

**Diego Liberalquino**

***Desenvolvimento de Plataforma de Comunicação  
GSM/GPRS para Sistemas Embarcados***

Recife - PE, Brasil

29/11/2010

**Diego Liberalquino**

***Desenvolvimento de Plataforma de Comunicação  
GSM/GPRS para Sistemas Embarcados***

Monografia apresentada para obtenção do Grau  
de Bacharel em Engenharia da Computação  
pela Universidade de Pernambuco.

Orientador:  
Sérgio Campello Oliveira

ENGENHARIA DA COMPUTAÇÃO  
ESCOLA POLITÉCNICA DE PERNAMBUCO  
UNIVERSIDADE DE PERNAMBUCO

Recife - PE, Brasil

29/11/2010

## *Resumo*

Sistemas embarcados são empregados usualmente em aplicações de telemetria e monitoramento. Esses dispositivos, muitas vezes, precisam utilizar comunicação sem fio para enviar ou receber comandos a um outro dispositivo que se localiza a quilômetros de distância. Redes GSM podem ser de grande valor nessas circunstâncias, uma vez que estão disponíveis em muitos lugares. Este trabalho propõe o projeto de uma plataforma de comunicação para transmissão e recepção de dados através da especificação GSM.

# *Abstract*

Embedded systems are often employed in telemetry and monitoring applications. These devices, many times, need to use wireless communication to send or receive commands to another device which may be located kilometers away. GSM networks can be of great value under these circumstances, because they are available in many places. This work proposes the design of a communication platform for data transmission and reception through the GSM specification.

# *Agradecimentos*

Muitas pessoas contribuíram para o meu crescimento durante os tempos em que eu passei na academia. A lista é grande, mas de grande significância. Neste espaço, tentarei brevemente retribuir com palavras a contribuição que me foi dada por elas.

Primeiramente, agradeço a meus pais e minha família pelo afeto, refúgio e apoio. Sem vocês, eu não teria chegado onde estou.

Agradeço também a meus companheiros de time em muitos projetos de faculdade, especialmente Jefferson Silva de Amorim, Péricles Miranda, Pedro Corrêa, com quem estive mais presente e meus colegas de iniciação científica Leandro Honorato e Ismael Mascarenhas. Vocês me mostraram a verdadeira essência da cooperação e amizade.

Agradeço ao meu irmão, Gabriel Liberalquino, meu principal conselheiro em assuntos de eletrônica, e, provavelmente, a pessoa em que compartilhei mais momentos nesta vida.

Por fim, agradeço a minha musa inspiradora, bela, inteligente e paciente. Pollyana Maciel me fez encontrar o caminho do amor, carinho e dedicação.

## *Lista de Figuras*

2.1	Estrutura de uma rede GSM (HEINE, 1999) . . . . .	p. 13
2.2	Modelo cliente-servidor de uma rede GPRS (SEURRE et al., 2003) . . . . .	p. 16
2.3	Arquitetura de uma rede GPRS (SEURRE et al., 2003) . . . . .	p. 17
2.4	Arquitetura de rede SMS (BODIC, 2005) . . . . .	p. 20
2.5	Diagrama de blocos do PIC18F4550 (MICROCHIP, 2007) . . . . .	p. 22
3.1	Diagrama elétrico da plataforma de comunicação . . . . .	p. 28
3.2	Interface de gravação e reset . . . . .	p. 29
3.3	Circuito de interface para USB . . . . .	p. 30
3.4	Circuito regulador de tensão . . . . .	p. 31
3.5	Pinagem do Nokia 2720 . . . . .	p. 33
3.6	Esquema para conversão de níveis de tensão (MBus) . . . . .	p. 34
3.7	Esquema para conversão de níveis de tensão (FBus) . . . . .	p. 34
5.1	Imagem da placa de circuito impresso . . . . .	p. 47
5.2	Conexão de linhas de transmissão do aparelho celular (Nokia 2720) . . . . .	p. 48

# *Lista de Tabelas*

2.1	Lista de comandos AT para operações com SMS (MULTITECHSYSTEMS, 2003) . . . . .	p. 24
4.1	Configurações do oscilador . . . . .	p. 38
4.2	Camadas de firmware da plataforma . . . . .	p. 38

# *Sumário*

<b>1</b>	<b>Introdução</b>	p. 9
1.1	Estrutura do Documento . . . . .	p. 11
<b>2</b>	<b>Tecnologias de software e hardware</b>	p. 12
2.1	GSM . . . . .	p. 12
2.1.1	Terminologia . . . . .	p. 14
2.1.2	GPRS . . . . .	p. 15
2.1.3	SMS . . . . .	p. 18
2.2	PIC 18F4550 . . . . .	p. 20
2.3	Protocolos de troca de mensagens . . . . .	p. 21
2.3.1	Comandos AT . . . . .	p. 23
2.3.2	Nokia MBus e FBus . . . . .	p. 24
2.3.3	Codificação de conteúdo . . . . .	p. 25
<b>3</b>	<b>Projeto de Hardware</b>	p. 27
3.1	Projeto de Hardware . . . . .	p. 27
3.2	Modem . . . . .	p. 31
<b>4</b>	<b>Projeto de Firmware</b>	p. 36
4.1	Configuração do microcontrolador . . . . .	p. 36
4.2	Organização da biblioteca . . . . .	p. 38
4.3	Descrição de funções e características da biblioteca . . . . .	p. 40
4.3.1	API para SMS . . . . .	p. 40

4.3.2	Funções de utilidade . . . . .	p.44
<b>5</b>	<b>Resultados e testes</b>	<b>p.46</b>
5.1	Hardware confeccionado . . . . .	p.46
5.2	Testes realizados . . . . .	p.48
<b>6</b>	<b>Conclusão</b>	<b>p.50</b>
	<b>Referências Bibliográficas</b>	<b>p.52</b>

# *1 Introdução*

Sistemas embarcados possuem diversas aplicações, de controle de máquinas a automação residencial. Muitos deles possuem capacidade de transmitir dados para outros dispositivos, seja outros sistemas embarcados, máquinas industriais ou computadores. A tecnologia utilizada depende do problema a ser resolvido, da disponibilidade do canal, do consumo de energia e do custo de implementação, manutenção e uso.

O problema descrito a seguir ilustra como a comunicação através da rede GSM pode ser útil como alternativa de comunicação em um projeto de sistemas embarcados.

Em um trabalho de pesquisa para a foi desenvolvido um sistema de monitoramento para cadeias de isoladores de alta tensão da Companhia Hidro-elétrica do São Francisco (CHESF) (OLIVEIRA; FONTANA, 2009). Essas instalações – as quais muitas vezes se localizam em locais afastados de regiões metropolitanas – costumam sofrer com o acúmulo de poluição e substâncias sólidas, que, depositadas na superfície dos isoladores e com o aumento da humidade, podem aumentar e causar descargas parciais cujo grau e gravidade podem aumentar e causar descargas completas conhecidas como *flashover*. Para realizar a medição da quantidade de resíduos, foi desenvolvido um sensor ótico que verifica a presença de regiões condutoras nas superfícies dos isoladores através da detecção de correntes de fuga, cujas formas de onda possuem correlação com a quantidade de poluentes acumulados. O sensor é conectado a um microcontrolador

que armazena dados coletados de pulsos de corrente de fuga em memória não volátil. O artefato construído pode ser acoplado às torres de transmissão com pouca dificuldade, e reduz os custos associados à inspeção.

O dispositivo construído possui algumas limitações. Os dados obtidos da corrente de fuga são armazenados em memória, ela pode funcionar ativamente por cerca de dois meses, e a coleta dos dados é realizada através de transmissão serial RS-232, entre plataforma e computador (OLIVEIRA; FONTANA, 2009). O sensor precisa ser retirado manualmente por uma equipe para que seja feita a análise dos dados coletados. Esse processo poderia ser automatizado caso fizesse uso de comunicação sem fio, que enviaria mensagens para os postos centrais da CHESF. Por esse propósito, foi proposta a utilização de vários protocolos de comunicação sem fio (GSM, Satélite e Bluetooth), de acordo com custo e disponibilidade de cada tecnologia.

A importância da rede GSM como meio de comunicação para o projeto mencionado se deve às limitações dos outros modelos: satélite e bluetooth. A transmissão via satélite é capaz de alcançar grandes distâncias, porém é lenta e cara. Já o protocolo bluetooth tem custo mínimo e seu funcionamento é simples, mas é incapaz de alcançar grandes distâncias. Por isso GSM é um meio termo entre as duas, mas útil para evitar os custos da transmissão via satélite quando terminais bluetooth não se encontram em lugares próximos.

Este trabalho propõe a elaboração de uma plataforma de software e hardware que possibilite que a comunicação seja realizada através das redes GSM, através de mensagens SMS ou conexão GPRS. A plataforma poderia funcionar independentemente, ou operar através de comandos de um outro sistema embarcado, e poderia ser usada para mais de um propósito, seja em aplicações comerciais – como no caso da CHESF – ou para ensino em um contexto acadêmico. Por isso é também objetivo a utilização de ferramentas de baixo custo e componentes disponíveis no mercado local, além da elaboração de software e hard-

ware sob licença que possibilite a modificação e distribuição do código fonte e esquema de hardware por qualquer pessoa ou empresa, com apenas algumas poucas restrições legais.

Um celular de modelo Nokia 2720 foi utilizado para acessar a rede GSM/GPRS. Porém, por falta de documentação e equipamentos para conectá-lo à plataforma proposta, não foi possível realizar o acesso à rede, como era esperado. É possível demonstrar a capacidade de enviar e receber SMS através de um aplicativo para computador pessoal, mas esta solução ainda não foi adequada com sucesso à plataforma.

## 1.1 Estrutura do Documento

O texto será dividido em cinco capítulos, incluindo esta introdução. A descrição de cada capítulo está disposta a seguir:

- **Capítulo 2 - Revisão bibliográfica:** Neste capítulo faremos uma revisão dos conceitos, tecnologias e ferramentas utilizadas para a elaboração e construção do dispositivo embarcado.
- **Capítulo 3 - Projeto de Hardware:** O hardware da plataforma, seu núcleo, periféricos e conexões externas serão abordados.
- **Capítulo 4 - Projeto de Firmware:** A configuração do firmware, sua arquitetura e bibliotecas serão discutidos.
- **Capítulo 5 - Resultados e Testes:** O design final da plataforma será ilustrado. Teremos algumas considerações sobre as funcionalidades do hardware e testes realizados.
- **Capítulo 6 - Conclusão:** Aqui serão realizadas algumas considerações finais a respeito dos resultados obtidos e sobre como esse trabalho pode ser estendido em outras pesquisas.

## ***2 Tecnologias de software e hardware***

Antes de apresentar detalhes do conteúdo que será discutido nos capítulos seguintes, faremos uma revisão breve dos assuntos que serão abordados neste projeto. Primeiramente serão apresentadas as características de uma rede GSM, seguida de informações sobre o microcontrolador empregado no projeto do hardware da plataforma, as ferramentas utilizadas para o desenvolvimento do firmware, e, finalmente, considerações sobre o modem GSM utilizado.

### **2.1 GSM**

GSM (Global System for Mobile Communications) é um protocolo de comunicação para dispositivos sem fio elaborado em 1982, destinado a homogeneizar as tecnologias de comunicação usadas nos países europeus que anteriormente eram puramente nacionais e por isso, caras. Em 1992, no entanto, a tecnologia – que já estava em operação em algumas partes da Europa e Estados Unidos – despertou interesse internacional (HEINE, 1999).

A arquitetura de uma rede GSM utiliza uma estrutura celular, onde cada célula – em cujo centro localiza-se uma estação base – utiliza uma faixa de frequência de maneira que células vizinhas não utilizem a mesma faixa, como pode ser ilustrado pela figura 2.1. O alcance de cada estação base é o menor possível, para que frequências escarsas sejam reusadas mais vezes. Vale lembrar que as faixas de frequência em que estações móveis operam sofrem bastante

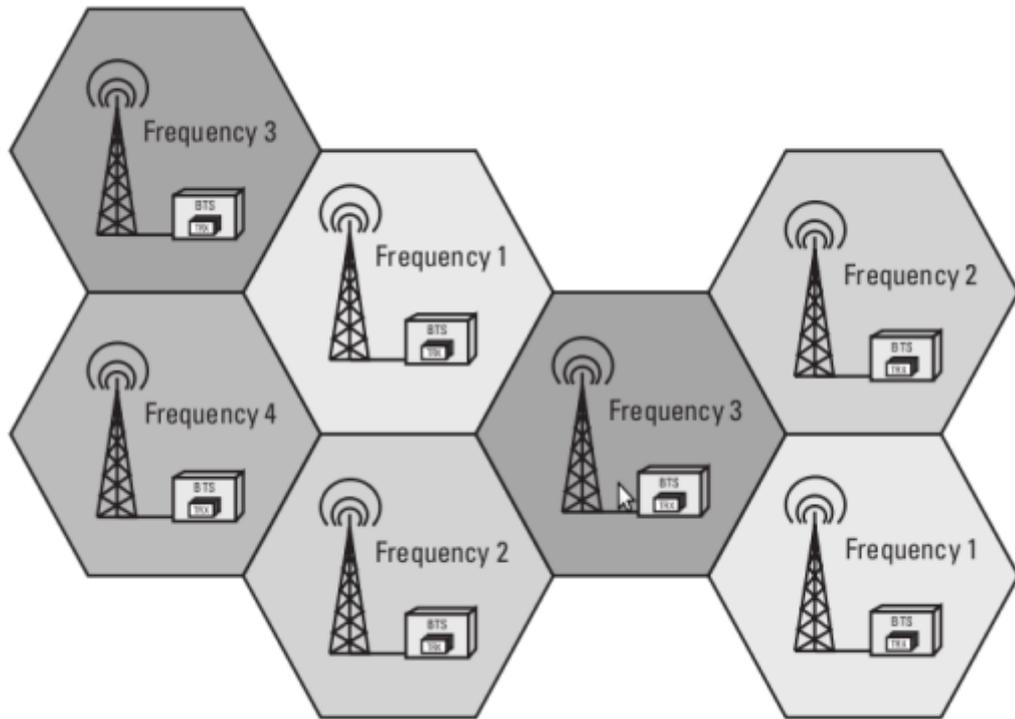


Figura 2.1: Estrutura de uma rede GSM (HEINE, 1999)

com atenuação por serem, geralmente, emissores de baixa potência. Por isso distâncias entre estação base e móvel de até 5 km são admitidas.

A estrutura celular também possui suas desvantagens. Um grande número de células significa um grande custo de infra-estrutura. Quando uma estação móvel se desloca, ela pode sair das fronteiras de uma célula para outra, e por isso, se faz necessária uma outra chamada para a nova estação base, um processo conhecido como *handover*. É preciso manter a localização aproximada de cada estação móvel dentro de uma célula, para que seja possível entregar uma chamada realizada por outro dispositivo. Por causa dos problemas de *handover* e localização, a sinalização – controle de emissão e recepção de sinais para que a comunicação seja estabelecida – se torna bastante complexa, com um volume de dados a ser analisado e, conseqüentemente, uma carga muito grande para ser processada por um único computador central.

### 2.1.1 Terminologia

A seguir está uma lista de termos e definições utilizados pela especificação GSM que são relevantes para este trabalho:

- **Mobile Station (MS) ou Estação Móvel:** Terminais de uma rede GSM, dispositivos portáteis que realizam chamadas dentro de uma célula.
- **Base Station Subsystem(BSS) ou Subsistema de Estação Base:** Gerenciam as operações de rádio, e provêm conectividade entre estações móveis e a rede GSM. De fato, a BSS é dividida entre outros dois módulos, *Base Transceiver Station* (BTS), que provê uma interface no ar e se encarrega das tarefas de rádio e *Base Station Controller* (BSC) que contacta vários BTS e exerce funções de controle.
- **Subscriber Identity Module (SIM):** Fornecem informações de identificação e provedor de serviço para estações móveis. Também servem como bancos de dados, onde guardam dados de chamada e número de diretório. Fisicamente, são representados por um chip (SIM card), que pode ser acoplado a um aparelho móvel.
- **Subscribers ou Assinantes:** São os clientes virtuais de operadoras das redes GSM. Cada assinante é identificado por um SIM. Embora um cliente físico possa ter vários SIM cards, diferentes SIM cards representam assinantes diferentes.

Ainda existem outros termos bastantes comuns na literatura GSM. O *Mobile Service Switching Center* (MSC) conecta vários BSC, e se encarrega de encaminhar as chamadas oriundas de dentro e fora de seu segmento de rede. Já o *Home Location Register* (HLR) age como um repositório, que armazena e administra dados de assinantes da rede GSM. O *Equipment Identity Register* (EIR) mantém registro dos dispositivos móveis e não dos SIM, para entre outros

propósitos, prevenir furtos de equipamentos e rastrear criminosos. Todos esses nós da rede citados, juntos, formam uma *Public Land Mobile Network* (PLMN).

### 2.1.2 GPRS

GPRS é uma sigla que significa *General Packet Radio Service*. Seu propósito é facilitar a interconexão entre redes móveis e redes de comutação de pacotes, e, principalmente, prover acesso à internet (SEURRE et al., 2003). Esta tecnologia abriu espaço para uma gama de novos serviços, entre eles *Multimedia Messaging Services* (MMS) e serviços de ação a distância, como sistemas de vigilância e monitoramento eletrônico (o tipo de aplicação proposto por este trabalho).

Os tipos de tráfego que mais se beneficiam de GPRS são:

- Transmissão não frequentes, como serviços de mensagem (SMS e MMS).
- Frequência de pequenos blocos de dados, com uma taxa inferior ou igual a 500 bytes por minuto, tais como um browser de internet.
- Baixa frequência de grandes blocos de dados, com poucas transações por hora.
- Taxa de transmissão assíncrona entre *downlink* e *uplink* (referentes ao envio e recebimento de mensagens), usados em browsers.

Diferentemente do conceito de chamada usado na telefonia, redes GPRS utilizam o modelo cliente-servidor existente em redes de computadores (figura 2.2). O dispositivo pode ser configurado como um cliente que faz requisições a um servidor, que pode estar localizado dentro de uma intranet ou internet. No entanto existe uma configuração menos comum, onde o móvel pode aceitar requisições como uma aplicação vertical para monitoramento por telemetria.

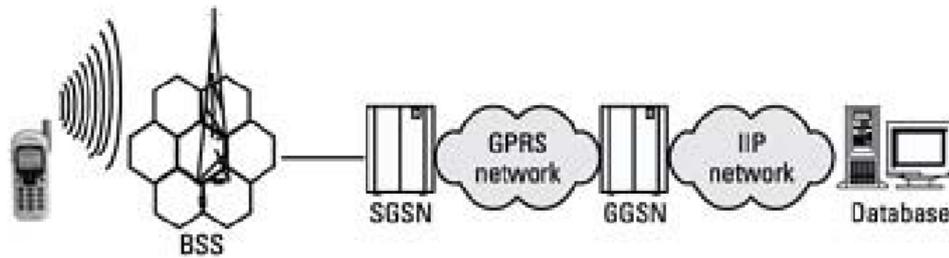


Figura 2.2: Modelo cliente-servidor de uma rede GPRS (SEURRE et al., 2003)

Nesse caso, o dispositivo precisa ser configurado para processar uma requisição e respondê-la ao seu recipiente.

A arquitetura de uma rede GPRS reusa muitos dos conceitos usados em GSM, como BSS, MSC e HLR, mas introduz novos nós, o *Gateway GPRS Support Node* (GGSN) e *Serving GPRS Support Node* (SGSN), que juntos formam o núcleo de uma rede GPRS. Com esses novos elementos, a estrutura lógica da rede GPRS (figura 2.3) está disposta da maneira descrita a seguir:

- **SGSN** (*Serving GPRS Support Node*): O nó da rede que serve uma estação móvel. Entrega pacotes para esta e comunica ao HLR para obter o perfil do assinante, mantendo um registro dos canais atribuídos a cada estação móvel para propósitos de roteamento. O SGSN é conectado a um ou vários BSS.
- **GGSN** (*Gateway GPRS Support Node*): Responsável pela conexão com redes de comutação de pacotes externas (como a internet). e estabelece comunicação com o SGSN através de uma rede IP de *backbone* (nó central que conecta outras redes IP menores).
- **HLR** (*Home Location Register*): Atua como banco de dados, armazenando dados de assinantes e informação de roteamento.
- **MSC** (*Mobile Services Switching Center*): Gerencia as camadas de origem e destino de usuários GSM e gerencia mobilidade. Pode encaminhar pagi-

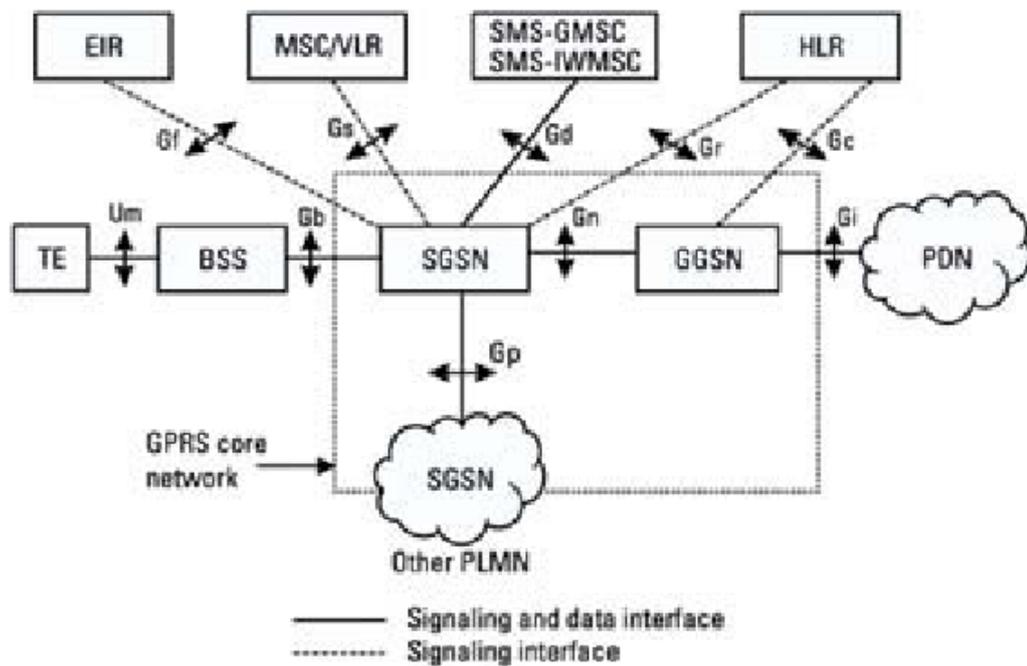


Figura 2.3: Arquitetura de uma rede GPRS (SEURRE et al., 2003)

nação de redes comutadas por circuito de MSs para SGSN. Não está diretamente envolvida na comunicação GPRS.

- **EIR** (*Equipment identity register*): Banco de dados que mantém registros das identidades de terminais da rede GPRS.
- **TE** (*Terminal Equipment*): Terminal da rede GPRS (um telefone celular, por exemplo).

Os termos *Um*, *Gf*, *Gs*, *Gb*, *Gp*, *Gn*, *Gd*, *Gr*, *Gc* e *Gi* representam interfaces entre dois equipamentos, que podem ser compostas de hardware, software, protocolos ou outros mecanismos para que seja possível encaminhar uma mensagem de um nó para outro. Um, por exemplo, representa a interface de ar entre um telefone celular e a estação base.

### 2.1.3 SMS

Um dos serviços disponíveis para redes GSM é conhecido como SMS (Simple Message Service) e permite a troca de mensagens de texto entre assinantes. SMS é usado não somente com fins de comunicação entre usuários, mas usados por aplicações comerciais para distribuição de conteúdo (promoções, avisos, notícias, etc), mesmo aplicações mais complexas, tais como monitoramento remoto e localização. Uma de suas principais vantagens é a sua disponibilidade na maioria dos dispositivos móveis, dos mais antigos aos mais recentes e mesmo alguns terminais fora de uma rede GSM (computadores, telex) podem ser usuários do serviço. Existem duas extensões para SMS: EMS (Enhanced Messaging Service) e MMS (Multimedia Messaging Service), que permitem a troca de tipos de mensagem mais sofisticados, como textos com animação e multimídia, respectivamente (BODIC, 2005).

O SMS introduz novos elementos na arquitetura da rede(GSM):

- **Short Message Entity (SME):** Terminais da rede que enviam ou recebem SMS, sejam dispositivos móveis ou terminais externos (servidores de internet, por exemplo). Um SME pode ser um servidor que faz a conexão entre o SMSC diretamente ou através de um gateway, e por isso, recebe a denominação ESME. Um exemplo de SME é um proxy/servidor WAP(*Wireless Application Protocol*). Na terminologia SMS, um SME que envia uma mensagem é um *originador*, enquanto aquele que recebe a mensagem um *recipiente*
- **SMS Center (SMSC):** O SMSC é responsável por receber e encaminhar mensagens de SME para SME, ou armazená-las quando o recipiente não está disponível. O SMSC pode fazer parte da rede GSM ou estar localizado fora dela. Um único SMSC poderia atender serviços de todas as operadoras, contudo, geralmente cada uma costuma manter o seu SMSC, fazendo

acordos para que mensagens sejam trocadas entre redes diferentes.

- **Email Gateway:** Provê interoperabilidade entre o serviço de email (Internet) e SMS (GSM), convertendo mensagens de um formato para outro e encaminhando-as de uma rede para outra.

A especificação SMS separa o processo de comunicação em quatro níveis de abstração, ou camadas, como será descrito a seguir (figura 2.4).

A primeira camada, *Short Message Application Layer (SM-AL)*, consiste em softwares que enviam, recebem e interpretam o conteúdo das mensagens. Já a segunda camada, *Short Message Transfer Layer (SM-TL)*, acrescenta cabeçalhos que permitem a entrega de uma mensagem SMS, tais como comprimento, remetente e receptor. O cabeçalho e o conteúdo são representados por uma sequência de octetos, chamados de *Transfer Protocol Data Unit (TPDU)*. Assim, como a camada de transporte do protocolo TCP/IP, a SM-TL usa mecanismos de controle para garantir a entrega de mensagens do SME de origem ao SME de destino. As duas camadas anteriores são de maior importância para o desenvolvimento de aplicações que operam sobre SMS.

Também merecem ser mencionadas as terceira e quarta camadas, *Short Message Relay layer (SM-RL)* e *Short Message Link Layer (SM-LL)*. Elas se encarregam, respectivamente, de informações utilizadas pelos mecanismos para transporte de mensagem através dos nós da rede SMS – SMEs, SMSCs etc – e da codificação das TPDU para a transmissão no meio físico (BODIC, 2005). Na camada SM-RL, dois nós da rede SMS merecem ser mencionados: *SMS Gateway MSC (SMS-GMSC)* e *SMS InterWorking MSC (SMS-IWMSC)*. O primeiro recebe a mensagem de um SMSC, interroga o HLR a respeito do caminho a ser tomado, e entrega a rede recipiente. O segundo nó realiza o trabalho inverso, onde recebe uma mensagem da rede GSM/GPRS e a encaminha para um SMSC.

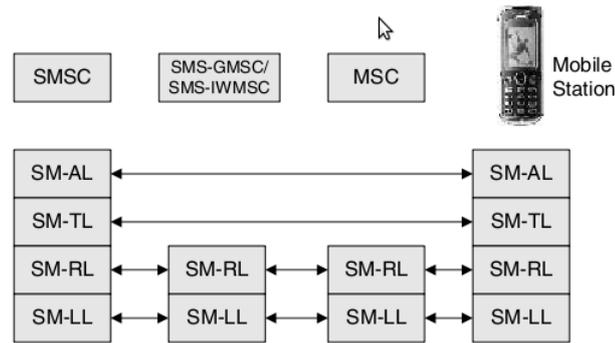


Figura 2.4: Arquitetura de rede SMS (BODIC, 2005)

## 2.2 PIC 18F4550

Computadores, em uma organização simplista e geral, são compostos de um ou mais processadores, memória principal e dispositivos de entrada e saída. Dessa forma, eles podem gerenciar uma grande gama de dispositivos, modificando os algoritmos lógicos que determinam sua operação – o software –, sem precisar, necessariamente, modificar o processador, o grande responsável pela administração de recursos da máquina. Sistemas embarcados não são diferentes, mas ao invés de usarem um computador de organização complexa, utilizam chips de propósito geral, com vários dispositivos – processador, memória principal e memória externa, e diversos módulos de entrada e saída – embutidos em uma única pastilha de silício. Esses chips são chamados de microcontroladores, e são empregados para controle de outros periféricos, tais como sensores térmicos e motores de passo.

O microcontrolador usado neste trabalho pertence à família PIC18, fabricado pela empresa Microchip, mais especificamente o modelo PIC18F4550. Possui as seguintes características (MICROCHIP, 2007):

- Baseado na arquitetura Harvard – com barramentos separados para memória de instrução e dados, pipeline de dois ciclos, um ciclo de leitura (*fetch*) e outro de execução (*execute*).

- A palavra de instrução tem tamanho de 16 bits, enquanto a palavra de dados, 8 bits. Pode endereçar até 16K palavras de instrução, e 2K palavras de dados. Memória de dados dividida em bancos
- Vários periféricos (USB, UART, MSSP, SPP, etc).
- Várias entradas e saídas analógicas.
- Interrupções externas e internas, timer, CCP e PWM

O modelo PIC18F4550 (figura 2.5) possui uma série de vantagens em relação a outros similares. É um dos poucos controladores da família PIC18 que possuem capacidade de realizar comunicação através de USB (*Universal Serial Bus*), possui várias instruções que facilitam o uso de linguagens de programação de alto nível (C, BASIC). A grande quantidade de periféricos disponíveis no chip é um dos diferenciais de microntroladores PIC, por reduzir a necessidade de uso de vários componentes externos para fornecer ao sistema embarcado a funcionalidade de cada tecnologia.

Existem muitos termos usados no modelo PIC18F4550 que não serão explicados neste texto. Para mais detalhes, seu datasheet deve ser consultado (MICROCHIP, 2007).

### **2.3 Protocolos de troca de mensagens**

Para que sistemas embarcados adquiram acesso à rede GSM, é possível utilizar um modem que se encarregue dos detalhes mais complicados da comunicação. Para que seja possível a troca de mensagens entre esses dois dispositivos, é comum que estes utilizem uma conexão serial, baseada no padrão RS232. O corpo da mensagem enviada utiliza uma gramática que defina quais partes de uma mensagem conterá comandos, dados, endereços etc, para que, assim, as

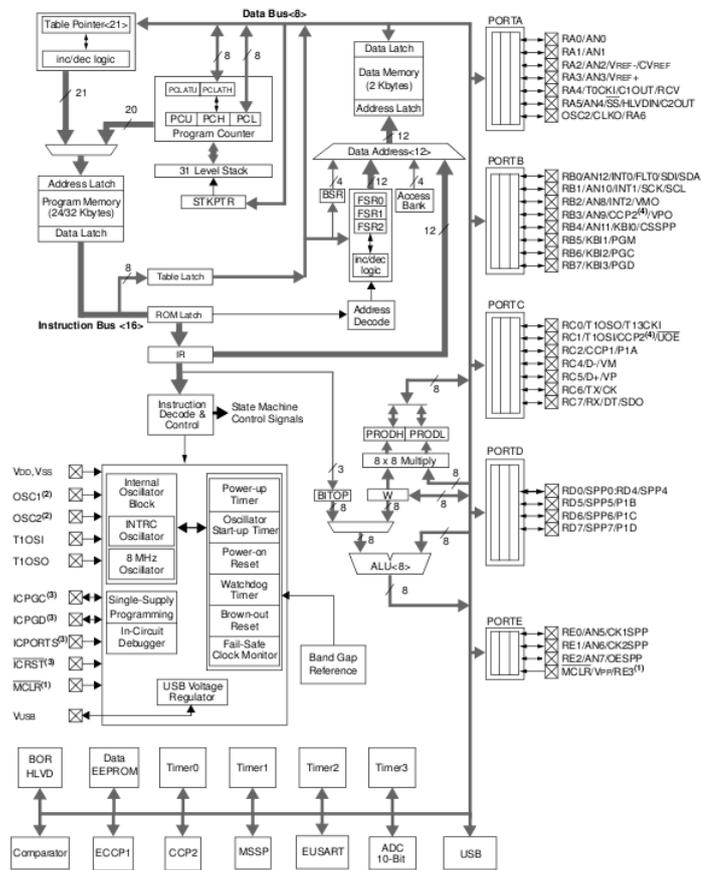


Figura 2.5: Diagrama de blocos do PIC18F4550 (MICROCHIP, 2007)

funcionalidades do modem estejam disponíveis de modo transparente para o núcleo do sistema embarcado.

Neste trabalho serão abordados dois tipos de protocolos de troca de mensagens: comandos AT e pacotes FBus.

### 2.3.1 Comandos AT

Muitos modems são operados por computadores e outros sistemas embarcados através de uma interface serial e um conjunto de comandos conhecidos como **Comandos AT**. O corpo desses comandos é constituído de apenas uma string de texto (geralmente ASCII ou outra codificação), iniciados pelo cabeçalho “AT+”, seguidos por um comando e alguns parâmetros. Todo comando AT é enviado como uma requisição, que é tratado e é devolvida uma resposta – no mesmo formato – ao emissor, com alguma mensagem de confirmação ou dados. Esse comportamento é similar ao encontrado em arquiteturas cliente-servidor, e de fato, modems sempre dependem de um host que gerencie sua operação em alguma aplicação específica.

Comandos AT permitem quatro tipos de operação (HOME, 2010):

- **Test:** Verifica se determinado comando é suportado pelo dispositivo.
- **Set:** Modifica configurações no modem para determinadas tarefas.
- **Read:** Verifica configurações de determinada operação no modem.
- **Execution:** Realiza uma ação ou retira informações de status sobre a execução do modem ou dispositivo móvel.

O padrão GSM define um conjunto de comandos AT que seja comum a todos os dispositivos que se conectem à rede celular, independente de fabricantes. A lista é bastante extensa, mas, na tabela 2.1, serão exibidos alguns comandos

para envio de SMS, acompanhado de uma breve descrição (MULTITECHSYSTEMS, 2003).

Tabela 2.1: Lista de comandos AT para operações com SMS (MULTITECHSYSTEMS, 2003)

+CMGS	Envia mensagem SMS
+CMGL	Lista mensagens armazenadas na memória
+CMGR	Lê mensagens armazenadas
+CMGL	Lista mensagens armazenadas na memória
+CMGF	Define modo de operação do modem (Texto ou PDU)
+CSCA	Gerencia número de SMSC

### 2.3.2 Nokia MBus e FBus

MBUS e FBUS são um protocolos de comunicação proprietários da Nokia que permitem a operação de dispositivos celulares a partir de comandos enviados por uma interface serial (UART). Esses protocolos estão presentes em qualquer celular Nokia, e utiliza um conjunto de comandos para enviar mensagens SMS, upload de arquivos para memória interna, configuração de chamadas, entre outros.

O primeiro utiliza um canal de comunicação half-duplex, taxa de transmissão de 9600bps, 8 bits de dados, paridade ímpar, um bit de parada. As linhas DTR (*Data Terminal Ready*) e RTS (*ready to send*) precisam estar em nível lógico 0. Quanto ao protocolo FBUS, a comunicação serial é realizada através de uma interface full-duplex, com taxas de transmissão de 115200 bps, 8 bits de dados, nenhum bit de paridade e um bit de parada. Os sinal DTR e RTS precisam estar em nível lógico 1 e 0, respectivamente. Os canais de entrada e saída do celular operam a 3.3V, uma vez que são voltados para o menor consumo de energia.

A Mensagem a ser transmitida é agrupada em um quadro de vários octetos. O primeiro octeto possui valor 0x1E, e representa o *frame id* da mensagem. A seguir, o segundo octeto contém o endereço do celular e o terceiro, o endereço de origem (no nosso caso o endereço do microcontrolador). Os quarto e quinto octetos representam o comprimento da mensagem, onde o quarto é o byte mais significativo e o quinto o menos significativo. O corpo da mensagem está localizado nos próximos octetos e, obrigatoriamente, precisa ter o mesmo comprimento (em bytes) que o valor descrito nos duas posições anteriores. Finalmente, os dois último octetos são o *checksum* da mensagem, o primeiro byte é obtido a partir da operação XOR de todos os bytes ímpares do quadro, enquanto o último, de todos os bytes pares.

Após a recepção do quadro pelo aparelho celular, este transmite dois quadros como resposta. O primeiro representa um **ACK** (de *Acknowledgement* ou confirmação) e o segundo quadro representa a resposta à requisição realizada anteriormente. Enquanto o microcontrolador não responder a mensagem com um outro ACK, o celular reenviará os quadros três vezes mais, até desistir.

### 2.3.3 Codificação de conteúdo

Aparelhos móveis muitas vezes não suportam o padrão de codificação estabelecido pelo padrão ASCII. Utilizam o alfabeto de símbolos definido na especificação GSM e, por isso, deve ser realizada a conversão das mensagens – no entanto, caracteres de A a Z e numerais (0 a 9) não necessitam de conversão. Caracteres em ambas especificações possuem tamanho de 7 bits, e por isso precisam ser agrupadas em octetos. Este procedimento é padrão em dispositivos celulares, principalmente ao usar protocolos que não suportam o modo texto (como é o caso de FBUS, ou de modems que operam somente em modo PDU). O agrupamento é realizado preenchendo os bits vazios à esquerda com os bits à direita do próximo caractere. O exemplo a seguir ilustra como a mensagem

“hello” é agrupada (exemplo retirado de EMBEDTRONICS (2010)).

h	e	l	l	o	(símbolos)
68	65	6C	6C	6F	(hexadecimal)
1101000	1100101	1101100	1101100	1101111	(binário)

Agora vamos mostrar a mesma mensagem ao contrário, para facilitar a compreensão do exemplo:

o	l	l	e	h	(símbolos)
6F	6C	6C	65	68	(hexadecimal)
1101111	1101100	1101100	1100101	1101000	(binário)

Agora o agrupamento será realizado:

06	FD	9B	32	E8	(hexadecimal)
110	<b>1111</b> 1101	<b>100</b> 11011	<b>00</b> 110010	<b>1</b> 1101000	(binário)

Logo, após o agrupamento a mensagem “hello” (em hexadecimal) será: E8 32 9B FD 06.

A Nokia não disponibiliza a documentação de seu protocolo de transmissão. No entanto, a partir de experimentação, grande parte dos comandos foram documentados (em código fonte) através do projeto *Gnokii*, que também disponibiliza um programa e informações de hardware (GNOKII, 2010).

## **3 *Projeto de Hardware***

Neste capítulo serão discutidos detalhes do projeto de hardware, do modem utilizado, e do firmware presente no sistema embarcado. Na primeira seção serão expostas as funcionalidades do hardware e seu propósito, para em seguida explicar mais detalhadamente suas particularidades. Na segunda seção serão expostas as características do modem utilizado e como foi realizada a interface entre este e sistema embarcado.

### **3.1 Projeto de Hardware**

Para demonstrar a capacidade de comunicação através de GSM/GPRS, este trabalho propõe a construção de uma plataforma de desenvolvimento que tenha as seguintes características:

- Interface de gravação e debug para ICD3 da microchip.
- Dispositivos de entrada e saída para facilitar o teste de funcionalidades do firmware.
- Conexão com barramento USB, para facilitar comunicação entre plataforma e computador.
- Barramento serial para comunicação com dispositivos seriais externos utilizando UART.
- Interface para comunicação com dispositivos embarcados externos.

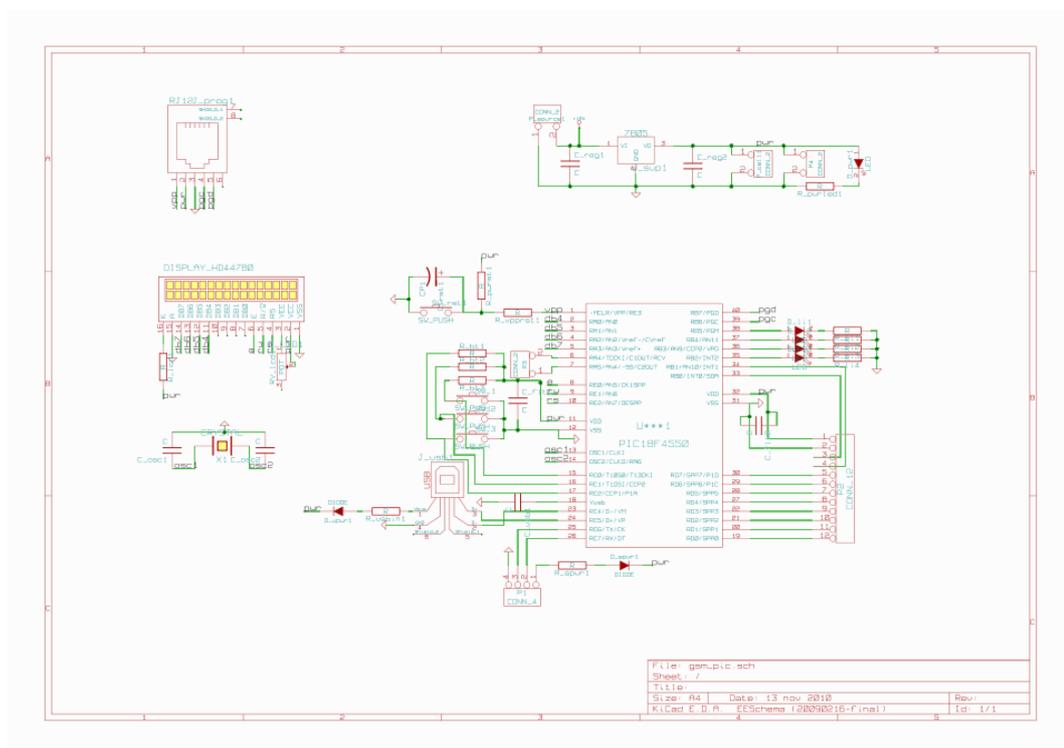


Figura 3.1: Diagrama elétrico da plataforma de comunicação

- Capacidade de operar sozinha através de uma fonte de até 9V, conexão USB, serial ou sistema embarcado externo.

No núcleo da plataforma, será usado o microcontrolador PIC18F4550, como foi mencionado no capítulo anterior. O diagrama elétrico do circuito mostra o acoplamento do microcontrolador a cada uma de suas interfaces (LCD, USB, ICSP etc), como está descrito na figura 3.1. Seus vários módulos serão descritos mais detalhadamente nas próximas seções.

Para acelerar o desenvolvimento da plataforma, foi utilizado um esquema de gravação in-circuit para que o firmware do controlador pudesse ser atualizado rapidamente. Ao pino vpp/mclr, que é usado para aplicar a tensão de gravação no controlador, foi construído um circuito de reset ativado por um botão que fica ativo com nível lógico 0 (ver figura 3.2), mas foi incluso um capacitor para evitar a queda de tensão muito brusca.

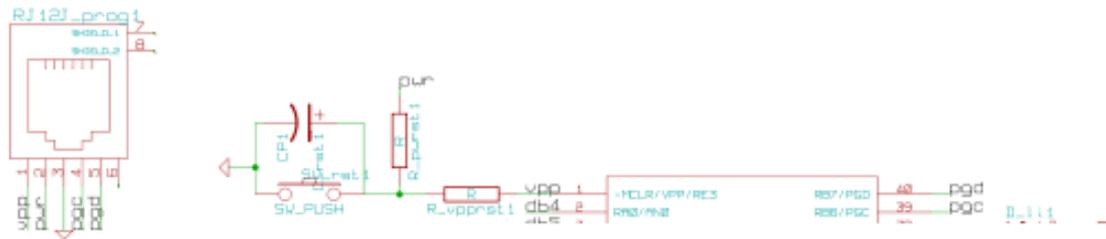


Figura 3.2: Interface de gravação e reset

Foram inclusos botões, LEDs e um display LCD ao circuito. Seu propósito é facilitar o teste de funcionalidades da plataforma, uma vez que há poucas maneiras eficientes de descobrir problemas no firmware quando este está em execução.

O circuito inclui uma interface USB (figura 3.3) para três propósitos:

- Permitir a transmissão de comandos do computador para a interface serial (modem).
- Ler, escrever e alterar memória do microcontrolador, incluindo memória flash e registradores que controlam dispositivos de entrada e saída. A interface USB pode ser mais um instrumento útil para a depuração do firmware e armazenar dados que foram armazenados durante sua execução.
- Realizar atualizações de firmware e inclusão de novas funcionalidades sem necessidade da interface de gravação. Essa característica é muito útil quando precisa-se adicionar ou corrigir uma funcionalidade do firmware quando o gravador icd3 não está disponível.

Adicionalmente, o USB pode alimentar o circuito caso a fonte não esteja presente. O chaveamento do circuito é bastante simples, realizado por um diodo comum.

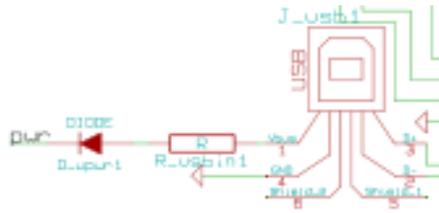


Figura 3.3: Circuito de interface para USB

O clock do microcontrolador precisa ser ajustado para que o USB possa funcionar. Por isso foi escolhido um oscilador de 20 MHz acoplado a dois capacitores de 15 pF. Isso fornece ao microcontrolador um clock inicial de 20 MHz. O PIC permite duas configurações para o uso de USB, full-speed e low-speed. Neste projeto será usada a comunicação full-speed, para que a taxa de transmissão disponível entre computador e microcontrolador seja capaz de alcançar os maiores valores possíveis (caso seja necessário). Mais detalhes sobre a configuração USB será explicado mais adiante.

O conector serial está ligado ao módulo UART do controlador. Utiliza quatro linhas, alimentação, terra, transmissão (TX) e recepção (RX). Já que vários modems diferem quanto ao nível de tensão utilizado durante a sua operação (alguns operam a 3.3 V, outros a 5 V, etc), pode ser necessário o uso de um adaptador para realizar algum ajuste, mas caso não seja necessário, pode ser ligado diretamente ao conector.

A plataforma construída não possui módulo de comunicação com outros sensores, e, por isso, ela necessita de um meio para trocar dados com outros sistemas embarcados. Logo, está a disposição um conector que está ligado a 10 linhas de transmissão, sendo que oito delas estão conectadas ao módulo SPP (Streaming Parallel Port), e outras duas a módulos sensíveis a interrupções externas. O uso dessas interrupções pode ser útil para sinalizar a comunicação entre os dois dispositivos, e servir como canais de controle para o uso das linhas SPP. No entanto, não é necessário o uso das 10 linhas de comunicação, nem

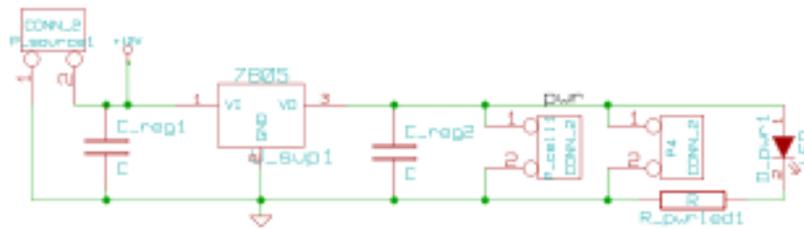


Figura 3.4: Circuito regulador de tensão

o módulo SPP para realizar a comunicação entre os dois dispositivos: os detalhes de como a troca de dados será realizada dependerá da implementação do firmware nos dois sistemas.

Assim como no caso da comunicação serial, o conector para comunicação externa pode depender de um adaptador, para que seja possível converter os níveis de tensão das linhas de transmissão. Por isso, o conector inclui duas linhas ligadas, a primeira ligada ao terra e outra ligada à alimentação do circuito.

A plataforma também pode ser alimentada através de uma fonte de 12 V comum, embora tensões mais fracas forneçam a potência suficiente. Como o PIC funciona com níveis de alimentação muito mais baixos – pode operar a 5V ou em aplicações em que se requiere preservação da bateria, 3V – a saída de tensão da fonte pode ser danosa para os componentes internos. Logo, se faz necessário o emprego de um regulador de tensão (7805) para limitar a corrente de entrada. O circuito pode ser encontrado na figura 3.4

## 3.2 Modem

Para a transmissão das mensagens através da rede GPRS, é necessário usar um instrumento que realize a conexão entre a estação base e estação móvel. Esse dispositivo deve cuidar dos detalhes de sinalização e chamada para que a aplicação desenvolvida trate apenas de detalhes da transferência dos dados. Por isso, nosso projeto fará emprego de um modem, que se encarregará dos detalhes

de comunicação para este circuito.

Como um dos objetivos deste trabalho é construir uma plataforma de comunicação GPRS através de equipamentos que podem ser encontrados facilmente no mercado local, fez-se uso de um dos modems mais comuns: telefones celulares. Muitos desses telefones possuem capacidade de se comunicar com a rede GSM, e até fornecem meios para efetuar a conexão com outros dispositivos – como por exemplo, a presença de uma linha de dados para para comunicação com computadores pessoais através da porta USB. Além disso, podem operar por muitas horas, com o uso moderado de seus recursos.

Todos os aparelhos celulares fabricados pela Nokia possuem capacidade para transmitir dados através de FBUS ou outros protocolos (comandos AT). Neste trabalho, optou-se pelo uso do modelo Nokia 2720, pois este suporta os protocolos WAP e GPRS, embora, para o uso exclusivo de SMS, modelos mais econômicos seriam suficientes.

Para enviar dados pelo celular, é necessário o uso de alguma interface entre ambos, e para tal, alguns pinos podem ser encontrados na parte posterior do celular, abaixo da bateria e próximo ao local onde é armazenado o SIM card. A figura 3.5 mostra a relação de pinos que podem ser usados para a comunicação com esta plataforma.

Os pinos Rx e Tx são os dois canais unidirecionais utilizados para comunicação através de FBus, enquanto o pino MBus é bidirecional e utilizado para o protocolo de mesmo nome. É preciso cuidado ao converter a tensão do celular para o sistema embarcado, já que este pode operar a uma tensão superior a 3.3 V.

Através de medição nos canais de transmissão, foi possível comprovar que o sinal em nível lógico alto – sem o uso de divisores de tensão – possui valor de aproximadamente 2.75 V. Segundo experimentos disponibilizados na internet,



Figura 3.5: Pinagem do Nokia 2720

a faixa de tensão aceitável para nível lógico é de 2 a 2.8 V, o que comprova a medição.

No dispositivo embarcado, a tensão em nível lógico alto das linhas Tx varia entre 3.7 e 4.3. É preciso reduzir a tensão para evitar danos no telefone. Os dois esquemas (figuras 3.6 e 3.7) mostram maneiras simples de fazer a regulação, onde Tx e Rx se referem as linhas do dispositivo embarcado (no celular, a configuração é inversa).

As relações  $\frac{R2}{R1}$  e  $\frac{R4}{R3}$  devem ser proporcionais a 1.8. Poderia-se usar uma dupla de resistores de 180 e 100  $\Omega$ , para não limitar severamente a corrente elétrica. O diodo schottky no esquema do MBus (figura 3.7) é empregado como retificador, mas, diferentemente dos diodos tradicionais, não acarreta em uma queda de tensão muito severa (de 0.15 até 0.46 V, comparados à faixa de 0.6 a 1.4 V, referentes a diodos de silício ou germânio mais comuns) e atua mais rapidamente. O Buffer 74HC541 é empregado para elevar o nível de tensão para a recepção no microcontrolador.

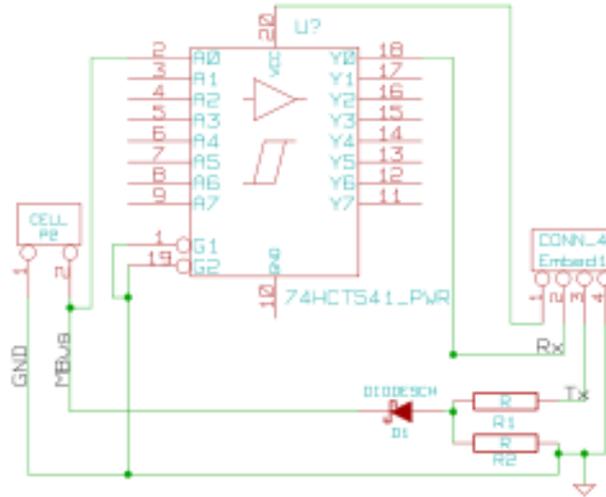


Figura 3.6: Esquema para conversão de níveis de tensão (MBus)

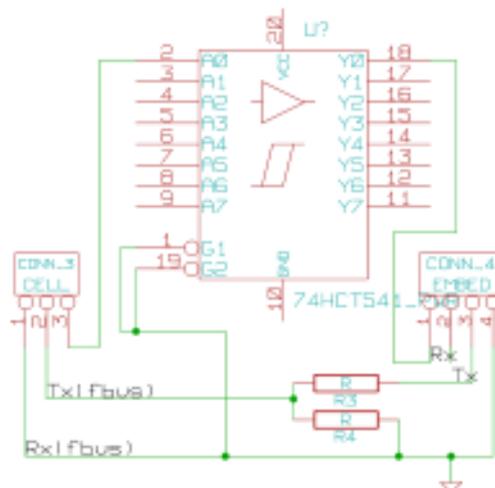


Figura 3.7: Esquema para conversão de níveis de tensão (FBus)

O celular não depende de alimentação externa. Sua alimentação é provida por sua bateria de íon lítio original.

## ***4 Projeto de Firmware***

O núcleo de sistema embarcado é um microcontrolador, que possui a responsabilidade de gerenciar quando um comando é acionado, ou quais parâmetros serão passados para os periféricos ligados as suas unidades de entrada e saída, quais unidades serão acionadas. O firmware – um programa escrito que pode ser escrito em linguagem de programação ou de montagem – contém a lógica de controle e execução de suas atividades, e está armazenado na memória de programa (não volátil, e lida durante a inicialização do mesmo). A lógica do firmware pode ser dividida em várias unidades lógicas (funções), que por sua vez compõem o conjunto de funcionalidades de um determinado módulo (biblioteca).

Uma das responsabilidades deste trabalho é a elaboração de uma biblioteca de funções que pode auxiliar programadores da plataforma a acessar as funcionalidades da rede GSM/GPRS sem precisar conhecer detalhes específicos dos protocolos de transmissão de dados. Nesta seção, veremos que funcionalidades estão disponíveis e qual a lógica relacionada a suas mensagens.

### **4.1 Configuração do microcontrolador**

O projeto do código fonte é baseado na linguagem C, segundo a especificação do compilador PICC da empresa CCS (Custom Computer Services). Este compilador não é open source, mas possui uma biblioteca extensa que reduz a

complexidade do desenvolvimento do firmware (como manipulação automática de acesso a bancos de memória). O CCS possui também uma série de módulos que se encarregam de periféricos como USB, UART, e virtual UART (porta serial implementada puramente em software). O compilador SDCC (*Simple Device C Compiler*), apesar de ser open source, ainda não possui uma versão estável.

Na primeira seção deste trabalho mencionamos a interface de gravação ICSP, que permite a gravação de um novo conjunto de instruções no microcontrolador sem que esse fosse removido da placa de circuito impresso que constitui a nossa plataforma. É no código fonte do firmware que armazenamos os parâmetros de configuração do controlador: módulos utilizados entre as unidades de entrada e saída, configurações de clock e timer, proteção de código, opções de reset, etc.

Como o PIC18F4550 utilizado faz uso de duas unidades de comunicação serial (RS 232 e USB) é necessário a configuração de um clock alto para evitar *buffer overruns* (esgotamento de espaço lógico que armazena as mensagens que chegam ao controlador, oriundas do modem ou computador, neste caso) em quaisquer desses módulos de transmissão de dados. O oscilador é configurado em modo HSPLL, que emprega cristais externos de alta frequência acoplados a um multiplicador de clock (PLL – *Phase Locked Loop*) e a um divisor de clock. Os bits PLLDIV2 do registrador PLL e CPUDIV1 do divisor de clock do processador, quando em nível lógico alto, acoplados a um cristal de 20 MHz produzem um clock de 48 MHz para o microcontrolador.

O módulo de USB possui também um clock, que é derivado da frequência de oscilação do processador. O modo full-speed requerido é obtido através da divisão do clock do processador (48 MHz). A tabela 4.1 mostra outras configurações de clock relativas ao PIC18F4550, para um oscilador de 20 MHz (MICROCHIP, 2007).

A frequência do microcontrolador destacada em negrito contém a única con-

Tabela 4.1: Configurações do oscilador

Modo de Clock	Divisão do Clock	Frequência do microcontrolador
HS, EC, ECIO	None (00)	20 MHz
	2 (01)	10 MHz
	3 (10)	6.67 MHz
	4 (11)	5 MHz
HSPLL, ECPLL, ECPIO	2 (00)	48 MHz
	3 (01)	32 MHz
	4 (10)	<b>24 MHz</b>
	6 (11)	5 MHz

figuração possível para que o módulo USB funcione no modo half-speed.

A porta serial precisa ser configurada com uma taxa de transmissão de 9600 bps, 8 bits de dados e um bit de parada. O modo FBus não utiliza paridade, enquanto no MBus a paridade é ímpar. A taxa de transmissão de 115200 bps – valor máximo para para a configuração do FBus – é muito elevada para o microcontrolador, e provoca um mal funcionamento, em virtude de buffer overruns. É possível configurar o controlador para ignorar esses erros. No entanto, para evitar transtornos, é mais seguro reduzir a taxa de transmissão para valores mais baixos, ou utilizar estritamente a linhas de transmissão para MBus (cujas taxa de transmissão é 9600 bps em configuração padrão).

## 4.2 Organização da biblioteca

Podemos definir a arquitetura do firmware em três camadas, como descrito na tabela 4.2.

Tabela 4.2: Camadas de firmware da plataforma

Aplicação
Transmissão
Sinalização

A primeira camada, **aplicação** contém a lógica de controle do firmware. Esta camada usa as funções definidas na camada de transmissão para enviar e receber mensagens através da rede GSM. A camada de aplicação conta com out-

ras funções utilitárias, como manipulação de periféricos de E/S (botões, LCD e LEDs).

A segunda camada, ou camada de **transmissão** possui diretivas para o encapsulamento de mensagens para o terminal serial (UART). Nesta camada, está presente a lógica de encapsulamento de conteúdo (como conversão de ASCII para PDU) e cabeçalhos da mensagem do protocolo GSM, para que possa ser transferida pelo protocolo de transmissão adequado (F/MBus ou comando AT, no caso de mensagens SMS). No entanto, todo o procedimento de encapsulamento deve ser transparente para a camada anterior.

Enquanto a camada anterior se encarrega do encapsulamento de conteúdo e cabeçalho, a camada de **sinalização** tem como papel realizar requisições e interpretar respostas. Diferentemente da camada anterior, a sinalização não se preocupa com o conteúdo das mensagens, mas de encaminhar a mensagem ao seu destino destino. No processo de sinalização, o controlador se encarrega das seguintes tarefas:

- Ler e escrever informações de sinalização da mensagem (cabeçalhos F/MBus), definir as configurações para o modem, antes que a mensagem a seja enviada (comandos AT).
- Verificar a presença de erros na mensagem através de checksum (F/MBus), ou formatação da string (comandos AT).
- Verificar e responder mensagens de confirmação enviadas pelo modem (F/MBus).
- Habilitar ou Desabilitar interrupções do microcontrolador.
- Enviar, receber vetores de dados através da porta serial.

## 4.3 Descrição de funções e características da biblioteca

As camadas de transmissão e sinalização são representada por uma API (*Application Programming Interface*) – um conjunto de funções e estruturas de dados que fornecem uma interface para algum tipo de funcionalidade. Estão também presentes na API interfaces para operação de periféricos da plataforma.

### 4.3.1 API para SMS

A seguir serão descritas um conjunto de variáveis, funções e estruturas de dados importantes para a camada de transmissão, em formato de lista, que inclui informações sobre a assinatura da função, estrutura ou variável relacionada, uma descrição de seus parâmetros e a descrição de seu uso. Se as assinaturas das funções começarem com o prefixo “at\_”, estão restritas ao envio de comandos AT. Caso o prefixo seja “fb\_”, são mensagens que podem ser transmitidas através dos protocolos FBus e MBus da nokia.

Houve problemas na confecção do hardware, por isso as funções relativas a FBus não foram testadas, e aquelas referentes a comandos AT foram testadas fora do ambiente da plataforma.

- int PDU\_SMSC:

**Descrição:**

Variável global que guarda o número do SMSC (agrupado em octetos) usado para transmissões SMS. Durante a inicialização da plataforma, esse número precisa ser ativado para que seja possível a transmissão de mensagens SMS.

- void set\_smsc(char\* smsc\_string)

**Parâmetros:**

`smc_string` – string em formato ASCII que representa o SMSC correspondente

**Descrição:**

Configura a variável global `PDU_SMSC` para o número do `smc` passado como parâmetro. Realiza a conversão da string para o formato PDU.

- `int at_send_sms(char* message, char* number):`

**Descrição:**

Transmite a mensagem para o número solicitado através de comandos AT para o modem. Retorna 0 em caso de sucesso.

- `int at_stored_sms():`

**Descrição:**

Retorna o valor que representa a quantidade de mensagens armazenadas no SIM card.

- `int at_read_sms( int index ):`

**Parâmetros:**

`index` – valor do índice da mensagem armazenada no SIM card

**Descrição:**

Armazena na variável global `MESSAGE`, a mensagem de posição *index* da memória do SIM card. Retorna 0 tenha sucesso.

- `void fb_send_sms(char* message, char* number):`

**Parâmetros:**

`message` – string de mensagem (em ASCII)

`number` – string correspondente ao número do telefone chamado (em ASCII).

**Descrição:**

Envia a mensagem em formato legível para o protocolo Nokia F/MBus, de forma similar à função `at_send_message`.

- `int fb_read_sms()`

**Descrição:**

Essa função deve chamada durante a recepção de dados pela interface serial. Armazena a mensagem recebida na variável global `MESSAGE`. Retorna 0 caso tenha sucesso.

- `struct sms_t decode_sms()`

**Descrição:**

Converte a mensagem armazenada em `MESSAGE` para uma `struct sms_t`

- `struct sms_t`

**Descrição:**

Estrutura que representa o corpo de uma mensagem sms. Os membros `m_number`, `m_time` e `m_message`, representam o número do destinatário, a hora em que a mensagem foi recebida, e a mensagem em si, todos em formato PDU.

- `int8 MESSAGE[BUFSIZE]`

**Descrição:**

Buffer para recepção de mensagens SMS. `BUFSIZE` é um macro com valor padrão 30.

- `int pdu_to_ascii(int8* pdu, int8* ascii, int pdu_size, int ascii_size)`

**Parâmetros:**

`pdu` – array de octetos em formato pdu.

`ascii` – array de caracteres ascii vazio.

`pdu_size` – tamanho do array pdu.

ascii\_size – tamanho do array ascii.

**Descrição:**

Converte valores de um array de pdu para uma string ascii. O tamanho do array em ascii deve ser maior que o de pdu em uma proporção de 8 para 7, arredondado para baixo, ou acarretará em erro. Retorna 0 em caso de sucesso.

As funções a seguir se referem ao processo de sinalização da mensagem através da interface rs232:

- int at\_config\_sms( int mode )

**Parâmetros:**

mode – O valor 0 configura o modem para usar o modo PDU. O valor 1, para modo texto.

**Descrição:**

Configura o modem para operar em modo PDU ou texto. É chamada em at\_sms\_send() e, neste caso, está configurado para operar em modo pdu por padrão.

- int at\_send\_command( char\* pdu, int length, char\* command, char\* response)

**Parâmetros:**

pdu – array de octetos em formato PDU.

length – comprimento do array de PDU.

command – comando at em formato ASCII

response – resposta obtida do comando enviado, em pdu.

**Descrição:**

Realiza a conversão do pdu em string ASCII (por exemplo se o valor 0xAF, converte para dois caracteres, 'A' e 'F' em ASCII), cria o corpo da mensagem, por exemplo AT+CMGS=14<CR>000103...<Ctrl-z> e envia o comando através do modem. O corpo da resposta é passada para a variável response, após a extração dos cabeçalhos do comando AT.

- char get\_ascii\_char ( char ch, int part )

**Parâmetros:**

ch – octeto em pdu.

part – 0 representa a parte baixa do octeto e 1 a parte alta.

**Descrição**

Função auxiliar que converte porções do octeto em ASCII.

- int fb\_decode\_msg ( char\* message )

**Parâmetros**

message – A mensagem que será extraída da serial.

**Descrição:**

Essa função é invocada todas as vezes que recebe uma resposta através da porta serial. Por padrão, retorna 0 quando o tipo da mensagem for um SMS. A string message será extraída da porção de conteúdo da mensagem fbus. Essa função realiza o checksum da mensagem recebida.

### 4.3.2 Funções de utilidade

Estas funções são usadas para manipulação de periféricos da plataforma. Por restrições de tempo do projeto, os módulos da API para escrita no monitor LCD não foram criados. Não há módulo para comunicação com dispositivos externos, já que a comunicação com outros dispositivos pode ter características bastante divergentes do que foi proposto neste projeto.

- void LED\_X\_ON(), LED\_X\_OFF()

**Descrição:**

Essas funções são macros para operação dos LEDs. X é um número que vai de 1 a 3.

- int BTN\_X\_PRESSED()

**Descrição:**

Macros que verificam se botões estão pressionados. Como na função anterior, X varia de 1 a 3.

- int usb\_msg\_receive( char\* message, int length )

**Parâmetros:**

message – buffer para por o conteúdo extraído da porta usb

length – comprimento máximo do buffer.

**Descrição:**

Verifica se usb foi enumerado e copia os dados obtidos da porta USB para message. Realiza a verificação de comprimento do buffer, para evitar que seu tamanho seja excedido. Retorna a quantidade de bytes lidos ou -1 caso tenha acontecido algum erro.

- int usb\_msg\_send( char\* message, int length )

**Parâmetros:**

message – buffer para por o conteúdo a ser enviado através da porta usb

length – comprimento máximo do buffer.

**Descrição:**

Faz o processo inverso da função anterior, enviando ao invés de receber.

## 5 *Resultados e testes*

Neste capítulo discutiremos sobre os resultados que foram alcançados com este trabalho, com imagens do hardware construído. A última seção inclui alguns testes realizados com a plataforma.

### 5.1 **Hardware confeccionado**

A plataforma foi construída sob a forma de uma placa de circuito impresso, utilizando o método de transferência térmica. Foram usados os seguintes componentes:

- Monitor LCD 16x2
- Conector USB tipo B
- Conector RJ-12
- Quatro chaves tácteis.
- Conector P4
- Oscilador cerâmico de 20MHz.
- 13 Resistores comuns, sendo seis de 470  $\Omega$ , três de 1 K $\Omega$ , dois de 15  $\Omega$ , um de 10 K $\Omega$  e um de 47  $\Omega$ .
- 6 capacitores cerâmicos, sendo dois de 100 nF, dois de 15 pF, um de 220 nF e um de 330 nF.

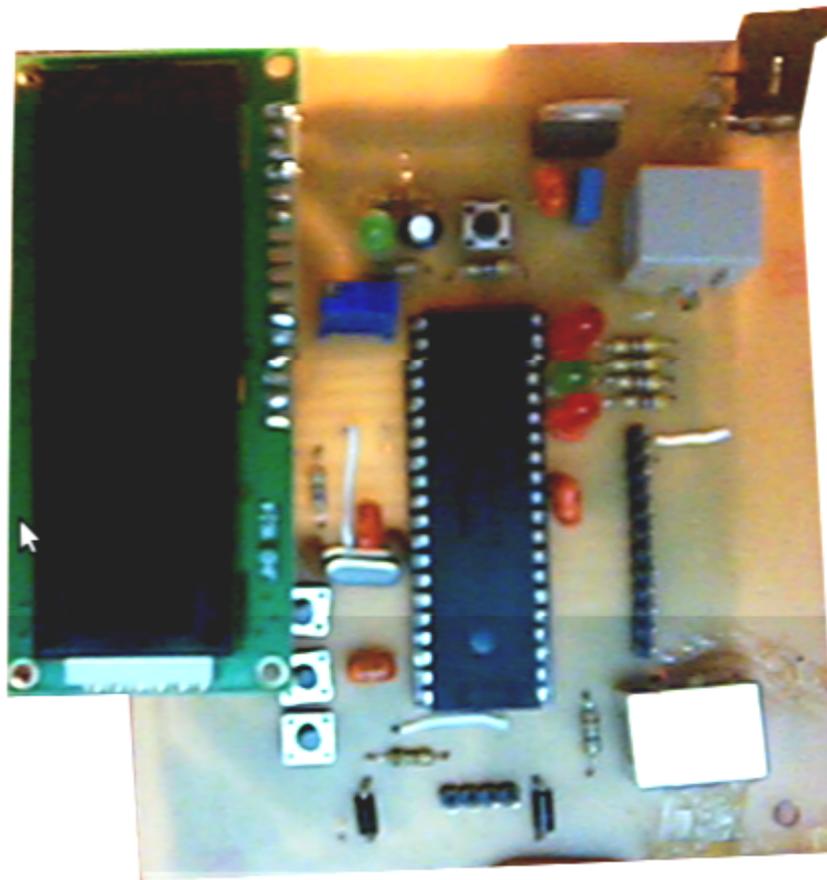


Figura 5.1: Imagem da placa de circuito impresso

- 1 capacitor eletrolítico de 1  $\mu$ F.
- 4 LEDs comuns
- Regulador de tensão 7805
- 2 diodos 1N4007
- 3 barras de pino

A figura 5.1 mostra uma imagem da placa confeccionada.

As linhas de transmissão MBus, Tx, Rx e GND do nokia 2720 foram acopladas a um conector molex. Foi possível conservar a bateria de íon-lítio no aparelho, removendo a necessidade de usar um circuito de alimentação para o telefone. A figura 5.2 ilustra a conexão realizada.



Figura 5.2: Conexão de linhas de transmissão do aparelho celular (Nokia 2720)

Não foi possível realizar a interface entre celular e plataforma, pois houve várias dificuldades. Apesar do nível de tensão ter sido convertido para um valor aceitável, não houve resposta do celular na linha de recepção do microcontrolador. Uma hipótese que justifique este comportamento é a possibilidade de que a requisição realizada não estivesse sendo decodificada corretamente e, por isso, não houvera resposta. Segundo Gnokii (2010), é possível haver variações no cabeçalho da mensagem FBus de acordo com o modelo de celular Nokia utilizado. Não há documentação, seja por parte da Nokia ou Gnokii, que justifique essa afirmação, contudo.

## 5.2 Testes realizados

Foram feitos testes para saber que funcionalidades da plataforma estavam operacionais. Os testes realizados foram:

- Piscar LEDs, para testar a operação desses módulos.

- Acender LEDs pressionando botões, para testar o funcionamento dos terminais de entrada.
- Acender e apagar LED através de comandos pela porta USB, para testar envio.
- Receber string de porta USB para testar a recepção.

Através de computador foi escrito um software que realiza o envio e recebimento de SMS através de funções similares às encontradas na biblioteca. Como as tarefas de envio e recepção de SMS eram simples, não houve problemas durante a operação.

## 6 *Conclusão*

Este trabalho propôs a construção de uma plataforma que pudesse ser usada para enviar dados através da rede GSM. Foi realizado projeto de software e hardware a partir de componentes encontrados facilmente no mercado local. Alguns testes foram realizados para verificar o funcionamento dos módulos de hardware. A comunicação pela plataforma através da rede GSM não foi realizada com sucesso, embora tenha sido demonstrado em software para PC que através de comandos AT e um modem é possível enviar e receber mensagens SMS.

Houve várias dificuldades no projeto da plataforma. Um primeiro protótipo possuía vários erros, alguns de projeto e outros do processo de fabricação. O hardware da plataforma foi construído em sua segunda tentativa.

Um dos desafios do projeto foi utilizar um celular como modem para acessar a rede GPRS. Como não havia cabos para a conexão entre celular e plataforma, foi necessária a criação de uma interface elétrica para operação através da porta serial. Embora houvesse uma documentação não-oficial na internet de como proceder na criação da interface, havia várias lacunas na literatura encontrada, especialmente quanto às capacidades do dispositivo celular utilizado no projeto. Esse fato mostra que o uso de modems externos voltados para aplicações GSM/GPRS ainda são a melhor alternativa, mesmo pela dificuldade de encontrá-los no mercado local ou a preços acessíveis.

Para dar continuidade ao desenvolvimento da plataforma proposta e fazer

com que ela atinja a meta proposta inicialmente, é necessário investir esforço na integração da plataforma com algum modem externo. Se essa meta for alcançada, teremos a disposição da academia e mercado um poderoso artefato, para uso em aplicações de telemetria e monitoramento.

## *Referências Bibliográficas*

BODIC, Gwenaël Le. *Mobile Messaging Technologies and Services*. 2nd. ed. [S.l.]: John Wiley & Sons, Ltd, 2005.

GNOKII. *gnokii*. 2010. URL: <http://www.gnokii.org/>. Visitado em: 20/11/2010.

HEINE, Gunnar. *GSM Networks: Protocols, Terminology and Implementation*. 1st. ed. [S.l.]: Artech House, 1999.

HOME, Developer's. *AT Command Operations: Test, Read, Set and Execution*. 2010. URL: <http://www.developershome.com/sms/atCommandOp.asp>. Visitado em: 20/11/2010.

MICROCHIP. *PIC18F2455/2550/4455/4550 Datasheet: 28/40/44-Pin, High Performance, Enhanced Flash, USB Microcontrollers with nanoWatt Technology*. [S.l.], 2007. 428 p.

MULTITECHSYSTEMS. *AT Commands: GSM Reference Guide*. [S.l.], 2003. 244 p.

OLIVEIRA, Sergio Campello; FONTANA, Eduardo. Optical detection of partial discharges on insulator strings of high voltage transmission lines. *IEEE Transactions on Instrumental and Measurement*, v. 58, p. 2328–2334, 2009.

SEURRE, Emmanuel; SAVELLI, Patrick; PIETRI, Pierre-Jean. *GPRS for Mobile Internet*. 1st. ed. [S.l.]: Artech House, 2003.