    }

    /**
     * This is the entry point of the application.
     * @see javax.microedition.midlet.MIDlet#startApp()
     */

    protected void startApp() throws MIDletStateChangeException {
        System.out.println ("Bar Code Scanner v "+VERSION+" _ "+BUILD_NR);
        try {
            httpUploadThread=new HTTPUploadThread(this);
            serialConnectionThread = new SerialConnectionThread (this);
        } catch (ATCommandFailedException e) {
```

```

        exceptionOccured(e);
    }
    httpUploadThread.start();
    serialConnectionThread.start();
    serialConnectionThread.addListener(this);
}

/**
 * this method does nothing, but we must implement it ...
 * @see javax.microedition.midlet.MIDlet#pauseApp()
 * */
protected void pauseApp() {}

/**
 * @see javax.microedition.midlet.MIDlet#destroyApp(boolean)
 */
protected void destroyApp(boolean arg0) throws MIDletStateChangeException {
    serialConnectionThread.stop();
    httpUploadThread.stop();
}

/**
 * this method is called, if at least one character is received from the serial
 * connection. The characters are forwarded to the
 * {@link HTTPUploadThread#upload(String) HTTPUploadThread.upload(String str)}
 * method.
 * @param str the received string
 * @see com.siemens.icm.ws.bcscanner.SerialConnectionListener#received(String)
 */
    public void received(String str) {
        if (DEBUG) System.out.println ("rcv="+str);
        httpUploadThread.upload(str);
    }

/**
 * @see
 com.siemens.icm.ws.bcscanner.SerialConnectionListener#exceptionOccured(java.lang.Exc
 eption)
 */
    public void exceptionOccured(Exception e) {
        System.out.println (e);
    }
}

```

Anexo 2.3 SerialConnectionListener

```
/*
 * Created on 05.08.2003
 */
package com.siemens.icm.ws.bcscanner;

/**
 * @author Anton Senger ICM WM AE CS
 * This is a listener interface for receiving characters from the serial
 * connection and notifications that an error
 * ({@link java.io.IOException IOException}) occurred
 */
public interface SerialConnectionListener {
/**
 * this method is called, if either a <code>line feed</code> has been
 * received from the serial connection, or the connection was closed. In each
 * case all characters except the finishing <code>CRLF</code> are returned.
 * @param line the received characters without <code>CRLF</code> or the
 * finishing <code>-1</code>(EOF) */
    public void received (String line);
// public void connectionClosed ();
/**
 * this method is called, if an exception, particularly
 * {@link java.io.IOException IOException} during the reading from the serial
 * port occurred
 * @param e the Exception */
    public void exceptionOccured (Exception e);
}
```

Anexo 2.4 SerialConncectionThread

```
/*
 * Created on 18.07.2003
 */
package com.siemens.icm.ws.bcscanner;
```

```

import java.io.IOException;
import java.io.InputStreamReader;
import java.util.Enumeration;
import java.util.Vector;
import javax.microedition.io.Connector;
import com.siemens.icm.io.CommConnection;

/**
 * @author Anton Senger ICM WM AE CS (WS)
 * an object of this class opens the serial connection, which is specified by
 * the parameter of the constructor and waits for incoming characters. On
 * receiving at least one character the listeners are notified. The listeners are
 * also notified, if an error occurs.
 */
public class SerialConnectionThread extends Thread {
    /**connection description as described in{@link CommConnection CommConnection}*/
    private final String CONNECTION;
    /** the default connection description.*/
    private final static String DEFAULT_CONNECTION =
        "comm:com1;baudrate=9600;bitsperchar=8;stopbits=1;parity=none;blocking=on";
    private boolean running = false;
    private Vector listeners = new Vector (5);
    private InputStreamReader in;

    /**
     * creates new instance of this thread.
     * At this place only the description of the connection is retrieved and stored
     * in the attribute {@link #CONNECTION CONNECTION}.
     * The description of the connection is either the result of the call
     * <code>main.getAppProperty("serial_connection_descr");</code> or, if that call
     * returns <code>null</code> or if the statement <code>blocking=on</code> in the #
     * return value is not found, the
     * {@link #DEFAULT_CONNECTION default connection description} is used
     * @param main the instance of the {@link Main Main class}, which initiated the
     * construction of this object.
     */
    public SerialConnectionThread (Main main) {
        String connection = main.getAppProperty("serial_connection_descr");
        if (connection != null) {
            connection = connection.toLowerCase();
            int i = connection.indexOf("blocking");
            if (i == -1) connection = DEFAULT_CONNECTION;
            else {
                int j = connection.indexOf(";", i);
                if (j == -1) j = connection.length() - 1;
                int k = connection.indexOf("on", i);
                if (k != -1 && k < j);
            }
        }
    }

```

```

        else connection = DEFAULT_CONNECTION;
    }
} else connection = DEFAULT_CONNECTION;
CONNECTION = connection;
if (main.DEBUG) System.out.println ("serial_connection_descr="+CONNECTION);
}

private void openConnection () throws IOException {
    CommConnection connection =
        (CommConnection)Connector.open(CONNECTION, Connector.READ);

    in = new InputStreamReader (connection.openDataInputStream());
}
/**
 * The run method of this thread. There a connection to serial port is opened,
 * then characters are read from this connection and forwarded to the listeners.
 * If an error ({@link java.io.IOException IOException}) occurs the listeners
 * are also notified by calling their method
 * {@link SerialConnectionListener#exceptionOccured(Exception)
 * SerialConnectionListener.exceptionOccured(Exception e)}.
 * @see Thread#run() Thread.run() */
public void run () {
    try {
        StringBuffer rcv = new StringBuffer ();
        openConnection();
        while (running) {
            if (readLine(rcv)) {
                in.close();
                openConnection();
            }
            for (int lastIndex = rcv.length()-1, chr = rcv.charAt(lastIndex);
                chr == '\n' || chr == '\r' ;
                lastIndex --, chr = rcv.charAt(lastIndex))
                rcv.deleteCharAt(lastIndex);

            for (Enumeration e = listeners.elements(); e.hasMoreElements();)
                ((SerialConnectionListener)e.nextElement()).received(rcv.toString());
        }
    } catch (IOException exception) {
        for (Enumeration e = listeners.elements(); e.hasMoreElements();)
            ((SerialConnectionListener)e.nextElement()).exceptionOccured(exception);
    }
}

/**
 * This method performs the actual reading from the serial connection.
 * It reads available bytes from the DataInputStream <code>in</code> and returns

```

```

* them as a String. It blocks, if no bytes are available, until at least one
* byte is available.
* @param in DataInputStream, from which to read
* @return bytes read from in as a String
* @throws IOException if the underlying read methods throw an IOException
* */
/*
private String read (DataInputStream in) throws IOException {
    byte[] bytes;          // this one may be tricky, so ill
    String rcvStr = "";    // describe it here :
    int readByte = -1;
    if (in.available() == 0) {    // nothing there, we must block
        readByte = in.read ();    // block, but dont forget the one, by
        int available = in.available(); // which we are woken up;
        bytes = new byte [available + 1]; // now we are woken up -> cache em all
        bytes[0] = (byte)readByte;    // and dont forget the first one
        if (available > 0) in.read (bytes, 1, available);
        rcvStr += new String (bytes);
    }
    while (in.available() > 0) {    // there are some in buffer
        bytes = new byte [in.available()]; // cache em all
        in.read (bytes);    // cache em all
        rcvStr += new String (bytes);    // and append to the return value
    }
    return rcvStr;
}
*/

private boolean readLine (StringBuffer rcv)
    throws IOException {
    if (rcv.length() > 0) rcv.delete(0, rcv.length());
    int chr;
    for (chr = in.read(); chr != '\r'  && chr !=1; chr=in.read())
        rcv.append((char)chr);
    if (chr == -1) rcv.deleteCharAt(rcv.length() - 1);
    return chr == -1;
}
/**
* starts this thread
* @see java.lang.Thread#start()
*/
public synchronized void start() {
    running = true;
    super.start();
}

/**

```



```

* stops this thread
**/
public synchronized void stop () {
    running = false;
    notify ();
}

/**
 * adds a new {@link SerialConnectionListener listener}
 * @param l the {@link SerialConnectionListener listener} to add
 */
public void addListener (SerialConnectionListener l) {
    if (!listeners.contains(l) && l != null) listeners.addElement(l);
}

/**
 * removes the specified { @link SerialConnectionListener listener}
 * @param l the { @link SerialConnectionListener listener} to remove
 */
public void removeListener (SerialConnectionListener l) {
    listeners.removeElement(l);
}
}

```

Anexo 2.5 HTTPUploadThread

```

/*
 * Created on 31.07.2003
 */
package com.siemens.icm.ws.bcscanner;

import java.io.IOException;
import java.util.Vector;

import javax.microedition.io.Connector;
import javax.microedition.io.HttpConnection;

import com.siemens.icm.io.ATCommandFailedException;

```

```

import com.siemens.icm.ws.SyncAtCommand;

/**
 * @author Anton Senger ICM WM AE CS (WS)
 * an object of this class is used to send a string to a http-server via
 * http-parameters. This class is configured by two parameters:
 * <table>
 * <tr><td>http_host</td><td>the url of the http host without any parameters</td></tr>
 * <tr><td>gprs_atc</td><td>the at command used to open the gprs
connection</td></tr>
 * </table>
 * this parameters are retrieved from the instance of the
 * {@link com.siemens.icm.ws.bcscanner.Main Main} class, that has to be passed
 * to the constructor of this class. In case that one of the parameters is not
 * found, the default value is used for that parameter.
 */
public class HTTPUploadThread extends Thread {
    private boolean running = false;
    private Vector uploadBuffer = new Vector (10);
    private final String HOST;
    /** the default value for the http host */
    private final static String DEFAULT_HOST = "www.getulio.eng.br/mestrado/get.php";
    /** the default at command to set dialup network access parameters */
    private final static String DEFAULT_ATC =
        "at^sjnet=gprs,claro.com.br,claro,claro,claro.com.br";
    // "at^sjnet=gprs,internet.t-dl.de,t-dl,gprs";
    private final String IMEI;
    private final Main main;

    public HTTPUploadThread (Main main) throws ATCommandFailedException {
        super ();
        this.main = main;
        String host = main.getAppProperty("http_host");
        this.HOST = (host != null ? host : DEFAULT_HOST);
        String atCommand = main.getAppProperty("gprs_atc");
        if (atCommand==null) atCommand = DEFAULT_ATC;
        else {
            atCommand = atCommand.trim();
            atCommand = atCommand.toLowerCase();
            if (! atCommand.startsWith("at^sjnet")) atCommand = DEFAULT_ATC;
        }
        if (main.DEBUG) {
            System.out.println ("http_host="+HOST);
            System.out.println ("gprs_atc="+atCommand);
        }
        SyncAtCommand atc = new SyncAtCommand (false);
        atc.setATDebug(main.DEBUG);
    }

```

```

        String response;
        response = atc.send("at+gsn\r").toLowerCase();
        IMEI = response.substring(0, response.indexOf("ok")).trim();
        response = atc.send(atCommand + "\r");
        response = response.toLowerCase();
        if (response.indexOf("ok") == -1)
            throw new ATCommandFailedException (response);
        atc.release();
    }

    /** starts this thread.
     * @see Thread#start() */
    public synchronized void start () {
        running = true;
        super.start();
    }

    /** stops this thread. */
    public synchronized void stop () {
        running = false;
        notify ();
    }

    /**
     * this method is used to notify this thread, that the string str must be
     * uploaded. after calling this method the url
     * <code>"http://"+HOST+"?"+"IMEI="+IMEI+"&"+ "value="+str</code> is created,
     * stored in the upload buffer and this thread is notified.
     * @see #run()
     * <h1 style="color:red">the string is not url-encoded. This may and will cause
     * problems</h1>
     * @param str the string, which is to be uploaded */
    public void upload (String str) {
        str = urlEncode(str);
        uploadBuffer.addElement("http://"+HOST+"?"+"IMEI="+IMEI+"&"+ "value="+str);
        if (main.DEBUG) System.out.println ("upload (" +str+ ")");
        synchronized (this) {
            notify ();
        }
    }

    /**
     * the thread sleeps until it is notified by the
     * {@link #upload(String) upload()} method. if this happens the contents of the
     * upload buffer are sent by retrieving the entries of the buffer, opening a
     * HttpConnection and opening a DataInputStream. Then the connection is closed
     * and either the next entry is sent or the thread sleeps again, if the upload

```

```

* buffer is empty
* @see Thread#run()
* @see HttpConnection
* @see Connector#open(java.lang.String, int, boolean) */
    public void run () {
        while (running) {
            if (uploadBuffer.isEmpty()) {
                synchronized (this) {
                    try { wait(); }
                    catch (InterruptedException e) {System.out.println (e);}// this cannot happen anyway
                }
            } else {
                String url = (String)uploadBuffer.elementAt(0);
                uploadBuffer.removeElementAt(0);
                try {
                    if (main.DEBUG) System.out.println ("connecting to "+url);
                    HttpConnection connection = (HttpConnection) Connector.open(url);
                    connection.openDataInputStream();
                    connection.close();
                } catch (IOException e) {
                    main.exceptionOccured(e);
                }
            }
        }
    }
}

```

```

private String urlEncode (String str) {
    if (str == null) return null;
    StringBuffer resultStr = new StringBuffer (str.length());
    char tmpChar;

```

```

    for (int ix = 0; ix < str.length(); ix ++) {
        tmpChar = str.charAt (ix);
        switch (tmpChar) {
            case ' ' : resultStr.append ("%20");
                break;
            case '-' : resultStr.append ("%2D");
                break;
            case '/' : resultStr.append ("%2F");
                break;
            case ':' : resultStr.append ("%3A");
                break;
            case '=' : resultStr.append ("%3D");
                break;
            case '?' : resultStr.append ("%3F");
                break;
            case '#' : resultStr.append ("%23");

```

```
break;
case '\r' : resultStr.append ("%0D");
break;
case '\n' : resultStr.append ("%0A");
break;
default : resultStr.append (tmpChar);
break;
}
}
return resultStr.toString();
}
}
```

Anexo 3 - Procedimento para a Configuração do Auto-Start

O procedimento é relativamente um pouco complexo, pois devem ser inseridas varias informações para o modulo:

1) Um Password (Senha):

- a. Não é indispensável para sua aplicação: Default é vazio =""
- b. Para mudar o Password (Senha):

Entrar o comando AT:

AT^SCFG="Userware/Passwd"[, <upwd>old, <upwd>new, <upwd>new]

2) Um delay para iniciar o programa:

Entrar o comando AT :

AT^SCFG="Userware/Autostart/Delay"[, <upwd>, <uad>]

**Importante: Um delay é indispensável para poder desabilitar o Auto-Start depois!!!
Pois uma vez a aplicação iniciada pode não ter mais possibilidade de mandar comandes AT na serial!**

3) O diretório onde é guardado seu programa .jar no modulo

Entrar o comando AT :

AT^SCFG="Userware/Autostart/AppName"[, <upwd>, <uaa>]

4) Finalmente a habilitação de Auto-Start:

- i. Entrar o comando AT :

AT^SCFG="Userware/Autostart"[, <upwd>, <ua>]

<upwd>old : Passwordo antigo
 <upwd>new : Password novo
 <upwd> : Password valido
 <uad>: tempo de 1 a 100000 em 100mS
 <uaa> : Path completo do arquivo .jar
 <ua> : habilita o Auto-Start : 1 habilita, 0 desabilita.

Na tela do Hyper-Terminal: (foi utilizado o password “jean” neste exemplo)

```
at^scfg="userware/passwd","","jean","jean"
^SCFG: "Userware/Passwd"
```

OK

```
at^scfg="userware/autostart/delay","jean","500"
^SCFG: "Userware/Autostart/Delay","500"
```

OK

```
at^scfg="userware/autostart/appname","jean","a:/bcScan.jar"
^SCFG: "Userware/Autostart/AppName","a: /bcScan.jar "
```

OK

```
at^scfg="userware/autostart","jean","1"
^SCFG: "Userware/Autostart","1"
```

OK

Para verificar a programação:

```
at^scfg?
^SCFG: "AutoExec","0","0","0","0",""
^SCFG: "AutoExec","0","1","0","0","","000:00:00","000:00:00"
^SCFG: "AutoExec","0","1","1","0","","000:00:00","000:00:00"
^SCFG: "AutoExec","0","1","2","0","","000:00:00","000:00:00"
^SCFG: "Userware/Autostart","1"
^SCFG: "Userware/Autostart/AppName","a: : /bcScan.jar "
^SCFG: "Userware/Autostart/Delay","500"
^SCFG: "Userware/Passwd"
^SCFG: "URC/Datamode/Ringline","off"
^SCFG: "URC/Ringline","local"
```

OK

Desligar o Modem, religar e esperar 50 Segundos. Na reinicialização o Auto-Start deverá funcionar.

Para tirar o Auto-Start, entrar rapidamente o comando AT:

```
ATE1  
at^scfg="userware/autostart","jean","0"
```

NOTA: - Para poder visualizar o que está sendo digitado, entrar com o comando ATE1 , assim estará “ecoando” as teclas.

Desligar o Modem depois para fazer efeito.

NOTA: -Se por acaso perder o controle do Modem, ligar o Modem sem SIM, esta aplicação volta no modo AT quanto não tem SIM. (isso é muito útil...)

Anexo 4 - Script de captura da informação na Web em PHP e seu armazenamento em um banco de dados MySQL

get.php

```
<html>
<head>
<title>Página PHP</title>
</head>
<body>
<?php

if (isset($_GET["IMEI"]) && isset($_GET["value"]))
{
    $IMEI = $_GET["IMEI"];
    $value = $_GET["value"];

}
else
;

$date = date("d-m-Y");
$time = date("H:i:s");

$link = mysql_connect(' 200.152.208.109' , ' getulioeba44feaa' );
if (!$link) {
    die(' Not connected : ' . mysql_error());
}

// make getulioengbr the current db
$db_selected = mysql_select_db(' getulioengbr' , $link);
if (!$db_selected) {
    die (' Cãt t use getulioengbr: ' . mysql_error());
}

$cadastra = "INSERT INTO telemetria
            (IMEI, value, data, hora)
            VALUES
            (' $IMEI' , ' $value' , ' $date' , ' $time' )";
$resultado = mysql_query($cadastra)
    or die("Falha na execução do cadastro");

echo "Dados adicionados com sucesso";
```

?>

</body>

</html>

Anexo 5 - Página em PHP que efetua Consulta ao Banco de Dados

```
<html>
<head>

<meta http-equiv="refresh" content="10;
URL=http://www.getulio.eng.br/mestrado/consulta.php">

<title>Monitoramento Remoto de Dados Via Internet</title>
</head>
<body>
<?php
$link = mysql_connect("200.152.208.109", "getulio", "ba44feaa")
    or die("Não foi possível conectar");

mysql_select_db("getulioengbr", $link)
    or die("Não foi possível selecionar o banco de dados");

$consulta = "SELECT * FROM telemetria ORDER BY numero DESC";
$resultado = mysql_query($consulta)
    or die("Falha na execução da consulta");
?>

<p align=center><font size="4" color="#000080"><b>UNESP-Ilha
Solteira/SP</b></font></P>
<p align=center><b>Monitoramento Remoto de Dados Via Internet usando Comunicação
sem Fio</b>
<align=center><i>http://www.getulio.eng.br/mestrado/consulta.php</i></center>

<p align=center><b> Resultado da Consulta</b>
<?php
echo "<table border=1 width=60%>";
echo "<tr>";
echo "<td width=10% align=center><b>Nº</b></td>";
echo "<td width=10% align=center><b>IMEI</b></td>";
echo "<td width=10% align=center><b>Valor</b></td>";
echo "<td width=10% align=center><b>Data</b></td>";
echo "<td width=10% align=center><b>Hora</b></td><p>";
echo "</tr>";
while ($linha = mysql_fetch_assoc($resultado))
{
    $numero = $linha["numero"];
    $IMEI = $linha["IMEI"];
```

```

$value = $linha["value"];
$data = $linha["data"];
$hora = $linha["hora"];
echo "<tr>";
echo "<td width=10% align=center>$numero</td>";
echo "<td width=10% align=center>$IMEI</td>";
echo "<td width=10% align=center>$value</td>";
echo "<td width=20% align=center>$data</td>";
echo "<td width=20% align=center>$hora</td><p>";

echo "</tr>";

}

echo "</table>"

?>

</body>
</html>

```