

UNIVERSIDADE SÃO FRANCISCO
CURSO DE ENGENHARIA ELÉTRICA

Plataforma de Telemetria Programável

Área de Telemetria

Por

Raul Rosetto Muñoz

Paulo Eduardo Silveira,
Orientador

Itatiba (SP), Novembro de 2012

UNIVERSIDADE SÃO FRANCISCO
CURSO DE ENGENHARIA ELÉTRICA

Plataforma de Telemetria Programável

Área de Telemetria

Por

Raul Rosetto Muñoz

Monografia apresentada ao Curso de Engenharia Elétrica da Universidade São Francisco, como requisito parcial para obtenção do título de Bacharel em Engenharia Elétrica.

Itatiba (SP), Novembro de 2012

*Aos meus pais e irmãos, por mostrarem todos
os dias o significado da palavra família.*

AGRADECIMENTOS

Agradeço primeiramente a minha família, aos meus pais por sempre me darem exemplos de perseverança, amor e caráter, além de me apoiarem em cada momento da minha vida e mostrarem claramente os melhores caminhos.

Aos meus queridos irmãos por estarem todos os dias ao meu lado mantendo nossa família fortemente unida.

Não posso deixar de falar nas pessoas que foram tão importantes em minha formação profissional. Agradeço ao Sr. Augusto Fattori por me ensinar a dar os primeiros passos em meu primeiro emprego.

Em seguida agradeço aos sócios da atual empresa em que trabalho, Phi Innovations, Flavio Castro Alves, Diego Thuler, Henrique Kuehne e Rafael Coutinho. Eles que me deram a oportunidade de fazer do meu *Hobby* um trabalho.

Também agradeço a minha namorada por me apoiar e acreditar em mim cada momento desde que nos conhecemos. Meus amigos Murilo Toledo, Italo Cunha e Vinicius Campos por estarem todos os dias ao meu lado vencendo cada desafio durante a graduação.

Para finalizar agradeço ao meu professor e orientador Paulo Silveira, que em momento algum deixa de mostrar seu caráter e profissionalismo, além de contribuir fortemente nesse trabalho.

Sumário

Lista de Siglas	vi
Lista de Figuras	vii
Resumo	viii
<i>Abstract</i>	ix
1 INTRODUÇÃO	1
1.1 Objetivo	2
2 REVISÃO BIBLIOGRÁFICA	3
2.1 Telemetria	3
2.1.1 Aquisição de dados	5
2.1.2 Controle	6
2.2 Sistemas Embarcados	7
2.3 Microprocessador	9
2.3.1 AM3358	10
2.4 Linux Embarcado	12
2.5 GPS	13
2.6 GPRS	14
3 METODOLOGIA	16
3.1 Desenvolvimento de <i>Hardware</i>	16
3.1.1 Placa GPS e GPRS	17
3.1.2 Circuito de Interface digital e analógica	21
3.2 Desenvolvimento de <i>Software</i>	24
3.2.1 Desenvolvimento do Linux	25
3.2.2 Desenvolvimento da Aplicação	26
3.2.3 Desenvolvimento do Configurador	27
3.3 Desenvolvimento da Interface WEB	28
4 RESULTADOS	31
5 CONCLUSÕES	35
6 REFERÊNCIAS BIBLIOGRAFICAS	36

Lista de Siglas

ADC	<i>ANALOG-TO-DIGITAL CONVERTER</i>
APN	<i>ACCESS POINT NAME</i>
CI	<i>CIRCUITO INTEGRADO</i>
CPU	<i>CENTRAL PROCESSING UNIT</i>
DMIPS	<i>DHRYSTONE MILLION INSTRUCTIONS PER SECOND</i>
DNS	<i>DOMAIN NAME SYSTEM</i>
GPIO	<i>GENERAL PURPOSE INPUT/OUTPUT</i>
GPRS	<i>GENERAL PACKET RADIO SERVICE</i>
GPS	<i>GLOBAL POSITIONING SYSTEM</i>
HTML	<i>HYPertext MARKUP LANGUAGE</i>
IO	<i>INPUT OUTPUT</i>
MMC	<i>MULTIMEDIACARD</i>
NMEA	<i>NATIONAL MARINE ELECTRONICS ASSOCIATION'S</i>
PCB	<i>PRINTED CIRCUIT BOARD</i>
PROFIBUS	<i>PROCESS FIELD BUS</i>
RF	<i>RADIOFREQUÊNCIA</i>
RTC	<i>REAL TIME CLOCK</i>
SD	<i>SECURE DIGITAL</i>
SIM	<i>SUBSCRIBER IDENTITY MODULE</i>
UART	<i>UNIVERSAL ASYNCHRONOUS RECEIVER/TRANSMITTER</i>
UCP	<i>UNIDADE CENTRAL DE PROCESSAMENTO</i>
URL	<i>UNIVERSAL RESOURCE LOCATOR</i>
USB	<i>UNIVERSAL SERIAL BUS</i>
WEB	<i>WORLD WIDE WEB</i>

Lista de Figuras

Figura 2.1 – Ilustração de um Equipamento de Telemetria.....	5
Figura 2.2 - Diagrama de blocos do processador AM3358	11
Figura 3.1 - Descrição do <i>Hardware</i>	17
Figura 3.2 - Conector P8 e P9 da <i>BeagleBone</i> respectivamente.	18
Figura 3.3 - Circuito do GPS.....	19
Figura 3.4 - Regulador de Tensão para o Modem GPRS.....	19
Figura 3.5 - Circuito do modem GPRS.	20
Figura 3.6 - SIM CARD para modem GPRS.....	20
Figura 3.7 - Layout da placa GPS e GPRS.....	21
Figura 3.8 - Circuito de Entrada.	22
Figura 3.9 - Circuito de Saída.....	23
Figura 3.10 - Circuito da Entrada Analógica.	23
Figura 3.11 - <i>Layout</i> da placa GPS e GPRS.....	24
Figura 3.12 - Arquitetura da Aplicação.....	27
Figura 3.13 - Servidor WEB	31
Figura 4.1 - Camada superior das placas.....	31
Figura 4.2 - Camada inferior das placas.....	32
Figura 4.3 – Placa GPS e GPRS montada.....	32
Figura 4.4 – Placa de interface digital e analógica.	33
Figura 4.5 - Equipamento completo.....	33

Resumo

Este trabalho apresenta o desenvolvimento de um equipamento de telemetria genérico, cujo objetivo é facilitar e dar maior agilidade ao desenvolvimento de novas soluções tecnológicas de mercado. Com a corrente evolução das tecnologias de controle e aquisição de dados a distancia, os profissionais de diversos seguimentos buscam soluções específicas de engenharia para implementar seus sistemas. O equipamento desenvolvido neste trabalho é uma solução de *Hardware* e *Software* que abrange um grande número de funcionalidades normalmente necessárias nessas soluções. Para isso, tem como plataforma uma distribuição Linux e conta com os seguintes recursos: interface ethernet, modem GPRS, modem GPS e entradas/saídas digitais e analógicas. Devido à variedade de periféricos, foram desenvolvidos todos os drivers, uma aplicação base que administra todos os dispositivos e, para visualização e registro das informações coletadas, foi implementada uma interface WEB. Assim sendo, as funcionalidades do equipamento desenvolvido fazem com que o desenvolvimento de novas soluções para um produto final em projetos de telemetria seja mais eficiente e flexível.

PALAVRAS-CHAVE: Telemetria; Aquisição de dados; Linux.

Abstract

This work presents the development of a generic telemetry device, which has the objective of bringing more agility and easiness to the creation of marketable technology solutions. With the ongoing evolution of distance data acquisition and control technology, professionals and practitioners from various market segments seek specific engineering solutions in order to implement their systems. The device here developed is a Hardware and Software solution which encompasses a large number of functionalities usually needed in such applications. For that purpose it has as a platform a Linux distribution and offers the following features: Ethernet interface, GPRS modem, GPS modem and analog and digital Input/Outputs. Due to the variety of peripherals, all the drivers and a base application that manages all the devices had to be developed. For the visualization and registering of all collected information, a WEB interface has been implemented. Therefore, the functionalities implemented for the device enable a more efficient and flexible development of new solutions for final products in telemetry.

KEY WORDS: *Telemetry; Data Acquisition; Linux.*

1 INTRODUÇÃO

Obter informações e poder controlar um sistema a distância vem se tornando a cada dia uma solução imprescindível.

Os recursos tecnológicos para desenvolver um sistema de telemetria vêm se tornando a cada dia mais eficiente e barato. Sendo assim, a possibilidade de se comunicar a longa distância até mesmo com áreas mais remotas, possibilitaram inúmeros segmentos de obter controle a distância em tempo real de suas atividades.

Atualmente a coleta de dados raramente se limita ao espaço de armazenamento, além do que, processar essas informações no equipamento embarcado e tomar decisões autônomas instantaneamente também se faz possíveis com os atuais processadores.

No entanto as demais informações necessárias podem ser enviadas via rede de telefonia móvel para uso da central. Com esses dados a central pode utilizá-los de diversas maneiras como, por exemplo, gerar um alarme ou obter o desempenho do sistema.

Sendo assim profissionais de diversos segmentos buscam soluções específicas de engenharia para manter seus sistemas cada vez mais competitivos. Para isso, investem em projetos de *Hardware* e *Software* para criar produtos específicos de controle e aquisição.

Apesar de inúmeros projetos específicos na área de telemetria ser desenvolvidos a cada dia, existem grandes semelhanças de *Hardware* e *Software* entre esses projetos. Basicamente o projeto utiliza-se de entradas e saídas analógicas, modem GPRS para comunicação utilizando a rede de telefonia móvel e GPS para identificar o posicionamento do equipamento.

O desenvolvimento de uma plataforma de telemetria com essas funcionalidades traria ao mercado agilidade no desenvolvimento de uma solução específica e também a redução de custo por peça, pois o mesmo seria utilizado em diversas áreas podendo ser produzido em larga escala.

1.1 Objetivo

O objetivo desse trabalho é criar uma plataforma de telemetria no qual poderá ser utilizado como base para outros projetos da área. No entanto será desenvolvido o projeto de *Hardware* e *Software* visando atender ao máximo a demanda do mercado por projetos de telemetria.

Para isso o desenvolvimento do *Hardware* busca conter os recursos essenciais para um equipamento completo. O equipamento utilizará de um modem GPRS para comunicação à distância, GPS para informar sua localização e entradas e saídas analógicas para o controle e aquisições de dados.

Tendo os requisitos de *Hardware* especificados, o projeto prevê o desenvolvimento de todos os drivers necessários, uma aplicação base para o equipamento e o servidor que receberá e controlará o equipamento remotamente.

Sendo assim, o equipamento em questão visa facilitar e dar maior agilidade ao desenvolvimento de novas soluções tecnológicas de telemetria, possibilitando assim uma redução de custo em novos projetos e no valor do equipamento, pois o mesmo poderá ser usado em inúmeros segmentos.

2 REVISÃO BIBLIOGRÁFICA

Serão apresentados neste capítulo os princípios de funcionamento de um equipamento programável que possibilita o controle e aquisição de dados remotos.

2.1 Telemetria

A palavra telemetria entrou em uso após a invenção dos meios de transmissão e é definida como a transmissão de medições coletadas a partir de um ponto distante.

A história da telemetria industrial vem acontecendo a mais de 200 anos. Durante todo esse tempo, eventos importantíssimos na história, como a Revolução Industrial, a Segunda Guerra Mundial e o desenvolvimento de computadores, tiveram impactos diretos sobre essa tecnologia. Atualmente, o número de equipamentos a disposição do engenheiro para realizar medições a distância é tão variado quanto o número de aplicações que os suportam. [1]

Basicamente, um sistema de telemetria se resume em coletar dados através de sensores, armazená-los momentaneamente, onde já são consideradas informações do sistema. Essas informações devem ser otimizadas e enviadas para uma central de monitoramento, onde serão usadas como parâmetros de certa atividade. [2]

Já os atuais sistemas de telemetria, permitem ao usuário não só a coleta dos dados, como também retornar ao sistema com rapidez a melhor ação a ser tomada.

Essa tecnologia também viabiliza o controle de diversas atividades, sendo conveniente ao usuário, pois não é necessário ter pessoas em campo e em diversos locais ao mesmo tempo para realizar essas atividades.

Um básico exemplo de aplicação que ilustra o funcionamento desse sistema é o monitoramento de transporte de carga em caminhões: instala-se em um determinado

caminhão um equipamento que coleta informações de posicionamento e de sensores responsáveis por reportar a violação da carga. Os dados são enviados em tempo real a uma central responsável pelo monitoramento da carga e, caso ocorra algum evento indesejável, o usuário ou o próprio sistema de monitoramento pode analisar e tomar alguma ação enviando comandos ao caminhão, como por exemplo, um bloqueio do motor.

O controle preciso e a manipulação dessas informações traz a diversos segmentos da indústria melhores resultados e produtividade, auxiliando em diversos ramos, tais como:

- Meteorologia;
- Agricultura;
- Eficiência energética;
- Consumo de água, eletricidade ou gás;
- Medicina;
- Testes de Voo.

A figura 2.1 mostra uma divisão em blocos de um sistema de telemetria.



Figura 2.1 – Ilustração de um Equipamento de Telemetria.

Os subitens desse capítulo trará uma visão geral de cada bloco que compõem o sistema de telemetria ilustrado na figura 1.

2.1.1 Aquisição de dados

A aquisição de dados é extremamente importante para um equipamento de telemetria, é nela que as informações são obtidas para serem transmitidas e monitoradas.

Identificar o item a ser mensurado é o primeiro passo. Toda grandeza física ou química possível de ser convertida em um sinal elétrico são dados possíveis de ser

coletado. São exemplos de variáveis mensuráveis pela instrumentação: vazão, densidade, corrente elétrica, temperatura, pressão, entre outros.

Podemos encontrar inúmeros sensores que façam a conversão dessas grandezas físicas e químicas para sinais elétricos, porém o importante é ter certeza do formato que o sensor disponibiliza as informações para o equipamento de coleta.

Os sensores podem disponibilizar essas informações de diversas maneiras, sendo a mais simples apenas um sinal digital que pode sinalizar, por exemplo, a ausência ou presença de uma pessoa no raio de cobertura do sensor.

Já os sensores analógicos convertem uma quantidade de grandeza física proporcionalmente em valores de tensão ou corrente elétrica, podendo ser facilmente lido em um canal de conversão analógica para digital.

Existem também dispositivos mais complexos que podem disponibilizar até mais de uma informação. Esses dispositivos normalmente disponibilizam seus dados para outros equipamentos através de comunicação paralela ou serial. Para melhor exemplificar, podemos citar o GPS, que pode coletar diversas informações como posicionamento, altitude, velocidade, entre outras e transferir via UART para outros equipamentos.

2.1.2 Controle

É recente a possibilidade de enviar aos equipamentos de telemetria um sinal de controle, isso ocorre devido a existência de meios de transmissão de dados de duas vias, ou seja, é possível além de coletar as informações remotas do equipamento, enviar instruções remotas para o equipamento.

Com essas instruções o equipamento pode controlar qualquer periférico previamente ligado a ele.

Além do controle remoto dos periféricos, é possível estabelecer regras que também as controlem, tais como, baseado no resultado de um sensor, ligar ou desligar uma válvula.

Podemos exemplificar como periféricos de controle: pistão, válvula, sirene, giroflex, entre outros.

2.2 Sistemas Embarcados

Com o advento dos microprocessadores na década de 1970, uma revolução muito grande ocorreu na área de controle. Pela primeira vez, sistemas relativamente complexos, poderiam ser construídos com um dispositivo simples, um microprocessador. [3]

O uso de microprocessadores possibilitou transformar diversos sistemas de malhas abertas, em sistemas de malhas fechadas. Com o uso de entradas e saídas que compõem um microprocessador, e o desenvolvimento de uma lógica interna, esses tipos de proezas foram facilmente implementados.

Um exemplo que ilustra essa evolução é notável nos motores dos carros. Nos últimos 20 anos, as emissões de poluentes pelos motores de combustão, reduziram cerca de 90%. Isso ocorre porque antigamente, com o uso do carburador, tínhamos um sistema de malha aberta. Hoje temos sistemas de malha fechada, que com base em informações de diversos sensores, otimizam o desempenho e minimizam as emissões. [3]

Basicamente um sistema embarcado é a combinação de *Hardware* e *Software*, podendo conter peças mecânicas ou elétricas adicionais para executar uma função específica. [4]

A concepção de um sistema embarcado está em contraste direto com um computador pessoal. Ele também é composto de *Hardware* e *Software* e componentes mecânicos. No entanto o mesmo não é projetado para executar funções específicas. Pelo contrário, é capaz de fazer muitas coisas diferentes. [4]

A lista a seguir apresenta algumas características específicas de um sistema embarcado. [3]

- Dedicados a tarefas específicas, enquanto que os PCs são plataformas de computação genérica;
- Suportados por uma ampla gama de processadores;
- Projetos de baixo custo;
- Restrições de tempo real;
- As consequências de uma falha de *Software* são muito mais graves em sistemas embarcados do que em sistemas *desktop*;
- Sistemas embarcados diversas vezes tem que operar sob condições ambientais extremas;
- Possuem muito menos recursos que sistemas *desktop*;
- Requerem ferramentas e métodos especializados para ser eficientemente projetado;
- Microprocessadores embarcados muitas vezes oferecem ferramentas dedicadas de depuração dos circuitos.

Conforme listado, existem diversos itens que provam a complexidade de um sistema embarcado.

Apesar de ser dedicado a atividades específicas, grande parte das aplicações de um sistema embarcado são muito críticas, não permitindo travamento ou um erro de *Software*.

Um exemplo é um medidor de glicemia usado diariamente por pessoas diabéticas. Esse aparelho tem a função de medir a taxa glicêmica, a partir de uma pequena amostra de sangue. Com base no valor reportado por esse pequeno aparelho, o usuário aplica a quantidade necessária de insulina para correção dessa taxa. Erros de *Software* ou *Hardware* nesse caso podem ocasionar até a morte do usuário. Isso mostra que esses tipos de sistemas embarcados não podem permitir erros, um dos motivos de ser um sistema unicamente dedicado a isso.

Hoje é possível encontrar no mercado uma grande quantidade de fabricantes e tipos de microprocessadores. Em consequência da grande concorrência e necessidade de mercado, inovações em tecnologia e custo surgem a cada dia.

Sendo assim, o desenvolvedor de *Software* embarcado tem que buscar cada vez mais o desenvolvimento de uma aplicação portátil para outros modelos e arquiteturas.

Outra importante característica que vem crescendo cada vez mais dentro do sistema embarcado é o fato dos processadores englobarem dentro do mesmo chip, cada vez mais periféricos, motivo este que os faz caminharem para um menor custo.

2.3 Microprocessador

O microprocessador é um circuito integrado com a capacidade de executar programas, fazer cálculos e tomar decisões, de acordo com as instruções armazenadas na memória. Por essas razões é popularmente conhecido como o cérebro de um computador. [5]

Conforme Vasconcelos, “Os microprocessadores formam uma parte importantíssima do computador, chamada de UCP (Unidade Central de Processamento), ou em inglês, CPU (Central Processing Unit). Antes da existência dos microprocessadores, as CPUs dos computadores eram formadas por um grande número de chips, distribuídos ao longo de uma ou diversas placas. Um microprocessador nada mais é que uma CPU inteira, dentro de um único CHIP”. [5]

Atualmente existem microprocessadores que além de um processador de alta performance e entradas e saídas digitais, integram elementos adicionais em sua estrutura interna como:

- UART - Interface serial amplamente utilizada por periféricos em sistemas embarcados. Possibilitando a comunicação do processador com modems 3G, GPRS, GPS, placas de aquisição de dados, placas IO, entre outros periféricos.
- RTC - Contador incrementado precisamente a cada segundo, permitindo ao sistema calcular precisamente o tempo e a associa-lo ao Horário Global.

- ADC - Conversor de dados analógicos em dados digitais, item imprescindível para leitura de sensores e dados analógicos.
- MMC – Interface com cartão de memória, disponibilizando memória flash para o sistema operacional e para o sistema de arquivos.

Além de todas essas funcionalidades em um único chip, já existem processadores com baixo consumo de energia e custo acessível.

É possível encontrar diversos fabricantes de microprocessadores, entre eles podemos citar a Atmel®, Freescale®, Intel Logic®, Texas Instruments® entre outras.

Tomaremos como base o processador AM3358 fabricado pela Texas Instruments, processador no qual conta com uma grande variedade de periféricos integrados no mesmo chip.

2.3.1 AM3358

Baseado no núcleo ARM Cortex-A8, são reforçados com processamento gráfico, periféricos e opções de interface como EtherCAT e PROFIBUS. Outra característica importante é o fato desse dispositivo suportar os sistemas operacionais de alto nível mais usados no mercado: [6]

- Linux;
- Windows CE;
- Android.

O uso de um sistema operacional traz para a aplicação em desenvolvimento inúmeras vantagens. Pois ele já prevê o gerenciamento de processo e memória, suporte a diversos protocolos, sistema de arquivos, entre outros.

A família Cortex-A8, foi desenvolvida visando atender uma ampla variedade de dispositivos móveis, incluindo telefones celulares, set-top boxes e sistemas de entretenimento automotivos.

Destacam-se no mercado pelo alto desempenho, fornecendo mais de 2000 Dhrystone MIPS (DMIPS) para aplicações destinadas ao usuário consumindo menos de 300 mW.

Na figura 2.2 [6] é mostrado o diagrama de blocos correspondente ao AM3358.

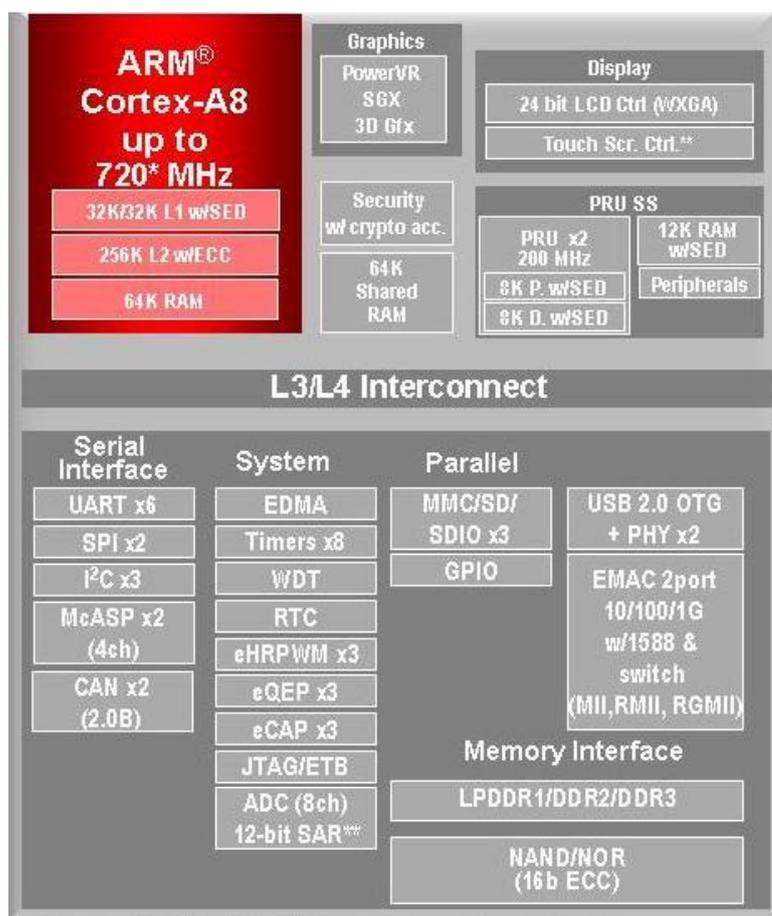


Figura 2.2 - Diagrama de blocos do processador AM3358

Após ter escolhido o processador que será usado, é recomendável conferir se o fabricante disponibiliza algum kit ou produto de desenvolvimento usando o mesmo.

Para o processador AM3358, a Texas Instruments disponibiliza uma CPU completa e com fácil acesso aos pinos do processador, além de disponibilizar ao usuário todo esquema elétrico, arquivos de fabricação e autorizar a reprodução.

Conferir se o fabricante ou alguma outra fonte disponibiliza o sistema operacional preparado para o processador e placa que será utilizado.

Esses tipos de cuidados antes de iniciar o projeto facilita o desenvolvimento do produto, pois, não será necessário criar o *Hardware* nem portar o sistema operacional.

2.4 Linux Embarcado

Atualmente o sistema operacional Linux é encontrado em grande quantidade de dispositivos eletrônicos. Ao se estudar o sistema é possível encontrar diversas fontes que mostram suas vantagens e como o esse sistema operacional tomou tanta proporção.

Inicialmente chamado de Unix, foi criado por Dennis Ritchie e Ken, no entanto, diversos motivos trouxeram o sistema para atual magnitude. [7]

O primeiro fato que colaborou para a difusão do Unix foi o desenvolvimento do sistema na linguagem C, desenvolvida por Ritchie, para substituir e superar as limitações das demais linguagens do momento.

Outra razão para seu crescimento e popularidade se deu através *Free Software Foundations* (FSF), organização sem fins lucrativos que deu inicio ao o que mais tarde seria conhecido como movimento *Open Source*. [8]

Há diversas razões para a escolha de construir um sistema embarcado usando Linux, seja pela sua flexibilidade, robustez, preço, a comunidade que o desenvolve ou até o grande numero de fornecedores. [8]

Ao mencionar a palavra Linux, nos referimos a um termo muito abrangente, rigorosamente falando se refere ao *Kernel* mantido pela “Linux Torvalds” e distribuído com o mesmo nome através do repositório principal. Essa base de código inclui apenas o *Kernel* e não utilitários de qualquer natureza.

Uma vez executado o *Kernel*, ele nunca é trocado ou removido do controle até ser desligado. Além de tudo ele fornece os recursos essenciais do sistema. Na verdade ele

controla todo o *Hardware*, e fornece maior nível de abstração, como por exemplo, processos, *sockets* e sistemas de arquivos para os diferentes *Softwares* que são executados no sistema. [8]

Para termos um sistema embarcado usando Linux é preciso adaptar o *Kernel* para o dispositivo do sistema embarcado. Outra opção é usar um *Kernel* já otimizado por terceiros, que já está adaptado para uma configuração de *Hardware* especial.

Abaixo estão algumas razões para a escolha do Linux como sistema operacional em projetos embarcados. [9]

Suportam diversos dispositivos de *Hardware*, provavelmente mais do que qualquer outro sistema operacional;

Suporte a uma grande variedade de aplicações e protocolos de rede;

Diferente dos sistemas operacionais proprietários o Linux pode ser usado sem custo algum;

Grande comunidade de desenvolvedores, permitindo um desenvolvimento rápido de novas arquiteturas de *Hardware*, plataformas e dispositivos;

Numero crescente de fornecedores de *Hardware* e *Software*, praticamente todos os fabricantes de primeira linha de chips e fornecedores de *Software* independentes, agora suportam Linux.

São inúmeras as características de um sistema operacional Linux, vale destacar que com todas essas vantagens e níveis de abstração, o desenvolvimento de um produto embarcado se torna muito semelhante com o desenvolvimento de *Software* para PCs, facilitando o desenvolvimento do projeto, pois, diversas coisas necessárias para o produto já são previstas no sistema operacional.

2.5 GPS

A sigla GPS vem do idioma inglês “*Global Positioning System*”, em outras palavras significa Sistema de Posicionamento Global. A tecnologia GPS começou a ser desenvolvida pela Marinha, Força Aérea e o Exército Americano por volta de 1957. Já em 1983, após a colisão de dois aviões deixando 269 mortes, foi proposta a disponibilização do GPS para uso Civil.

Basicamente a tecnologia GPS é capaz de obter a sua localização com o uso de um receptor de radio. Ao obter o sinal de diversos satélites que orbitam a Terra é possível calcular a posição exata do receptor. Atualmente o sistema é livre, e de baixo custo, basta comprar um receptor liga-lo corretamente e será possível obter a localização exata do mesmo. [10]

Para o desenvolvimento de sistemas eletrônicos, o conhecimento a fundo da tecnologia não se faz necessário visto que, para obter as informações de um GPS existem modems no mercado que preparados para receber o sinal e fazer a interface com equipamentos eletrônicos.

Existem no mercado modems que utilizam diversas interfaces, entre as mais utilizadas estão USB (*Universal Serial Bus*) e UART (*Universal Asynchronous Receiver/Transmitter*). As informações fornecidas pelos modems podem ser obtidas em diversos protocolos.

O protocolo mais utilizado é o NMEA, além as informações de latitude e longitude são possíveis obter diversas outras informações como data, hora, altitude, numero de satélites, velocidade, entre outros.

Esse trabalho será desenvolvido com o modem GPS MAX 6 fabricado pela u-blox. Esse modem conta com uma interface UART, meio o qual será feita a aquisição de informações e em formato NMEA[11].

2.6 GPRS

Tem-se hoje a telefonia móvel como umas das tecnologias mais crescentes e investidas no mercado de comércio e pesquisa no mundo inteiro, nota-se este fato verificando o número crescente de operadoras de celular. [12]

Sendo assim a cobertura de telefonia móvel vem aumentando cada vez mais, tornando possível o uso da tecnologia não apenas para telefones celulares, mais também, para comunicação com equipamentos remotos.

Teve início em 1920, onde em caráter experimental departamentos de policias nos Estados Unidos começaram a usar radio telefonia. A tecnologia teve um grande sucesso com embarcações marítimas não sendo muito adequado para comunicação em terra. Alguns problemas ainda impactavam na tecnologia, o equipamento era volumoso, prédios e obstáculos atrapalhavam na propagação do sinal. [12]

Já na década de 1930, com o desenvolvimento de frequência modulada, utilizada na segunda guerra mundial, alguns avanços foram realizados a ponto de em 1940 ser disponibilizado em algumas grandes cidades o serviço de telefonia móvel. O sistema ainda tinham limitações, e foram necessários muitos anos para a telefonia móvel se tornar uma tecnologia acessível. [12]

A evolução da tecnologia de telefonia móvel teve grandes evoluções, evoluções divididas em 3 gerações caminhando para 4 geração. Basicamente a cara geração evoluções como velocidade de transmissão foi sendo evoluídas.

Aplicar essa tecnologia em equipamentos eletrônicos que não sejam celulares vem se tornando cada dia mais comum. Modens 2G e 3G atualmente são componentes acessíveis no mercado.

Com o uso de interfaces USB ou UART para comunicar com o modem é possível enviar e receber informações a distancia utilizando a rede de telefonia móvel.

Para esse projeto será utilizado modem 2G LEON-100, fabricado pela u-blox. Para comunicação com o modem será utilizado a interface UART. [13]

3 METODOLOGIA

Este capítulo tem o intuito de mostrar os principais passos para construção do equipamento de telemetria programável, cujo projeto abrange uma ampla gama de atividades em *Hardware* e *Software*, nas quais serão descritas nesse capítulo.

3.1 Desenvolvimento de *Hardware*

Tomando como base para o projeto de *Hardware*, a placa *BeagleBone* descrita no capítulo anterior, fez-se necessário o desenvolvimento e produção de duas novas placas.

No conceito em que a *BeagleBone* é comercializada, tem-se uma placa enxuta, porém com dois conectores usados para expansão, os quais podem ser usados para o desenvolvimento de placas anexas conforme a necessidade de cada projeto.

Assim sendo e visando o desenvolvimento de um equipamento de telemetria com capacidade de transferência de dados remotos, posicionamento global e interfaces digitais e analógicas, fez-se necessário o desenvolvimento e produção de duas novas placas anexas, nas quais uma contém conter GPS / GPRS e a outra interface digital / analógica conforme ilustra a figura 3.1.

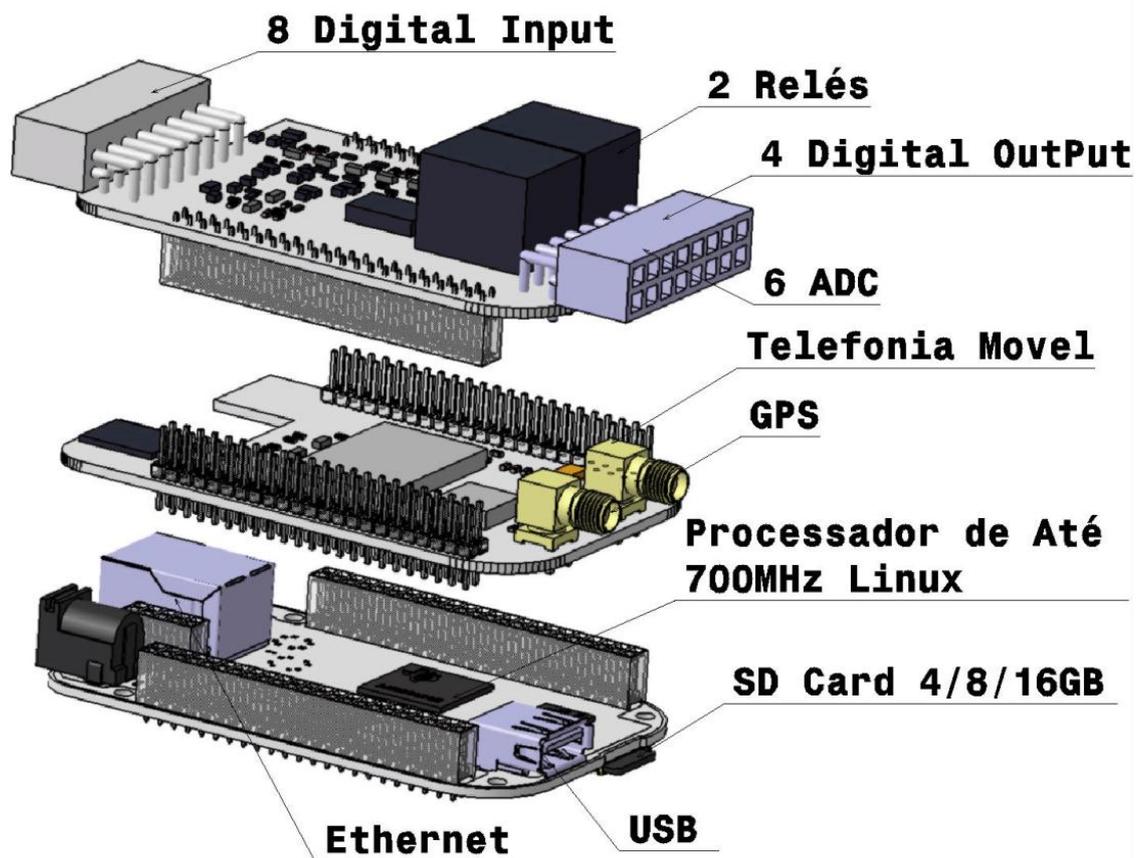


Figura 3.1 - Descrição do *Hardware*.

3.1.1 Placa GPS e GPRS

Tomando como base os kits de desenvolvimento do fabricante U-blox, *datasheets* dos componentes e o manual da *BeagleBone*, foi desenvolvido o esquema elétrico para que o GPS MAX 6 e o GPRS LEON-100, sejam anexados ao projeto.

O desenvolvimento do esquema elétrico necessitou atenção em três itens:

- Alimentação;
- Interface de Comunicação;

- Antena.

Para a alimentação e comunicação foi necessário definir os pinos ligados a *BeagleBone*. No manual de referencia da *BeagleBone* é possível ver os pinos disponíveis nos conectores externos.

A figura 3.2 mostra os dois conectores de expansão disponíveis na *BeagleBone* e , destacado em vermelho, todos os pinos usados para comunicação com a placa GPS e GPRS.

SIGNAL NAME	PROC	CONN	PROC	SIGNAL NAME	SIGNAL NAME	PIN	CONN	PIN	SIGNAL NAME
	GND	1	2	GND		GND	1	2	GND
GPIO1_6	R9	3	4	T9	GPIO1_7	VDD_3V3EXP	3	4	VDD_3V3EXP
GPIO1_2	R8	5	6	T8	GPIO1_3	VDD_5V	5	6	VDD_5V
TIMER4	R7	7	8	T7	TIMER7	SYS_5V	7	8	SYS_5V
TIMER5	T6	9	10	U6	TIMER6	PWR_BUTTON*	9	10	A10
GPIO1_13	R12	11	12	T12	GPIO1_12	UART4_RXD	11	12	U18
EHRPWM2B	T10	13	14	T11	GPIO0_26	UART4_TXD	13	14	U14
GPIO1_15	U13	15	16	V13	GPIO1_14	GPIO1_16	15	16	T14
GPIO0_27	U12	17	18	V12	GPIO2_1	I2C1_SCL	17	18	B16
EHRPWM2A	U10	19	20	V9	GPIO1_31	I2C2_SCL	19	20	D18
GPIO1_30	U9	21	22	V8	GPIO1_5	UART2_TXD	21	22	A17
GPIO1_4	U8	23	24	V7	GPIO1_1	GPIO1_17	23	24	D15
GPIO1_0	U7	25	26	V6	GPIO1_29	GPIO3_21	25	26	D16
GPIO2_22	U5	27	28	V5	GPIO2_24	GPIO3_19	27	28	C12
GPIO2_23	R5	29	30	R6	GPIO2_25	SPI1_D0	29	30	D12
UART5_CTSN	V4	31	32	T5	UART5_RTSN	SPI1_SCLK	31	32	VDD_ADC(1.8V)
UART4_RTSN	V3	33	34	U4	UART3_RTSN	AIN4	33	34	GNDA_ADC
UART4_CTSN	V2	35	36	U3	UART3_CTSN	AIN6	35	36	A5
UART5_TXD	U1	37	38	U2	UART5_RXD	AIN2	37	38	A7
GPIO2_12	T3	39	40	T4	GPIO2_13	AIN0	39	40	C7
GPIO2_10	T1	41	42	T2	GPIO2_11	CLKOUT2	41	42	C18
GPIO2_8	R3	43	44	R4	GPIO2_9		43	44	GND
GPIO2_6	R1	45	46	R2	GPIO2_7		45	46	GND

Figura 3.2 - Conector P8 e P9 da *BeagleBone* respectivamente.

Toda alimentação foi ligada ao conector P9 da *BeagleBone*, onde para o GPS é alimentado direto com 3,3V. Já sua comunicação é ligada na interface UART2. A figura 3.3 mostra o esquema elétrico correspondente a essas ligações.

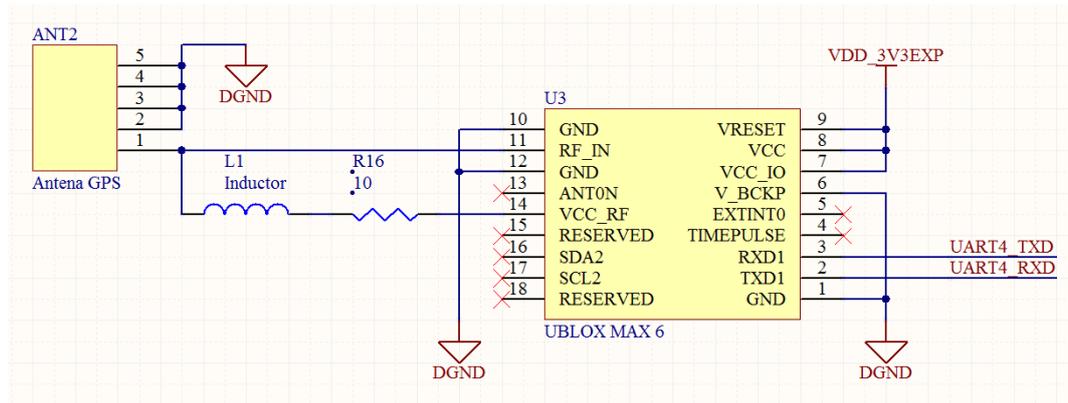


Figura 3.3 - Circuito do GPS.

O modem GPRS também foi alimentado pela *BeagleBone* porém, para sua alimentação é necessário 3,8V, tensão não encontrada na placa base. No entanto, foi necessário o uso do regulador LP38501, componente no qual converte 5V em 3,8V.

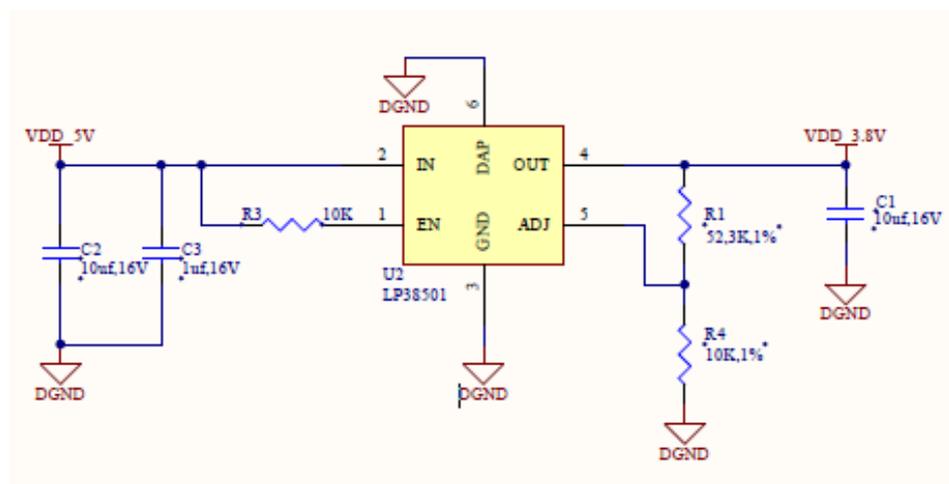


Figura 3.4 - Regulador de Tensão para o Modem GPRS.

O modem se comunica usando a mesma interface do GPS. A única diferença entre a comunicação do GPS e do GPRS é o nível de tensão em que o GPRS opera em Tx e Rx, 2,8V. Para fazer com que a interface UART da *BeagleBone*, que opera em 3,3V fique no nível de 2,8V foi adicionado resistores de 100Ω em serie com a linha.

Além da comunicação e alimentação do modem, foi necessário a ligação de dois pinos de controle, um de *power on/off* e outro para reset, a figura 3.5 mostra a comunicação UART, alimentação e pinos de controle usados no circuito do modem Leon-100.

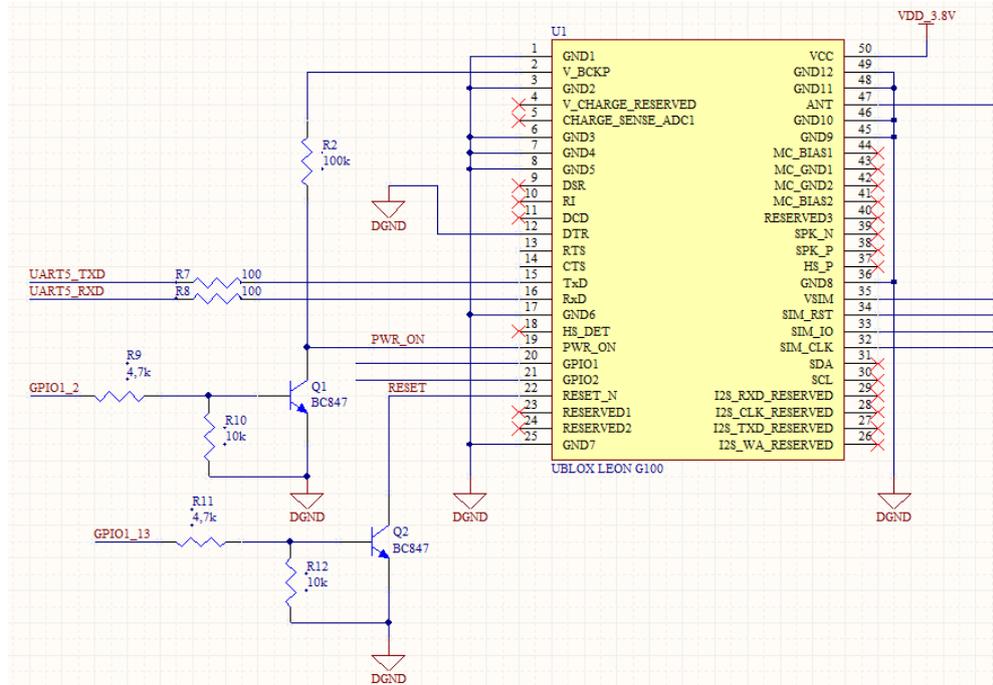


Figura 3.5 - Circuito do modem GPRS.

Para Ligação do SIM CARD ao modem GPRS, foram apenas seguidas as instruções do fabricante, a figura 3.6 mostra o esquema elétrico dessa ligação.

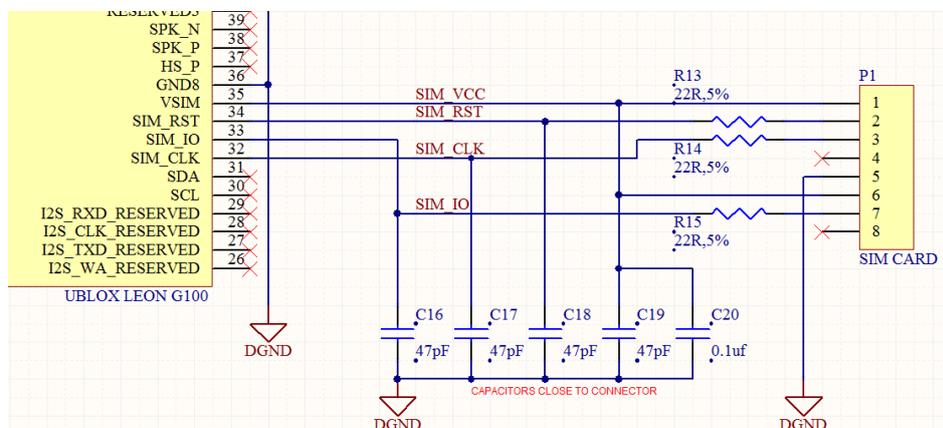


Figura 3.6 - SIM CARD para modem GPRS.

Tendo todo o circuito desenhado, foi possível realizar o layout da placa de circuito impresso. A placa foi desenhada em duas camadas e com especificações de produção buscando obter um bom preço de fabricação e sem deixar sair do tamanho original da BeagleBone.

A figura 3.7 mostra o layout da placa final, em vermelho está a camada superior e em azul a camada inferior.

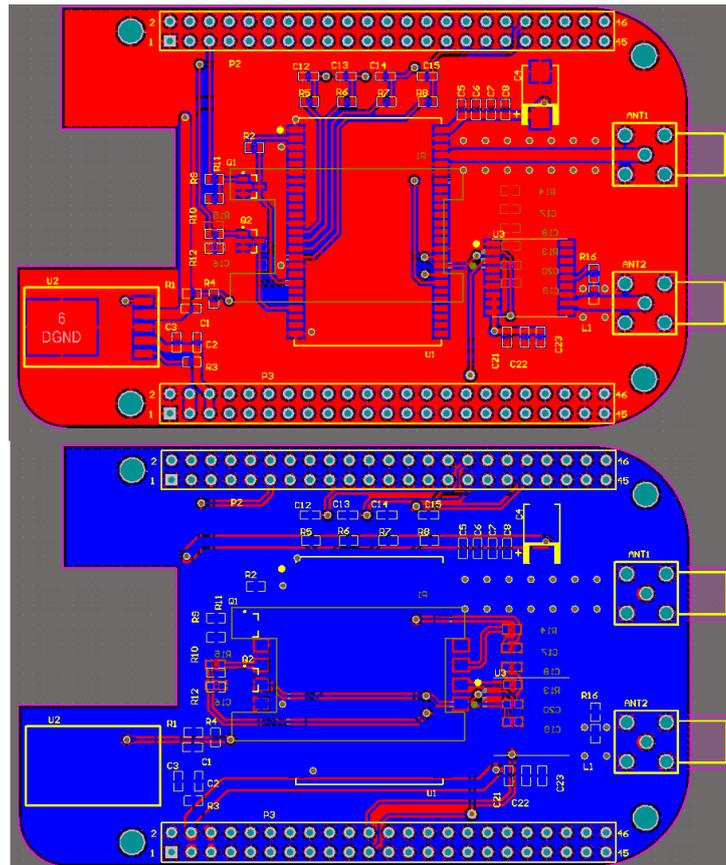


Figura 3.7 - Layout da placa GPS e GPRS.

3.1.2 Circuito de Interface digital e analógica

Tomando como base a placa anterior, foi necessário o desenvolvimento de outro circuito anexo para o projeto. Esse circuito tem o intuito de realizar a interface entre as GPIO e ADC da *BeagleBone* sensores e atuadores do usuário.

A placa de interface conta com 8 entradas digitais, 4 saídas digitais *open-drain*, 2 relés com contatos de até 15A e 6 entradas analógicas de 12bits.

A figura 3.8 ilustra um dos 8 circuitos de entrada, esse circuito foi desenvolvido para manter isolar a entrada digital do processador da entrada disponível para ligação do usuário, protegendo assim o circuito do processador de dano por corrente ou tensão elevada.

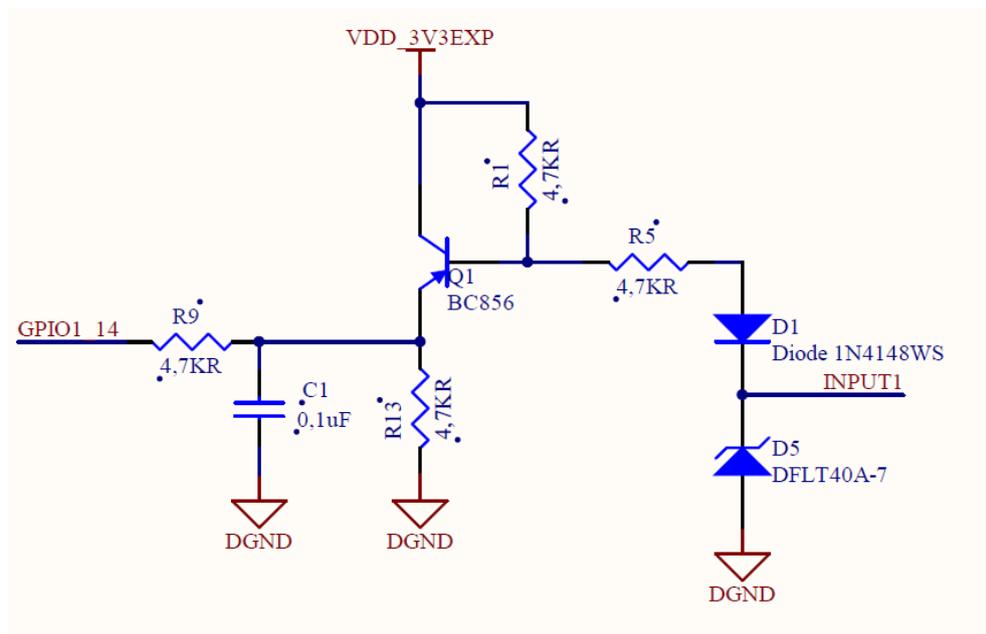


Figura 3.8 - Circuito de Entrada.

Para o circuito de saídas foi usado o CI ULN2803 para realizar a interface entre circuito digital de baixa potencia com a saída de alta corrente. A figura 3.9 mostra o circuito desenvolvido assim como a ligação dos dois relés ligados na placa.

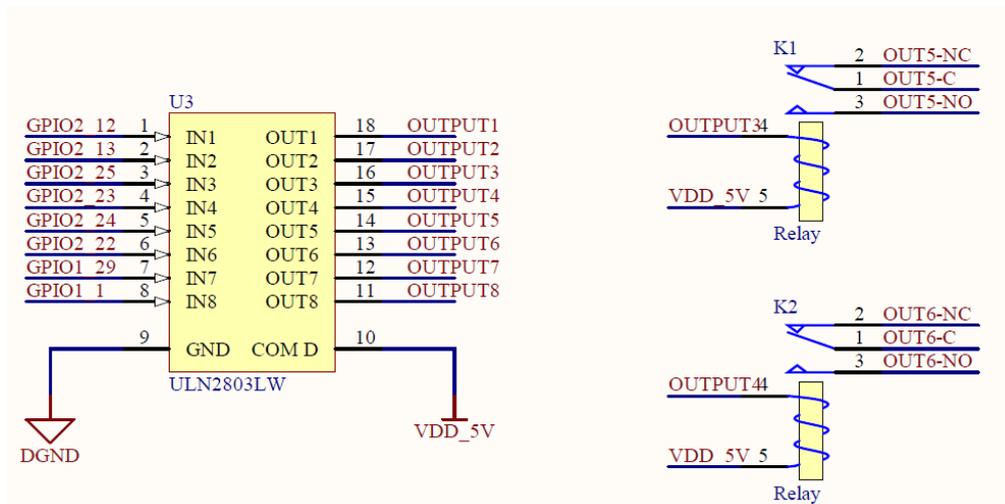


Figura 3.9 - Circuito de Saída.

Nas entradas analógicas disponíveis na *BeagleBone*, foi necessário fazer um divisor resistivo para poder elevar o nível de tensão da entrada analógica. As entradas analógicas da placa trabalham de 0V a 1,8V. Com o circuito da Figura 3.10 esse nível passa a ser de 0V a 10V, possibilitando assim diversificar o número de sensores conectados no mesmo.

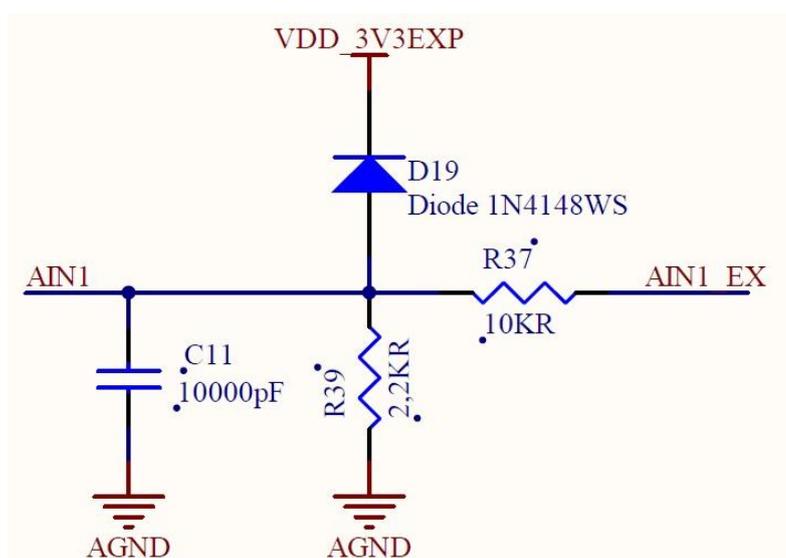


Figura 3.10 - Circuito da Entrada Analógica.

Com as mesmas especificações da placa GPS e GPRS, foi desenhada e fabricada a placa de Interface.

A figura 3.11 mostra o *layout* da placa final, em vermelho está a camada superior e em azul a camada inferior.

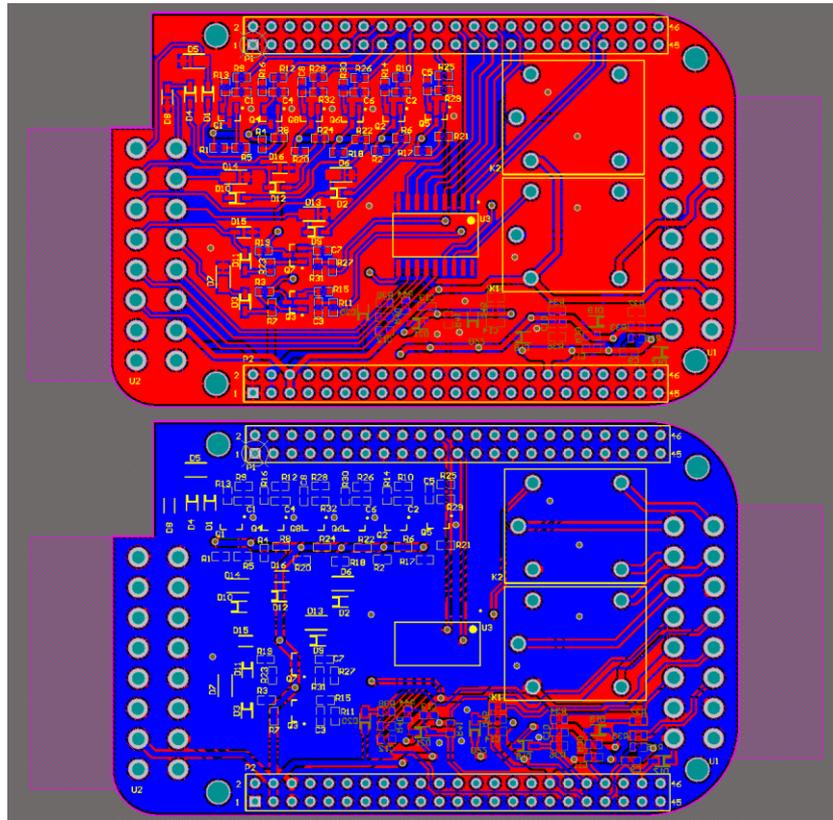


Figura 3.11 - *Layout* da placa GPS e GPRS.

3.2 Desenvolvimento de *Software*

O desenvolvimento do *Software* foi dividido em três atividades principais: primeiramente a preparação do Linux otimizado para a placa *BeagleBone*, em seguida o desenvolvimento da aplicação responsável por toda a lógica do produto e ao final um configurador WEB para fornecer parâmetros para a aplicação.

3.2.1 Desenvolvimento do Linux

Tendo como objetivo criar um Linux para rodar na *BeagleBone*, uma placa relativamente popular nos sistemas embarcados, é possível encontrarmos o suporte a *BeagleBone* facilmente nas ferramentas de compilação.

Para esse projeto a ferramenta usada foi o Yocto, projeto *open source* que fornece modelos, métodos e ferramentas para criar um sistema baseado em Linux personalizado para seu produto embarcado.

Todo o Linux Embarcado criado para *BeagleBone* foi configurado e compilado usando um computador com sistema operacional Linux, especificamente o Ubuntu 12.04.

Para incluir suporte a placa *BeagleBone* ao Yocto foi necessário adicionar duas camadas ao projeto, meta-ti mantida pela Texas Instruments e meta-openembedded, camada *open source* para sistemas embarcados.

Tendo as camadas necessárias a configuração de compilação para *BeagleBone* será encontrada e compilada.

Concluído a compilação foram gerados quatro arquivos fundamentais para o funcionamento do sistema operacional:

- MLO;
- u-boot.img;
- ulmage;
- core-image-minimal-*BeagleBone*.tar.bz2.

Esses arquivos foram devidamente carregados no cartão SD e testados com o terminal serial a inicialização do sistema operacional.

Tendo o sistema operacional funcionando, o próximo passo foi personalizar o sistema operacional com itens necessários para o produto.

Entre as configurações feitas, está a inicialização automática da aplicação, configurações de rede padrão, suporte ao modem GPRS, habilitado o acesso a entradas e saídas digitais através de arquivos (*user spaces*), entre outros.

Depois de realizadas as personalizações do sistema operacional, toda infraestrutura para aplicação estava desenvolvida.

3.2.2 Desenvolvimento da Aplicação

O desenvolvimento da aplicação principal foi estruturada em módulos principais, tendo o objetivo de obter uma arquitetura onde cada módulo fica responsável por suas atividades.

Todos os módulos só se comunicam com o módulo administrador, responsável por toda organização de atividades entre os demais módulos.

Na camada abaixo do administrador foram desenvolvidos os módulos responsáveis por cada base da aplicação.

O módulo GPS tem apenas realiza aquisição das informações do GPS, decodifica e disponibiliza para o administrador as informações de GPS validas.

Toda interface com entrada e saída digital e analógica é feita pelo módulo de alarme. Através do driver `/sys/class/` é possível alterar e ler os valores das entradas e saídas. Outra função importando desse modulo é gerar os eventos em situações configuráveis de alarme, um exemplo é ao identificar a mudança de estado em uma entrada, gera-se um evento de alarme para o administrador.

Os eventos gerados com suas respectivas informações, data, hora, coordenada GPS, entre outras, são passados ao modulo de armazenamento para que o mesmo imediatamente salve as informações no sistema de arquivo.

O equipamento conta também com um módulo dedicado apenas com trocas de informações com o servidor, sendo possível enviar eventos a central, identificar o posicionamento do equipamento e também acionar uma saída do equipamento.

A estrutura de módulos é ilustrada na figura 3.12, a aplicação foi desenvolvida de modo que a utilização do equipamento seja flexível, para isso o modulo administrador lê informações extraídas do configurador e modela a aplicação para operar da maneira configurada.

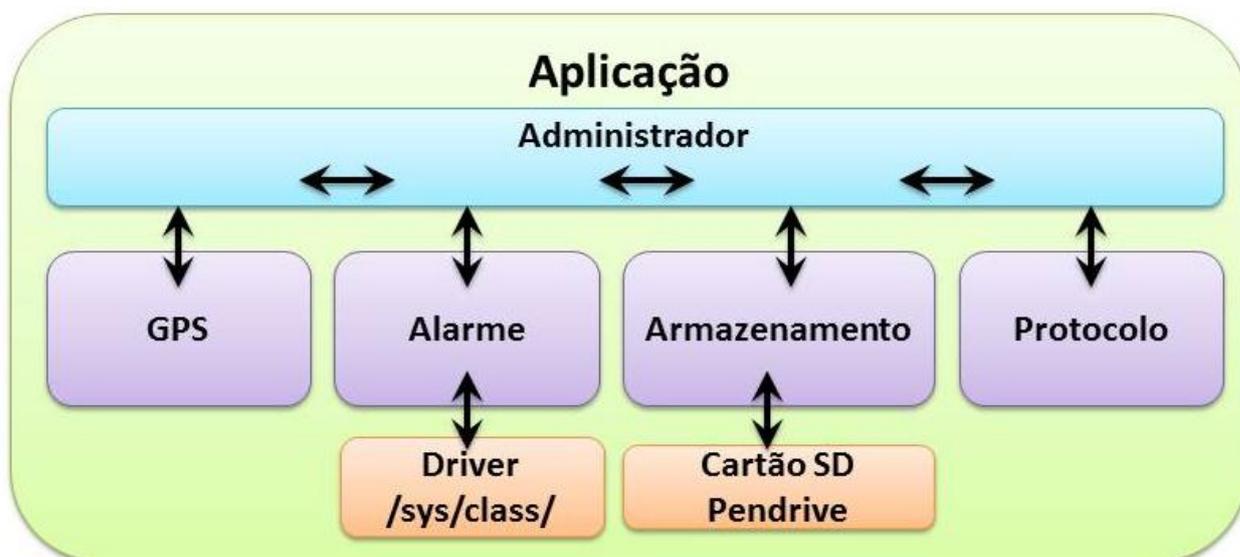


Figura 3.12 - Arquitetura da Aplicação.

3.2.3 Desenvolvimento do Configurador

Com o intuito de tornar o equipamento mais versátil, diversas informações importantes para a aplicação e também para o sistema operacional, foram desenvolvidas de modo que seja possível alterá-las.

Para isso foi desenvolvido um servidor WEB, hospedado diretamente no Linux do equipamento, tornando possível o acesso local e remoto para alteração dos parâmetros embarcados.

O servidor foi desenvolvido usando a linguagem “javascript” do lado do servidor com a biblioteca “node js” e saída em HTML, onde ao acessar a URL do equipamento na porta correta é possível visualizar o site com as informações já configuradas no equipamento.

Após ter acesso a tela inicial é possível alterar cada informação prevista e atualizar o equipamento. Essas informações alteram funcionalidades como:

- Configurações de rede;
- Configurações de DNS;
- Configurações de Servidor;
- Uso de GPS;
- Uso de ethernet ou GPRS;
- Configurações de APN;
- Configurações de Entradas;
- Configurações de Saídas;
- Configurações de ADC.

3.3 Desenvolvimento da Interface WEB

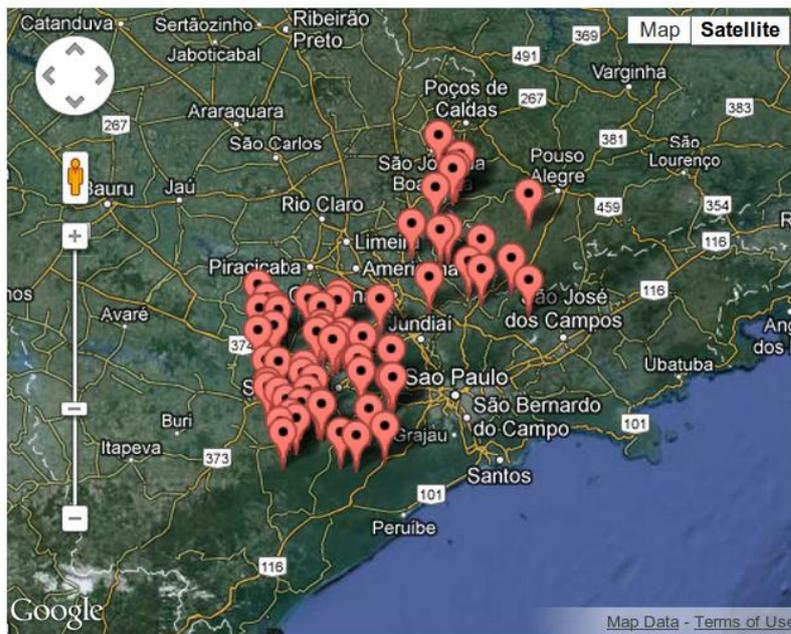
Tendo o equipamento desenvolvido e operando em campo, a visualização das informações e alarmes coletados se torna um grande diferencial para o sistema. Sendo assim, foi desenvolvido e preparado um servidor para coletar e mostrar as informações coletadas pelo equipamento em uma página WEB.

O servidor conta com uma aplicação em C que possibilita múltiplas conexões na qual é responsável por decodificar as mensagens do protocolo e inseri-las em um banco de dados MySQL.

Com base nas informações inclusas no banco de dados, foi desenvolvido uma aplicação em Java EE, sobre o servidor Tomcat 7 e o *framework* MVC VRaptor. Nesta aplicação é possível verificar os dados do GPS, as entradas analógicas, as entradas digitais e enviar comandos para o equipamento.

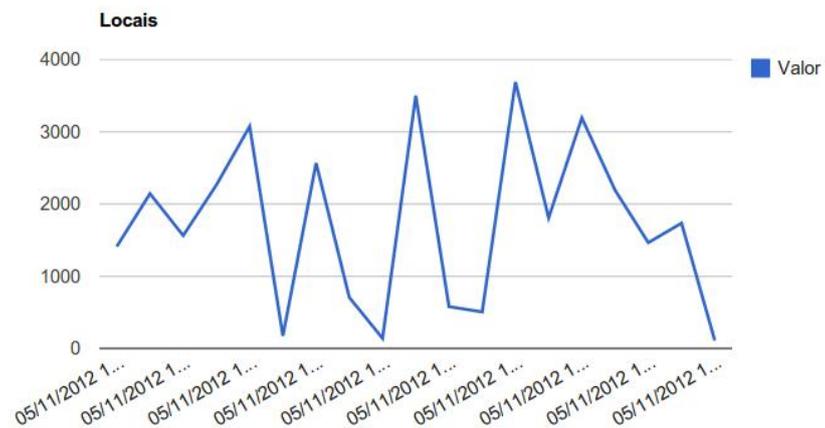
A figura 3.13 mostra o servidor web contendo dados coletados do equipamento desenvolvido.

Map



Entradas analógicas

Entrada: AN2



Entradas digitais

Entrada: IN3

Data/Hora	Dipositivo	Entrada	Valor
05/11/2012 12:02:52	Raul	IN3	1
05/11/2012 12:02:55	Raul	IN3	0
05/11/2012 12:08:12	Raul	IN3	1
05/11/2012 12:08:15	Raul	IN3	1
05/11/2012 12:08:17	Raul	IN3	1
05/11/2012 12:08:19	Raul	IN3	0
05/11/2012 12:08:22	Raul	IN3	0

Outputs

Out 1
 Out 2
 Out 3
 Out 4
 Out 5
 Out 6

Figura 3.13 - Servidor WEB

4 RESULTADOS

A primeira etapa do projeto foi desenvolvido os esquemas elétricos e layouts das placas de circuito impresso, como ferramenta para desenvolvimento dos mesmos foi usado o *Software Altium Designer*.

Tendo os *layouts* concluídos foi decidido fabricar as PCBs em uma empresa especializada, pois as mesmas continham trilhas de RF críticas e também duas camadas.

A figura 4.0 ilustra a camada superior das placas de circuito impressos e a figura 4.1 a camada inferior após a produção.

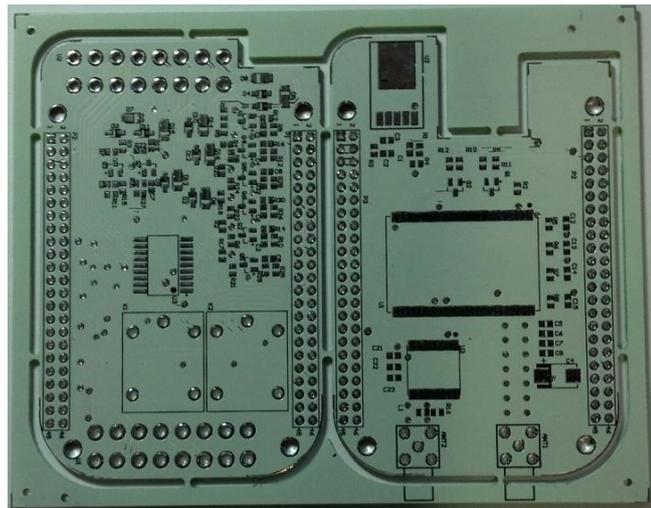


Figura 4.1 - Camada superior das placas.

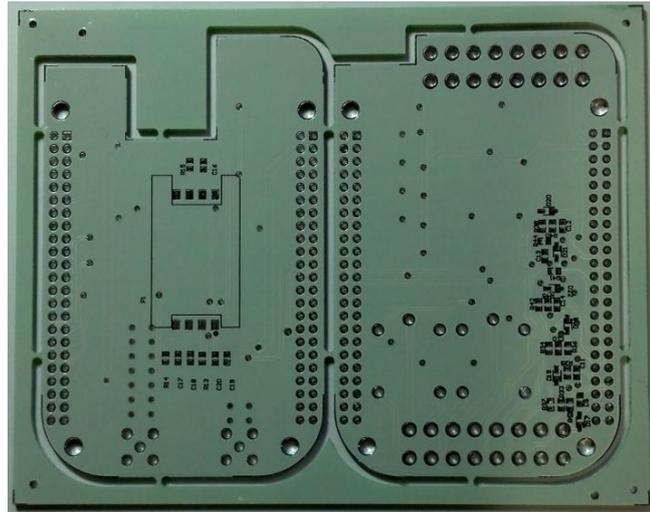


Figura 4.2 - Camada inferior das placas.

Após a produção das placas foram comprados todos os componentes e soldados nas suas respectivas placas. A figura 4.3 e 4.4 mostram a placa contendo GPS e GPRS e também a placa de interface digital e analógica respectivamente.



Figura 4.3 – Placa GPS e GPRS montada.

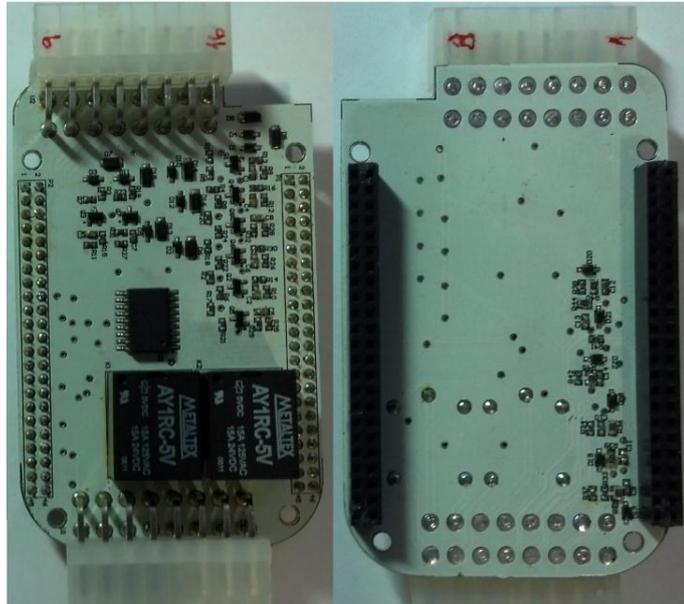


Figura 4.4 – Placa de interface digital e analógica.

Tendo todo o *Hardware* montado as placas foram integradas a *BeagleBone* tendo ao final o *Hardware* pequeno e completo. A figura 4.5 mostra o *Hardware* final do projeto.

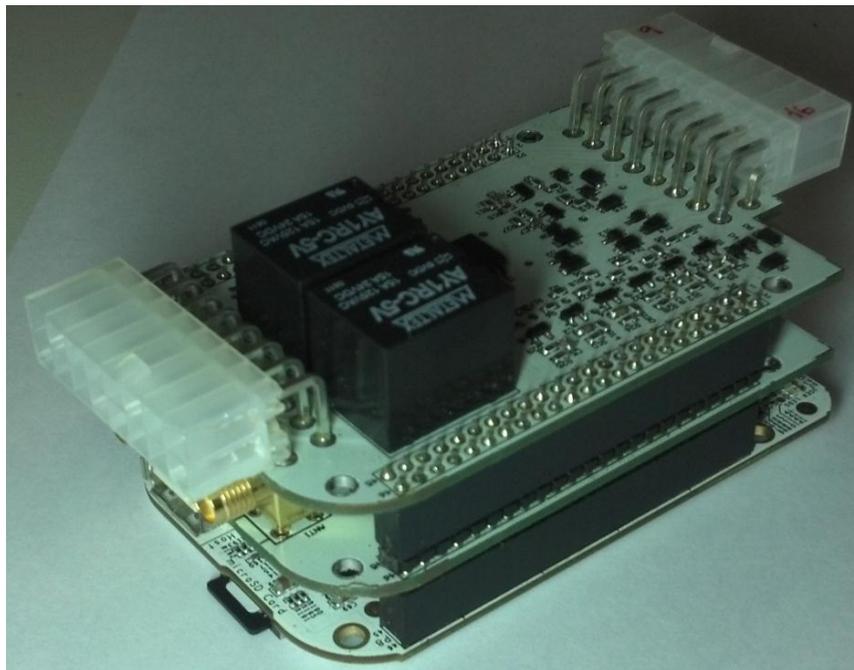


Figura 4.5 - Equipamento completo.

Tendo o *Hardware* completo foi possível preparar o Linux e desenvolver o *Software*. Todos os recursos do sistema operacional previstos para o equipamento foram instalados corretamente e tiveram o funcionamento esperado.

Para configuração do equipamento foi desenvolvido um servidor WEB que fica hospedado no equipamento e o mesmo tornou possível a configuração de todos os parâmetros previstos na metodologia.

Com base em todas as configurações a aplicação principal desenvolvida em C embarcada no equipamento obteve os resultados previstos. A leitura de todas as entradas digitais e analógicas, as principais regras configuradas para gerar alarmes e o envio dessas informações e também de coordenadas GPS para o servidor foram testados e validados.

Além das funcionalidades previstas no equipamento embarcado, foi preparado o servidor e desenvolvido uma interface WEB para visualização de todas as informações que o equipamento pode coletar.

A interface conta com mapa para visualizar a posição do equipamento, gráficos para ilustrar o comportamento das entradas analógicas e uma tabela que mostra o estado das entradas digitais.

Tendo como principal funcionalidade informar o servidor de eventos estimulados pelas entradas digitais e analógicas, foi feito o principal teste configurando o equipamento para gerar evento ao acionar determinada entrada digital e ao entrar em um nível de conversão na entrada analógica.

Esses eventos foram reportados no servidor e o mesmo atualizou o posicionamento do equipamento no mapa e os valores de gráficos e tabelas.

5 CONCLUSÕES

Foi desenvolvido o projeto de Hardware e Software para construção da plataforma de telemetria, assim como produzido o protótipo e realizados os testes necessários para validação do equipamento proposto.

Buscando atender o maior numero de aplicações na área de telemetria, foram estudados os requisitos necessários para o desenvolvimento do produto. Sendo assim o projeto buscou abranger o maior numero de funcionalidades tanto em Hardware como em Software.

Ao final obteve-se um equipamento completamente funcional, apto a receber determinada configuração do usuário, controlar o sistema aplicado remotamente e também disponibilizar os dados coletados para a visualização remota.

A visualização foi feita através de um servidor WEB no qual possibilita o usuário de forma fácil o entendimento das informações coletadas pelo equipamento.

6 REFERÊNCIAS BIBLIOGRÁFICAS

1. Frank Carden, Russell P. Jedlicka, Robert Henry. Telemetry systems engineering. Editora Artech House telecommunications library 2002
2. Pelichek, Daniel. Estudo da Telemetria para Aquisição, Processamento e Transmissão de dados em Sistemas Remotos, 2009, 51 f. Trabalho de Conclusão de Curso – Curso de Engenharia Elétrica, Universidade de São Paulo, São Carlos, 2009.
3. Arnold S. Berger. Embedded Systems Design: An Introduction to Processes, Tools, and Techniques. Editora CMP Books 2002.
4. Michael Barr, Anthony Massa. Programming Embedded Systems. Editora O'Reilly & Associates, Inc, 2006.
5. Vasconcelos Laércio. *Hardware na Prática*, 2ª edição. Laércio Vasconcelos Computação, Av. Rio Branco, 146/1229, Centro, Rio de Janeiro. Disponível em: < www.laercio.com.br >. Acesso em: 15 de abril, 2012.
6. Texas Instruments. Datasheet [AM335x ARM Cortex-A8 Microprocessors \(MPUs\). Rev. B.](#) : Texas Instruments, Post Office Box 655303, Dallas, Texas 75265 Copyright © 2012, Texas Instruments Incorporated, 2012. Disponível em: < <http://www.ti.com/lit/ds/symlink/am3358.pdf>>. Acesso em: 11 de abril, 2012.
7. Love, Robert. Linux *Kernel* Development, 3ª edição. Editora Developer's Library, 2010.
8. Yaghmour, Karim. Building Embedded Linux Systems. Editora O'Reilly & Associates, Inc, 2003.
9. Hallinan, Christopher Embedded Linux Primer, 2ª edição. Editora Prentice Hall Professional Technical Reference, 2011.
10. Mcnamara Joel. Gps For Dummies . Editora Wiley Publishing, Inc, 2004.
11. U-BLOX, Datasheet. MAX-6 u-blox 6 GPS Modules Data Sheet. U-blox AG, Zuercherstrasse,68, CH-8800 Thalwil, Switzerland, U-BLOX, 2012. Disponível

em: < http://www.u-blox.com/images/downloads/Product_Docs/MAX-6_DataSheet_%28GPS.G6-HW-10106%29.pdf>. Acesso em: 15 de setembro, 2012.

12. Smith, Clint. 3G Wireless Network, 1ª edição. Editora McGraw-Hill, Inc, 2001.

13. U-BLOX, Datasheet. LEON-G100/G200 quad-band GSM/GPRS Data and Voice Modules Data Sheet. U-blox AG, Zuercherstrasse,68, CH-8800 Thalwil, Switzerland, U-BLOX, 2012. Disponível em: < http://www.u-blox.com/images/downloads/Product_Docs/LEON-G100_G200_DataSheet%28GSM.G1-HW-10004%29.pdf >. Acesso em: 15 de setembro, 2012.