

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA  
ENGENHARIA INDUSTRIAL ELÉTRICA:  
ÊNFASE EM ELETRÔNICA E TELECOMUNICAÇÕES**

**FABIAN ERIK FINK  
JOÃO LUCAS RONZANI  
VINICIUS CARDOSO DURGIEWICZ  
WEBERTON SHIGUEYUKI NATSUAKI**

**SISTEMA DE AMOSTRAGEM PARA CÁLCULO DE  
HORÁRIO DE TRANSPORTE PÚBLICO**

**TRBALHO DE CONCLUSÃO DE CURSO**

**CURITIBA 2013**

**FABIAN ERIK FINK  
JOÃO LUCAS RONZANI  
VINICIUS CARDOSO DURGIEWICZ  
WEBERTON SHIGUEYUKI NATSUAKI**

**SISTEMA DE AMOSTRAGEM PARA CÁLCULO DE  
HORÁRIO DE TRANSPORTE PÚBLICO**

Trabalho de conclusão de curso de graduação apresentado à disciplina de Trabalho de Conclusão de Curso 2, do Curso Superior de Engenharia Industrial Elétrica Ênfase em Eletrônica e Telecomunicações do Departamento Acadêmico de Eletrônica – DAELN – da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Engenheiro.

Orientador: Rubens Alexandre de Faria

**CURITIBA 2013**

**FABIAN ERIK FINK**

**JOÃO LUCAS RONZANI**

**VINICIUS CARDOSO DURGIEWICZ**

**WEBERTON SHIGUEYUKI NATSUAKI**

**SACH:  
Sistema de Amostragem e Cálculo de Horários**

Este Trabalho de Conclusão de Curso foi julgado e aprovado como requisito parcial para obtenção do título de Engenheiro Industrial Eletricista com ênfase em Eletrônica e Telecomunicações pela Universidade Tecnológica Federal do Paraná.

Curitiba, de de 2013.

---

Prof. Dr. Hilton José da Silva de Azevedo  
Coordenador de Curso  
Departamento Acadêmico de Eletrônica

---

Prof. Dr. Dario Eduardo Amaral Dergint  
Responsável pelo Trabalho de Conclusão de Curso  
Departamento Acadêmico de Eletrônica

**BANCA EXAMINADORA**

---

Prof. M.Sc. Daniel Rossato

---

Prof. Dr. Rubens Alexandre de Faria  
Orientador

---

Prof. Dr. Kleber Kendy Horikawa Nabas

## RESUMO

Este trabalho apresenta um sistema de cálculo e amostragem, de forma aproximada, do horário de chegada e saída de transportes públicos. O projeto, iniciado através da observação das falhas do sistema de transporte público, tem como foco o auxílio ao usuário do transporte público coletivo para sua melhor utilização. Estão contempladas neste documento todas as etapas do desenvolvimento desde o estudo do problema em questão e das tecnologias a serem utilizadas, passando pela integração dos módulos do projeto até seu funcionamento e testes. O sistema pronto integrará completamente o usuário do transporte público, com o mais próximo possível do horário de chegada e saída do mesmo, aumentando a sua eficiência.

**Palavras-chave:** Transporte Público. Amostragem. Cálculo de Horário.

## **ABSTRACT**

This work presents a sampling and calculation system, to estimate the arrival and boarding time of the public transportation vehicles. The project, started by the observation of the critical aspects of the public transportation system, aim to provide a more effective utilization for the public transportation users. This document covers all development steps, since the study of the problem and the technologies to be used, through the integration of the modules and finally to its operation and tests. The final product will integrate completely, user and the public transportation system, providing the closest arrival and boarding time, improving its efficiency.

**Keywords:** Public Transportation. Sampling. Time Calculation.

## LISTA DE FIGURAS

Figura 1 – Funcionamento do sistema proposto .....	11
Figura 2 - Constelação de Satélites. ....	15
Figura 3 - Mapa das estações de monitoramento. ....	15
Figura 4 – Exemplo de DER.....	18
Figura 5 – Entidade .....	19
Figura 6 - Representação de atributos .....	20
Figura 7 - Exemplo de relacionamento.....	20
Figura 8 - Representação de participação mínima e máxima (min,máx) de E em R. ....	21
Figura 9 - Modelo Cliente/Servidor.....	23
Figura 10 - Diagrama do Banco de Dados. ....	32
Figura 11 - Interface do usuário. ....	36
Figura 12 – Módulo Jupiter JN3 e sua pinagem.....	38
Figura 13 - Configuração dos pinos do módulo GL865 QUAD.....	40
Figura 14 - Exemplo de fonte chaveada com saída regulável para o LM2596.....	41
Figura 15 - Exemplo de utilização do módulo Jupiter JN3 .....	42
Figura 16 – Placa base do microcontrolador Cortex-M4. ....	44
Figura 17 - Parte Frontal da PCB desenvolvida .....	45
Figura 18 - Parte traseira da PCB desenvolvida .....	45
Figura 19 – Escolha da linha de ônibus. ....	47
Figura 20 - Escolha da Estação desejada.....	47
Figura 21 - Página mostrando botão para executar cálculo de tempo. ....	48
Figura 22 - Tempo disponibilizado para o usuário.....	49
Figura 25 – Face superior da placa de circuito impresso com placa STMF32F4 acoplada.....	58
Figura 26 - Face inferior da placa de Circuito impresso. ....	58

## LISTA DE TABELAS

Tabela 1 - Estrutura padrão de uma sentença .....	16
Tabela 2 - Principais elementos da notação Chen.....	21
Tabela 3 - Tabelas do Banco de Dados.....	33

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>10</b>
1.1	JUSTIFICATIVA E MOTIVAÇÃO	10
1.2	OBJETIVOS	11
1.3	ESCOPO	12
1.3.1	Módulo do Ônibus	12
1.3.2	Banco de Dados	12
1.3.3	Interface ao usuário	12
1.4	METODOLOGIA	13
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>14</b>
2.1	GLOBAL POSITIONING SYSTEM (GPS)	14
2.1.1	Segmento Espacial	14
2.1.2	Segmento de Controle	15
2.1.3	Segmento de Usuário	16
2.2	PROTOCOLO NMEA 0813	16
2.3	BASE DE DADOS	17
2.3.1	Modelo Entidade-Relacionamento	17
2.3.2	Entidades	18
2.3.3	Atributos	19
2.3.4	Relacionamentos	20
2.3.5	Cardinalidade	21
2.4	HTTP ( <i>Hypertext Transfer Protocol</i> )	22
2.5	API GOOGLE MAPS	24
2.5.1	Regras Gerais:	25
2.5.2	Exceções:	25
2.5.3	Exemplos:	25
2.6	GENERAL PACKET RADIO SERVICE (GPRS)	26
2.7	GLOBAL SYSTEM FOR MOBILE COMMUNICATIONS (GSM)	28
<b>3</b>	<b>DESENVOLVIMENTO</b>	<b>29</b>
3.1	SERVIDOR	29
3.1.1	Apache	29
3.1.2	PHP	30
3.1.2.1.	phpMyAdmin	30
3.2	BANCO DE DADOS	30
3.3	DYNAMIC DNS	34



3.4	WEB SERVICE .....	35
3.5	DESENVOLVIMENTO de HARDWARE.....	36
3.5.1	Módulo GPS.....	36
3.5.2	Módulo GPRS.....	38
3.5.3	Fonte de Alimentação .....	40
3.5.4	Radio Frequência em placas de circuito impresso.....	41
3.5.5	Microcontrolador Cortex-M4 STM32F4.....	43
3.5.6	Placa de circuito impresso .....	44
<b>4</b>	<b>RESULTADOS OBTIDOS .....</b>	<b>46</b>
4.1	TESTES REALIZADOS EM CAMPO .....	46
4.2	CONSIDERAÇÕES.....	50
<b>5</b>	<b>CONCLUSÃO .....</b>	<b>51</b>
	<b>REFERÊNCIAS.....</b>	<b>53</b>
	<b>APÊNDICES .....</b>	<b>55</b>

## 1 INTRODUÇÃO

O Sistema de Amostragem para Cálculo de Horário de Transporte Público (SACH) é um sistema que visa aumentar a interação entre o usuário e o sistema de transporte público de uma cidade, visando aperfeiçoar e também evitar o desperdício de tempo por parte da população que depende deste sistema.

O sistema explora a tecnologia de navegação *Global Positioning System* (GPS), utilizando os sinais de posicionamento via satélite, a qual foi agregada através de um módulo ao *hardware* desenvolvido para o produto. O sistema apresenta um *hardware* que consiste em uma base microcontrolada na qual opera o *software*, um banco de dados para armazenamento e cálculo de tempo estimado para chegada do ônibus ao ponto. Tudo isso disposto em uma interface ao usuário para facilidade de acesso.

### 1.1 JUSTIFICATIVA E MOTIVAÇÃO

Com o crescimento horizontal das grandes metrópoles, a integração total da população com o transporte público é fundamental. Um estudo divulgado pelo Instituto de Pesquisa Econômica Aplicada (IPEA) aponta que 65% da população das capitais usam transporte público para se deslocar. Sob algumas condições, 71% dos brasileiros aceitariam deixar o carro ou a moto na garagem e utilizar transporte público. A maioria, 20,8%, usaria o transporte coletivo se fosse mais rápido. Há ainda hoje, uma deficiência no conhecimento de horários exatos de chegada e saída dos meios de transporte, por isso a necessidade da construção de um sistema que atenda a população nesse aspecto (IPEA, 2011).

Com base nas estatísticas mencionadas e aproveitando a crescente demanda por dispositivos eletrônicos portáteis, tais como *smartphones*, *tablets* e *palmtops*, e paralelamente, o rápido desenvolvimento de tecnologias de comunicação móvel como o 3G e *Wi-fi*, observa-se a viabilidade da introdução de um sistema que integre informações sobre o sistema de transporte público e

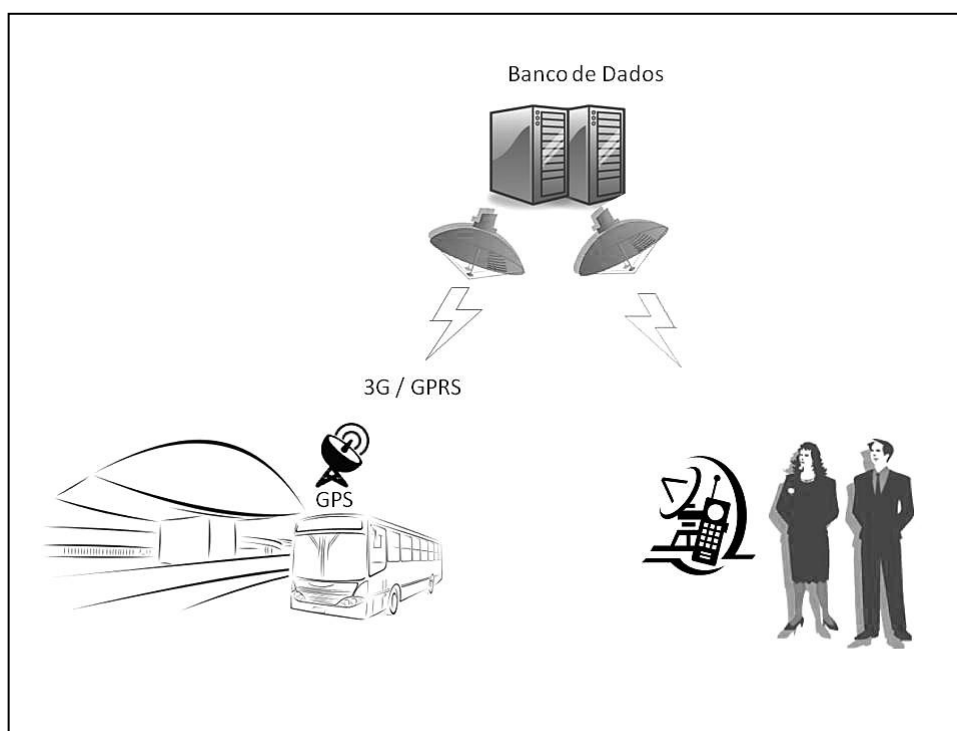
disponibilize aos usuários de maneira flexível, explorando toda a comodidade que as tecnologias móveis podem proporcionar.

## 1.2 OBJETIVOS

O objetivo do projeto é desenvolver um sistema que amostra e calcule, de forma aproximada, o horário de chegada e saída de transportes públicos.

Para o desenvolvimento do sistema, foi necessário estudar o transporte público de uma cidade (nesse caso Curitiba). O sistema consistirá de três módulos: O módulo GPS, *hardware* e *firmware*. Estes são integrados ao transporte onde farão aquisição de dados; que serão enviados para o banco de dados e tratados por *software*, que além de fazer os cálculos, irá disponibilizá-los ao usuário.

O sistema pronto integrará completamente o usuário do transporte público ao possível horário de chegada e saída do mesmo, aumentando a sua eficiência. O esquema mostrado na figura 1 ilustra, de uma maneira ampla, o funcionamento do sistema.



**Figura 1 – Funcionamento do sistema proposto**  
**Fonte: Autoria própria**

## 1.3 ESCOPO

### 1.3.1 Módulo do Ônibus

- Módulo GPS;
- Recebe os dados do satélite e envia para o microcontrolador a posição encontrada;
- Microcontrolador;
- Controlador faz todas as rotinas de recebimento de dados do satélite, tratamento de dados e preparação para o envio pelo módulo 3G;
- Módulo 3G;
- Recebe os dados do microcontrolador e envia pela rede de dados até o banco de dados;

### 1.3.2 Banco de Dados

- Computador que armazena os dados recebidos pelo módulo 3G, e as partes que compõe o banco de dados;
- *Software* de interface ao administrador do sistema;
- *Software* para administração e manutenção do sistema;
- Banco de dados;
- Banco de dados que armazena as amostras enviadas pelo módulo do ônibus.

### 1.3.3 Interface ao usuário

- Aplicação *Web* construída em PHP (*Hypertext Preprocessor*);
- Interface para usuário acessar as informações disponibilizadas.

#### 1.4 METODOLOGIA

##### **1ª Etapa:** Estudar a viabilidade do projeto

Etapa na qual foi levantada toda a informação relevante sobre o sistema de transporte público em questão, no caso Curitiba, para que se torne possível obter uma visão sobre o desempenho do SACH e sua viabilidade.

##### **2ª Etapa:** Estudo do problema

Para compreensão das necessidades de projeto foi estudado o funcionamento do módulo GPS para aquisição dos dados a serem processados, assim como uma maneira para a transmissão dos mesmos para o banco de dados através de uma tecnologia de transmissão de dados wireless mais comum no mercado (GPRS ou 3G).

##### **3ª Etapa:** Implementação e teste das seções do projeto

Nesta etapa as ações do projeto foram tratadas independentemente. Assim foram desenvolvidas as seções de *hardware*, *firmware* e *software*. Em todos os segmentos do projeto foram especificados os requisitos para um desenvolvimento mais objetivo, organizado e eficiente.

Na parte de *hardware*, primeiramente, foi projetado o circuito esquemático, em seguida confeccionadas as placas para a montagem para, enfim fazer as ligações dos componentes.

Para o *firmware*, foram estudados os *datasheets* e uma linguagem apropriada para desenvolvimento, para a programação do microcontrolador. Na parte de *software*, foi implementado o código para tratamento dos dados e formulação do banco de dados.

##### **4ª Etapa:** Integração das seções do projeto

Após todas as etapas do projeto concluídas, foi realizada a integração das seções desenvolvidas independentemente, previamente testadas.

##### **5ª Etapa:** Teste da funcionalidade do sistema

Após integrar as seções do projeto, foram realizados mais testes para verificar se o sistema está funcionando da maneira proposta, e no caso de possíveis erros, estes serão devidamente tratados e corrigidos.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo será mostrada toda a fundamentação teórica envolvida no projeto, detalhando todos os aspectos relevantes para a aplicação no projeto.

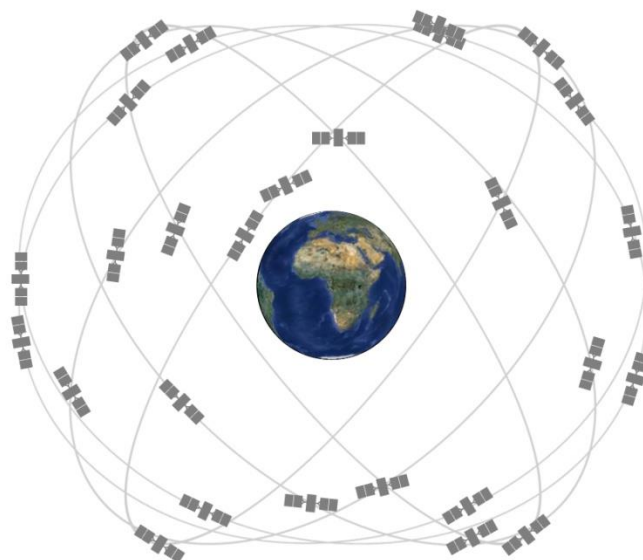
### 2.1 GLOBAL POSITIONING SYSTEM (GPS)

O *GPS* é um sistema de navegação via satélite de uma direção que fornece dados para localização, navegação e hora. Esse sistema é constituído de três segmentos: o segmento espacial, o segmento de controle e o segmento de usuário (The U.S. Government website: GPS.gov, 2013).

#### 2.1.1 Segmento Espacial

O segmento espacial consiste em uma constelação de satélites emitindo sinais de rádio para os usuários. Segundo a força aérea norte-americana, existem pelo menos 24 satélites disponíveis em 95% do tempo.

Os satélites se posicionam na órbita terrestre média – MEO na sigla em inglês- a uma altitude de aproximadamente 20200 km. Cada satélite dá uma volta completa na terra, cerca de duas vezes por dia. Esses satélites são arranjados em seis planos orbitais espaçados igualmente, onde cada plano possui quatro *slots* que são ocupados pelos satélites. Esse arranjo garante que há pelo menos quatro satélites visíveis virtualmente em cada ponto do planeta (*The U.S. Government website: GPS.gov, 2013*).

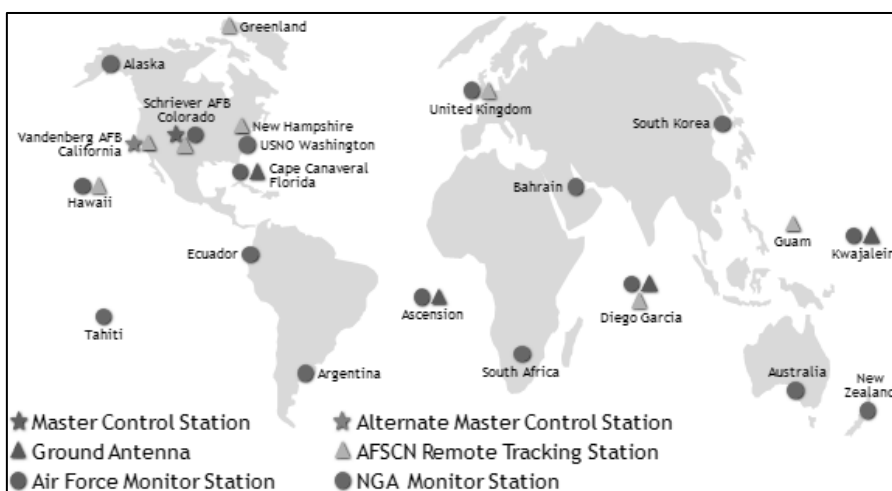


**Figura 2 - Constelação de Satélites.**  
 Fonte: The U.S. Government website: GPS.gov (2013)

### 2.1.2 Segmento de Controle

O segmento de controle consiste de vários locais que rastreiam os satélites, monitoram suas transmissões, realizam análises e enviam comandos e dados para as constelações.

O segmento de controle atual inclui uma estação de controle central, uma estação de controle central alternativa, 12 antenas e 16 locais de monitoramento (The U.S. Government website: GPS.gov, 2013).



**Figura 3 - Mapa das estações de monitoramento.**  
 Fonte: The U.S. Government website: GPS.gov (2013)

### 2.1.3 Segmento de Usuário

O segmento de usuário é constituído dos aparelhos que recebem os sinais do satélite. Esses aparelhos fazem os cálculos necessários para o posicionamento e trazem outras funcionalidades como mapas e navegação por voz (*The U.S. Government website: GPS.gov, 2013*).

## 2.2 PROTOCOLO NMEA 0813

Protocolo desenvolvido pela *National Marine Electronics Association* (NMEA) para estabelecer uma série de especificações a fim de obter uma interface padronizada entre os equipamentos eletrônicos utilizados para navegação marítima. O sistema GPS utiliza esse protocolo para transmissão dos dados para se determinar o posicionamento dos receptores GPS (Joe Mehaffey, 2012).

O protocolo caracteriza as mensagens, chamadas de sentenças, enviadas durante a comunicação. Essas sentenças consistem em um link serial transmitido a uma taxa de 4800 bits/segundo. A seguir são descritas as principais regras de formatação das sentenças (Joe Mehaffey, 2012):

- Todas as sentenças iniciam com \$;
- Em seguida é composta de cinco caracteres;
- Todos os campos de dados da sentença são separados por vírgula;

**Tabela 1 - Estrutura padrão de uma sentença**

Taxa de Transmissão (bits/Segundo)	Número de Bits de Dados	Paridade	Stop Bits	Handshake
4800	8	Nenhum	1	Nenhum

Fonte: Autoria própria



## 2.3 BASE DE DADOS

Uma base de dados é a coleção organizada de dados, usualmente armazenada em forma digital. Na década de 70 surgiram os primeiros sistemas gerenciadores de bases de dados, os SGBDs (ELMAZARI-NAVATHE, 2003). Anteriormente, as aplicações utilizavam o próprio sistema de arquivos do sistema operacional para organizar suas informações. Tal organização e eficiência na recuperação desses dados eram altamente dependentes do desenvolvedor da aplicação.

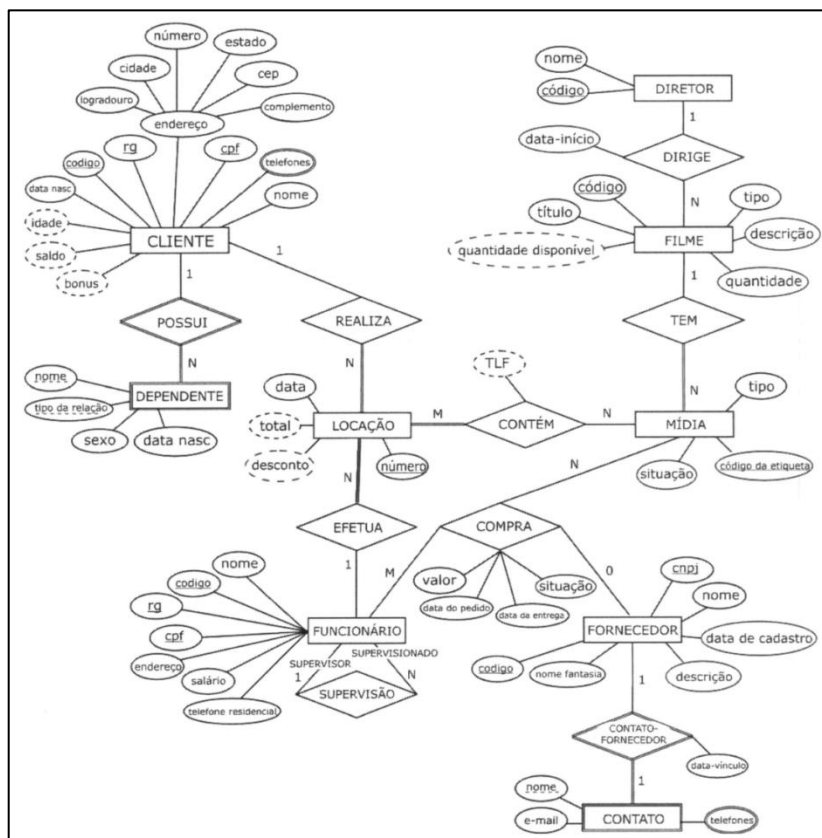
Com o aumento do volume de dados em aplicações específicas, como sistemas bancários e sistemas empresariais, fez-se necessária a criação dos SGBDs. Alguns dos principais sistemas atualmente são Oracle, Microsoft SQL Server, PostgreSQL, MySQL e SQLite. Para administrar tais sistemas, a linguagem SQL é largamente difundida (ELMAZARI-NAVATHE, 2003).

### 2.3.1 Modelo Entidade-Relacionamento

O modelo entidade-relacionamento é uma abstração com a finalidade de descrever conceitualmente os dados pertencentes a um domínio. O diagrama entidade-relacionamento (DER) é a maneira gráfica de se representar este modelo. Esta notação é largamente utilizada na modelagem de bancos de dados (CHEN, 1976). Outras notações que merecem destaque são (ELMAZARI-NAVATHE, 2003):

- IDEF1X;
- Notação de Martin;
- Notação de Bachman;
- Notação de Setzer;
- Uso da UML para representar modelos de dados não-orientados a objetos.

O objetivo de Chen na criação dessa notação foi modelar os elementos do mundo real e representá-los da maneira mais clara possível. A figura 4 ilustra um exemplo de DER:



**Figura 4 – Exemplo de DER**  
Fonte: UNESP, 2012.

Para o entendimento da DER, é necessário o conhecimento das simbologias e elementos que o compõem. A seguir, serão descritas as mesmas.

### 2.3.2 Entidades

É o conjunto de elementos com características próprias. Seus atributos não dependem de outras entidades (CHEN, 1976). Podem ser compreendidos como os substantivos na língua portuguesa. Ela pode ser algo concreto (mesa, ônibus), ou até mesmo algo abstrato (mentira, pensamento). Exemplos de entidades: objetos, locais, pessoas e etc.

A representação de uma entidade é feita através de um retângulo e um texto interno nomeando a entidade, conforme a Figura 5.



**Figura 5 – Entidade**  
**Fonte: Autoria própria**

### 2.3.3 Atributos

Os atributos modelam as características de uma entidade ou relacionamento. Os atributos de uma entidade não dependem de nenhuma outra entidade. Os atributos podem ser do tipo: determinante, composto ou multi-valorado (CHEN, 1976).

Um atributo determinante é aquele que identifica de forma única uma entidade, não podendo haver repetições. Os atributos determinantes serão as chaves primárias no banco de dados. Ele é identificado pelo seu texto com formatação sublinhada. Exemplos: CPF, CNPJ, Título de Eleitor.

Atributos compostos são aqueles que podem ser formados por vários itens menores. Um excelente exemplo é a entidade “Endereço”: seu conteúdo poderá ser dividido em vários outros atributos como, logradouro, CEP, cidade, estado.

Por fim, atributos multivalorados, são aqueles que podem assumir mais de um valor para determinado elemento de uma entidade. Por exemplo, para uma entidade “Consumidor” poderá haver zero ou mais endereços de e-mail cadastrados. Ou seja, possui uma relação 1 para N com a entidade. É indicado colocando-se um asterisco precedendo o nome do atributo. Na Figura 6, são exemplificadas suas notações.

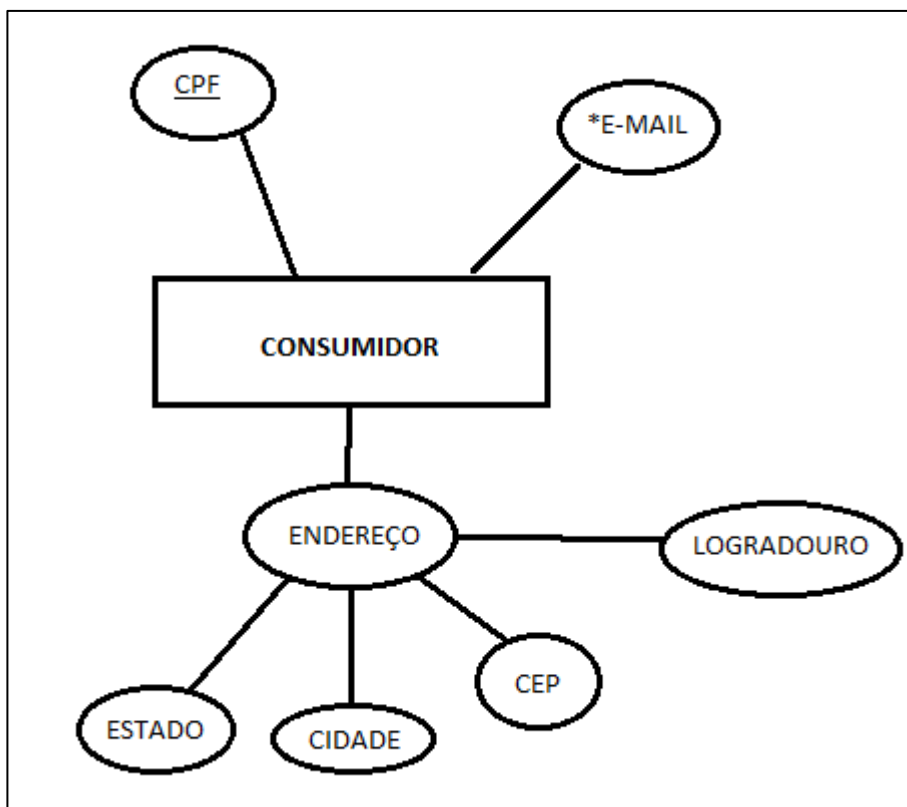


Figura 6 - Representação de atributos  
Fonte: Autoria própria

#### 2.3.4 Relacionamentos

Relacionamentos são associações entre entidades, ou até mesmo entre uma única entidade (Auto-Relacionamento). Um exemplo de relacionamento é demonstrado a seguir: um fornecedor é vinculado a um produto através do relacionamento de venda. É representado por meio de um losango, como ilustrado na Figura 7.



Figura 7 - Exemplo de relacionamento  
Fonte: Autoria própria

### 2.3.5 Cardinalidade

Identifica quantas vezes cada elemento de uma entidade pode participar de um relacionamento. No exemplo anterior, cada fornecedor pode vender N produtos, mas cada produto pode ser vendido por apenas um fornecedor, caracterizando uma relação 1 para N. Outra variação mais precisa para esta notação está representada na Figura 8, na qual são especificadas a participação mínima e máxima (min,máx) de E em R. Deve-se atentar que a ordem de leitura da cardinalidade passa a ser invertida em relação à notação anterior.

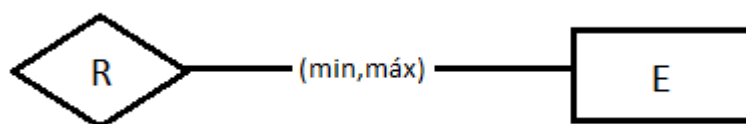

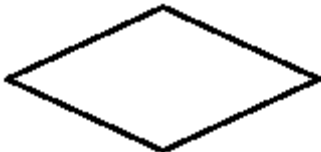


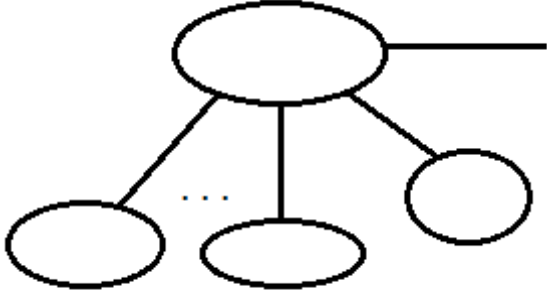
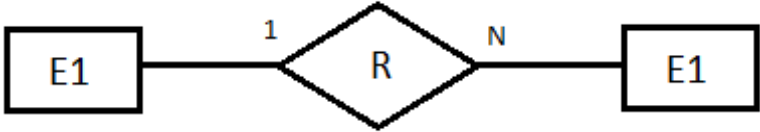
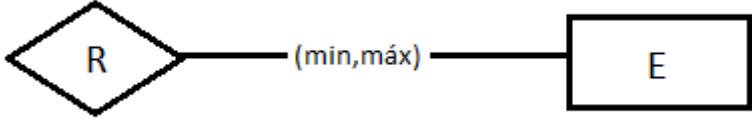


Figura 8 - Representação de participação mínima e máxima (min,máx) de E em R.  
Fonte: Autoria própria

A tabela 2 relaciona os principais elementos da notação Chen.

Tabela 2 - Principais elementos da notação Chen.

Símbolo	Significado
	Entidade
	Relacionamento
	Atributo

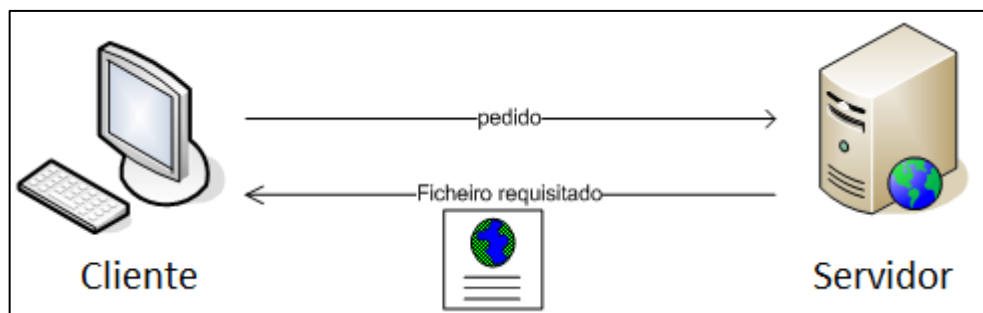
	<p>Atibuto chave</p>
	<p>Atributo composto</p>
	<p>Cardinalidade 1:N para E1:E2 em R</p>
	<p>Representação de participação mínima e máxima (min,máx) de E em R</p>

Fonte: Autoria própria

## 2.4 HTTP (*HYPertext TRAnSFER PROTOCOL*)

O HTTP ou protocolo de transferência de hipertexto, é um protocolo de rede que atua na camada de aplicação no modelo OSI. O HTTP é a base para a *World Wide Web*. A padronização desse protocolo foi coordenada pela *Internet Engineering Task Force (IETF)*, principalmente por meio da RFC 2616, de julho de 1999 (*NETWORK WORKING GROUP*, 1999).

O funcionamento do HTTP se dá por meio de um modelo “cliente-servidor” operando com um sistema de requisição e resposta. Basicamente, existe a figura do servidor, que é o elemento que hospeda as páginas *web*. Sendo o cliente, o agente que irá requisitar o conteúdo do servidor. Desta maneira, um cliente (em geral um navegador *web*) pode requisitar um conteúdo, serviço ou a execução de alguma função pelo servidor. O servidor, por sua vez, retorna uma resposta indicativa do *status* da requisição, podendo conter qualquer outro conteúdo no corpo da mensagem, conforme ilustrado na Figura 9. O HTTP é um protocolo em modo texto.



**Figura 9 - Modelo Cliente/Servidor**  
 Fonte: Autoria própria

Uma vez que o HTTP encontra-se na camada de aplicação e sendo ele concebido dentro do *framework* TCP/IP, o mesmo não possui serviços para prover confiabilidade de transmissão de dados. Sendo assim, os protocolos das camadas inferiores devem garantir esses requisitos. O HTTP presume a utilização de protocolos confiáveis na camada de transporte. Usualmente é utilizado em conjunto com o TCP.

Um cliente em geral é referenciado como um *User Agent*, ou simplesmente UA. Os recursos HTTP são identificados e localizados por meio de um *Uniform Resource Identifier (URI)* (*NETWORKING WORK GROUP*, 1998), mais especificamente, do tipo *Uniform Resource Locator (URL)* (*NETWORKING WORK GROUP*, 1994). Por exemplo, a URL <http://sach.webhop.org> indica o recurso HTML referente à página inicial do *website* do projeto SACH.

As requisições são feitas por meio dos métodos HTTP. A seguir estão apresentados os principais métodos.

- HEAD: método similar ao GET porém sem a necessidade de um corpo na mensagem;
- GET: requisição de recuso. É um método básico do HTTP;
- POST: envia dados a serem processados pelo servidor a um recurso identificado, por exemplo, o envio de dados de preenchimento de um formulário;
- PUT: faz o upload da representação de um recurso especificado.
- DELETE: remove o recurso especificado;
- TRACE: envia ao cliente a requisição recebida. Utilizado para verificação se há algum servidor intermediário manipulando as mensagens;

- OPTIONS: retorna os métodos HTTP suportados pelo servidor.

As respostas enviadas pelo servidor são conhecidas por status codes e podem ser divididas nas classes 1xx, 2xx, 3xx, 4xx e 5xx. Onde x representa um dígito entre 0 e 9. A classe 1xx indica uma resposta informacional e não é uma resposta final. A classe 2xx indica sucesso. Por exemplo, a resposta “200 OK” indica o sucesso da transação. A classe 3xx indica uma resposta de redirecionamento de recurso. As respostas 4xx indicam erros na requisição do cliente. Finalmente a classe 5xx indica que o servidor recebeu uma requisição válida, mas por um erro interno não pode completá-la.

## 2.5 API GOOGLE MAPS

Segundo Svennerberg, Gabriel (*Beggining Google Maps API 3*, 2010), o Google Maps é uma ferramenta amplamente usada para visualizar a localização de coisas, procurar um endereço, conseguir direções enquanto dirige. Aplicações que usam mapas vêm crescendo nos últimos anos e revolucionam como as informações são vistas e usadas. A API do Google Maps permite mostrar dados de uma forma mais eficiente e usável.

A natureza dinâmica do Google Maps deve ao uso de HTML, CSS e JavaScript trabalhando juntos. Os quadros do mapa são imagens carregadas usando chamadas Ajax e inseridas em uma <div> na página HTML. Conforme o usuário navega pelo mapa, informações de coordenadas e nível de zoom são enviadas pela API para o Ajax, que retorna as novas imagens a serem mostradas (*Beggining Google Maps API 3*, 2010).

A API consiste basicamente em um arquivo JavaScript que contém classes com métodos e propriedades que dizem ao mapa como ele deve se comportar. Para o uso do Google Maps, deve-se respeitar alguns termos de serviços. Traduzido de “*Google Maps/Google Earth APIs Terms of Service*”, alguns dos requisitos são:



### 2.5.1 Regras Gerais:

- Acesso Livre (sem pagamentos): A implementação usando a API deve ser acessível aos usuários livre de cobranças, assinaturas pagas, ou outras formas de acesso restrito por pagamentos. Esta regra vale para todo o conteúdo implementado agora e para qualquer conteúdo que venha a ser vinculado no futuro;
- Acesso público (sem *firewall*): A implementação usando a API não deve operar somente atrás de um *firewall*, ou somente dentro de uma rede interna (exceto durante fase de testes e desenvolvimento), ou dentro de uma comunidade fechada (por exemplo, acesso somente a usuários convidados) (*Beggining Google Maps API 3*, 2010).

### 2.5.2 Exceções:

- Versão Enterprise do Google: As regras das sessões anteriores não se aplicam caso se obtenha a versão Enterprise do Google, ou que seja concedida uma permissão escrita da Google.
- Aplicações Móveis: A regra de “Acesso Livre” descrita anteriormente não se aplica caso esteja se desenvolvendo aplicações para aparelhos móveis que possam ser vendidos por uma taxa através de uma loja online e que possa fazer o *download* da aplicação para um aparelho que consiga acessar essa loja (*Beggining Google Maps API 3*, 2010).

### 2.5.3 Exemplos:

- A aplicação pode solicitar que usuários realizem um login, porém não pode cobrar para isso;

- Pode ser cobrada uma taxa para a sua aplicação caso esta esteja disponível para download para um aparelho rodando o Sistema Operacional *Android* através da Android Market.

Se forem recebidas mais de 2.500 solicitações de geocódigo feitas por um mesmo endereço IP em um período de 24 horas, ou se as solicitações forem enviadas com muita frequência por um único endereço IP, o geocodificador da Google Maps API começará a responder com o código de status 620. Se o uso excessivo do geocodificador continuar, o acesso ao geocodificador da API do Google Maps a partir desse endereço IP pode ser bloqueado permanentemente.

Ao usar uma das classes de geocodificação da API do *Google Maps*, as solicitações são feitas pelo navegador da web do usuário e são contabilizadas na cota do endereço IP do usuário. Isso significa que o uso excessivo por parte de um usuário não terá impacto em outros usuários do site. No entanto, se muitos usuários acessarem as APIs do *Google Maps* por meio de um único proxy, eles compartilharão a cota do endereço IP do *proxy*.

Quando as solicitações do geocodificador são feitas com o Serviços de Geocodificação, elas são contabilizadas na cota do endereço IP do sistema que está fazendo a solicitação HTTP. Porém, não existe um limite para o número de visualizações de páginas de mapas no site.

## 2.6 GENERAL PACKET RADIO SERVICE (GPRS)

O *GPRS* é um serviço de comutação por pacotes para redes móveis. Essa tecnologia oferece um aumento das taxas de transferência nas redes de dados, quando comparado as tecnologias anteriores que utilizavam comutação por circuitos. O *GPRS* oferece taxas teóricas máxima de 56 kbps para envio de dados e 28 kbps para recepção. Nas redes com comutação por circuitos os recursos de rede são alocados exclusivamente para o usuário, independente de estar em uso ou não, enquanto que na comutação por pacotes o recurso é alocado estatisticamente, conforme demandado pelo usuário. Isso implica em aumento da complexidade de rede, visto que há atraso na entrega dos pacotes, os pacotes são enviados da

origem e são remontados no destino. Assim para algumas aplicações é necessário implementar qualidade de serviço para garantir o funcionamento correto. Como vantagem ele apresenta uma economia nos recursos de redes e redução de custos, visto que a tarifação pode ser feita pelo tempo de conexão (*3rd Generation Partnership Project, 2013*).

O GPRS permite conexão com redes móveis 2G, 3G e WCDMA para transmitir pacotes IP para redes externas como a *Internet*.

O advento da tecnologia GPRS tornou alguns serviços possíveis, tais como:

- Acesso a *Internet* “Always on”;
- Serviço de mensagem multimedia (MMS);
- *Push-to-talk* em telefones celulares;
- Acesso a *Internet* para dispositivos móveis através de *Wireless Application Protocol* (WAP).

Alguns exemplos dos protocolos que a tecnologia GPRS suporta:

- Internet Protocol (IP);
- Point-to-Point Protocol (PPP);

Conexões X.25, que são tipicamente usadas para aplicações como terminais de pagamentos sem fio, apesar de ter sido removida do padrão.

Quando TCP/IP é usado, cada dispositivo GPRS pode ter um ou mais endereço IP alocado.

Os dispositivos que utilizam a tecnologia GPRS podem ser divididos em três classes, conforme o seu *hardware*.

- Classe A: dispositivos classe A podem ser conectados aos serviços GSM e GPRS ao mesmo tempo;
- Classe B: dispositivos classe B podem ser conectados aos serviços GSM e GPRS, porém não ao mesmo tempo. Durante o serviço GSM, o serviço GPRS é suspenso sendo concluído automaticamente logo após o término do serviço GSM. A maioria dos dispositivos hoje em dia é classe B;

- Classe C: dispositivos classe C podem ser conectados aos serviços GSM e GPRS porém não ao mesmo tempo. A troca dos serviços deve ser feita manualmente.

## 2.7 GLOBAL SYSTEM FOR MOBILE COMMUNICATIONS (GSM)

Sistema Global para Comunicações Móveis (*Global System for Mobile Communications*) é um padrão de tecnologia para redes móveis desenvolvido pela *3rd Generation Partnership Project (3GPP)* para telefones celulares. O GSM utiliza canais de voz digitais nas frequências de 900MHz em dois conjuntos de frequências de 25MHz cada, sendo as frequências de 890MHz até 915MHz para transmissões do terminal e nas frequências de 935MHz até 960MHz para transmissões de rede. É utilizada uma combinação de TDMA (*Time Division Multiple Access*) e o FDMA (*Frequency Division Multiple Access*). Os canais de 25MHz são divididos em 124 canais com 200kHz, onde cada canal tem largura de banda de 270 kbps. Essas frequências são utilizadas pelas estações bases que fazem multiplexação por tempo em oito espaços de tempo (*timeslots*) (*3rd Generation Partnership Project, 2013*).

### 3 DESENVOLVIMENTO

#### 3.1 SERVIDOR

Foi necessária a implementação de um servidor web que hospeda o banco de dados e a interface que estaria disponível ao usuário (página HTML). Tal servidor foi confeccionado com a utilização de ferramentas amplamente difundidas e usadas. Os servidores web são a espinha dorsal da Internet, são os responsáveis por hospedar todas as páginas, os mecanismos de busca, base de dados e servem para todo tipo de aplicativo via web. vez mais os aplicativos *desktop*.

No início da criação da internet, eram utilizadas apenas páginas html estáticas e scripts CGI. A solução adotada pela equipe foi a combinação popularmente chamada de LAMP (Linux + Apache + MySQL + PHP). Sempre que é solicitada uma página em PHP, ou outra linguagem, entra em ação o módulo apropriado, que faz o processamento necessário e devolve ao Apache a página HTML que será exibida. Entram em ação, então, os gestores de conteúdo, que combinam os recursos do PHP com um banco de dados como o MySQL, acessado através dele.

##### 3.1.1 Apache

Apache é geralmente reconhecido como o servidor *Web* mais popular do mundo (servidor HTTP). Originalmente concebido para ambientes Unix, o servidor Web Apache foi portado para outros sistemas operacionais de rede *Windows*. O nome "Apache" deriva da palavra "*patchy*" que os desenvolvedores do Apache usaram para descrever as primeiras versões do seu *software* (APACHE Software Foundation, 2012).

O servidor Web Apache fornece uma gama completa de recursos de serviços da *Web*, incluindo CGI, SSL, e domínios virtuais. Apache também suporta módulos de plug-in para extensão. Apache é um *software* livre, distribuído pela

*Apache Software Foundation*, que promove várias tecnologias Web avançadas de código livre e aberto.

### 3.1.2 PHP

PHP (um acrônimo recursivo para PHP: *Hypertext Preprocessor*) é uma linguagem de *scripting* de uso geral amplamente utilizada aberta que é especialmente interessante para desenvolvimento web e pode ser incorporado em HTML (The PHP Group, 2013).

#### 3.1.2.1. phpMyAdmin

O phpMyAdmin é uma ferramenta de *software* livre escrito em PHP, destinado a lidar com a administração do MySQL sobre a *World Wide Web*. Suporta uma ampla gama de operações com o MySQL. As operações mais utilizadas são suportadas pela interface do usuário (gerenciamento de banco de dados, tabelas, campos, relações, índices, usuários, permissões, etc.), enquanto ainda tem a capacidade de executar diretamente qualquer declaração SQL.

Esta ferramenta foi utilizada para alimentar a base de dados fixos durante a fase de testes preliminares, auxiliando no desenvolvimento e melhora das rotinas implementadas.

## 3.2 BANCO DE DADOS

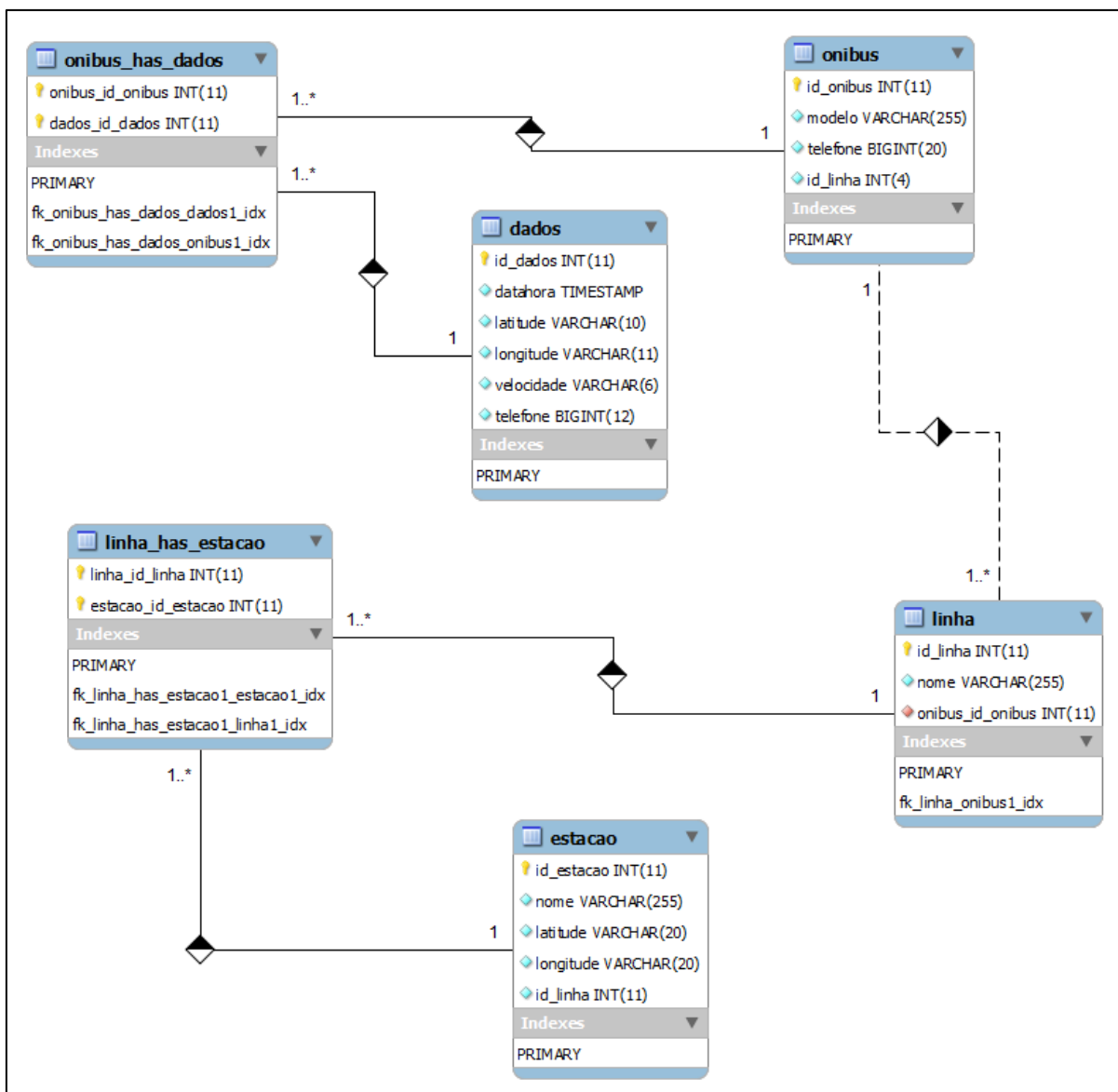
A opção de se utilizar um banco de dados para armazenar as informações coletadas através dos módulos presentes em diversos veículos leva em conta a quantidade de dados transmitidos continuamente por diversos módulos, robustez, velocidade e maior confiabilidade para o armazenamento e processamento das mesmas.

Entre os SGBDs algumas das principais alternativas existentes no mercado são:

- Microsoft SQL Server;
- Oracle;
- MySQL;
- PostgreSQL.

Das opções citadas acima, os sistemas MySQL e PostgreSQL fazem parte de projetos de código aberto e são alternativas gratuitas. Os sistemas Oracle e Microsoft SQL Server são opções pagas. Para o projeto SACH escolheu-se o SGBD MySQL por tratar-se de um sistema altamente difundido no mercado.

Para a modelagem do banco de dados foi utilizado o *software* MySQL WorkBench que implementa de maneira gráfica as tabelas e as relações entre as mesmas facilitando a visualização e o funcionamento do banco de dados. Na figura 10 é mostrada a formatação do banco de dados.



**Figura 10 - Diagrama do Banco de Dados.**

Fonte: Autoria própria

O banco de dados foi modelado de maneira simples com poucas tabelas para melhorar a performance, robustez e evitar a inserção de dados repetidos, o que em um futuro poderia sobrecarregá-lo com informações duplicadas.

Com base neste modelo as tabelas foram estruturadas da seguinte maneira:



Tabela 3 - Tabelas do Banco de Dados.

Tabela dados						
Coluna	Tipo	Colação	Atributos	Nulo	Padrão	Extra
<u>id_dados</u>	int(11)			Não	None	AUTO_INCREMENT
datahora	timestamp		on update CURRENT_TIMESTAMP	Não	CURRENT_TIMESTAMP	ON UPDATE CURRENT_TIMESTAMP
latitude	varchar(10)	latin1_swedish_ci		Não	None	
longitude	varchar(11)	latin1_swedish_ci		Não	None	
velocidade	varchar(6)	latin1_swedish_ci		Não	None	
telefone	bigint(12)			Não	None	

Tabela estacao						
Coluna	Tipo	Colação	Atributos	Nulo	Padrão	Extra
<u>id_estacao</u>	int(11)			Não	None	AUTO_INCREMENT
nome	varchar(255)	latin1_swedish_ci		Não	None	
latitude	varchar(20)	latin1_swedish_ci		Não	None	
longitude	varchar(20)	latin1_swedish_ci		Não	None	
id_linha	int(11)			Não	None	

Tabela linha						
Coluna	Tipo	Colação	Atributos	Nulo	Padrão	Extra
<u>id_linha</u>	int(11)			Não	None	AUTO_INCREMENT
nome	varchar(255)	latin1_swedish_ci		Não	None	
onibus_id_onibus	int(11)			Sim	NULL	

Tabela onibus						
Coluna	Tipo	Colação	Atributos	Nulo	Padrão	Extra
<u>id_onibus</u>	int(11)			Não	None	AUTO_INCREMENT
modelo	varchar(255)	latin1_swedish_ci		Não	None	
telefone	bigint(20)			Não	None	
id_linha	int(4)			Não	None	

Fonte: Autoria própria

Observação: as tabelas “linhas\_has\_estacao” e “onibus\_has\_dados” não foram comentadas pelo fato de serem criadas automaticamente pelo *software* MySQL WorkBench apenas para indicar os relacionamentos do tipo “muitos para muitos”.

### 3.1.3.1. Instruções SQL

Por último, para implementação do banco de dados é necessária a conversão das tabelas em código SQL para ser executado no SGBD. Tal código está descrito no Anexo 1 da lista de Apêndices.

## 3.3 DYNAMIC DNS

Serviço de *Domain Name System (DNS)* é utilizado para a conversão de endereços IP públicos a partir de nomes de domínio para o formato numérico (IP). Isto significa que não há nenhuma necessidade de se ter em mente o endereço 177.99.189.216, mas é fácil de se lembrar [www.google.com.br](http://www.google.com.br), por exemplo. A tradução do nome do domínio para o conjunto de números é executada pelo servidor de DNS adequado, à qual o computador envia um pedido para a tradução do endereço. Depois de receber a resposta com o endereço numérico, o computador se conecta ao computador / servidor relevante.

Tal comunicação é possível quando o servidor de destino tem um endereço IP estático público a que esteja sempre acessível. O registro, uma vez colocado no banco de dados, ligando o endereço 177.99.189.216 com [www.google.com.br](http://www.google.com.br), está sempre atualizado e constante. Portanto, o cliente pode se comunicar com o servidor, quer pelo seu endereço de domínio (usando a tradução DNS) ou diretamente pelo endereço numérico (sem ajuda do serviço de DNS).

Porém, no caso de não possuir um endereço IP fixo, é necessário a utilização de um serviço de *Dynamic DNS (DDNS)*, da mesma forma que o servidor DNS, que fornece um banco de dados contendo as relações entre domínio e endereços numéricos, esta base pode ser atualizado a pedido do proprietário do domínio. Graças a esse recurso, o servidor está disponível no endereço do domínio, independentemente do seu endereço IP real. Nesse caso, o acesso ao servidor é limitado ao seu nome de domínio traduzido pelo servidor DDNS.

Para este projeto foi utilizado os serviços da DynDNS.com, que faz a tradução do endereço [sach.webhop.org](http://sach.webhop.org) e encaminha os pedidos de acesso ao IP do

servidor correspondente que hospeda o banco de dados e a aplicação *web* deste projeto.

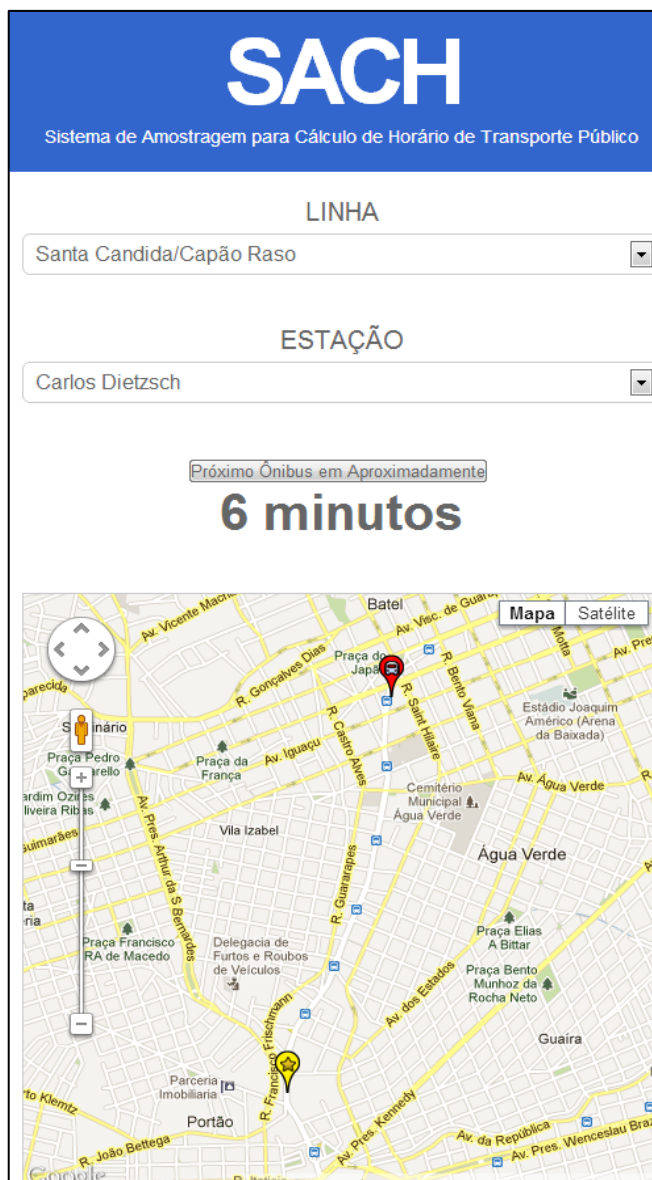
### 3.4 WEB SERVICE

Foi desenvolvida uma página *web* para que o usuário pudesse fazer as consultas ao banco de dados e ser informado em aproximadamente quanto tempo o próximo ônibus chegaria a estação/ponto de ônibus desejado.

A escolha do formato de uma página *web* foi feita pelo fato da ampla difusão do uso de *smartphones*, *tablets* que possuem acesso a *Internet*, desta maneira o serviço estaria disponível a qualquer momento e em qualquer lugar que tenha acesso a *Internet*. A página *web* pode ser vista na figura 11. O usuário apenas deve informar qual a linha de ônibus, qual estação/ponto de ônibus ele deseja informações e clicar no botão “Próximo Ônibus em Aproximadamente”. Com isso, uma requisição será enviada ao banco de dados que irá buscar quais ônibus pertencem a esta linha e calculará qual está mais próximo do ponto desejado.

O servidor irá retornar ao usuário o tempo aproximado e um mapa que informa em tempo real a posição do ônibus mais próximo (ícone vermelho) e a localização da estação/ponto de ônibus pesquisada (ícone amarelo).

O *Web Service* pode ser acessado pelo seguinte endereço: <http://sach.webhop.org>.



**Figura 11 - Interface do usuário.**  
**Fonte: Autoria própria**

### 3.5 DESENVOLVIMENTO DE HARDWARE

#### 3.5.1 Módulo GPS

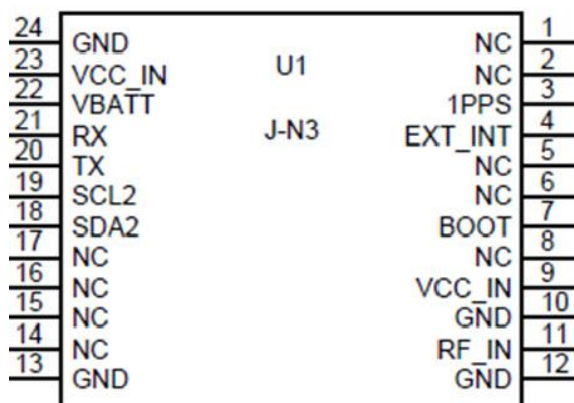
O módulo utilizado no projeto do *hardware* foi o módulo fabricado pela *Telit Wireless Solutions*, o Jupiter JN3. Esse módulo construído com arquitetura *SiRFstarIV* possui 48 canais que possibilitam a captura de sinais com potência de

até -163dBm, tamanho extremamente reduzido, suporte à tecnologia i2c, baixo consumo, suporte à efemeridades e *Satellite Based Augmentation System (SBAS)* (TELLIT WIRELESS SOLUTIONS, 2012). O módulo ainda oferece um *Time to First Fix (TTFF)* em torno de dois segundos com o uso de uma bateria para memória não volátil, e informações GPS utilizando o protocolo padrão NMEA através porta serial *Transistor-Transistor Logic (TTL)*.

O fato de o módulo oferecer uma saída serial com sentenças NMEA, torna a interface com um processador extremamente simples. A saída serial do Jupiter JN3 oferece quatro sentenças padrão NMEA. As sentenças estão definidas da seguinte forma.

- **RMC:** *Recommended Minimum Data for GPS*. Essa sentença traz informações essenciais sobre posicionamento, velocidade, data e hora além de algumas informações complementares. Estão disponíveis na saída serial do módulo a cada um segundo.
- **GGA:** *Fix Information*. Fornece dados de posicionamento em 3D, que contém além de latitude e longitude a altitude. Descreve também a quantidade de satélites rastreados, a qualidade e a hora do *Fix*, Estão disponíveis a cada um segundo.
- **GSA:** *Overall Satellite Data*. Dispõe os dados sobre os satélites rastreados, bem como a precisão do *Fix*. Está disponível a cada um segundo.
- **GSV:** *Detailed Satellite Data*. Mostra dados sobre os satélites que o aparelho consegue rastrear. Está disponível a cada cinco segundos.

O módulo Jupiter JN3 conta com um pino 1PPS (*Time Mark Pulse*). Esse pino disponibiliza uma saída com um pulso de largura de 200ms caso haja um *Fix3D* Esse pino está ligado ao processador para monitoramento do funcionamento e também a um LED para indicação visual do funcionamento do módulo. A figura 12 ilustra o módulo e seus pinos.



**Figura 12 – Módulo Jupiter JN3 e sua pinagem.**  
**Fonte: Telit Wireless Solutions, 2012.**

### 3.5.2 Módulo GPRS

Para a comunicação móvel, foi utilizado o módulo fabricado pela Telit Wireless Solutions. Esse módulo é um dispositivo GSM/GPRS *quad band* (850, 900, 1800, 1900 MHz), baixo consumo, tamanho reduzido. Ele inclui várias funcionalidades como pilha TCP/IP, multiplexador serial e executa comandos AT, que são bastante difundidos para programação de modems. Dentre os comandos AT, há alguns específicos para configuração do dispositivo além de permitirem acesso completo ao SIM card. Possui ainda, oito portas para I/O, áudio analógico e digital, dois conversores analógico-digital e um digital-analógico. Internamente, possui um interpretador de *scripts* Python com 1,9MB de memória não-volátil para armazenamento e 1MB de memória RAM (TELLIT WIRELESS SOLUTIONS, 2013).

O módulo faz interface com o processador através de portas seriais assíncronas. As portas seriais possuem detecção automática do *baud-rate* que pode variar de 300 a 115200 bits por segundo (TELLIT WIRELESS SOLUTIONS, 2013). Como o módulo disponibiliza saídas para o controle de comunicação por *hardware*, foi feita também a interface dos respectivos pinos ao processador. Apesar das conexões existentes, o controle da comunicação foi realizado em *software*. A principal razão para existirem as conexões e não usá-las, é a possibilidade de aliviar o processamento em melhorias futuras e deixar o controle da comunicação a cargo do *hardware*.

As portas de I/O podem se usadas tanto como entrada geral como para funções alternativas. Essas funções podem ser ativadas ou desativadas na configuração do módulo. No caso desse projeto, foi usada somente uma dessas portas configurada como indicativo de estado. Essa porta oferece uma saída para a conexão com um LED, que por sua vez pisca com frequências diferentes para indicar o status do módulo.

- Sempre desligado: Módulo desligado;
- Frequência alta: Procura de rede, não registrado, processo de desligamento;
- Frequência baixa: Registrado na rede;
- Sempre ligado: Chamada ativa.

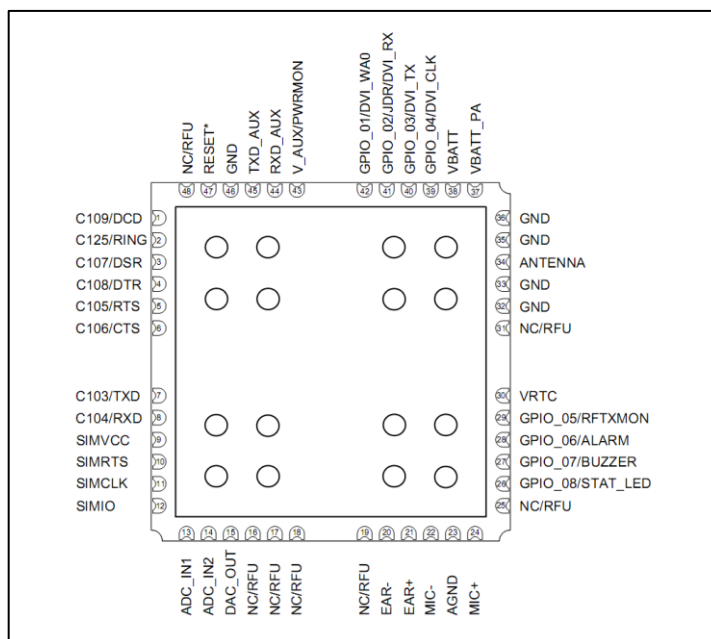
Ao contrário do módulo GPS, que possui uma porta semelhante, essa saída particular do módulo não foi conectada ao processador, pois é possível receber o status enviando um comando AT pela porta serial. Porém, a saída que oferece uma tensão auxiliar de 2,8V que foi ligada ao processador para monitorar se o módulo está funcional. Somente após essa saída estar em nível lógico alto, é possível enviar comandos AT e receber suas respostas.

O GL865 permite a conexão com um cartão do tipo *SIM card*, ou USIM. A principal diferença entre os dois é o fato de o *SIM card* ser um *chip* comprado com operadoras de telefone, e ligado ao módulo através de um conector específico. O USIM tem, virtualmente, a mesma funcionalidade de um *SIM card*, porém vem encapsulado no formato de um circuito integrado e é soldado à placa.

Para melhorias futuras, o USIM pode ser utilizado pois não possibilita a troca do número de telefone atrelado a ele, diferente do *SIM card*, que pode ser substituído. No protótipo construído, foi utilizado o *SIM card*, pois é o formato oferecido pela maioria das operadoras de telefone.

Não foram usadas as partes do módulo que tratam de conversores analógico-digitais e áudio. Essas partes não fazem parte da ideia inicial do projeto, porém fazem parte de melhorias planejadas futuramente. A existência dessas funcionalidades tornam essas melhorias mais fáceis, pois não precisam envolver um projeto para agregar um novo componente, somente para agregar as funções em si.

Na figura 13 estão os pinos existentes no módulo e sua configuração física.



**Figura 13 - Configuração dos pinos do módulo GL865 QUAD**  
**Fonte: GL865 Hardware User Guide, 2013.**

### 3.5.3 Fonte de Alimentação

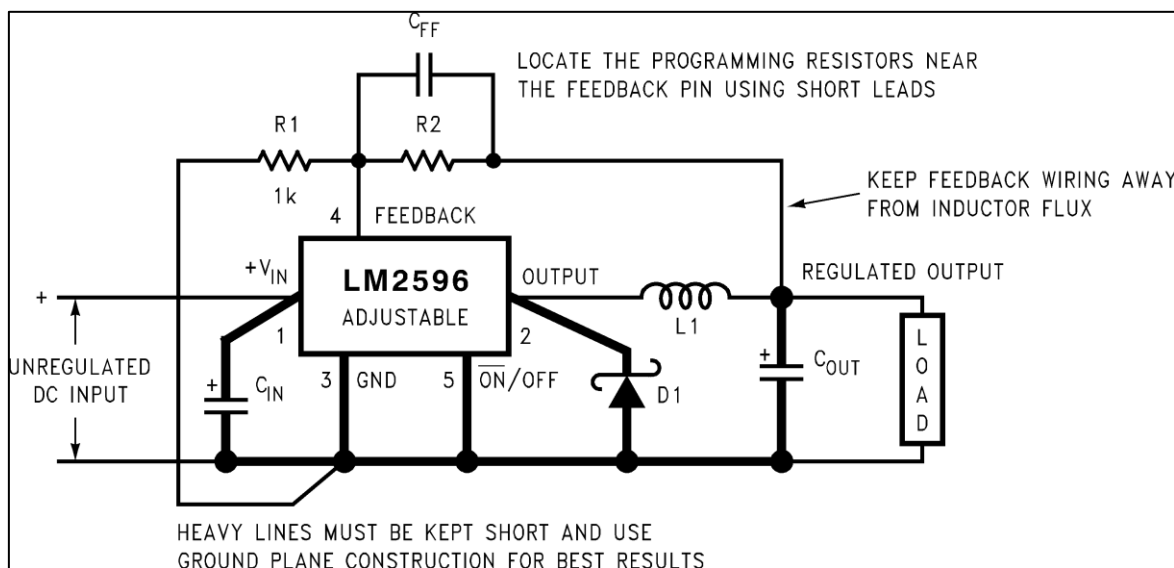
A alimentação do circuito foi projeto levando em consideração a fonte de energia existente em automóveis. A tensão fornecida pela maioria das baterias existentes no mercado hoje em dia é de 12V.

Os componentes do projeto pedem uma alimentação de três tensões diferentes. O módulo GPS funciona com uma alimentação padrão de 3,3V, bem como o processador. O módulo GPRS funciona com uma grande extensão de tensões, porém sua documentação sugere uma tensão de 3,8V. Uma tensão auxiliar de 5V foi disponibilizada para futuras melhorias, como display gráfico, e também para o processador, que tolera níveis lógicos de até 5V, necessitando então de tal alimentação.

Para fornecer as tensões citadas, foram utilizados três conversores chaveados do tipo *step-down*. O circuito integrado escolhido para fornecer as tensões foi o LM2596 fabricado pela Texas Instruments. Ele opera com uma frequência de 150kHz, o que o faz precisar de poucos componentes auxiliares (Texas Instruments, 2004). Suas especificações também atendem certos requisitos necessários para o projeto do circuito.



- Saídas com tensões ajustáveis;
- Corrente de saída de 3 A;
- Tensão de entrada de até 40 V;
- Alta eficiência.



**Figura 14 - Exemplo de fonte chaveada com saída regulável para o LM2596**  
 Fonte: TEXAS INSTRUMENTS - LM2596 Datasheet (1999).

### 3.5.4 Radio Frequência em placas de circuito impresso

Para o bom funcionamento dos módulos GPS e GPRS, é necessário que as trilhas que levam o sinal recebido até as entradas respectivas sigam algumas regras. A razão para que essas regras sejam seguidas, é o fato de tanto o GPRS quanto o GPS trabalharem com frequências muito elevadas, da ordem de até 2GHz. De maneira geral, as regras são descritas nos manuais dos módulos.

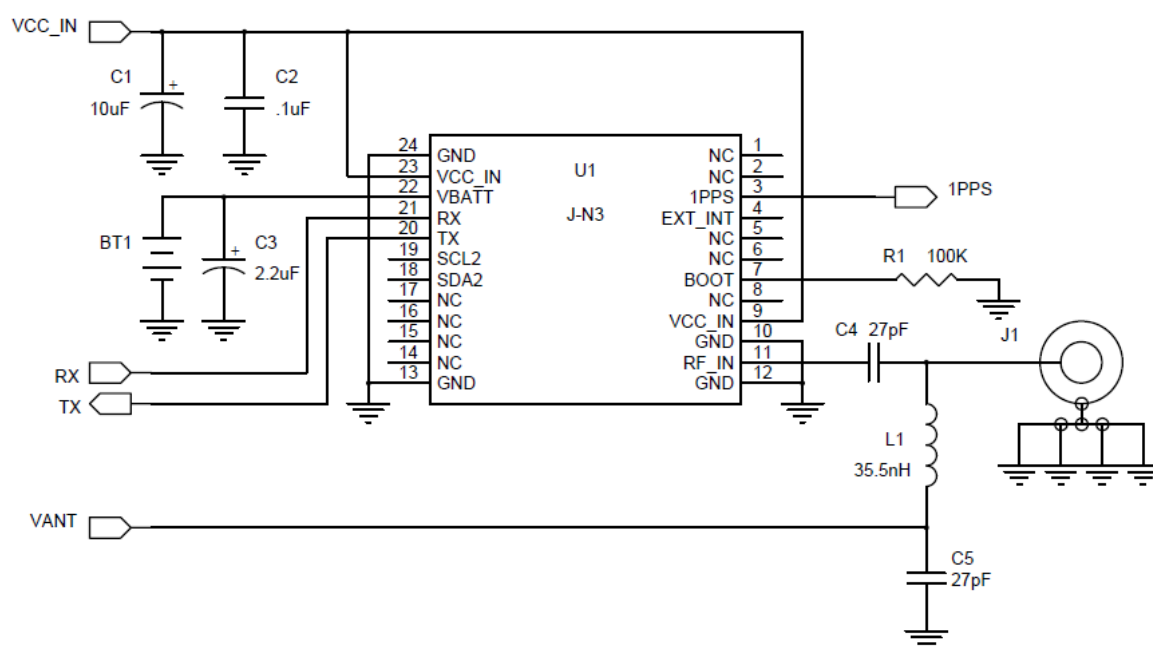
Para o caso do módulo GPRS GL865, as seguintes recomendações são sugeridas:

- Impedância da trilha 50Ω;
- A trilha deve a mais curta possível;
- Envolver a trilha com o plano de referência;

- Deve ser posicionar vias de conexão para o plano de referência a cada 2 mm ao redor da trilha;
- Evitar posicionar componentes que possam gerar interferência eletromagnética, próximos à trilha.

No caso do módulo GPS Jupiter JN3, o manual aconselha a usar uma antena ativa, com um amplificador de baixo ruído embutido, devido ao fato que ela será fisicamente posicionada longe do módulo. A impedância requerida para a trilha também é de  $50\Omega$  e as recomendações para o seu projeto são as mesmas citadas no caso do módulo GPRS. Porém, como a antena requer uma alimentação externa, um circuito extra é necessário para o correto funcionamento. O manual sugere um exemplo de aplicação para a utilização do módulo.

No exemplo abaixo, o capacitor C5 é posicionado de forma a evitar que a tensão de alimentação da antena entre no módulo. O indutor é escolhido de tal forma a ser ressonante com a frequência do sinal GPS, para minimizar a carga, por fim, o capacitor C6 é um filtro que permite a passagem do sinal RF, impedindo a entrada de frequências baixas. O sinal VANT é a alimentação da antena. Para o projeto do PCB do projeto em questão, foram seguidas as recomendações e o circuito descrito na figura 15, no que se trata de recepção do sinal RF.



**Figura 15 - Exemplo de utilização do módulo Jupiter JN3**  
 Fonte: Jupiter JN3, Manual de Referência de Hardware

### 3.5.5 Microcontrolador Cortex-M4 STM32F4

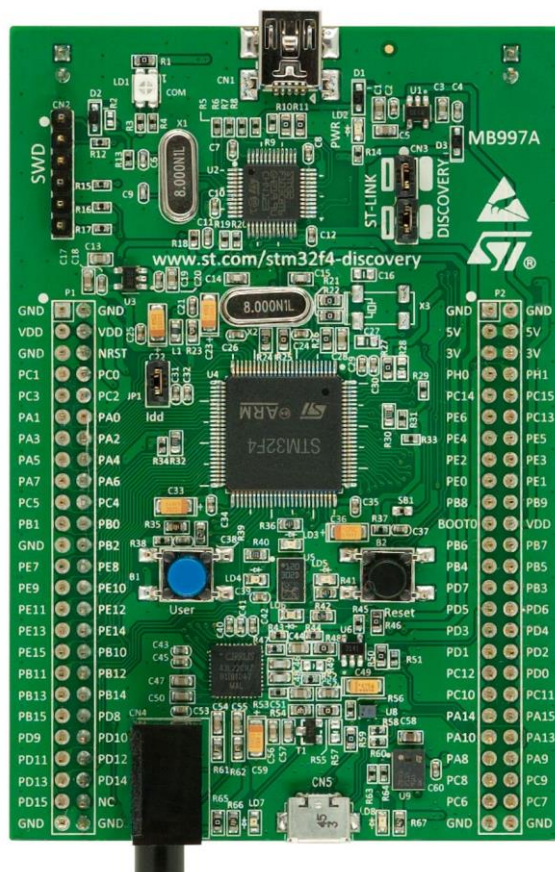
Para fazer a interface entre os módulos foi utilizado o processador STM32F407VG fabricado pela ST Microelectronics. Optou-se por utilizar um kit de desenvolvimento contendo o processador, devido à facilidade do desenvolvimento do protótipo.

O microcontrolador utilizado possui 1MB de memória *Flash* para código, baixo consumo, 140 portas para I/O, interface USB, 15 interfaces para comunicação, suporte a *Ethernet*, entre outras funcionalidades (STMicroelectronics, 2012). Foram utilizadas duas portas USART para a conexão com os dois módulos, utilizando interrupções e tratamento por *software* para o gerenciamento da comunicação entre eles. Foi deixado preparado, porém não foram implementadas, conexões para um *display* gráfico, que ocupa cerca de 20 pinos do microcontrolador.

A utilização desse microcontrolador é a possibilidade de adicionar funcionalidades extras, sem necessidade de uma modificação significativa do *hardware*. A seguir algumas funcionalidades possíveis.

- Leitor de falhas do motor. Com a padronização dos códigos de falhas e a utilização da interface *Controller Area Network (CAN)* nas centrais eletrônicas dos motores, é possível fazer uma interface direta com o processador para a leitura de falhas;
- *Display* para mostrar informações, podendo ser personalizado para atender às necessidades de cada cliente;
- Botões para utilização do usuário. O usuário pode utilizar os botões para alternar funções e visualizar informações relevantes. É possível adicionar um número fixo para chamadas de emergência, sem a necessidade do uso de um telefone pessoal.

A figura 16 ilustra o *kit*.



**Figura 16 – Placa base do microcontrolador Cortex-M4.**  
**Fonte: STM32F4 Discovery User Manual, 2012.**

### 3.5.6 Placa de circuito impresso

Para agregar todos os componentes e circuitos necessários, foi projetada uma placa de circuito impresso. Essa placa une os dois módulos, a fonte de alimentação, conectores para as antenas e a placa com o microcontrolador. Seu tamanho foi o menor possível, levando em conta o limitante do tamanho dos componentes e as regras para construção das trilhas. A placa possui dois lados funcionais e todas as trilhas estão rodeadas por planos de referência o que aumenta a proteção contra ruídos externos.

A placa foi desenvolvida com a finalidade de embarcar os componentes no interior de um automóvel, portando a sua robustez foi mais considerada em relação a tamanho. Os conectores da antena têm saída externa, para a fixação das mesmas em uma posição que tenha o “céu visível”. O conector de potência também está disponível externamente.

A confecção da mesma foi feita na UTFPR e possui acabamento em *Hot Air Leveling (HAL)*, e máscara da serigrafia azul. As figuras 17 e 18 mostram a placa confeccionada.

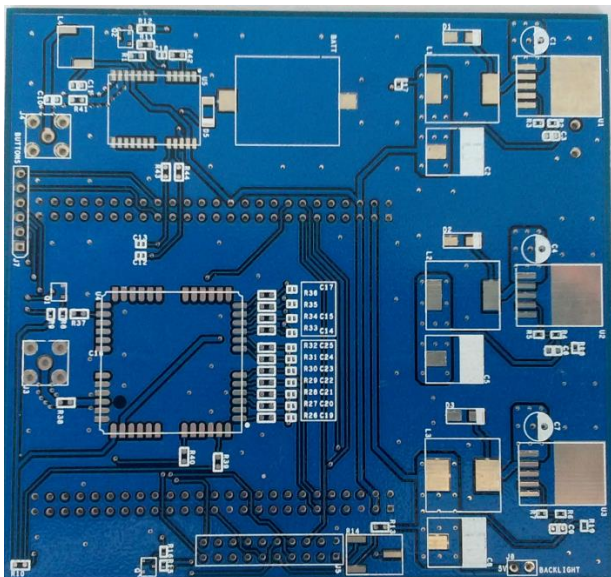


Figura 17 - Parte Frontal da PCB desenvolvida  
Fonte: Autoria Própria

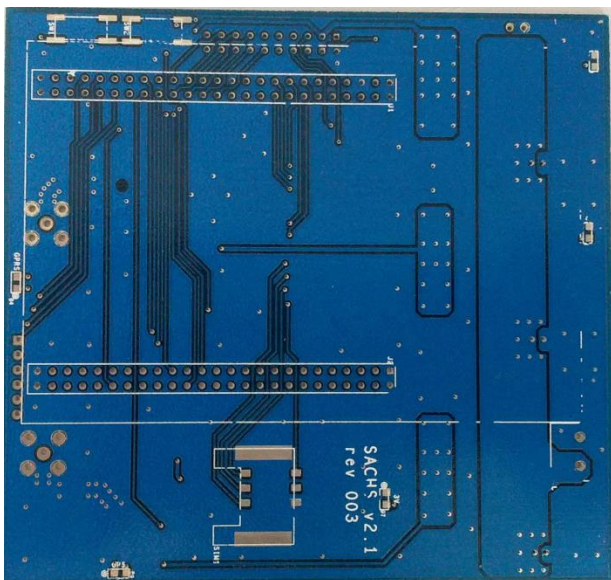


Figura 18 - Parte traseira da PCB desenvolvida  
Fonte: Autoria Própria

## 4 RESULTADOS OBTIDOS

Neste capítulo serão descritos os resultados de uma simulação de utilização do sistema em campo. Serão extraídas e mostradas as tabelas geradas no Banco de Dados.

A simulação consiste em um percurso padrão de uma determinada linha de ônibus do sistema de transporte de Curitiba, realizada com um veículo particular. Durante este percurso foi acessada a página *web* para teste do acesso de um usuário.

### 4.1 TESTES REALIZADOS EM CAMPO

O percurso realizado no teste foi entre dois pontos da Linha Santa Cândida/Capão Raso. O ponto de partida do carro foi da Estação Catedral da Fé e o ponto de chegada a Estação Bento Viana, cuja distância é de 1,5km.

- Ponto Inicial: Estação Catedral da Fé
- Ponto Final: Estação Bento Viana

As figuras a seguir, mostram as interfaces de acesso à web, disponibilizadas para o usuário.

A figura 17 mostra a interface inicial do *website* ao ser acessado inicialmente. É mostrado um campo com uma barra de rolagem contendo as linhas do Sistema de transporte público.

**SACH**  
Sistema de Amostragem para Cálculo de Horário de Transporte Público

LINHA

Santa Candida/Capão Raso

Trabalho Conclusão de Curso  
Alunos  
Fabian Fink; João Ronzani; Vinicius Durgiewicz; Weberton Natsuaki  
Orientador  
Rubens Alexandre de Faria  
Todos Direitos Reservados – UTFPR - Universidade Tecnológica Federal Do Paraná - 2013

**Figura 19 – Escolha da linha de ônibus.**  
Fonte: Autoria própria.

Após a escolha da Linha, no caso, Santa Cândida/Capão Raso, é disponibilizada outro campo contendo as estações que se encontram nessa linha. Par ao teste foi escolhida a Estação Bento Viana.

**SACH**  
Sistema de Amostragem para Cálculo de Horário de Transporte Público

LINHA

Santa Candida/Capão Raso

ESTAÇÃO

Selecione uma Estação Tubo

Selecione uma Estação Tubo

Trabalho Conclusão de Curso  
Alunos  
Fabian Fink; João Ronzani; Vinicius Durgiewicz; Weberton Natsuaki  
Orientador  
Rubens Alexandre de Faria  
Todos Direitos Reservados – UTFPR - Universidade Tecnológica Federal Do Paraná - 2013

Selecione uma Estação Tubo

Alferes Poli (Catedral da Fé)  
Antônio Cavalheiros  
Antônio Lago  
Bento Viana  
Bom Jesus  
Carlos Dietzsch  
Central  
Constantino Marochi  
Coronel Dulcídio  
Dom Pedro I  
Fernando de Noronha  
Gago Coutinho  
Herculano de Araújo  
Holanda  
Hospital do Trabalhador  
Itajuba  
Joaquim Nabuco  
Maria Clara (Perpétuo Socorro)

Trabalho Conclusão de Curso  
Alunos  
Fabian Fink; João Ronzani; Vinicius Durgiewicz; Weberton Natsuaki  
Orientador  
Rubens Alexandre de Faria  
Todos Direitos Reservados – UTFPR - Universidade Tecnológica Federal Do Paraná - 2013

**Figura 20 - Escolha da Estação desejada.**  
Fonte: Autoria própria.

Escolhida a Linha, o mapa contendo a Estação é mostrado, e um botão para mostrar a previsão do próximo carro é mostrado.



Figura 21 - Página mostrando botão para executar cálculo de tempo.  
Fonte: Autoria própria.



Após o pressionamento do botão, o cálculo utilizando os dados sobre o posicionamento do ônibus, que é constantemente atualizado no banco de dados, é efetuado e o tempo estimado é mostrado na interface.

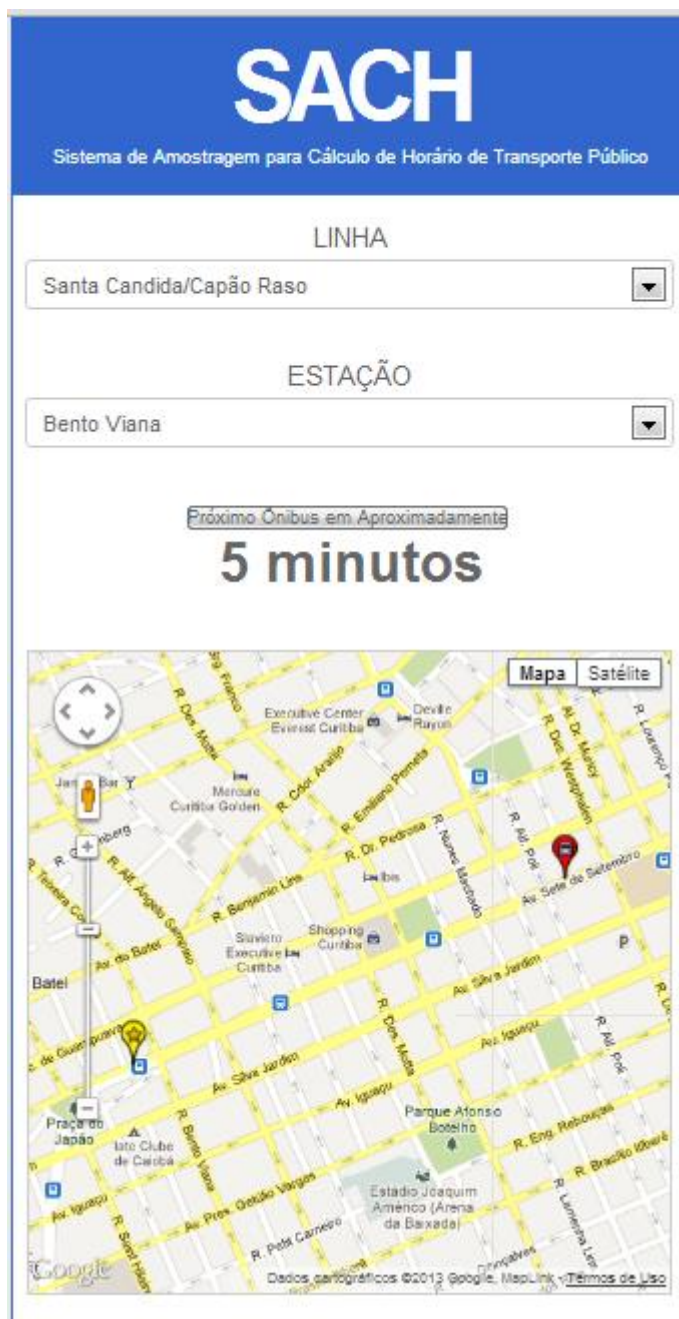


Figura 22 - Tempo disponibilizado para o usuário.  
Fonte: Autoria própria.

## 4.2 CONSIDERAÇÕES

O valor obtido no teste com o carro feito pela equipe foi de 4 minutos e 46 segundos, muito próximo do valor calculado de 5 minutos, pelo sistema proposto. Esse valor calculado foi comparado ao simulador disponibilizado pelo aplicativo de mapas do grupo Google, o qual apresentou uma estimativa de 5 minutos.

Os comparativos entre um teste de trajeto real e a simulação com outro sistema com proposta semelhante foi bem animador já que mostram que o SACH obtém valores muito próximos do esperado e apresenta bom desempenho no serviço que se propõe a executar.

Uma rotina de limpeza é executada para que os dados das tabelas sejam limpos todos os dias para a manutenção do bom funcionamento do sistema. Em uma implementação real ela deve ser programada pra executar a limpeza em horário nos quais o fluxo de ônibus e, conseqüentemente de dados, é menor.

## 5 CONCLUSÃO

Este trabalho contemplou o desenvolvimento de um Sistema de Cálculo de horário de veículos para aplicação em um sistema de transporte público.

A ideia do projeto surgiu de uma observação sobre os problemas diariamente relatados sobre o sistema de transporte público de grandes cidades, principalmente no Brasil, onde a grande maioria da população sofre pela falta de estrutura desse sistema. Um estudo para levantar esses problemas foi realizado, sendo um deles a falta de informação sobre os horários e também das linhas que realizam os ônibus.

Várias opções em relação à tecnologia foram estudadas na busca por um sistema que fosse eficiente, robusto e principalmente de fácil acesso para o usuário. Assim se optou pela utilização das tecnologias GPS e GPRS, aproveitando a utilização amplamente difundida dessas tecnologias. Os módulos GPS e GPRS, fabricados pela *Telit*, oferecem grande facilidade na montagem no PCB e na utilização por oferecerem interface serial, já amplamente utilizada.

Quanto ao microcontrolador, escolheu-se um com arquitetura ARM por ser amplamente difundida e de fácil utilização. O microcontrolador STM32F407 atendeu à todas as especificações. Das interfaces que o microcontrolador possui, muitas não foram utilizadas para esse projeto, porém são uteis para futuras melhorias do sistema.

A criação de um banco de dados se fez necessária devido ao imenso tráfego de dados que são gerados pelas centenas de veículos e demais componentes pertencentes a um sistema de transporte real de um grande centro. A opção por utilizar a sistema de Banco de Dados *MySQL* se deve ao fato de ser uma plataforma robusta para o tamanho do sistema projetado além de ser muito difundida e de constante atualizações para sua melhoria.

A escolha do formato de uma página *web* foi feita pelo fato da ampla difusão do uso de *smartphones*, *tablets* que possuem acesso a *Internet*, desta maneira o serviço estaria disponível a qualquer momento e em qualquer lugar que tenha acesso a *Internet*. A utilização da linguagem de programação *Java Script* se fez necessária pois é possível uma fácil integração com a API de mapas fornecida pelo Google.

As maiores dificuldades encontradas estão na má qualidade do PCB, desenvolvido na UTFPR, com verniz da máscara se soltando com facilidade e problemas com as trilhas, principalmente em situações de retrabalho dos componentes, e a falta de confiabilidade nas conexões à rede das operadoras de telefonia móvel, muitas vezes demoradas e não respondendo às solicitações enviadas.

O estudo da transformação deste projeto em um negócio vendável foi realizado através da elaboração do plano de negócios já apresentado neste relatório anteriormente. Apesar das dificuldades e do alto valor de investimento inicial para a estruturação de uma empresa, o negócio se mostra promissor, pois é visto através do gráfico da figura 22, as linhas de gastos e receitas cruzando no terceiro ano de empresa, ou seja, o *break even point* ocorre em três anos de operação, que segundo estudos de viabilidade econômica, é o limite máximo para que uma empresa tenha possibilidades de prosperar.

## REFERÊNCIAS

3rd GENERATION PARTNERSHIP PROJECT, THE (3GPP) – **GPRS & EDGE** – Disponível em: <<http://www.3gpp.org/Technologies/Keywords-Acronyms/article/gprs-edge>>. Acesso em: 01/03/2013.

APACHE SOFTWARE FOUNDATION. **APACHE HTTP SERVER PROJECT** – <[http://httpd.apache.org/ABOUT\\_APACHE.html](http://httpd.apache.org/ABOUT_APACHE.html)>. Acesso em: 18/03/2013.

CUNHA, Cristóvão José da. **Diagrama Entidade-Relacionamento**. Disponível em: <[http://www.feg.unesp.br/~crisov/CTIG/banco\\_dados/04%20-%20Diagrama%20Entidade%20Relacionamento%20\(DER\).jpg](http://www.feg.unesp.br/~crisov/CTIG/banco_dados/04%20-%20Diagrama%20Entidade%20Relacionamento%20(DER).jpg)>. Acesso em: 25/02/2013.

IPEA – INSTITUTO DE PESQUISA ECONÔMICA APLICADA. **JORNAL DO COMÉRCIO: Deficiências De ônibus e metrô aceleram o uso de transporte individual**. Disponível em: <[http://www.ipea.gov.br/portal/index.php?option=com\\_content&view=article&id=7107:jornal-do-comercio-rs-deficiencias-de-onibus-e-metros-aceleram-o-uso-do-transporte-individual-parte-1&catid=159:clipping&Itemid=75](http://www.ipea.gov.br/portal/index.php?option=com_content&view=article&id=7107:jornal-do-comercio-rs-deficiencias-de-onibus-e-metros-aceleram-o-uso-do-transporte-individual-parte-1&catid=159:clipping&Itemid=75)>. Acesso em: 20/05/2012.

GARMIN INTERNATIONAL INC. **What is GPS?** Disponível em: <<http://www8.garmin.com/aboutGPS/>>. Acesso em: 07/12/2012.

Mahaffey, Joe - GPSINFORMATION.NET. **NMEA data information**. Disponível em: <<http://www.gpsinformation.org/dale/nmea.htm>>. Acesso em: 10/01/2013.

PHP GROUP, THE. **Manual do PHP** - <<http://www.php.net/manual/en/preface.php>>. Acesso em: 18/02/2013.

STMICROELECTRONICS. **STM32F4 Datasheet**. STMicroelectronics, 2012. 167p.

STMICROELECTRONICS. **UM1472 User Manual** – STM32F4 Discovery, 2012. 38p.

TELLIT WIRELESS SOLUTIONS. **GL865 Hardware User Guide**. Telit 2013, 76p.

TELLIT WIRELESS SOLUTIONS. **GL865-QUAD** – Disponível em:  
<[http://www.telit.com/en/products/gsm-gprs.php?p\\_id=12&p\\_ac=show&p=110](http://www.telit.com/en/products/gsm-gprs.php?p_id=12&p_ac=show&p=110)>.  
Acesso em: 11/01/2013.

TELLIT WIRELESS SOLUTIONS. **JN3 EVK User Manual**. Tellit Wireless Solutions 2012, 34p.

TEXAS INSTRUMENTS. **LM2596 Datasheet** – LM2596 *Simple Switcher*, 2004. 44p.

TOMTOM, SISTEMAS GPS PORTÁTEIS DE NAVEGAÇÃO PARA AUTOMÓVEL.  
**Como funciona o GPS?** Disponível em:  
<<http://www.tomtom.com/howdoesitwork/page.php?ID=8&CID=2%20&Language=17>>.  
>. Acesso em: 07/12/2012.

U.S. GOVERNMENT WEBSITE, THE: GPS.gov. **Space Segment**. Disponível em:  
<<http://www.gps.gov/systems/gps/>>. Acesso em: 09/12/2012.

## APÊNDICES

### APÊNDICE A: Código para conversão das tabelas em código SQL

```
--  
-- Banco de Dados: `sach`  
--  
-----  
  
--  
-- Estrutura da tabela `dados`  
--  
  
CREATE TABLE IF NOT EXISTS `dados` (  
  `id_dados` int(11) NOT NULL AUTO_INCREMENT,  
  `datahora` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE  
CURRENT_TIMESTAMP,  
  `latitude` varchar(10) NOT NULL,  
  `longitude` varchar(11) NOT NULL,  
  `velocidade` varchar(6) NOT NULL,  
  `telefone` bigint(12) NOT NULL,  
  PRIMARY KEY (`id_dados`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=13 ;  
  
-----  
  
--  
-- Estrutura da tabela `estacao`  
--  
  
CREATE TABLE IF NOT EXISTS `estacao` (  
  `id_estacao` int(11) NOT NULL AUTO_INCREMENT,  
  `nome` varchar(255) NOT NULL,  
  `latitude` varchar(20) NOT NULL,  
  `longitude` varchar(20) NOT NULL,  
  `id_linha` int(11) NOT NULL,  
  PRIMARY KEY (`id_estacao`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT=''  
  
-----  
  
--  
-- Estrutura da tabela `linha`
```

```

--
CREATE TABLE IF NOT EXISTS `linha` (
  `id_linha` int(11) NOT NULL,
  `nome` varchar(255) NOT NULL,
  `onibus_id_onibus` int(11) DEFAULT NULL,
  PRIMARY KEY (`id_linha`),
  KEY `fk_linha_onibus1_idx` (`onibus_id_onibus`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-----

--
-- Estrutura da tabela `linha_has_estacao`
--

CREATE TABLE IF NOT EXISTS `linha_has_estacao` (
  `linha_id_linha` int(11) NOT NULL,
  `estacao_id_estacao` int(11) NOT NULL,
  PRIMARY KEY (`linha_id_linha`,`estacao_id_estacao`),
  KEY `fk_linha_has_estacao1_estacao1_idx` (`estacao_id_estacao`),
  KEY `fk_linha_has_estacao1_linha1_idx` (`linha_id_linha`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-----

--
-- Estrutura da tabela `onibus`
--

CREATE TABLE IF NOT EXISTS `onibus` (
  `id_onibus` int(11) NOT NULL AUTO_INCREMENT,
  `modelo` varchar(255) NOT NULL,
  `telefone` bigint(20) NOT NULL,
  `id_linha` int(4) NOT NULL,
  PRIMARY KEY (`id_onibus`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;

-----

--
-- Estrutura da tabela `onibus_has_dados`
--

CREATE TABLE IF NOT EXISTS `onibus_has_dados` (
  `onibus_id_onibus` int(11) NOT NULL,

```



```
`dados_id_dados` int(11) NOT NULL,  
PRIMARY KEY (`onibus_id_onibus`,`dados_id_dados`),  
KEY `fk_onibus_has_dados_dados1_idx` (`dados_id_dados`),  
KEY `fk_onibus_has_dados_onibus1_idx` (`onibus_id_onibus`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
  
--  
Restrições para as tabelas dumpadas  
--  
  
--  
-- Restrições para a tabela `linha`  
--  
ALTER TABLE `linha`  
  ADD CONSTRAINT `fk_linha_onibus1` FOREIGN KEY (`onibus_id_onibus`)  
  REFERENCES `onibus` (`id_onibus`) ON DELETE NO ACTION ON UPDATE NO ACTION;  
  
--  
-- Restrições para a tabela `linha_has_estacao`  
--  
ALTER TABLE `linha_has_estacao`  
  ADD CONSTRAINT `fk_linha_has_estacao1_estacao1` FOREIGN KEY (`estacao_id_estacao`)  
  REFERENCES `estacao` (`id_estacao`) ON DELETE NO ACTION ON UPDATE NO ACTION,  
  ADD CONSTRAINT `fk_linha_has_estacao1_linha1` FOREIGN KEY (`linha_id_linha`)  
  REFERENCES `linha` (`id_linha`) ON DELETE NO ACTION ON UPDATE NO ACTION;  
  
--  
-- Restrições para a tabela `onibus_has_dados`  
--  
ALTER TABLE `onibus_has_dados`  
  ADD CONSTRAINT `fk_onibus_has_dados_dados1` FOREIGN KEY (`dados_id_dados`)  
  REFERENCES `dados` (`id_dados`) ON DELETE NO ACTION ON UPDATE NO ACTION,  
  ADD CONSTRAINT `fk_onibus_has_dados_onibus1` FOREIGN KEY (`onibus_id_onibus`)  
  REFERENCES `onibus` (`id_onibus`) ON DELETE NO ACTION ON UPDATE NO ACTION;
```

## APÊNDICE B – Fotos do Sistema

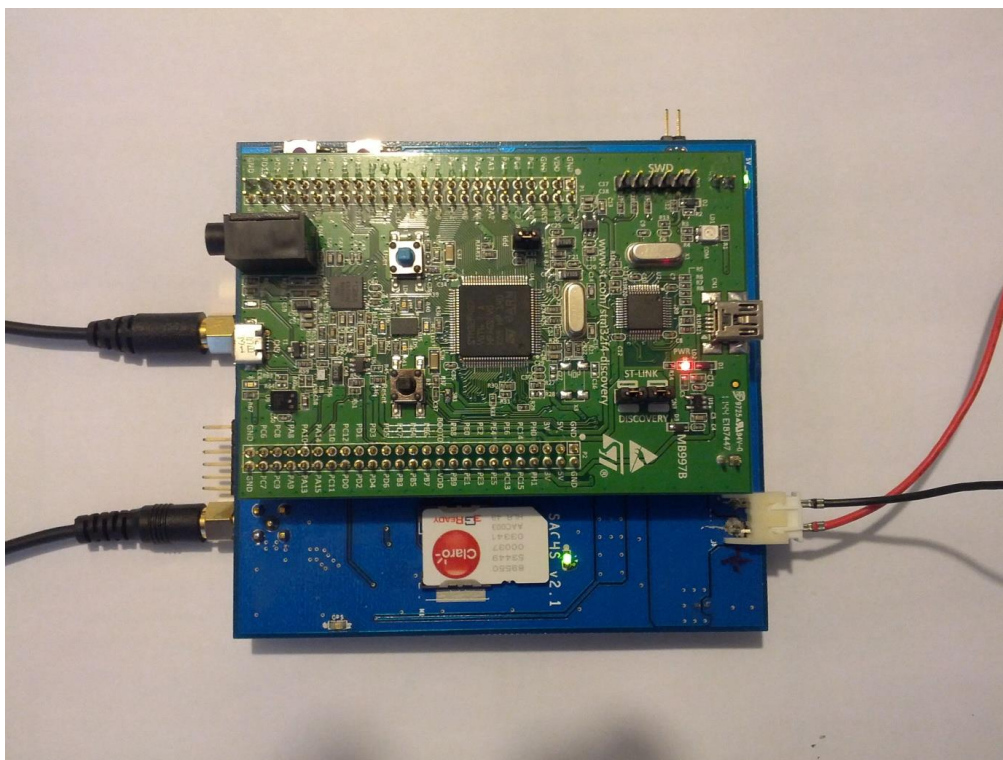


Figura 23 – Face superior da placa de circuito impresso com placa STM32F4 acoplada.  
Fonte: Autoria Própria

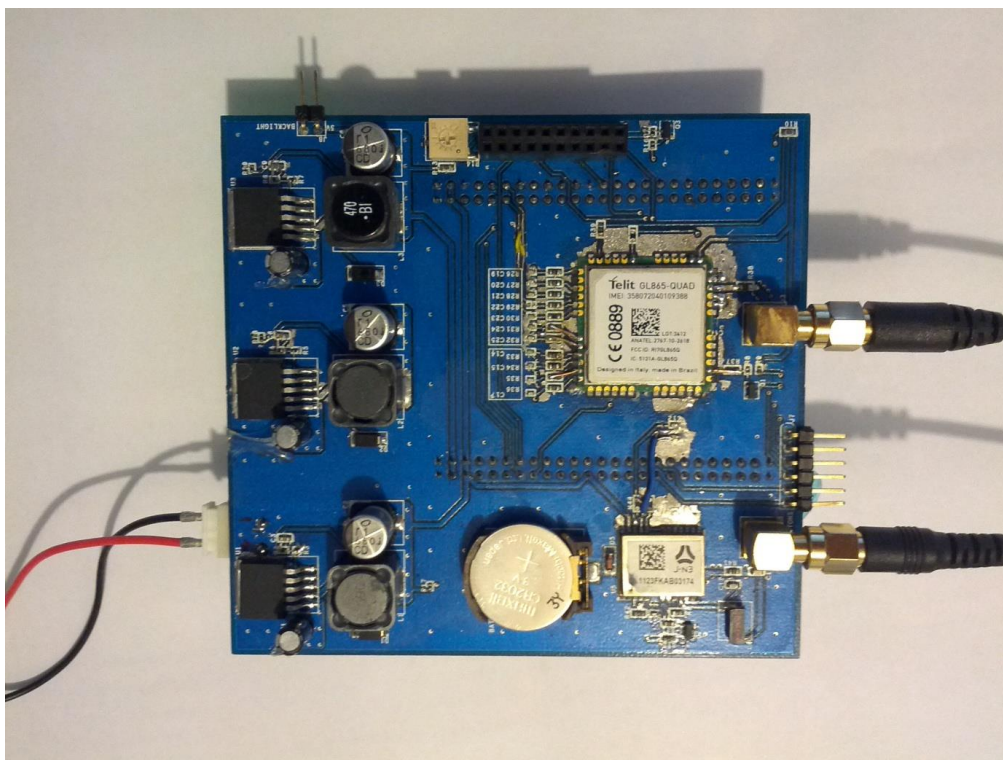
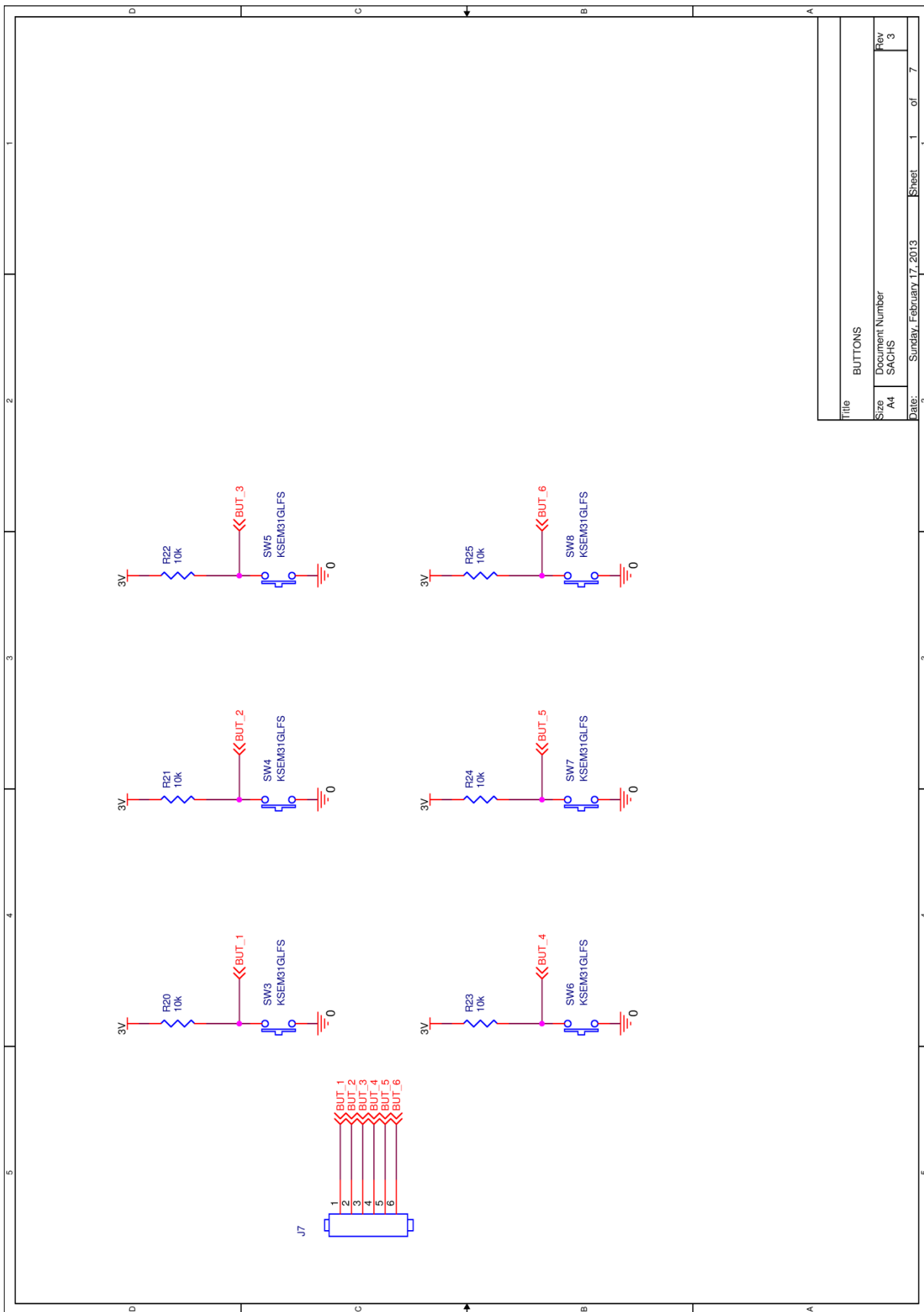
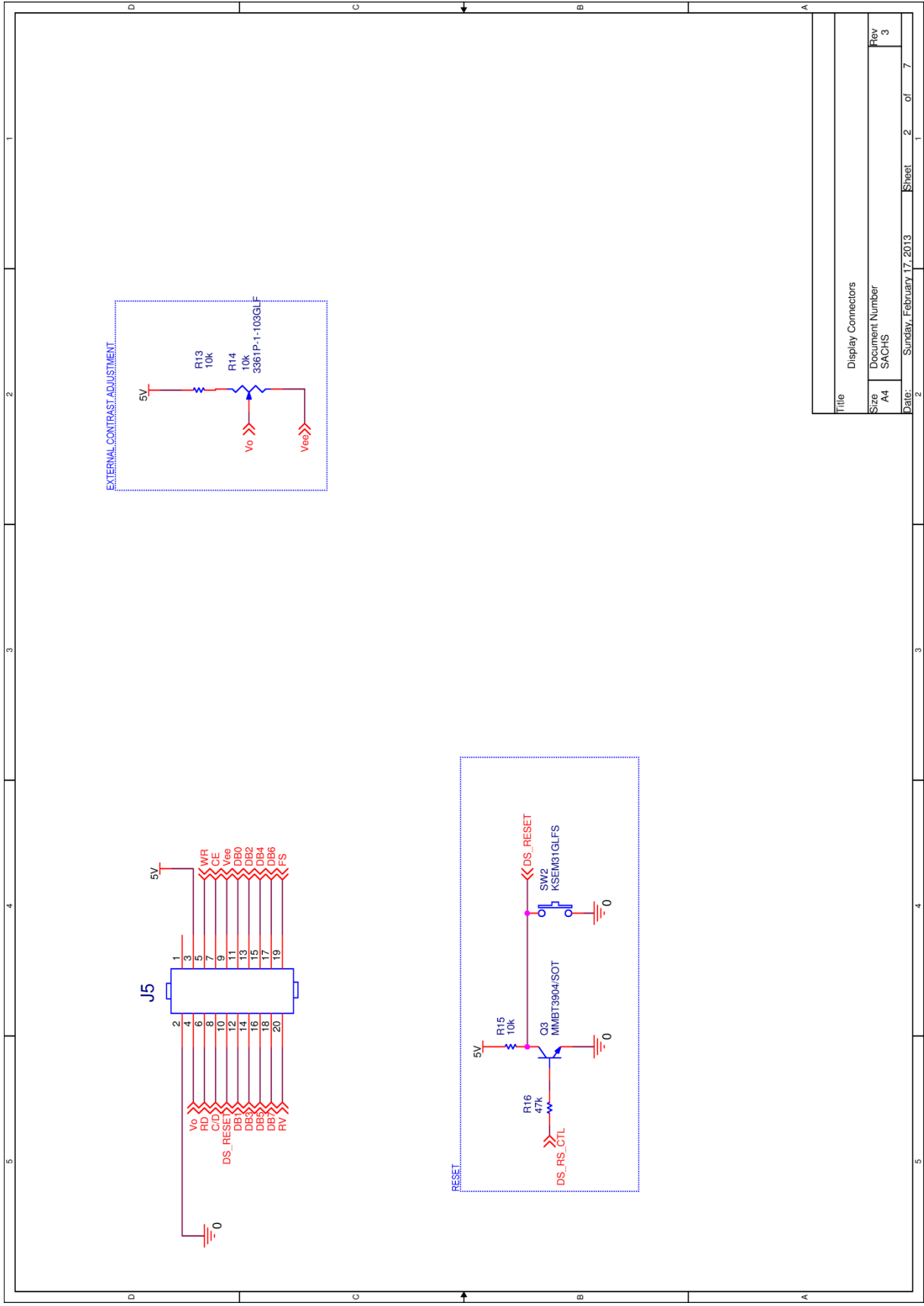


Figura 24 - Face inferior da placa de Circuito impresso.  
Fonte: Autoria Própria.

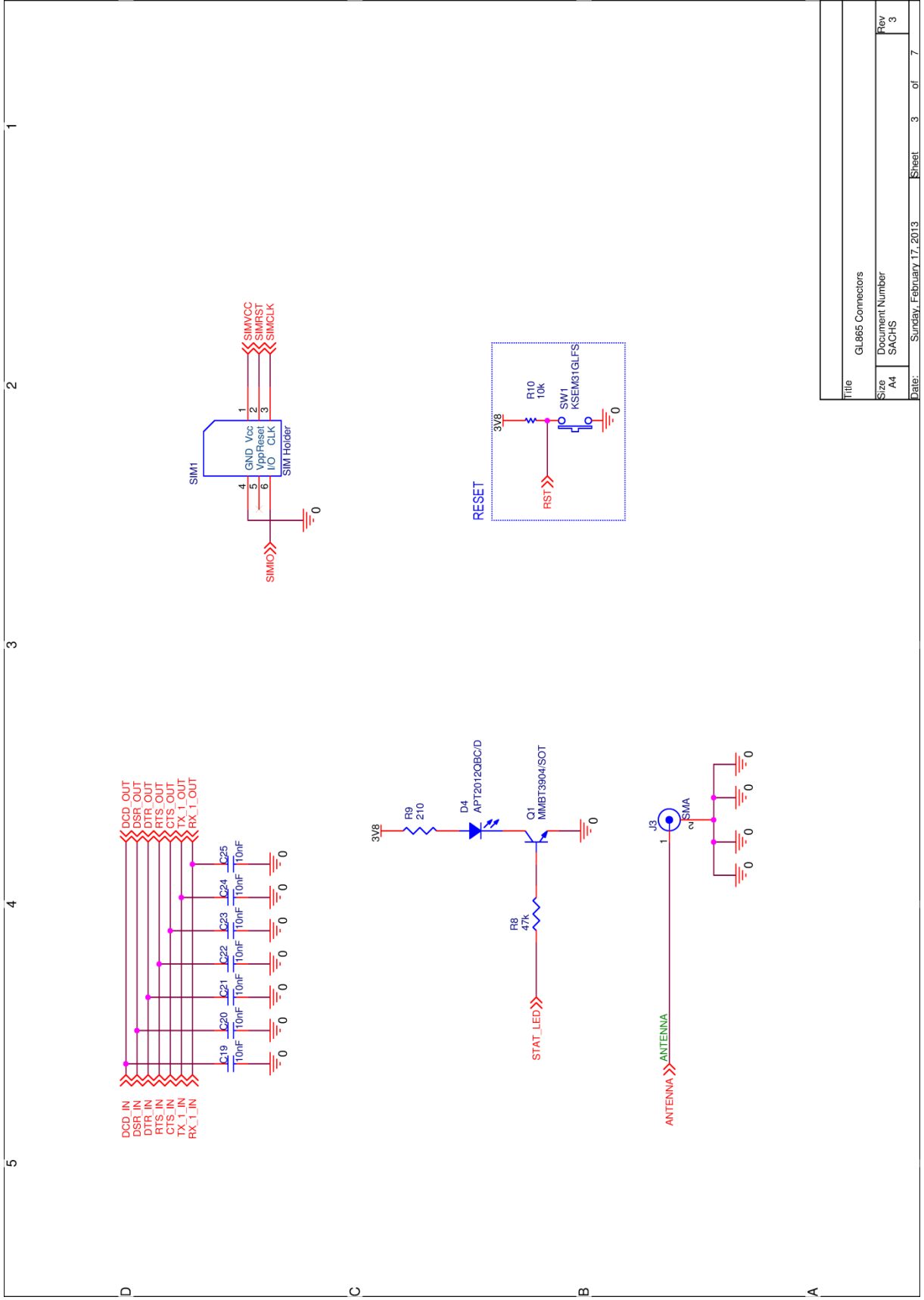
Apêndice C- Esquemáticos



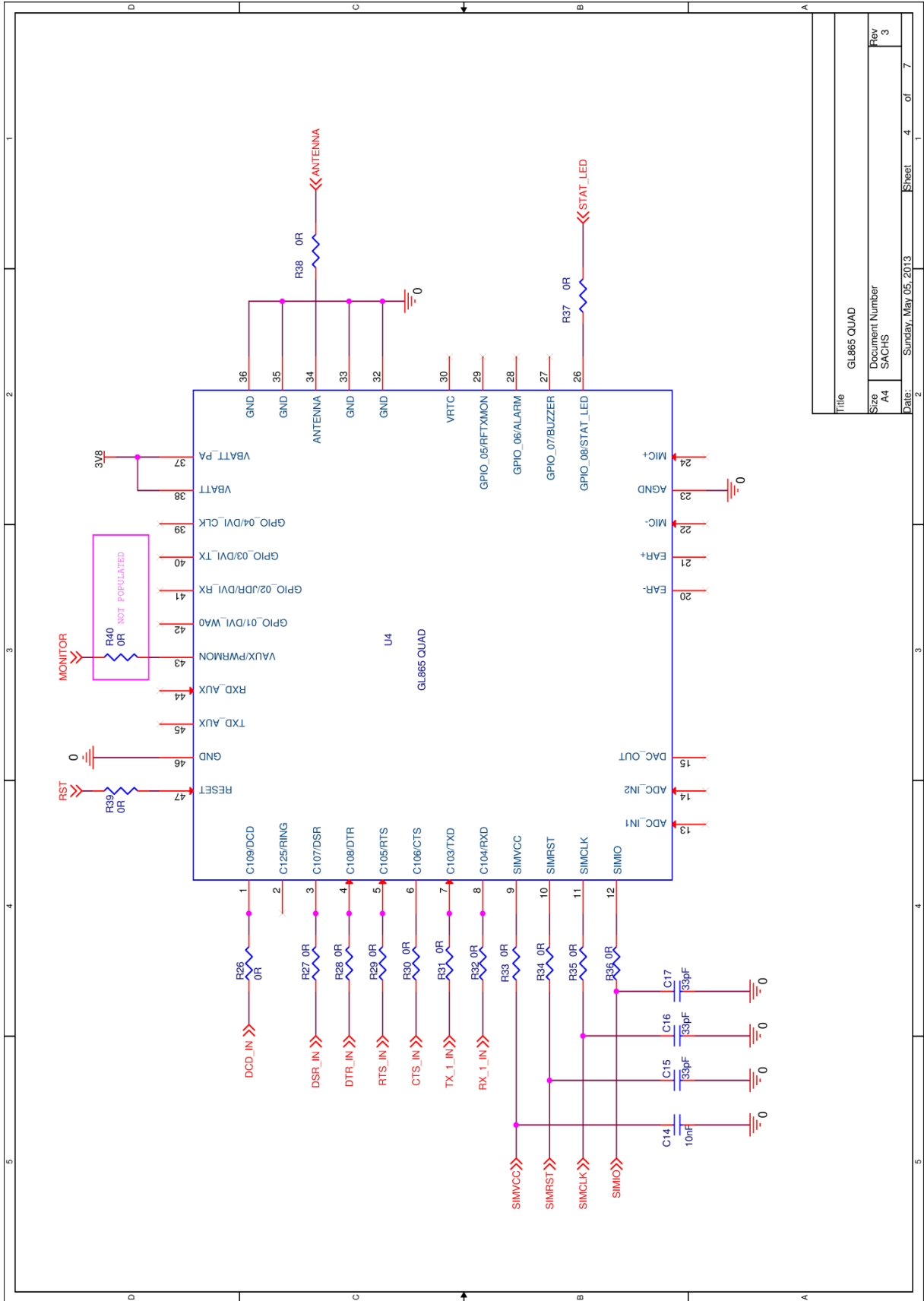
Title		BUTTONS
Size	Document Number	Rev
A4	SACHS	3
Date:	Sunday, February 17, 2013	Sheet 1 of 7



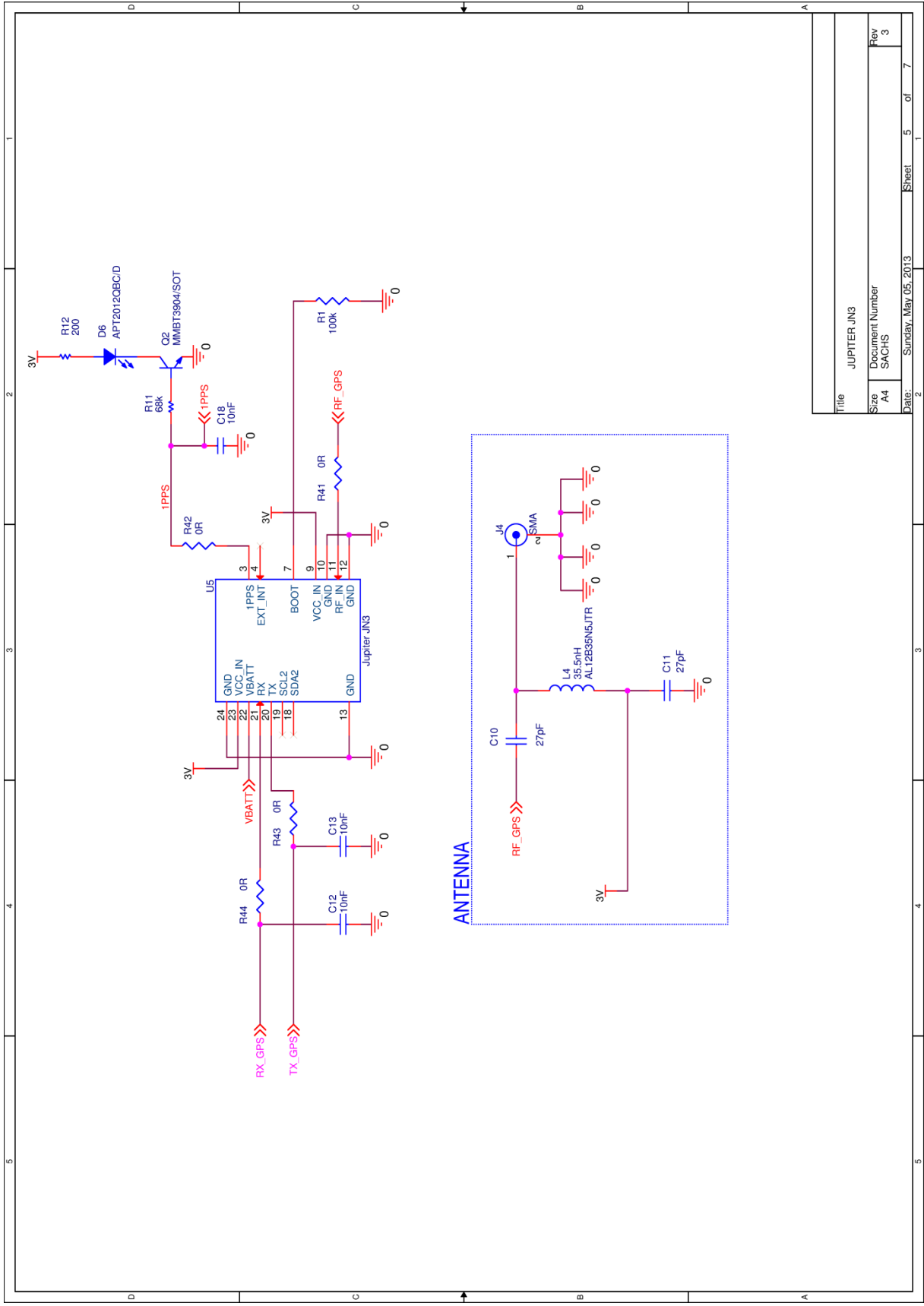
Title	Display Connectors
Size	A4
Document Number	SACHS
Rev	3
Date:	Sunday, February 17, 2013
Sheet	2 of 7



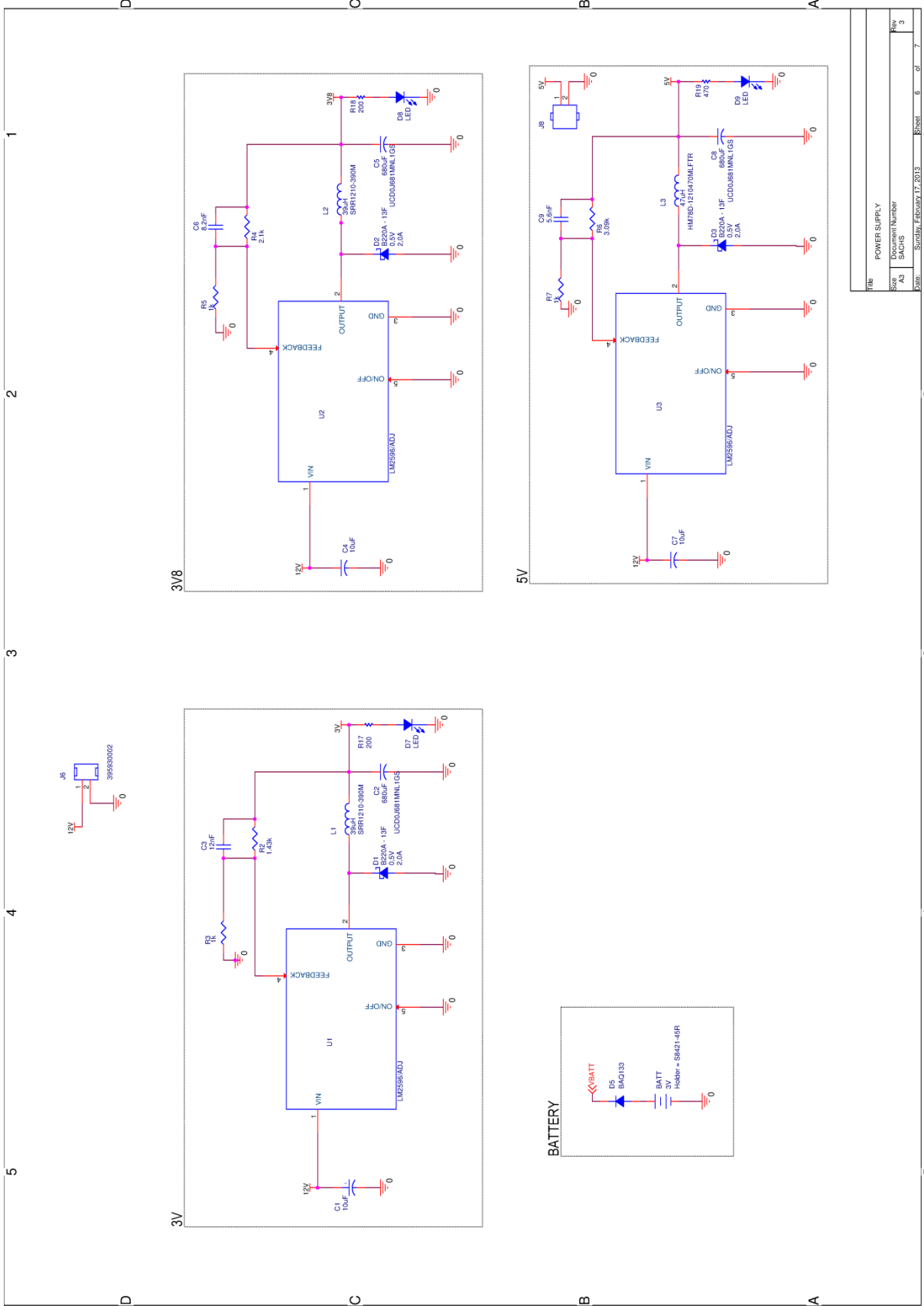
Title		GL865 Connectors	
Size	A4	Document Number	SACHS
Rev	3	Date	Sunday, February 17, 2013
Sheet		3	of 7



Title		GL865 QUAD
Size	Document Number	
A4	SACHS	
Rev	3	
Date:	Sunday, May 05, 2013	Sheet 4 of 7

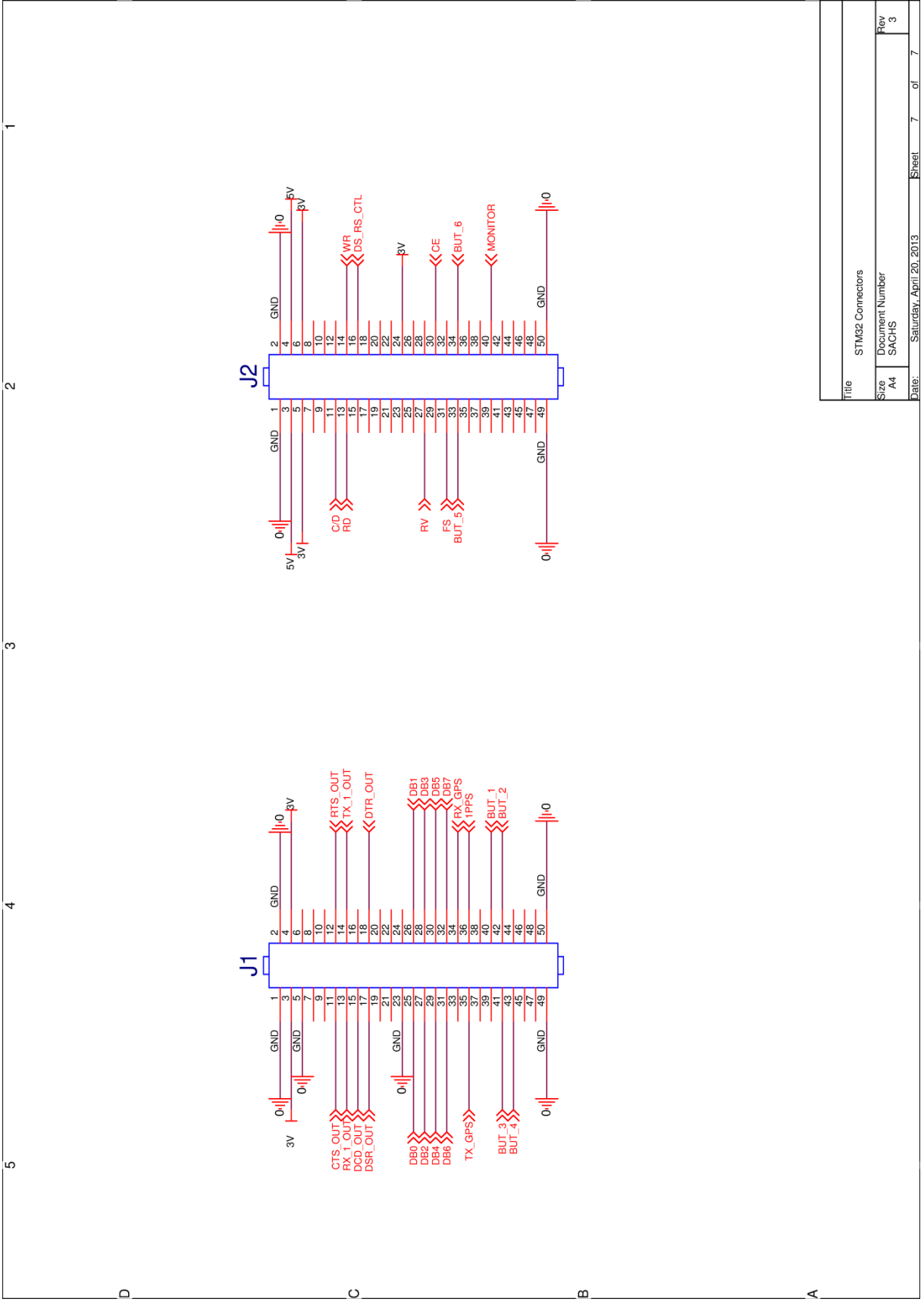


Title		JUPITER JN3
Size	Document Number	A4 SACHS
Date:	Rev	Sunday, May 05, 2013 3 of 7
Sheet		5 of 7



Tab	POWER SUPPLY
Size	Document Number
AD	SACHS
Date:	Sunday, February 17, 2013
Sheet	6 of 7
Rev	3





Title		STM32 Connectors	
Size	A4	Document Number	SACHS
Rev	3	Date	Saturday, April 20, 2013
Sheet		7	of 7

Apêndice D – Fluxograma de operação do *Firmware*