



**Universidade Federal da Paraíba**  
**Centro de Energias Alternativas e Renováveis**  
Departamento de Engenharia Elétrica

VICTOR MIRANDA FERNANDES

**DESENVOLVIMENTO DE EQUIPAMENTO PARA TESTES DE  
RASTREADORES VEICULARES**

João Pessoa, Paraíba  
Fevereiro de 2014

VICTOR MIRANDA FERNANDES

# DESENVOLVIMENTO DE EQUIPAMENTO PARA TESTES DE RASTREADORES VEICULARES

*Relatório de Estágio submetido ao  
Departamento de Engenharia Elétrica da  
Universidade Federal da Paraíba como parte  
dos requisitos necessários para a obtenção do  
título de Engenheiro Eletricista.*

Área de Concentração: Sistemas Embarcados

Orientador:  
Professor Isaac Maia Pessoa, Dr.

João Pessoa, Paraíba  
Fevereiro de 2014

VICTOR MIRANDA FERNANDES

DESENVOLVIMENTO DE EQUIPAMENTO PARA TESTES DE  
RASTREADORES VEICULARES

*Relatório de Estágio submetido ao Departamento de  
Engenharia Elétrica da Universidade Federal da Paraíba  
como parte dos requisitos necessários para a obtenção do  
título de Engenheiro Eletricista.*

Área de Concentração: Sistemas Embarcados

Aprovado em        /        /

**Professor Euler Cássio Tavares de Macêdo, M. Sc.**  
Universidade Federal da Paraíba  
Avaliador

**Tiago Sousa Rocha**  
Zênite Tecnologia e Teleinformática  
Avaliador

**Professor Isaac Maia Pessoa, Dr.**  
Universidade Federal da Paraíba  
Orientador, UFPB

Dedico este trabalho à minha mãe.

## AGRADECIMENTOS

À minha família, pela criação, educação, amor, carinho e cuidado, além de todo apoio em todos os momentos. Aos meus colegas, por todos esses anos de stress e noites sem dormir. Aos meus companheiros de desenvolvimento no estágio, que contribuíram diretamente neste trabalho. Aos meus professores, pelo conhecimento. E, sobretudo, à minha namorada, pelo apoio, incentivo e paciência.

*“It’s over 9000!”*

VEGETA.

## RESUMO

Uma das etapas no processo de desenvolvimento de produtos é a realização da validação funcional. O objetivo desse estágio foi o desenvolvimento de um equipamento para efetuar testes em rastreadores automotivos a fim de verificar o seu correto funcionamento. Os testes feitos nos rastreadores se resumem em validar o funcionamento de: entradas e saídas, entrada analógica, comunicação serial, módulo GPS e módulo GSM. O equipamento de testes não só servirá na linha de produção, como também para efetuar testes em campo, diretamente no local do cliente.

**Palavras-chave:** Rastreador Automotivo, GPS, Raspberry Pi.

## ABSTRACT

One of the steps of product development process is the functional validation. The objective of this internship was the development of a device for testing the proper functioning of automotive trackers. The tests on the trackers are summarized in the following operations: inputs and outputs, analog input, serial communication, GPS and GSM modules. The testing equipment will not only serve on the production line, as well as for field testing, directly at the customer location.

**Palavras-chave:** Automotive tracker, GPS, Raspberry Pi.

# SUMÁRIO

Agradecimentos .....	v
Resumo .....	vii
Abstract .....	viii
Sumário .....	ix
Lista de Ilustrações .....	x
1 Introdução.....	1
1.1 Testador automático.....	2
2 Metodologia.....	4
2.1 RaspberryPi.....	4
2.2 Linux Embarcado.....	4
2.3 Desenvolvimento .....	5
2.3.1 Software Embarcado.....	5
2.3.2 Realização dos testes .....	7
2.3.3 Placa de circuito Impresso .....	10
3 Resultados .....	11
4 Conclusão .....	12
Referências .....	13
Anexo – Esquema Elétrico .....	14

## LISTA DE ILUSTRAÇÕES

Figura 1- Diagrama do testador automático .....	3
Figura 2 – Telas iniciais e telas de testes durante a utilização do testador automático.....	6
Figura 3 - Telas de resultados.....	7
Figura 4 – Esquema elétrico para teste das entradas .....	7
Figura 5 - Esquema elétrico para teste de saídas .....	8
Figura 6 - Filtro passa baixa para teste da entrada analógica .....	9
Figura 7 - Placa de circuito impresso desenvolvida .....	10
Figura 8 - Produto final montado .....	11

# 1 INTRODUÇÃO

A Zênite é uma empresa inovadora, consolidada no mercado brasileiro, que fornece, com tecnologia própria, softwares e equipamentos de alta qualidade para as áreas de telecomunicações, segurança e rastreamento. Fundada em 1991, tem como missão prover soluções inovadoras, para que seus clientes possam usar a tecnologia celular de forma mais econômica, fácil e segura. Para atender as necessidades do mercado, a Zênite conta com uma equipe de pesquisa e desenvolvimento altamente qualificada, encarregada do contínuo desenvolvimento de seus produtos. Os revendedores e distribuidores da Zênite, recebem atendimento comercial e técnico personalizado, garantindo agilidade, eficiência e lucratividade nos seus negócios. Os produtos Zênite estão presentes em todo o território nacional e no exterior através de uma rede com mais de 380 empresas que atendem a mais de 16.000 clientes, com mais de 140.000 equipamentos comercializados. Em 2007, confirmando a sua vocação para a produção de equipamentos com tecnologia celular, criou uma linha de produtos para segurança e rastreamento, composta por rastreadores GPS para veículos, pessoas, cargas e animais, como também o comunicador GPRS para painéis de alarme.

Os rastreadores desenvolvidos pela Zênite Tecnologia e Teleinformática utilizam o sistema GPS (*Global Positioning System*) para obter a localização dos veículos. GPS é um sistema de rádionavegação mundial formada a partir da constelação de 24 satélites e suas estações terrestres [1]. Os rastreadores automotivos podem ser configurados para enviar informações ao servidor da empresa de acordo com eventos[2] que podem ser gerados em intervalos de tempo, por exemplo, de três em três minutos, baseados em alguma informação do veículo, por exemplo, a variação no ângulo de sua posição (cada curva feita pelo veículo gera um novo evento) ou baseados no próprio sistema de localização, pois em alguns casos, o rastreador limita a área que o veículo pode se movimentar (cerca de movimentação). Caso o veículo ultrapasse esta tal cerca, um novo evento também será gerado. Esta última opção é bastante utilizada por empresas que exigem que determinado caminho seja tomado pelos seus funcionários.

O rastreador é composto por uma placa de circuito complexa contendo processador, entradas e saídas digitais, entrada analógica, módulo GPS e GSM. Esse

equipamento de rastreamento possui uma interface de comandos para verificar o estado de cada entrada e saída, verificar o sinal do GPS bem como a quantidade de satélites conectados, configurar entradas e saídas, configurar eventos automáticos para envio de informações ao servidor, etc. Cada funcionalidade possui um respectivo comando que pode ser manualmente configurado com o intuito de verificar algum estado do equipamento. Atualmente essa etapa de execução de comandos para verificação é realizada de forma manual por um operador experiente que tem em memória tais comandos. Todo esse processo leva muito tempo para configurar o equipamento e realizar cada tipo de verificação, como, por exemplo, uma configuração para verificar as entradas, outra configuração para verificar as saídas, outra para o sinal do GPS, etc. Além disso, é necessário manuseio de alguns equipamentos eletrônicos como fontes de alimentação e multímetros.

Pensando em melhorar a eficiência dos testes, a proposta do estágio foi elaborada para desenvolver um equipamento capaz de realizar de forma automática a verificação e validação do rastreador. A proposta então é o desenvolvimento de um equipamento envolvendo partes de hardware e software, o testador automático. O qual permite que o rastreador seja conectado a ele e os testes possam ser feitos automaticamente e em um tempo consideravelmente menor, possibilitando uma maior velocidade na produção dos rastreadores e conseqüentemente um aumento no lucro da empresa.

## 1.1 TESTADOR AUTOMÁTICO

Além do tempo consideravelmente menor, existem diversas vantagens no testador automático. A primeira vantagem é a possibilidade de testar os rastreadores em campo, ou seja, um cliente não mais precisa levar o seu produto para um local específico de testes, pois não há mais a necessidade de se utilizarem fontes de alimentação, multímetros e outros equipamentos eletrônicos para efetuar os testes. O testador automático também pode ser utilizado pelas empresas revendedoras dos rastreadores, pois não é preciso conhecimento técnico para efetuar os testes, o próprio equipamento avisa se houver algum problema no rastreador que está sendo testado e avisa onde acontece, facilitando eventuais reparos. Um diagrama do testador automático proposto no estágio é mostrado na Figura 1. Pode-se observar que ele é composto por um

display, um teclado, uma PCB e uma RaspberryPi (equipamento de processamento). A função da PCB é fazer as conexões da RaspberryPi ao teclado e display, as conversões de tensão entre a RaspberryPi e o rastreador, pois a RaspberryPi opera de 0 a 3,3V e o rastreador, como um equipamento veicular, opera em 12V (carros convencionais) ou 36V (caminhões).

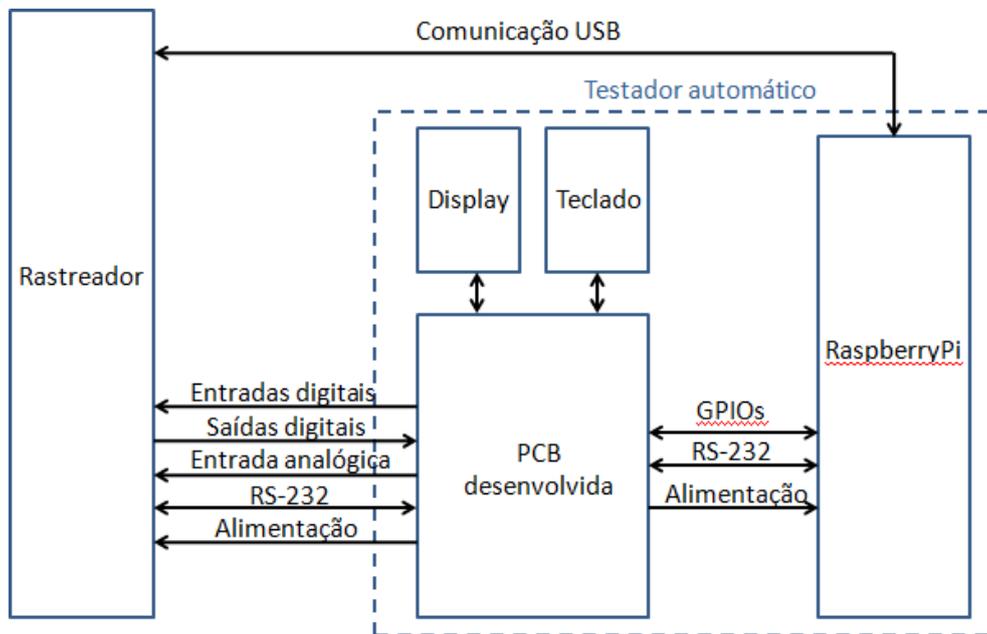


Figura 1- Diagrama do testador automático

## 2 METODOLOGIA

No decorrer deste capítulo serão apresentadas as etapas de desenvolvimento do testador automático dos rastreadores.

### 2.1 RASPBERRYPI

RaspberryPi, é referenciada neste trabalho como unidade de processamento, e foi escolhida por oferecer diversos recursos tais como: Interface de programação POSIX [7], possibilidade de programação em diversas linguagens, instalação de serviços como bancos de dados, servidor WEB, agendamento de tarefas, etc.

RaspberryPi foi desenvolvida pela *RaspberryPiFoundation* e trata-se de um computador completo de baixo custo, podendo chegar até 1 GHz com *overclock*. A *RaspberryPi* ainda possui 512 MB de memória RAM e utiliza um cartão SD como memória secundária para armazenamento persistente do S.O (Sistema Operacional), aplicativos e outros arquivos do usuário. Os desenvolvedores ainda fornecem a distribuição Linux para processadores ARM, a Raspbian que pode ser baixada gratuitamente no site do fabricante.

Dos recursos de hardware fornecidos pela RaspberryPi, foram utilizados 17GPIOs, incluindo o barramento I<sup>2</sup>C para o display.

### 2.2 LINUX EMBARCADO

O S.O. mais popular na plataforma RaspberryPi é o Linux que foi naturalmente o escolhido devido a sua flexibilidade em projetos embarcados.

Dentre as atividades desenvolvidas dentro do Linux, as principais foram: Compilação de software em C, gerenciamento de arquivos e diretórios, agendamento de tarefas (cron), criação de pacotes instaladores, etc.

## 2.3 DESENVOLVIMENTO

### 2.3.1 SOFTWARE EMBARCADO

A programação do testador automático foi feita em linguagem C. O código desenvolvido usa diversas bibliotecas, entre elas, biblioteca para comunicação serial (rs232.h) [4], biblioteca para utilização e gerenciamento de *log* [5], biblioteca para uso de display LCD via I<sup>2</sup>C [6], entre outras. A função principal do sistema é a comunicação com o rastreador para fazer a sua configuração e possibilitar os testes automáticos. Esta comunicação é feita por meio do protocolo USB como está mostrado na Figura 1. De uma forma geral, o código desenvolvido segue os seguintes passos:

1. Inicia a API para gerar o arquivo do *log*<sup>1</sup>;
2. Inicia a API para a comunicação I<sup>2</sup>C com o display;
3. Aguarda o usuário pressionar ENTER no teclado matricial para iniciar o teste;
4. Configura os pinos de entrada e saída da RaspberryPi;
5. Inicia a comunicação USB com a porta /dev/ttyACM0;
6. Aguarda o usuário escolher com qual APN<sup>2</sup> deseja realizar o teste;
7. Salva os parâmetros atuais do rastreador que será testado;
8. Inicia o teste na sequência: entradas digitais, saídas digitais, entrada analógica, comunicação RS-232, GPS e GSM, e mostra o andamento de cada um deles no display;
9. Reconfigura o rastreador para as configurações iniciais que foram salvas no passo 7;
10. Indica quais testes deram erro em uma tela de resultados no display, onde é possível navegar em várias páginas ao pressionar as setas do teclado;
11. Limpa os dados do teste realizado e volta para a tela inicial para realizar um novo teste.

---

<sup>1</sup> Cada etapa do teste gera um evento no arquivo de *log*.

<sup>2</sup> *Access Point Name* (APN) é o nome de um gateway entre GPRS ou 3G e um computador conectado a internet [8]. Este serviço está diretamente ligado ao plano de dados do *simcard* inserido no rastreador. Caso o teste seja feito com um SimCard da empresa, o programa carrega um arquivo de configuração que já está embarcado na RaspberryPi. Caso seja utilizado a APN do cliente, o rastreador já deve estar pré-configurado com essas informações.

O passo 9 é importante, pois não é interessante que o rastreador esteja com uma configuração diferente ao término do teste. Pois cada cliente precisa de uma configuração específica que é determinada pela sua aplicação. Os passos citados acima estão exemplificados nas Figuras 2 e 3 por meio das telas mostradas no equipamento.



Mesmas telas para o teste das entradas 1, 2 e 3



Mesmas telas para o teste das saídas 1, 2 e 3



Mesmas telas para outros status da  
comunicação GSM/GPRS

Figura 2 – Telas iniciais e telas de testes durante a utilização do testador automático

Caso não ocorra erros:



Caso ocorram erros:



Figura 3 - Telas de resultados

### 2.3.2 REALIZAÇÃO DOS TESTES

Para o teste das entradas, foi necessário testar os níveis lógicos alto e baixo das entradas do rastreador. Isso foi feito por meio de uma variação em um pino de saída da RaspberryPi que está conectada a um acoplador óptico de acordo com a Figura 3 para efetuar a conversão de tensão. É possível observar que existe uma lógica inversora na conversão, ou seja, para a saída da RaspberryPi (conector TESTE\_IN\_ISO0) em nível alto, o acoplador óptico chaveia a tensão de entrada para o rastreador (IN\_ISO0) para zero. E para a saída da RaspberryPi em nível zero, o acoplador óptico não é chaveado e a tensão SUPPLY é passada para a entrada do rastreador. Para cada um dos níveis (alto e baixo) foi preciso enviar um comando de leitura de pino de entrada para o rastreador para comparar com o valor real enviado para ele.

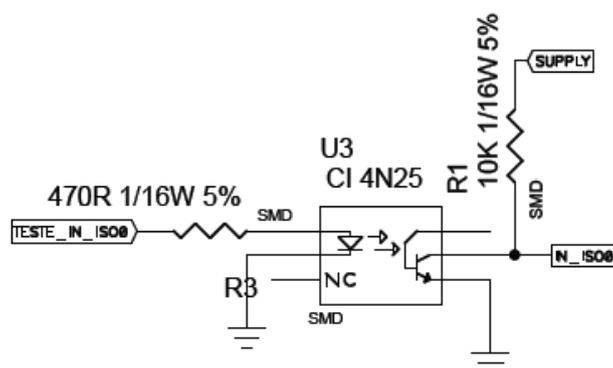


Figura 4 – Esquema elétrico para teste das entradas

Para o teste das saídas, também foi necessário testar os níveis lógicos alto e baixo. Porém, algumas versões dos rastreadores possuem saídas negativas que funcionam da seguinte forma: quando habilitadas valem zero e quando não habilitadas ficam em alta impedância. Já as saídas positivas, quando habilitadas valem 12 V e quando desabilitadas ficam em alta impedância. Dessa forma, foi necessário desenvolver um circuito capaz de detectar as variações das entradas negativas (terra e alta impedância) e das entradas positivas (12V e alta impedância) para funcionar em qualquer versão do rastreador. Tal circuito está mostrado na Figura 5. Assim como no teste das entradas, foi preciso enviar um comando para setar o pino de saída do rastreador e compará-lo com o valor lido no pino de entrada da RaspberryPi (saída do circuito da Figura 5).

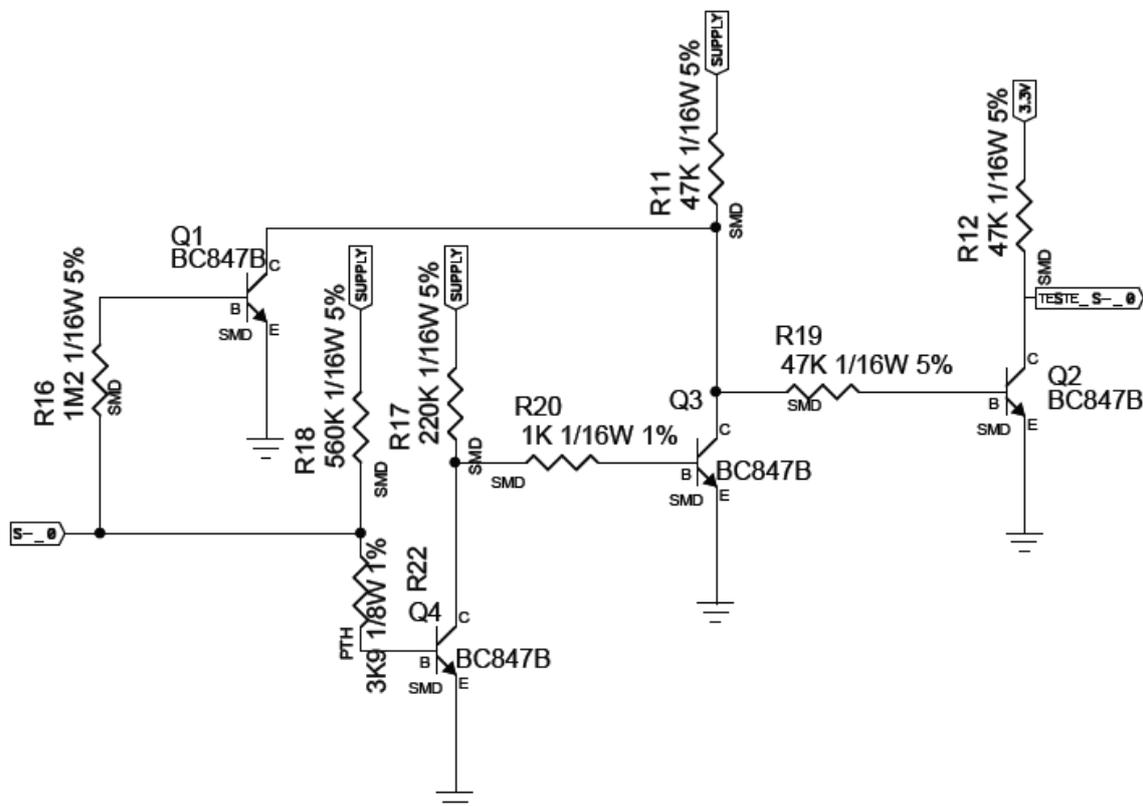


Figura 5 - Esquema elétrico para teste de saídas

Para o teste da entrada analógica foi necessário configurá-la no rastreador com uma faixa de 0 até 3,3 V. E para gerar uma tensão analógica com a RaspberryPi, foi utilizado um PWM (Modulação por Largura de Pulso) em um de seus pinos. Ainda foi preciso projetar um filtro passa-baixa, com componentes existentes no estoque da empresa, como está mostrado na Figura 6. Para cada um dos níveis foi preciso enviar um comando de leitura de pino de entrada analógica para o rastreador para comparar com o valor real enviado para ele.

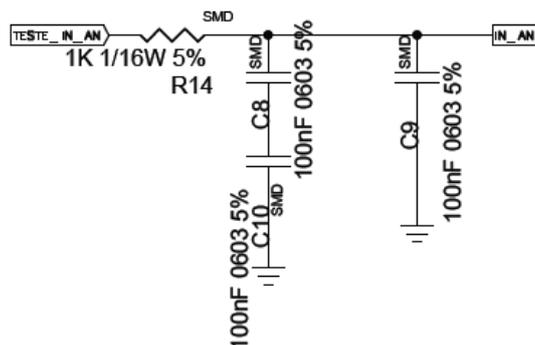


Figura 6 - Filtro passa baixa para teste da entrada analógica

O PWM foi feito com dois *duty cycles* diferentes, um para gerar uma média de 1 V (aproximadamente 30%) e outro para gerar 2 V (aproximadamente 60%). De tal forma que dois níveis analógicos pudessem ser testados.

Para o teste da comunicação serial RS-232, foi preciso utilizar o CI (Circuito integrado) MAX3223 para realizar a conversão de tensão TTL (0 a 5 V ou 0 a 3,3 V) para RS-232 (+12 V e -12 V) e vice-versa. Como mostrado na Figura 2, este teste envolve a conexão da RaspberryPi com o rastreador que é feita por meio da porta serial /dev/ttyAMA0 na RaspberryPi. Caso não seja possível efetuar a conexão, haverá um erro e o testador nem fará o teste de comunicação. No teste de comunicação, um comando é enviado e sua resposta é analisada. Caso a resposta esteja sem erros, o teste indica que a comunicação está funcionando normalmente.

Para o teste do GPS, foi preciso verificar se o módulo do GPS utilizado pelo rastreador estava habilitado, este é o teste de status mostrado na Figura 2. Já para o teste de satélites, foi preciso verificar a quantidade de satélites que o rastreador consegue se conectar em até 30 segundos. Se a quantidade de satélites conectados for maior do que 4, o rastreador passará no teste.

Para o teste do GSM (*Global System for Mobile*) e GPRS (*General Packet Radio Service*), é necessário que haja pelo menos um *Simcard* inserido no rastreador em qualquer um dos dois *slots* existentes. A RaspberryPi precisa configurar o rastreador para enviar o status da conexão GSM/GPRS automaticamente e fica apenas analisando estes status. Se receber o status CONNECTION OK em até 5 minutos, o módulo GSM do rastreador está funcionando normalmente.

### 2.3.3 PLACA DE CIRCUITO IMPRESSO

Para o desenvolvimento do produto final foi necessário projetar uma placa de circuito impresso baseando-se no protótipo funcional previamente desenvolvido na protoboard. O ambiente usado foi PCAD 2006, que é o utilizado pela empresa e já possui as bibliotecas incluindo os componentes do estoque. A placa desenvolvida está mostrada na Figura 7. Alguns erros foram cometidos no desenvolvimento da placa, os principais problemas foram: diâmetro errado na criação do componente “barra de pinos” na biblioteca do PCAD, falta de conexão do pino *enable* do CI MAX232 e a barra de pino para o teclado foi colocada invertida. Tais erros foram corrigidos utilizando estilete e *jumpers*. O esquema elétrico da placa de circuito impresso mostrada na Figura 7 está anexado a este trabalho.

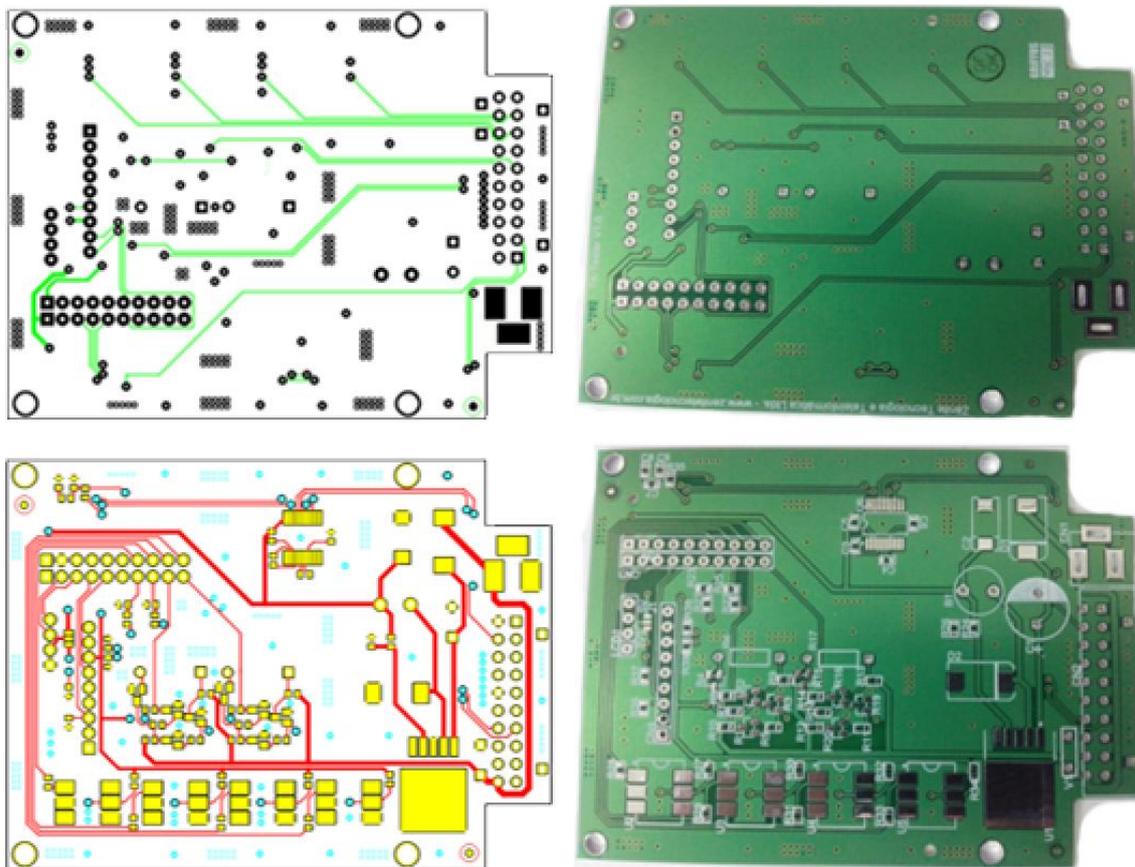


Figura 7 - Placa de circuito impresso desenvolvida

### 3 RESULTADOS

Após a confecção da placa, os componentes foram soldados e os cabos flat foram montados para serem utilizados nas devidas conexões, como está mostrado na Figura 8 (b). Alguns testes foram efetuados para validar o funcionamento do testador automático, como por exemplo: a verificação de continuidade nas trilhas da PCB e a verificação dos resultados do teste quando erros são inseridos no rastreador<sup>3</sup>.

O produto final foi montado, como mostrado na Figura 8 (a), e já está pronto para ser utilizado pela equipe da produção para testar os rastreadores antes da venda.



Figura 8 - Produto final montado

<sup>3</sup> Por exemplo, a retirada de acopladores óticos para deixar alguma entrada do rastreador sempre em alta impedância,

## 4 CONCLUSÃO

A solução proposta neste estágio resolveu o problema do tempo gasto na verificação do funcionamento dos rastreadores. Além disso, a solução ainda serviu para ajudar na manutenção de equipamentos em campo. Alguns assuntos específicos do curso de engenharia elétrica foram utilizados no decorrer deste trabalho e principalmente no projeto da placa de circuito impresso, como por exemplo, projeto de filtro passa-baixa e projeto de circuitos com transistores operando em corte e saturação (como chave). É importante ressaltar que o produto desenvolvido é inovador, pois não existem produtos similares no mercado.

## REFERÊNCIAS

- [1] EETIMES. **How does a GPS tracker work?**. Acesso em 20 de Janeiro, 2014. Disponível em: <[http://www.eetimes.com/document.asp?doc\\_id=1278363](http://www.eetimes.com/document.asp?doc_id=1278363)>.
- [2] ZENITE. **Protocolo avançado**. Acesso em 19 de Janeiro, 2014. Não disponível na internet.
- [3] RASPBERRY PI. **Quick Start Guide**. Acesso em 19 de Janeiro, 2014. Disponível em: <[http://www.raspberrypi.org/wp-content/uploads/2012/04/quick-start-guide-v2\\_1.pdf](http://www.raspberrypi.org/wp-content/uploads/2012/04/quick-start-guide-v2_1.pdf)>.
- [4] TEUNIS VAN BEELEN. **RS-232 for Linux and Windows**. Acessado em 19 de Janeiro, 2014. Disponível em: <<http://www.teuniz.net/RS-232/>>.
- [5] LINUX JOURNAL. Syslog configuration. Acessado em 19 de Janeiro, 2014. Disponível em: <<http://www.linuxjournal.com/article/5476>>.
- [6] GORDON PROJETS. **WiringPi**. Acessado em 19 de Janeiro, 2014. Disponível em: <<https://projects.drogon.net/raspberry-pi/wiringpi/>>.
- [7] IEEE. **POSIX**. Acessado em 20 de Janeiro, 2014. Disponível em: <<http://pubs.opengroup.org/onlinepubs/9699919799/>>.
- [8] WIKIPEDIA. **Access Point Name**. Acessado em 21 de Janeiro, 2014. Disponível em: <[http://en.wikipedia.org/wiki/Access\\_Point\\_Name](http://en.wikipedia.org/wiki/Access_Point_Name)>.

# ANEXO – ESQUEMA ELÉTRICO

