



## **TRABALHO DE GRADUAÇÃO**

# **Desenvolvimento de uma Placa Eletrônica do Sistema Mínimo da plataforma LAICANSAT**

**Alex Kraus de Castro**

**Brasília, Dezembro de 2015**

**UNIVERSIDADE DE BRASÍLIA**

**FACULDADE DE TECNOLOGIA  
CURSO DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E  
AUTOMAÇÃO**

UNIVERSIDADE DE BRASÍLIA  
Faculdade de Tecnologia  
Curso de Graduação em Engenharia de Controle e Automação

## TRABALHO DE GRADUAÇÃO

# Desenvolvimento de uma Placa Eletrônica do Sistema Mínimo da plataforma LAICANSAT

**Alex Kraus de Castro**

Relatório submetido como requisito parcial para obtenção do grau de Engenheiro de Controle e Automação.

### **Banca Examinadora**

Prof. Dr. Renato Alves Borges, UnB/ ENE  
*Orientador*

---

Prof. Dr. Simone Bastistini, FGA/ UnB  
*Coorientador*

---

Prof. Dr. Chantal Cappelletti, FGA/UnB  
*Examinador Externo*

---

Brasília, Dezembro de 2015

## FICHA CATALOGRÁFICA

Alex, Kraus de Castro  
Desenvolvimento de uma Placa Eletrônica do Sistema Mínimo da plataforma LAICANSAT

[Distrito Federal] 2015.

xi, 66p., 297 mm (FT/UnB, Engenheiro, Controle e Automação, Ano). Trabalho de Graduação – Universidade de Brasília.Faculdade de Tecnologia.

1.Microcontrolador  
3.Esquemático

2.Sensores  
4.Layout

I. Mecatrônica/FT/UnB

II. Título (série)

## REFERÊNCIA BIBLIOGRÁFICA

Castro,A.K. de, (2015). Desenvolvimento de uma Placa Eletrônica do Sistema Mínimo da plataforma LAICANSAT. Trabalho de Graduação em Engenharia de Controle e Automação, Publicação FT.TG-nº 18, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 66p.

### 1.1 CESSÃO DE DIREITOS

AUTOR: Alex Kraus de Castro.

Desenvolvimento de uma Placa Eletrônica do Sistema Mínimo da plataforma LAICANSAT: Aplicação do conhecimento em eletrônica para o desenvolvimento da placa de circuito impressa do projeto LAICANSAT.

GRAU: Engenheiro

ANO: 2015

É concedida à Universidade de Brasília permissão para reproduzir cópias deste Trabalho de Graduação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desse Trabalho de Graduação pode ser reproduzida sem autorização por escrito do autor.

---

Alex Kraus de Castro  
SMPW Quadra 18 Conjunto 05 Lote 02 – Parkway  
71741-805 Brasília – DF – Brasil.

## AGRADECIMENTOS

Aos meus pais, Jorge e Eliane, pela criação e pelo exemplo de perseverança e dedicação, integridade, amor e dedicação.

Aos meus orientador Renato e coorientador Simone pelos ensinamentos durante o desenvolvimento do Projeto.

A Universidade de Brasília – UnB, pela oportunidade. Em particular agradeço aos professores do curso.

A Autotrak, pela oportunidade de desenvolver o meu aprendizado durante o meu estágio, em especial ao Fernando e André que foram fundamentais no desenvolvimento deste projeto.

Aos meus colegas de curso, em especial Paulo, Helton e Gabriel, pela amizade e companheirismo durante todo o curso.

Ao CNPq, pela oportunidade de participar do Ciências sem Fronteira.

*Alex Kraus de Castro.*

## RESUMO

O presente trabalho apresenta o desenvolvimento de uma placa eletrônica do sistema mínimo da plataforma LAICAnSat. O objetivo do trabalho é padronizar, implementar e desenvolver uma placa eletrônica que consiste no computador de bordo do projeto. Um dos requisitos dessa padronização foi a utilização do padrão de placa PC104 que se encaixa perfeitamente no padrão *CubeSat*. O sistema embarcado possui um receptor GPS, Xbee, sensor de temperatura, pressão, umidade e sensores de posicionamento. É apresentado um histórico das missões passadas e, em seguida, é apresentado as decisões tomadas em relação aos novos requisitos do projeto gerados pelos experimentos passados. É apresentado em detalhes o novo hardware do projeto junto com os testes feitos na placa. O resultado do estudo apresentou uma placa compatível com o objetivo do projeto.

Palavras Chave: Microcontrolador; Sensores; Esquemático; Layout.

## ABSTRACT

This work presents the development of an electronic board of the minimum LAICAnSat platform system. The objective is to standardize, deploy and develop an electronic board consisting of the project on-board computer. One of the requirements of this standardization was using the PC104 board pattern that fits perfectly into the CubeSat standard. The embedded system has a GPS receiver, Xbee, temperature sensor, pressure, humidity and positioning sensors. A history of the past missions is displayed and then is presented the decisions taken in relation to the new project requirements generated by past experiments. It is presented in detail the new hardware design along with the tests done on the board. The study results showed a board compatible with the project objective.

Keywords: Microcontroller; Sensors; Schematics ; Layout.

# SUMÁRIO

<b>1. INTRODUÇÃO</b> .....	<b>1</b>
1.1 Contextualização.....	1
1.1.1 <i>CanSat</i> .....	2
1.1.2 BalloonSats.....	3
1.2 O Projeto LAICAnSat.....	4
1.2.1 <i>Cubesat</i> .....	5
1.2.2 PC104.....	7
1.3 Objetivos.....	8
1.4 Apresentação do Manuscrito.....	9
<b>2. VISÃO GERAL</b> .....	<b>10</b>
2.1 Plataforma LAICAnSat-1.....	10
2.2 Requisitos.....	11
2.3 Protocolos de Comunicação.....	13
2.3.1 UART.....	13
2.3.2 I2C.....	14
2.3.3 SPI.....	15
2.3.4 2-Wire-Bus.....	17
2.4 Capacitor de Desacoplamento.....	17
2.5 Resistor utilizados com a função Pull-Up.....	17
<b>3. FERRAMENTAS DO PROJETO</b> .....	<b>19</b>
3.1 ALTIUM.....	19
3.1.1 Esquemático.....	19
3.1.2 Layout.....	21
3.2 Arduino IDE.....	22
3.3 WEBENCH DESIGN.....	23
3.4 PCB SATURN.....	23
<b>4. PROJETO LAICAnSAT-3</b> .....	<b>24</b>
4.1 Esquemático.....	24
4.1.1 SUMÁRIO.....	25
4.1.2 GNSS-XBEE-SDCARD.....	25
4.1.2.1 GNSS.....	25
4.1.2.2 XBEE.....	27
4.1.2.3 SDCARD.....	27
4.1.3 IMU+SENSORES.....	28
4.1.3.1 Sensor <i>variometer</i> .....	28
4.1.3.2 Sensor Angular de Três Eixos.....	28
4.1.3.3 Acelerômetro 3D e Magnetômetro 3D.....	29
4.1.3.4 Sensor de Pressão e Temperatura.....	30
4.1.3.5 Sensor de Umidade e Temperatura.....	31
4.1.4 <i>Power Supply</i> .....	31
4.1.5 MCU.....	34
4.2 Placa de Circuito Impressa.....	37
4.2.1 COTAS.....	39
4.2.2 TOPLAYER.....	40
4.2.3 BOTTOMLAYER.....	45
4.2.4 TOPOVERLAY.....	46
4.2.5 FABINFO.....	48
4.3 Testes e Análise dos Resultados.....	48
<b>5. CONCLUSÃO</b> .....	<b>54</b>
<b>REFERÊNCIA BIBLIOGRÁFICA</b> .....	<b>56</b>
<b>ANEXO 1 – Lista de Materiais</b> .....	<b>60</b>
<b>ANEXO 2 – Códigos de Testes</b> .....	<b>62</b>
<b>ANEXO 3 – Esquemático</b> .....	<b>64</b>

# LISTA DE FIGURAS

Figura 1 – Exemplo de CanSat e suas subcamadas [8]	3
Figura 2 - Exemplo de plataforma BalloonSat [9].	4
Figura 3 - Primeiro protótipo de um CubeSat mais lançador [13]	6
Figura 4 - Exemplo de Cubesat e P-POD [14]	7
Figura 5 - Especificações mecânicas PC104[16]	8
Figura 6 - Conexão de 3 placa no padrão PC104 [16]	8
Figura 7 – Comunicação UART [30]	14
Figura 8 – Conexão UART [30]	14
Figura 9 – Canal de comunicação I2C [31]	15
Figura 10 – Canal de comunicação SPI [32]	16
Figura 11 - 2-Wire-Bus[33]	17
Figura 12 – Criação do transistor 2N3904 [35]	20
Figura 13 – Esquemático Altium [35]	20
Figura 14 – Layout Altium [35]	21
Figura 15 – IDE Arduino [36]	22
Figura 16 - Bootloader Teensy [18]	22
Figura 17 – Webench Design [37]	23
Figura 18 - PCB Saturn [38]	23
Figura 19 – Flowchart placa LAICAnSat	24
Figura 20 – Sumário	25
Figura 21 – GNSS	26
Figura 22 – Xbee	27
Figura 23 – Micro Cartão SD	27
Figura 24 – Sensor variometer MS5611	28
Figura 25 – Sensor de torção e giro L3GD20H	29
Figura 26 – Sensor acelerômetro e magnetômetro LSM303DLHC	30
Figura 27 - Sensor de pressão e temperatura BMP180	31
Figura 28 - Sensor de humidade e temperatura SHT15	32
Figura 29 - Regulador de tensão TPS62142	33
Figura 30 - Step Down Converter	33
Figura 31 - Simulação TPS62142	34
Figura 32 - Gráfico de tensão Vout e Vsw	35
Figura 33 - Esquemático do Microcontrolador	36
Figura 34 – LaicanSat PCB	38
Figura 35 - Divisões da placa	39
Figura 36 – Cotas	40
Figura 37 - Toplayer PCB	40
Figura 38 - Toplayer MCU	41
Figura 39 - Trilha USB	41
Figura 40 – Toplayer Antena	42
Figura 41 - MCU com camadas sobrepostas	43
Figura 42 - Toplayer fonte de tensão	43
Figura 43 - Toplayer Mini54Tan	44
Figura 44 - Toplayer SD Card + SHT15	44
Figura 45 – Sensores	45
Figura 46 – Bottomlayer	46
Figura 47 – Bottomlayer sensores	46
Figura 48 - Topoverlay	48
Figura 49 - Topoverlay MCU	48
Figura 50 – Fabinfo	49
Figura 51 - Placa LAICAnSat Produzida pela GRIFFUS	50
Figura 52 - Teste barramento I2C	51
Figura 53 - Teste Cartão Micro SD	52

<i>Figura 54 - Teste GPS</i>	<u>52</u>
<i>Figura 55 – Teste dos sensores armazenando no cartão de memória</i>	<u>53</u>
<i>Figura 56 - Placa eletrônica LaicanSat</i>	<u>55</u>



# LISTA DE TABELAS

Tabela 1 - Informações dos sensores ..... 12

# LISTA DE SÍMBOLOS

## Siglas

ABNT	Associação Brasileira de Normas Técnicas
ISS	International Space Station
APRS	Automatic Packet Reporting System
IMU	Inertial Measurement Unit
MCU	Microcontroler
GPS	Global Position System
GNSS	Global Navigation Satellite System
GAUSS	Group of Astrodynamics for the Use of Space Systems
OPAL	Orbiting picosatellite automated launcher
P-POD	Poly-Picosatellite Orbital Deployer
SBC	Single Board Computer
IDE	Integrated Development Environment
UART	Universal Asynchronous Reiceiver/Transmitter
GND	Ground
I2C	Inter-Integrated Circuit
SCL	Clock Signal
SDA	Data Signal
SPI	Serial Peripheral Interface
SS	Slave Select
MOSI	Master Out Slave In
MISO	Master In Slave Out
SCK	Clock Signal
PCB	Printed Circuit Board
PCI	Placa de Circuito Impresso

# 1. INTRODUÇÃO

Este trabalho é parte do projeto LAICAnSat, desenvolvido pela Universidade de Brasília e tem como objetivo o desenvolvimento de uma placa eletrônica capaz de adquirir dados da atmosfera em alta altitude. Com isso, obter um sistema de controle confiável que possa utilizar os dados coletados pela placa na recuperação dos dados de forma segura, através de um pouso em uma região pré-determinada, garantindo assim o sucesso da missão.

Neste capítulo faz-se a contextualização deste trabalho com o projeto LAICAnSat. Em seguida são esclarecidas as características principais dos *CanSats* e *BallonSats* e como eles se conectam com o trabalho presente. Logo após, destaca-se quais são os objetivos principais do trabalho e menciona-se dois projetos de satélites. Depois, os padrões Cubesat e PC104 são descritos finalizando com uma apresentação geral deste manuscrito.

## 1.1 Contextualização

Os Estados Unidos da América (EUA) tem investigado uma opção para aplicações de distribuição de suprimentos através do uso do sistema lançados do ar objetivando a recuperação de uma carga útil [1]. Como também é investigado este método para a recuperação de cargas da Estação Espacial Internacional (ISS) [2].

Uma plataforma altamente difundida pelo mundo para a experimentação científica e para o desenvolvimento de novos conceitos na área aeroespacial é o balão de alta altitude [3]. A área aeroespacial sempre foi acompanhada de altos custos, por isso foi preciso buscar novos meios para que reduzissem os orçamentos. Uma das soluções foi a utilização de balões de alta altitude. Um dos principais atrativos para a pesquisa e experimentação científica com esses balões é o baixo custo de desenvolvimento e lançamento [4].

O projeto LAICAnSAT é um esforço realizado dentro da Universidade de Brasília no sentido de utilizar a ideia do modelo *Ballonsat* (do inglês *ballon*, balão, *sat*, satélite) somado ao *CanSat* (do inglês *can*, lata, e *sat*, satélite) para aprimorar o conhecimento no campo de tecnologia espacial [5]. Desta forma é possível juntar os desafios do balonismo com as competições *CanSat*.

O projeto LAICAnSat tem por objetivo desenvolver experimentos com balões meteorológicos para fins científicos e de sensoriamento remoto. Foram realizados dois experimentos de balão com a carga útil e sistema de rádio embarcados. O primeiro lançamento ocorreu em 2 de maio de 2014 e o segundo no dia 24 do mesmo mês, ambos na cidade de Padre Bernardo, Goiás. Os lançamentos foram feitos sem ter um sistema de controle embarcado. No segundo lançamento foi necessária a ajuda do Corpo de Bombeiros para a recuperação da carga útil, uma vez que esta acabou presa na copa de uma árvore dentro de uma mata

densa. Assim, pode-se perceber que um sistema de controle e navegação eficiente resultaria em ações de recuperação da carga útil mais simples e eficientes.

A carga útil possuía duas câmeras de alta definição além de sensores que mediam a temperatura, pressão, umidade, nível de luz UV, altitude, posição, velocidade, direção, e aceleração, bem como monitores de tensão, corrente e temperatura. Ambos lançamentos foram importantes também no sentido de testar os equipamentos utilizados neste trabalho, como por exemplo o sistema APRS (do inglês *Automatic Packet Reporting System*, Sistema de Notificação Automática de Pacotes).

### **1.1.1 CanSat**

Os *CanSats* são dispositivos eletrônicos desenvolvidos para serem embarcados em foguetes de sondagem e são muito utilizados para introdução na área espacial, com o foco em estudantes [6]. Normalmente são equipados com paraquedas para reduzir a velocidade do impacto na hora da queda do dispositivo.

Existem diversas competições de *CanSats* hoje em dia no mundo todo. As competições na Europa, por exemplo, estipulam para que as dimensões do dispositivo sejam de 66 mm de diâmetro, 115 mm de altura e uma massa de 350 g. Essas dimensões se assemelham a de uma lata de refrigerante. O voo deve ter menos de 120 s, a taxa de descida entre 8 m/s e 11 m/s entre outras características [7].

Em geral, utilizam diversos dispositivos de sensoriamento – transdutores (temperatura, pressão, etc.), acelerômetro ou unidade de medidas inerciais (IMU), câmeras fotográficas/vídeo, GPS e microcontrolador (MCU). Na Figura 1 são apresentados alguns exemplos de dispositivos embarcados. Pode-se dizer que os *CanSats* são modelos funcionais e projeto de satélites reais. Com o seu baixo custo, curto tempo de preparação e simplicidade, tornam este dispositivo uma ferramenta de introdução para alunos na área da ciência espacial.

O *Cansat* trabalha com conceitos principalmente de engenharia de projetos, elétrica/eletrônica e engenharia de software. Todas elas são cruciais para o desenvolvimento de sistemas espaciais reais e estimulam a formação de novos profissionais no setor aeroespacial.

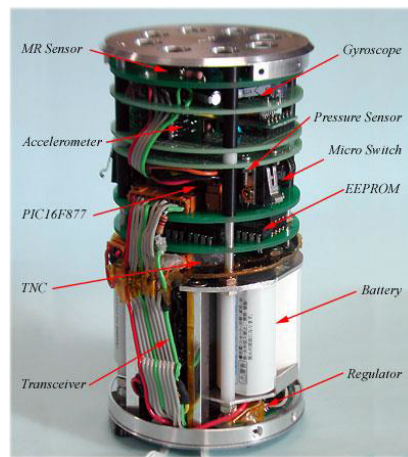


Figura 1 – Exemplo de CanSat e suas subcamadas [8]

### 1.1.2 BalloonSats

Por representar uma alternativa de baixo custo para o desenvolvimento de tecnologias espaciais as plataformas de balões de alta altitude ganharam uma nova perspectiva na área aeroespacial. A utilização desses balões viabilizaram a criação de uma plataforma de experimentação científica e demonstração de novos conceitos e tecnologias. Esta plataforma é composta pelo balão e por uma carga útil. Esta carga útil é onde se encontra o experimento e o equipamento de suporte e paraquedas.

O LAICAnSat pode ser considerado um *balloonsat* por apresentar uma carga útil inferior a 4 Kg e ser lançado por um balão sonda [9]. Frequentemente, a utilização dos balões de alta altitude levam a carga útil ate uma altitude de 18 Km e se estende até 50 Km. A estratosfera possui características interessantes que justificam o estudo de sistemas nesta região. Por exemplo, a temperatura pode variar de  $-50^{\circ}\text{C}$  a  $10^{\circ}\text{C}$ .

Na carga útil destes balões podem ser inclusos diversos tipos de sensores como câmeras, acelerômetros, giroscópios, magnetômetros, sensores de pressão, umidade, temperatura, raios ultravioleta, infravermelho, detectores de gases e micro-organismos. Por essa grande versatilidade tais plataformas têm se tornado muito interessante para investigações científicas. A Figura 2 apresenta um exemplo de diversos *BalloonSats* sendo elevados através de um balão de alta altitude em série com um paraquedas que auxilia na recuperação da carga após o balão estourar.

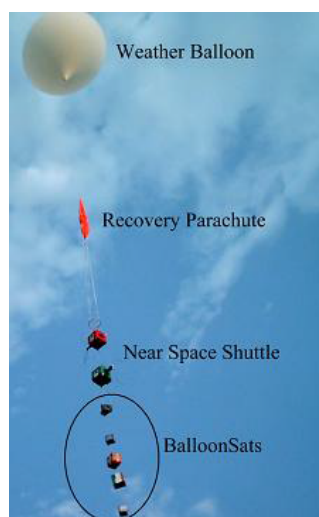


Figura 2 - Exemplo de plataforma *BalloonSat* [9].

## 1.2 O Projeto LAICAnSat

O projeto LAICAnSat tem como objetivo desenvolver uma plataforma de experimentos em alta altitude. Ele engloba o desenvolvimento de *balloonsats* e *cansats* e seu desenvolvimento é dividido em vertentes: alta altitude e baixa altitude.

O principal objetivo da vertente de baixa altitude é implementar um sistema de controle capaz de conduzir uma carga útil para uma área de pouso. Em altas altitudes, os paraquedas não são eficazes no controle da carga útil devido a baixa pressão atmosférica. Sendo assim, o desenvolvimento do sistema de controle do paraquedas fica restrito a vertente de baixa altitude.

A vertente de alta altitude tem como objetivo desenvolver um sistema embarcado capaz de resistir a variações de temperatura, pressão e umidade, capaz de fornecer dados meteorológicos e imagens de sensoriamento remoto. Inclui-se no projeto o sistema APRS responsável por rastrear a carga útil.

Com a realização de duas missões (LAICAnSat-1 e LAICAnSat-2), realizadas em 2014, houve desenvolvimentos em várias áreas, além de testes de sistemas e subsistemas, antes dos lançamentos. Os objetivos iniciais definidos no projeto foram:

- Obter dados atmosféricos e imagens da superfície terrestre;
- Obter dados relativos a altitude da carga útil;
- Rastrear o balão em todos os momentos da missão;
- Obter telemetria dos sensores embarcados.

Os requisitos foram:

- Garantir autonomia mínima do sistema de rastreamento (mínimo de duas vezes o tempo esperado de voo);

- Manter a temperatura dos componentes eletrônicos entre -5°C e +50°C;
- Taxa de descida igual ou inferior a 5 m/s, abaixo de 3 km de altitude;
- Massa total inferior a 4 kg.

O desenvolvimento deste trabalho tem como objetivo realizar a terceira missão do projeto, o LAICAnSat-3. Um dos focos do trabalho foi a padronização do sistema embarcado. O G.A.U.S.S (*Group of Astrodynamics for the Use of Space Systems*) desenvolveu junto com a ajuda da universidade de Roma um computador de bordo para microssatélites chamado de ABACUS. A finalidade deste computador é controlar todos os equipamentos embarcados no satélite, armazenar todas as informações coletadas em memória, enviar e receber informações da estação central. Este dispositivo já realizou diversas missões em órbita [10]. Este computador de bordo é disposto em um *Cubesat* e a sua placa de circuito impressa é compatível com o formato PC104. Nos tópicos a seguir serão discutidos em mais detalhe o modelo *Cubesat* e o padrão PC104. Para o processamento dos dados, a plataforma ABACUS possui dois cores (MCU MSP430 e uma FPGA Spartan-3E), uma IMU com magnetômetro, acelerômetro e giroscópio e 3 sensores de temperatura entre outros componentes para aquisição de dados, como também, possui um dispositivo para transmitir essas informações para as estações terrestres [10].

A Universidade de Brasília também participa de outro projeto na área aeroespacial, o nanosatélite Serpens – sigla para Sistema Espacial para Realização de Pesquisa e Experimento com Nanosatélites. Entretanto este projeto visa o desenvolvimento de um nanosatelite que será lançado em órbita. O satélite testa conceitos simples do uso *CubeSat* para o recebimento, armazenamento e retransmissão de mensagens por sistema de rádio. Sua proposta é ilustrar que utilizando esses experimentos, pequenos satélites podem ser usados para agregar funcionalidade ao sistema de coleta de dados ambientais no país. O principal objetivo é a capacitação de recursos humanos e a consolidação dos novos cursos de engenharia espacial brasileiro. Este projeto conta com a participação de outras universidades federais como do ABC (UFABC) e também de universidades do exterior como a *Sapienza Universita di Roma* (Itália). O lançamento do satélite ocorreu no dia 17 de setembro de 2015 e os seus sinais já foram captados por vários radioamadores do país [17].

### **1.2.1 Cubesat**

O custo para desenvolver e lançar um satélite de grande porte é muito alto, por isso, com o objetivo de desenvolver pesquisas e experimentos científicos, procurou-se uma alternativa. Um dos primeiros satélites feito por estudantes na universidade de *Stanford* foi em abril de 1994. Este satélite possuía uma forma hexagonal com 20 kg, 457 mm de diâmetro e 305

mm de altura. Em fevereiro de 2000 a Universidade de *Stanford* fez um lançamento do satélite *OPAL (orbiting picosatellite automated launcher)* que fez a sociedade entrar em uma nova era de experimentação no espaço [13]. O *OPAL* foi construído em *The Aerospace Corporation in El Segundo, CA (California)* e demonstrou um novo espaço de comunicação com estações terrestres utilizando rádios e teste de *MEMS RF switches and intersatellite*. Este dispositivo é considerado um microsatélite que possui de 25 kg a 50 kg e carrega dentro dele diversos picosatelites . Estes picosatelites possuem um peso menor do que um 1 kg e tem dimensões de 4x3x1 polegadas e são liberados pelo *OPAL* quando entrado em órbita [13].

Depois do lançamento desse novo microsatélite, foi determinado que estes satélites poderiam ser usados de diversas formas em práticas e experiências espaciais para programas de estudo espacial. A universidade de *Stanford* e a universidade politécnica da Califórnia (*Cal Poly*) fizeram uma relação colaborativa e em conjunto realizaram o primeiro protótipo do novo modelo de satélite chamado de *CubeSats* [13]. Veja o primeiro protótipo de *CubeSat* na Figura 3.



Figura 3 - Primeiro protótipo de um *CubeSat* mais lançador [13]

*CubeSat* vem da união da palavra em inglês *cube*, em português cubo, e *sat* que é uma abreviação de satélite. *CubeSat* é um tipo de satélite miniaturizado usado para pesquisas espaciais e comunicação para estudantes e o seu padrão exige que o projeto seja um cubo de 10 cm de aresta com um peso menor do que 1 kg. Um dos fatores que determinaram esse padrão foi o tamanho dos componentes disponível para serem comprados a um preço baixo e as dimensões para encaixar na no P-POD (*Poly-Picosatellite Orbital Deployer*). O mecanismo responsável por proteger os *CubeSats* no seu lançamento e ejetá-los em órbita é o P-POD [14]. Veja na Figura 4 um exemplo de *CubeSat* e um P-POD.



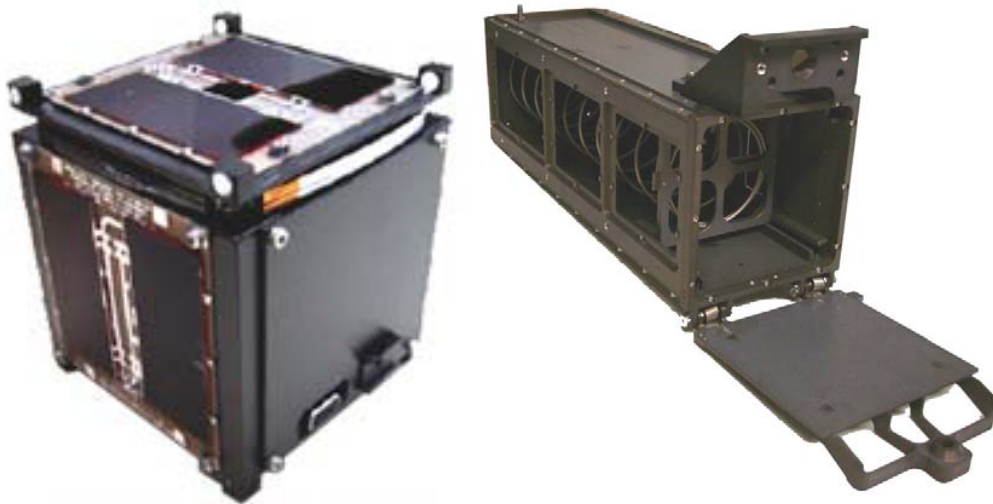


Figura 4 - Exemplo de Cubesat é P-POD [14]

### 1.2.2 PC104

O rápido crescimento de computadores embarcados, usados para fins industriais, exigiu uma forma de ampliar recursos de plataformas SBC (Single Board Computer) sem um grande custo ou espaço disponível. Em fevereiro de 1992, com a participação de 12 empresas, foi criado o consórcio do padrão PC104 que tinha como intuito adaptar a tecnologia de computadores de bordo para aplicações embarcadas [15]. A primeira especificação foi *open design* que oferece a potência e a flexibilidade de um computador IBM com um tamanho que encaixa perfeitamente em projetos embarcados. Este projeto consta com o barramento ISA do computador IBM que foi estabelecido pelo IEEE P996[15].

O padrão PC104 pode ser dividido em mecânico e elétrico. O primeiro exige que a placa possua um padrão bem definido de 90x96 cm, conforme na Figura 5 todas as especificações mecânicas [16]. O segundo exige que o consumo da placa seja de cerca de 1W-2W, limitando a corrente do barramento em 4 mA [16]. Outra característica interessante desse padrão é a possibilidade de conectar um módulo PC104 com outro, veja a conexão de 3 placas na Figura 6. É por meio do barramento ISA que os módulos irão trocar informações entre si. O barramento ISA utilizado no projeto deste manuscrito foi o de 16 bits, veja na Figura 5 a posição dos conectores que é estabelecida pelo padrão mecânico [16].

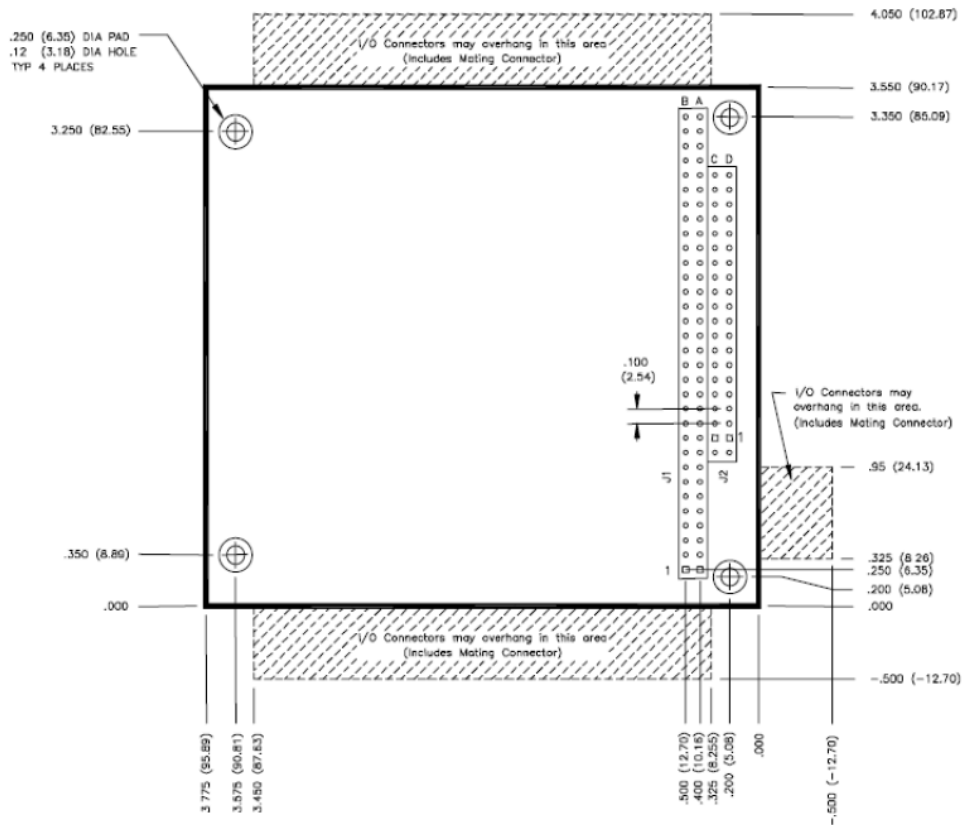


Figura 5 - Especificações mecânicas PC104[16]

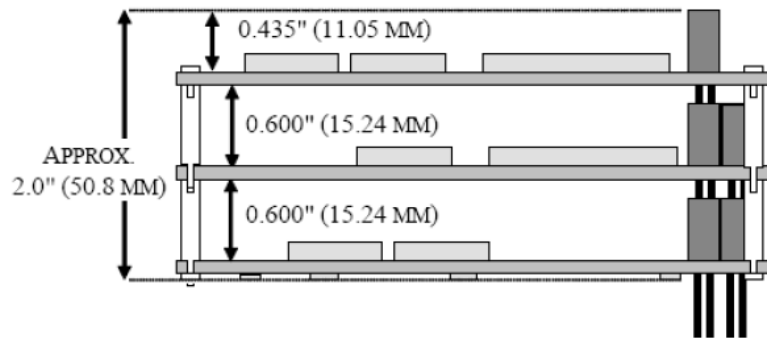


Figura 6 - Conexão de 3 placa no padrão PC104 [16]

### 1.3 Objetivos

O objetivo deste trabalho é gerar uma padronização da placa eletrônica do computador de bordo da plataforma LAICAnSat. Um dos requisitos dessa padronização foi a utilização do padrão de placa PC104 que se encaixa perfeitamente no padrão *Cubesat*.

Como objetivos específicos foram enumerados:

- Estudar os componentes utilizado no projeto LAICAnSat-1;

- Desenvolver uma reconfiguração e agrupamento do esquemático a partir dos componentes já implementados na primeira missão;
- Aprimorar o funcionamento da placa redefinindo uma nova fonte de alimentação e remover sensores redundantes tornando a placa mais eficiente;
- Desenhar um layout de placa robusto ao sistema projetado no esquemático;
- Realizar testes de funcionamento da placa.

## 1.4 Apresentação do Manuscrito

Este documento está dividido em 5 capítulos, que descreverão o caminho percorrido a fim de atingir os objetivos finais, a saber:

- O Capítulo 2 apresentará uma visão geral do projeto. Será discutido os experimentos realizados anteriormente e como foram utilizados para o desenvolvimento do atual projeto. Os componentes escolhidos serão descritos junto com uma explicação dos protocolos de comunicação utilizados;
- O Capítulo 3 descreve a funcionalidade das ferramentas de projeto utilizados. Os softwares utilizados para o desenvolvimento do projeto serão apresentados e logo será feita uma explicação sobre resistores *pull-up* e capacitores de desacoplamento;
- No Capítulo 4 será descrito todo o desenvolvimento do projeto. Primeiramente, será apresentado o sumário do projeto e, em seguida, o esquemático e descrito. Logo após, é descrito o desenvolvimento da placa de circuito impresso e para finalizar são mostrados os testes feitos na placa;
- Por fim, o Capítulo 5 descreve as conclusões deste trabalho, repassando os objetivos iniciais. Ainda neste capítulo, projeções de trabalhos futuros na área deste projeto são levantados, de forma a fomentar a continuidade da pesquisa nesta área de estudo.

## 2. VISÃO GERAL

Para o entendimento deste trabalho será dada uma visão geral sobre o projeto, explicando os requisitos gerados a partir das missões anteriores e introduzir os conceitos básicos utilizados para o desenvolvimento do projeto de forma a conceituar, definir e contextualizar a eletrônica envolvida nas soluções propostas. Assim sendo, é necessário informar como funcionam os protocolos de comunicação utilizados pelo microcontrolador. Estes protocolos foram implementados utilizando o software do Arduino IDE. Em seguida é explicado o funcionamento de um resistor *pull-up* e finalizando com a explicação sobre capacitores de desacoplamento.

### 2.1 Plataforma LAICAnSat-1

O projeto LAICAnSat realizou dois experimentos utilizando a plataforma de experimentos em alta altitude. A partir das missões anteriores detectou-se uma necessidade de desenvolver uma placa de circuito impressa do circuito eletrônico.

O projeto desenvolvido por Nehme [5] apresentava os seguintes componentes:

- Duas plataformas de microcontroladores teensy[18];
- Uma IMU desenvolvida em um *shield* para Arduino [19];
- Um sensor *variometer* MS5611 desenvolvida em um *shield* para Arduino[20];
- Um sensor de temperatura e umidade SHT15 desenvolvida em um *shield* para Arduino[21];
- 3 sensores de temperatura DS18B20[22];
- um sensor de corrente INA219 desenvolvida em um *shield* para Arduino [23];
- Um regulador de tensão MIC2941 [24];
- Um sensor de radiação infravermelha TSL2561 desenvolvida em um *shield* para Arduino[25];
- Um sensor de radiação ultravioleta UVM-30A desenvolvida em um *shield* para Arduino[26];
- Um GPS da LEA 6 desenvolvido pela Ublox [28];
- Um radio Xbee.

A forma de conectar os diversos elementos com o microcontrolador e com a bateria era por meio da utilização de uma protoboard. Uma parte dos componentes foram colocados por meio de soquetes nesta placa, os outros foram conectados por meio de fios. Todos os dispositivos foram colocados dentro de uma caixa de isopor para a realização da primeira missão. Entretanto esse tipo de procedimento pode comprometer todo o sistema

desconectando alguma parte do circuito ou até mesmo gerando um curto circuito. No tópico a seguir é descrito os requisitos gerados a partir desse primeiro experimento.

## 2.2 Requisitos

O circuito eletrônico utilizado neste projeto foi desenvolvido pelo Nehme [5] e utilizava duas plataformas de microcontrolador Teensy e muitas placas de desenvolvimento conectadas a eles. Com a perspectiva de aprimorar o computador de bordo os seguintes requisitos foram gerados:

- Escolher componentes eletrônicos que funcionem dentro de uma faixa de temperatura de  $-5^{\circ}\text{C}$  e  $+50^{\circ}\text{C}$ ;
- Rastrear a plataforma de experimento em todos os momentos do vôo;
- Obter telemetria dos sensores embarcados;
- Taxa de descida igual ou inferior a 5m/s, abaixo de 3km de altitude;
- Massa total inferior a 4000g.

Para o desenvolvimento da placa de circuito impressa, adotou-se as seguintes etapas:

- Realizar um esquemático da plataforma Teensy e dos Shields de arduino;
- Checar a finalidade dos componentes;
- Excluir os componentes não necessários;
- Alterar a fonte de alimentação;
- Desenvolver um projeto de esquemático conectando todos os componentes;
- Desenvolver uma placa de circuito impressa de forma a conectar todos os componentes sem a necessidade de fios.

Analisando os componentes, o primeiro dispositivo a ser trocado é a fonte reguladora de tensão. Como um dos possíveis objetivos posteriores do projeto poderá ser controlar o paraquedas, o acionamento de motores pode vir a ser uma necessidade, por isso optou-se por uma fonte reguladora de tensão mais robusta. A nova fonte de tensão é um *Step-Down-Regulator* da *Texas Instruments* TPS62142 [27]. Esta fonte é uma fonte chaveada capaz de alimentar o circuito em 3,3 volts e disponibilizar em até 2 Amperes a corrente elétrica. Utilizou-se o software WEBENCH DESIGN da *Texas* para projetar a fonte de alimentação.

Observou-se que para o processamento de dados da placa não era necessário a utilização de dois microcontroladores. Por isso o projeto foi desenvolvido utilizando-se somente um microcontrolador. Todo o esquemático desenvolvido foi realizado utilizando o software ALTIUM.

O sensor de corrente elétrica INA219, o sensor de infravermelho TSL2561 e o sensor ultravioleta UVM-30A foram removidos do projeto por não apresentarem informações de

relevância para coleta de dados. Os 3 sensores DS18B20 foram removidos por serem redundantes no projeto pois a placa já possui outros sensores executando a mesma tarefa.

As informações na primeira missão LAICAnSat eram armazenadas em *datalogger* (um shield para arduino que possuía um *slot* para cartão micro-sd). Entretanto este componente utiliza-se do protocolo *UART* para se comunicar com o microcontrolador. A comunicação *UART* é utilizada por muitos componentes e suas portas no microcontrolador são restritas, por isso, optou-se por gravar diretamente todos os dados em um cartão micro-sd utilizando o protocolo *SPI*.

Para uma coleta mais robusta da posição do microsatélite, optou-se por trocar o módulo GPS pelo módulo GNSS M8T da Ublox[29]. Este dispositivo é capaz de receber tanto informações GPS e GLONASS. Outra característica importante deste dispositivo é a capacidade de enviar para o microcontrolador os dados brutos da posição. Manteve-se o Xbee para comunicações a rádio para pequenas distâncias.

O módulo GNSS M8T da Ublox e o Xbee utilizam do protocolo UART para se comunicarem com o microcontrolador. Foi projetado um conector externo para comunicação via UART com outros dispositivos fora da placa.

Observe na Tabela 1 todos os sensores utilizados na placa. Estes sensores são alimentados pela tensão de 3,3 V. Os dispositivos possuem um consumo de corrente muito baixo sendo ideais para a missão. Na tabela encontra-se os protocolos utilizado por cada um dos sensores, quais as unidades físicas que eles disponibilizam, precisão e o nome da empresa que produz o sensor.

Tabela 1 – Informações dos sensores

Componente	Tensão	Corrente	Protocolo	Unidades físicas	Precisão	Empresa
BMP180	3.3V	32uA	I2C	Pressão(hPa) e Temperatura(°C)	0.12 hPa   1,5°C	Bosch
L3GD20H	3.3V	5mA	I2C	<i>Twist and Turn Motion</i>	2000dps	ST Microelectronics
LSM303DLHC	3.3V	110uA	I2C	Aceleração Campo Magnético	2g   1,3 gauus	ST Microelectronics
MS5611	3.3V	12.5uA	I2C	<i>Variometer</i>	10 cm   1,5m bar	Measurement Specialties
SHT15	3.3V	28uA	2-wire-bus	Umidade e temperatura	2 %RH   0,3°C	Sensirion

## 2.3 Protocolos de Comunicação

Nesta seção serão descritos os protocolos de comunicação utilizados pelo microcontrolador. É importante ter uma noção sobre os protocolos de comunicação para a confecção de um layout mais robusto da placa eletrônica. Por exemplo, o protocolo I2C utiliza um barramento capaz de conectar diversos dispositivos, indicando que aqueles componentes fiquem próximos em uma determinada região da placa.

### 2.3.1 UART

O protocolo de comunicação *UART* (*universal asynchronous receiver/transmitter*, que significa “Transmissão e Recepção Assíncrona Universal”), é dada de forma serial e assíncrona, sem a necessidade de que o microcontrolador gere um *clock* de comunicação para o dispositivo com quem está se comunicando [30]. O módulo GNSS e Xbee utilizam este protocolo de comunicação para trocar informações com o microcontrolador.

Como o *clock* externo foi retirado da comunicação, diminuindo a quantidade de conexões físicas, algumas definições precisam ser seguidas a risca para o sucesso da comunicação. São elas:

- *Data Bits*
- Bits de Sincronização
- Bit de paridade
- *Baud Rate*

Neste protocolo, os dados são enviados em pacotes. Cada um desses pacotes podem possuir de 10 bits a 12 bits. Iniciando necessariamente com um start bit seguido de 8 bits de dados. Pode-se escolher de 1 a 2 bits de paridade depois dos 8 bits de dados para checar se houve ruptura em algum bit enviado. Para informar ao dispositivo que o pacote foi entregue, dentro do próprio pacote existe um Stop bit informando o fim da comunicação.

Para que os dispositivos se comuniquem, ambos precisam ter exatamente o mesmo *Baud Rate*. *Baud Rate* é a frequência com que cada bit vai ser enviada. O *Baud Rate* utilizado no projeto foi de 9600 bits por segundo.

Com o *Baud Rate* configurado, o dispositivo que receber um start bit vai saber exatamente quanto tempo vai demorar para que, dentro daquele canal de comunicação, um outro bit for enviado. Veja Figura 7.

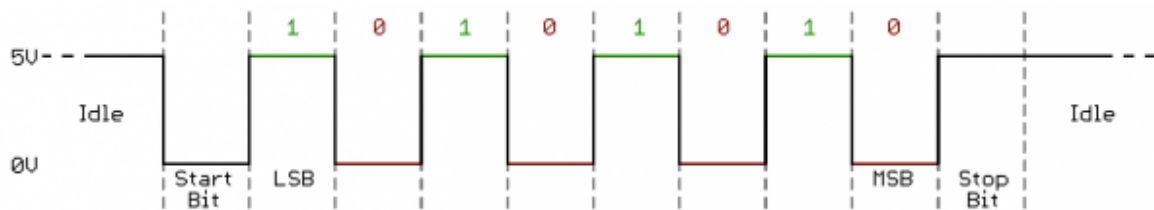


Figura 7 – Comunicação *UART* [30]

É possível visualizar na comunicação da Figura 7 que ela possui um start bit seguido de 8 bits de dados e um stop bit. Analisando esses dados percebe-se que o start bit é definido no nível lógico de tensão em 0 volts e o stop bit em nível de tensão alta de 5 volts.

O protocolo *UART* tem a vantagem de ser Half-Duplex. Ou seja, a comunicação pode ocorrer tanto do microcontrolador para outro dispositivo ou do dispositivo para o microcontrolador. Entretanto a comunicação não pode ser simultânea e algumas considerações físicas precisam ser feitas. Veja Figura 8.

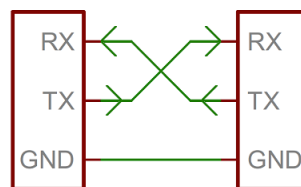


Figura 8 – Conexão *UART* [30]

Analisando a Figura 8 observa-se que o canal de comunicação RX (*Receive Data*) de um dispositivo tem que ser conectado no canal TX (*Transmission Data*) do outro dispositivo. Os dois dispositivos devem ser conectados na mesma referência de tensão, essa conexão é feita pela porta GND (Terra).

### 2.3.2 I2C

O *I2C* (*Inter-integrated Circuit*) é um protocolo de comunicação que pode conectar vários dispositivos no seu canal de comunicação. É um protocolo utilizado somente para curtas distâncias sendo requerido apenas dois canais de comunicação entre o microcontrolador e os dispositivos ligados no canal. No desenvolvimento do projeto, utilizou-se esse protocolo de comunicação para que o microcontrolador se comunicasse com os sensores.

Este tipo de comunicação pode conectar até 1008 dispositivos em seu canal. Entretanto esse tipo de dispositivo requer que apenas um dispositivo seja o mestre e os outros sejam escravos. Ou seja, os escravos apenas respondem alguma informação quando o mestre pede.



Nesta comunicação, é necessário que seja gerado um clock de comunicação. Um dos canais é apenas para que todos os dispositivos sejam conectados a ele e saibam qual o clock gerado pelo mestre, no caso, o microcontrolador. O outro canal de comunicação é estritamente para dados. O nome dado ao primeiro canal é SCL (*Clock Signal*) e ao segundo SDA (*Data Signal*) [31].

Este protocolo funciona como se fosse um broadcast. O microcontrolador quando quer saber alguma informação sobre um certo sensor, ele envia pelo canal de dados o endereço desse sensor. Esse sensor responde pelo mesmo canal de dados que está ouvindo e responde a informação pedida pelo microcontrolador.

A taxa de comunicação pode ir de 100 kHz até 400 kHz. As informações são passadas em pacotes de 8 bits podendo ter um extra para paridade. Diferentemente da comunicação *UART* não há a necessidade dos bits de start e stop.

Nos canais de comunicação e *clock* adiciona-se um resistor *pull-up* para evitar que aquele canal fique com uma tensão flutuante. Esses resistores são conectados no canal de comunicação e na tensão de 3.3 Volts para manter sempre o barramento em alta tensão. Para realizar um envio de pacote de dados, o dispositivo chaveia a tensão do barramento para 0 volts. Dessa forma os dados podem ter os níveis lógicos de baixa e alta tensão sendo o primeiro 0 volts e o segundo 3.3 volts.

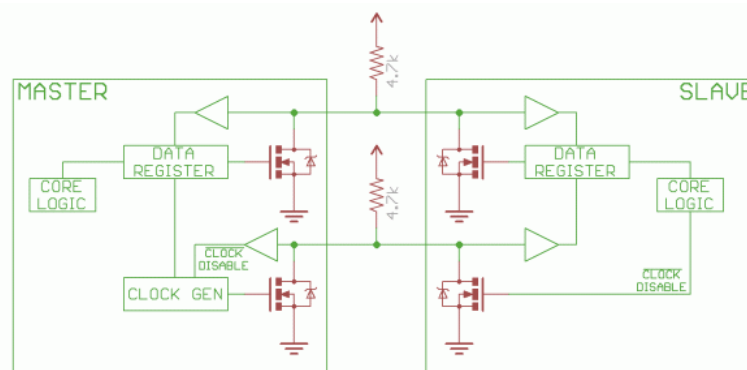


Figura 9 – Canal de comunicação I2C [31]

Analisando a Figura 9 observamos como é dado em hardware a conexão de dois dispositivos no protocolo I2C. Observa-se que a porta de comunicação dos dois dispositivos de clock estão ligadas e o de dados estão ligados entre eles. É possível notar que existe em cada canal um resistor pull-up.

### 2.3.3 SPI

Serial Peripheral Interface (SPI) é um protocolo de comunicação usualmente utilizado entre microcontroladores e seus periféricos. Neste projeto o protocolo SPI foi utilizado para se comunicar com o cartão SD. É utilizado canais separados para *clock*, dados, *SS* (*Slave Select*) [32].

Este protocolo é entendido como um broadcast como o I2C, entretanto em vez de ter que enviar o endereço do dispositivo no canal de dados ele apenas seleciona apenas o *SS* do dispositivo com qual deseja troca informações. Dessa forma quando o dispositivo tem o seu *SS* selecionado ele fica ciente de que dados que estão trafegando no canal de comunicação são para ele.

Normalmente neste protocolo utiliza-se a definição de mestre e escravo. Neste projeto o microcontrolador é o mestre que gera o *clock* e seleciona com qual escravo está falando.

SPI tem a vantagem de ser o protocolo com a maior taxa de comunicação entre os mencionados neste projeto, entretanto ele tem uma limitação de que os dispositivos conectados são limitado ao número de portas que o microcontrolador consegue suportar para selecionar o *SS*. Outra desvantagem é a grande quantidade de canais necessários para se comunicar.

Existem mais dois canais de comunicação, *MOSI* e *MISO*, no protocolo SPI. O primeiro significa Master Out Slave In. É apenas neste canal que o microcontrolador envia seus dados. O segundo significa Master In Slave Out. Este é o canal que os escravos respondem o mestre enviando os dados requeridos.

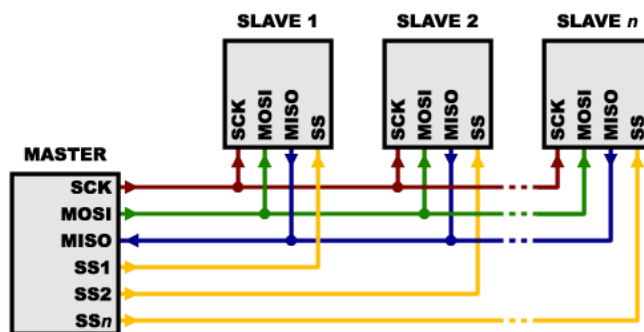


Figura 10 – Canal de comunicação SPI [32]

É possível analisar na Figura 10 como é feito o barramento de dados no protocolo SPI. O canal de clock (*SCK*) é conectado a todos os slaves saindo do master. O canal *MOSI* sai com dados do master e vai para todos os slaves e o canal *MISO* sai de todos os slaves e entram no master. Apenas o canal slave select (*SS*) é enviado individualmente, saindo do master, para os slaves.

### 2.3.4 2-Wire-Bus

O protocolo *2-Wire-Bus* utiliza dois canais de comunicação. Um para o *clock* gerado pelo microcontrolador e outro para dados.

A comunicação *2-Wire-Bus* utiliza o mesmo *clock* da comunicação SPI. A transmissão de dados ocorre por um canal de dados. Apenas o sensor SHT15 utiliza este protocolo de comunicação no projeto.

Analisando a Figura 11 percebemos como é feita a conexão entre o sensor SHT15 e o microcontrolador. Outra observação é a presença de um resistor de Pull-Up apenas no canal de dados.

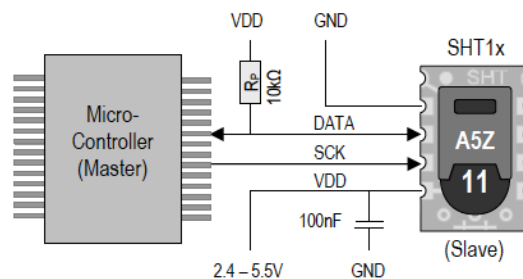


Figura 11 - 2-Wire-Bus[33]

### 2.4 Capacitor de Desacoplamento

Capacitores de desacoplamento são amplamente usados na eletrônica. Circuitos eletrônicos possuem uma demanda de corrente elétrica que não é constante, ela varia de acordo com as condições impostas pelo circuito. Por isso a necessidade de corrente pode aumentar consideravelmente em pequenos instantes de tempo. Para esse tipo de demanda, as vezes, a fonte de alimentação não consegue prover uma carga em tempo hábil. Tendo isso em vista, utiliza-se capacitores de desacoplamento que suprem essa necessidade de corrente do circuito integrado quando ela é exigida. Os capacitores são conectados próximos ao circuito integrado na entrada de tensão VCC e a tensão terra.

### 2.5 Resistor utilizados com a função Pull-Up

Os resistores de *pull-up* são resistores bastante utilizados no projeto de circuitos eletrônicos. Quando uma entrada do microcontrolador é configurada como input e um dispositivo é conectado aquela porta, a tensão em sua porta fica em estado flutuante, ou seja, sua tensão fica indeterminada. No canal de comunicação é necessário um nível de tensão lógico

esperado. Para evitar esse tipo de situação utiliza-se um resistor *pull-up*. O resistor utilizado com uma configuração pull-up garante que a entrada do microcontrolador e do canal de comunicação se ajustem a um nível lógico esperado. Neste manuscrito utilizou-se resistores nesta configuração no barramento do protocolo I2C para realizar a comunicação dos sensores com o microcontrolador. Veja um exemplo de resistor pull-up na Figura 9.

A ideia do resistor *pull-up* é que ele fracamente “puxe” a tensão no condutor que ele esta se conectando para a tensão 3,3 volts. Contudo, o resistor é intencionalmente de valor alto, geralmente de 10k ohms, suficiente para que caso o microcontrolador ou o próprio sensor queira “puxar” a tensão para 0 volts ele consiga.

## 3. FERRAMENTAS DO PROJETO

Para o desenvolvimento do projeto foram utilizado softwares de desenvolvimento de PCB (Printed Circuit Board), simulação de circuitos, desenvolvimento do firmware e uma calculadora de impedância. Para desenvolvimento da PCB foi utilizado o Altium design, para simulação de circuito o da Texas instruments, criação do firmware foi utilizado a IDE do Arduino e para alguns cálculos de impedância foi utilizado o PCB Saturn.

Este capítulo visa explicar como funcionam os software e alguma de suas características importantes para o projeto.

### 3.1 ALTIUM

Altium é uma ferramenta poderosa para o desenvolvimento de PCB's. Utilizando este software é possível desenhar o esquemático de todo o circuito e desenvolver a placa de circuito impressa [34].

Primeiramente será explicado como é feito o esquemático junto com algumas aplicações essenciais para reproduzir o projeto. Depois será explicado a ferramenta de criação de PCB's.

#### 3.1.1 Esquemático

A ferramenta Altium permite criar um projeto com vários subprojetos. Estes projetos são conectados por um projeto superior que assemelha ao um diagrama de blocos informando como os subprojetos se comunicam.

Este projeto superior é chamado no projeto de Sumário. Ele se assemelha a um diagrama de blocos que informa quais são os blocos pertencentes do projeto inteiro e como eles se comunicam. Exemplo seria como foi feito no projeto, onde o bloco do microcontrolador se comunica com os sensores e todos eles são alimentados pela fonte de tensão.

No mundo da eletrônica existem diferentes tipos e formas como os componentes são descrito e utilizados. Este software já possui uma biblioteca de componentes discretos como por exemplo resistores e capacitores. Nesta biblioteca se encontra o desenho técnico dos componentes mais o seu footprint. Footprint é o desenho da região aonde o componente vai ser soldado na placa de circuito impresso. Uma ferramenta auxiliar é a possibilidade de colocar junto ao footprint o desenho em 3D, dessa forma facilita a visualização total da sua placa.

Como o mundo dos componentes eletrônicos é muito abrangente a biblioteca padrão não atende todos os componentes, por isso o programa oferece uma ferramenta que possibilita a criação desses componentes. Nessa ferramenta é possível criar o desenho técnico e o seu footprint. Veja na Figura 12 o desenho técnico e o *footprint* criados de um transistor 2N3904.

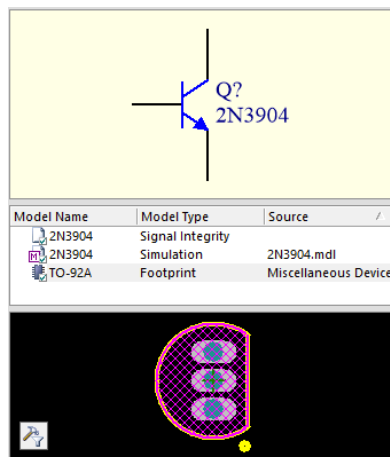


Figura 12 – Criação do transistor 2N3904 [35]

Possuindo todos os componentes do projeto basta coloca-los na parte do esquemático para a criação do projeto. Um exemplo de como o circuito pode ser criado é mostrado na Figura 13.

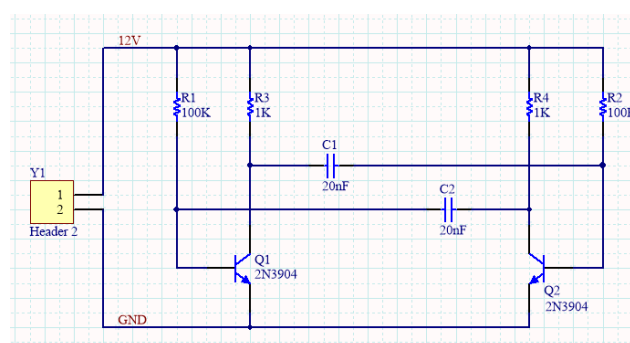


Figura 13 – Esquemático Altium [35]

Para a criação do circuito basta colocar os componentes necessários para o projeto e conecta-los. Dentro deste esquemático é possível criar portas que serão conectadas em outros blocos do projeto. Outra utilização importante é a criação de net labels. Analisando a Figura 13 é possível perceber que existem duas net labels, um de 12V e outro de GND. É importante nomear todas as conexões importantes utilizando o netlabel. Exemplo seriam a

tensão de 12 volts, a tensão terra e também todas as trilhas de comunicação como I2C, *UART* entre outras.

Depois de criado todo o circuito, utiliza-se um compilador para checar todas as conexões estão coerentes. Passando pela análise do compilador pode-se começar a desenhar o layout da PCB. [35]

### 3.1.2 Layout

Para começar o Layout da placa, primeiramente, é importante modelar o tamanho da placa. O Altium disponibiliza a ferramenta para desenhar a placa com as dimensões exigidas pelo projeto. Outra função importante é a possibilidade de criar zonas de *Keep out*, ou seja, o programa vai gerar uma violação caso algum componente esteja no perímetro delimitado como ilegal.

Possuindo a placa já nos padrões corretos coloca-se os componentes dentro da placa nas posições desejadas.

O programa possui uma característica que auxilia muito na construção da placa. O software cria uma fina linha branca em linha reta realizando todas as conexões entre elementos.

As conexões, nas placas de circuito impresso, são feitas por trilhas ou por polígonos. O Altium permite que as trilhas sejam traçadas diretamente, com angulação de 45 graus ou até mesmo curvas. É possível também gerar polígonos em áreas onde seja necessário ter uma área maior de conexão como planos terra e planos de tensão.

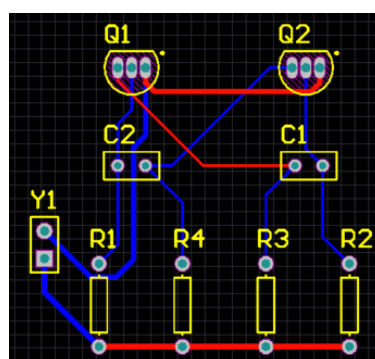


Figura 14 – Layout Altium [35]

Analisando a Figura 14 percebemos como o software realiza as conexões do projeto mostrado na Figura 13. Neste exemplo temos conexões em linha reta e linhas com 45 graus. É possível analisar também que as trilhas em azul estão em uma camada e as trilhas em vermelho em outra camada.

No projeto da PCB é possível determinar quantas camadas de cobre serão utilizadas junto com sua espessura. É possível também determinar a espessura do dielétrico utilizado [35].

### 3.2 Arduino IDE

Para a realização da programação do firmware foi utilizado a IDE (*Integrated Development Environment*, em português, Ambiente de desenvolvimento Integrado) do Arduino. A linguagem do Arduino é basicamente em C++ com pequenas modificações [36].

A linguagem em C++ é uma abstração para os seres humanos poderem programar um microcontrolador. Utilizando a linguagem C++ podemos colocar as ideias do projeto em uma linguagem entendida por pessoas e depois reproduzi-las em linguagem de máquina.

A utilização da IDE do arduino permite a criação do nosso código e a reprodução desse código em linguagem de máquina. Observe na Figura 15 uma ilustração da IDE do arduino.

A screenshot of the Arduino IDE window titled "Blink | Arduino 1.0". The window shows a code editor with the following C++ code for a blink program:

```
/*  
 * Blink  
 * Turns on an LED on for one second, then off for one second, repeatedly.  
 *  
 * This example code is in the public domain.  
 */  
  
void setup() {  
  // initialize the digital pin as an output.  
  // Pin 13 has an LED connected on most Arduino boards:  
  pinMode(13, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(13, HIGH); // set the LED on  
  delay(1000);           // wait for a second  
  digitalWrite(13, LOW); // set the LED off  
  delay(1000);           // wait for a second  
}
```

Figura 15 – IDE Arduino [36]

Foi utilizado a linguagem do arduino pois ela é uma linguagem mais alto nível do que assembly. Por ser uma linguagem mais alto nível a programação fica de certa forma mais simples. O arduino também disponibiliza diversas bibliotecas prontas que auxiliam na programação dos sensores e outros dispositivos.

Apos a criação do código, podemos compila-lo e envia-lo para o microcontrolador. Como foi utilizado a plataforma Teensy utilizamos o bootloader da teensy [18].

Este bootloader apenas pega o programa compilado na IDE do Arduino e transmite ele para o microcontrolador via USB. Veja uma ilustração na Figura 16.

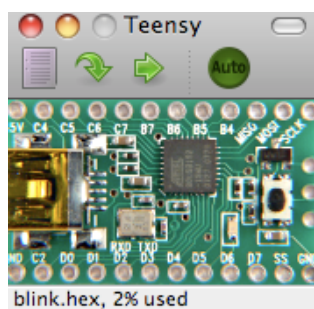


Figura 16 - Bootloader Teensy [18]



### 3.3 WEBENCH DESIGN

Webench Design é uma ferramenta de simulação de circuitos desenvolvida pela Texas Instruments. Esta ferramenta é muito útil para a simulação de circuito envolvendo conversores de tensão [37].

Este software disponibiliza o esquemático, a forma do layout da placa junto com a dissipação de calor que aquele circuito vai produzir. É importante sempre simular o circuito para testar se ele realmente vai atender as especificações do projeto. Observe a ilustração do software na Figura 17.

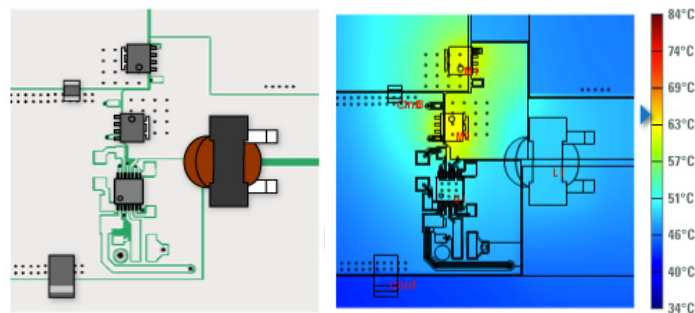


Figura 17 – Webench Design [37]

Observando a ilustração da Figura 17 percebemos que o software informa como posicionar os componentes e também como devem ser traçadas as trilhas. Observando a imagem da direita observamos a dissipação de calor.

### 3.4 PCB SATURN

O software PCB Saturn é uma calculadora utilizada para diversos cálculos para PCB [38]. Foi utilizado no projeto este software para calcular a impedância de 50 ohms necessária para a antena e a impedância de 90 ohms para a impedância para o par diferencial da comunicação USB. Dentro do software é possível indicar a distância entre camadas, como a trilha esta sendo traçada entre outras opções para calcular a largura da trilha para casar a impedância necessária. Veja uma ilustração do programa na Figura 18.

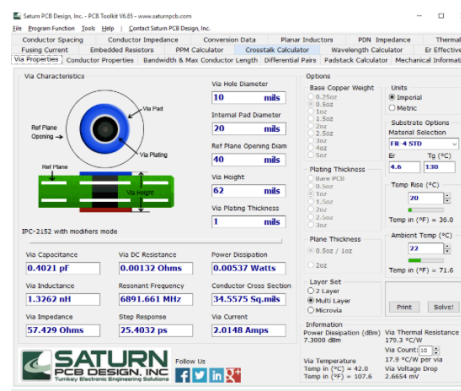


Figura 18 - PCB Saturn [38]

## 4. PROJETO LAICAnSAT-3

Neste capítulo será abordado a concepção do esquemático e do layout da placa de circuito impressa. Logo após será feita uma análise dos dados obtidos através de códigos de teste implementados na placa. Encontra-se na Figura 19 uma descrição, por meio de um diagrama de blocos, de como funciona a placa.

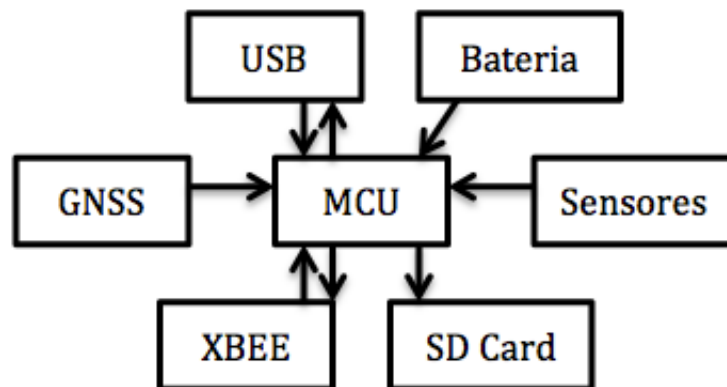


Figura 19 – Flowchart placa LAICAnSat

O microcontrolador (MCU) é o principal elemento da placa. É ele que vai enviar e receber todas as informações na placa. Para programa-lo utiliza-se a USB que conectada no computador pode transmitir o código da placa. A bateria alimenta o dispositivo e a todos os outros componentes da placa. Os sensores e o GNSS enviam os seus dados para o microcontrolador que pode armazená-los no cartão de memória micro SD ou pode enviá-los via rádio por meio do Xbee.

### 4.1 Esquemático

Nesta sessão será abordado sobre o esquemático produzido para o projeto. O projeto foi dividido em blocos cada um com suas devidas características e unidos após por um bloco Sumário. Cada um dos blocos será discutido em tópicos separados. Os blocos são nomeados de:

- Sumário
- GNSS-Xbee-SDCard
- IMU+Sensores
- Power Supply
- MCU

### 4.1.1 SUMÁRIO

O sumário é o documento principal do projeto por ser responsável por conectar todas as partes separadas do projeto. Veja Figura 20. Dentro deste documento encontram-se diversos subprogramas como IMU, MCU GNSS\_XBEE\_SDCARD e PowerSupply. É neste documento que as diferentes partes do projeto se comunicam. É possível observar na Figura 20 que o esquemático da IMU se comunica com o MCU por meio das portas I2C\_SCL, I2C\_SDA, SHT15\_DATA e MCU\_SCK. Essas portas indicam que dentro de cada bloco de esquemático vai ter uma porta aonde componentes se conectaram a ela. Dessa forma fica mais simples criar projetos grandes e interligar diferentes partes no mesmo projeto. O único bloco que não possui conexão é o da fonte de alimentação, entretanto como ele fornece só tensão para os componentes, não é necessário criar uma porta só para isso.

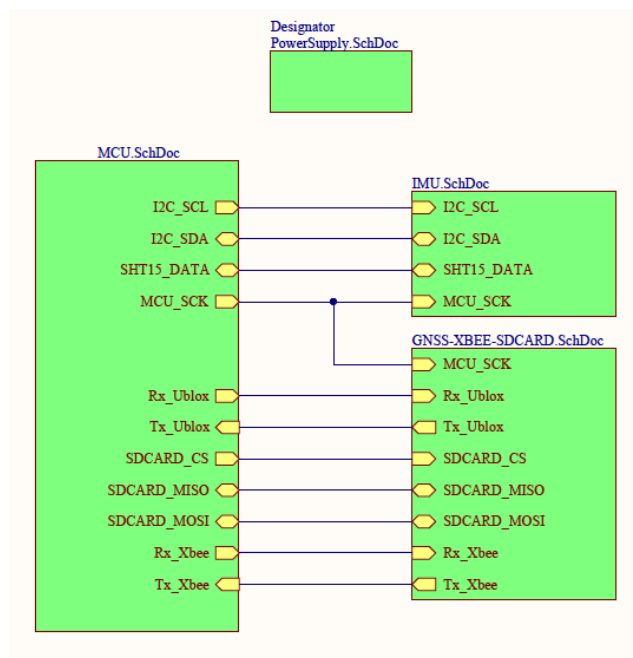


Figura 20 – Sumário

### 4.1.2 GNSS-XBEE-SDCARD

Este bloco possui um módulo GNSS LEA M8T, um Xbee e um SDCard. O módulo GNSS é responsável por receber a posição geográfica e o Xbee por fazer transmissão via rádio. Ambos trabalham com uma frequência alta e devem ser separados no desenvolvimento da PCB. O SDCard armazena os dados enviados pelo microcontrolador e

foi adicionado nesta sessão porque criar um bloco apenas pro cartão de memoria não seria algo muito vantajoso. Cada um dos componentes será explicado separadamente.

#### 4.1.2.1 GNSS

Utilizou-se o módulo GNSS LEA-M8T produzido pela Ublox [29]. Este dispositivo suporta as constelações de satélites do GLONASS, GPS e BeiDou. A utilização deste equipamento é importante para a pesquisa pois ele enviará para o microcontrolador informações sobre o tempo e a posição geográfica do experimento. Existe ainda a possibilidade de obter os dados brutos adquiridos pelo módulo.

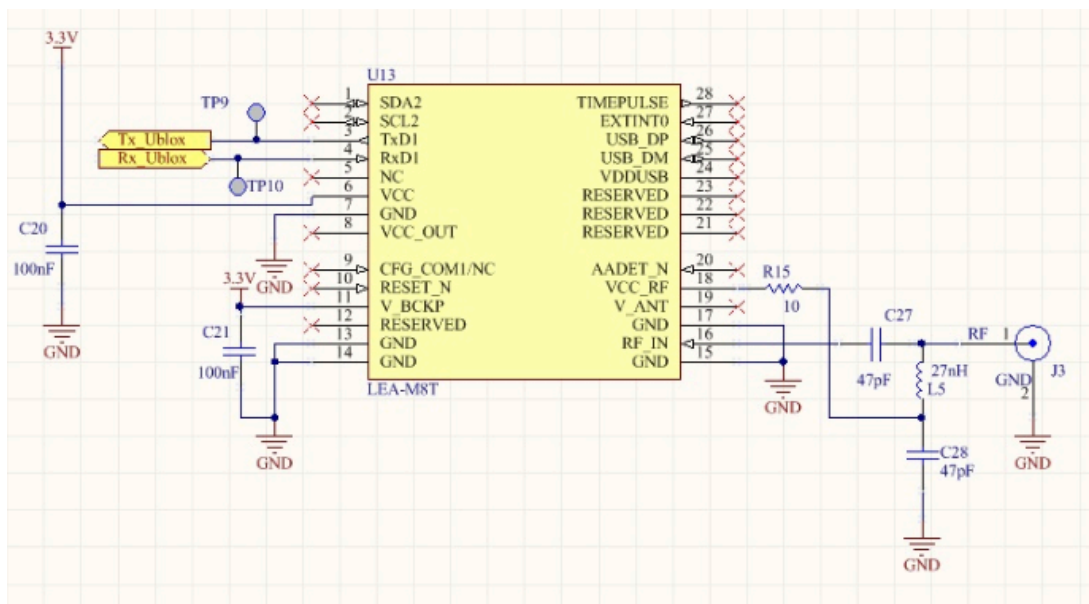


Figura 21 – GNSS

Na Figura 21, encontra-se o esquemático utilizado para o GNSS. O dispositivo é alimentado por uma alimentação de 3,3 Volts e se comunica com o microcontrolador via serial pelas entradas 3 e 4 utilizando o protocolo UART. A porta 3 se conecta a uma outra porta chamada de Tx\_Ublox. Essa porta cria uma conexão entre elementos de esquemáticos diferentes dentro de um mesmo projeto.

Os capacitores C20 e C21 são capacitores de desacoplamento. O circuito a direita da imagem é utilizado para alimentar a antena externa (conector J3). Este circuito é estipulado pelo datasheet do componente. O Capacitor C27 é utilizado para filtrar uma tensão DC, deixando apenas que o sinal RF seja transmitido para dentro do módulo. É o próprio modulo que alimenta a antena pela porta 18. Existe um resistor R15 para limitar a corrente que sai do modulo para a antena e um indutor L5 e um capacitor C28 para filtrar e controlar a tensão na antena.

A junção do capacitor junto com o indutor L5 permite que a antena consiga transmitir informações pelo mesmo canal por onde é alimentada. Essa informação é enviada em uma certa frequência que o capacitor C27 permite passar para a entrada 16 (RF\_IN) e o indutor em conjunto com o capacitor não permite que essa variação no canal afete a fonte de alimentação vinda do módulo.

#### 4.1.2.2 XBEE

O Xbee é um dispositivo utilizado para se comunicar com outros dispositivos Xbee via radio frequência. Como o Xbee possui o mesmo formato e as mesmas entradas, não se especificou um Xbee tornando possível conectar qualquer modelo de Xbee. Veja Figura 22.

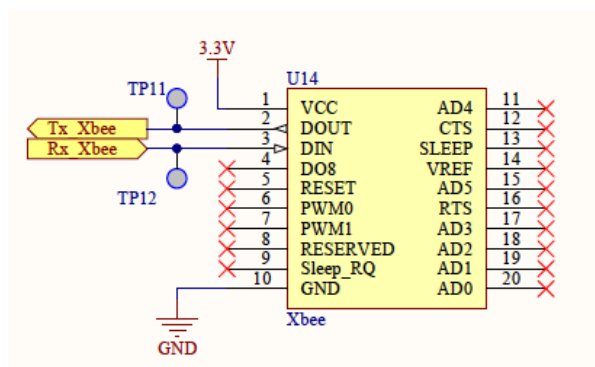


Figura 22 – Xbee

O Xbee necessita somente de uma alimentação 3,3 Volts. Ele se comunica com o microcontrolador via serial pelas portas 2 e 3 utilizando o protocolo de comunicação UART.

#### 4.1.2.3 SDCARD

O SD Card é um cartão micro SD aonde serão armazenadas as informações coletadas pelo microcontrolador.

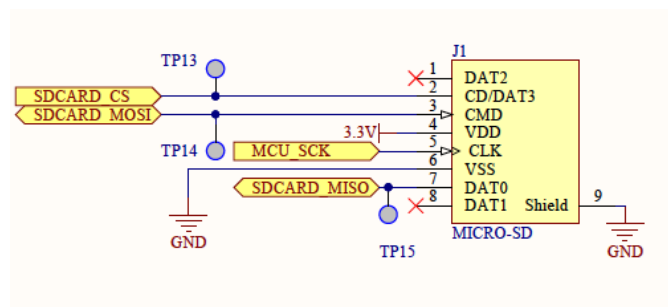


Figura 23 – Micro Cartão SD

O esquemático do micro SD encontra-se na Figura 23. O cartão se comunica com o microcontrolador utilizando o protocolo SPI. O cartão SD necessita somente de uma alimentação de 3,3 Volts e utilizar as conexões de acordo com a Figura 23. O slot do cartão de memória no esquemático é referenciado pelo conector J1.

### 4.1.3 IMU+SENSORES

Todos os sensores foram colocados nesta sessão. A IMU constitui de um sensor 3D acelerômetro e 3D magnetômetro(LSM303DLHC), um sensor de torção e giro (L3GD20H) e mais um sensor de temperatura e pressão (BMP180). São somados a essa sessão um sensor de temperatura e umidade (SHT15) mais um sensor de altímetro e variômetro (MS5611).

A IMU mais o sensor MS5611 são conectados ao barramento I2C de comunicação com o microcontrolador, já o SHT15 se comunica utilizando o protocolo 2-wire-bus.

#### 4.1.3.1 Sensor variometer

O MS5611, produzido pela *Measurement Specialties*, é um altímetro de alta precisão que possui internamente um elemento de medição de pressão linear. Este sensor é muito utilizado em equipamentos de avião e equipamento pessoais de navegação [20].

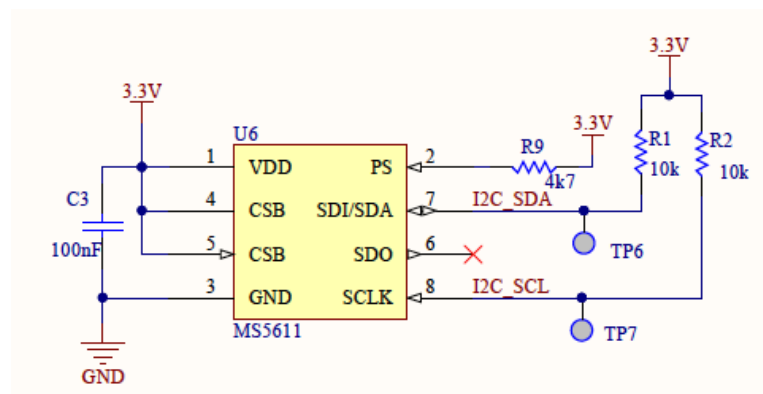


Figura 24 – Sensor variometer MS5611

O esquemático do sensor MS5611 encontra-se na Figura 24. Este dispositivo é alimentado por uma tensão 3,3 Volts pela porta 1. É conectado um capacitor de desacoplamento C3 de 100nF.

Para este sensor é necessário utilizar um resistor de pull-up na entrada PS (porta 2) do dispositivo. Isto é necessário para que o sensor saiba que a comunicação com o microcontrolador será dada pelo protocolo I2C.

É necessário também conectar a entrada CSB (portas 4 e 5) do dispositivo em alta tensão. Isso é feito para alterar o ultimo numero do endereço utilizado pelo sensor no protocolo I2C. Se essa porta fosse aterrada teríamos o problema do sensor possuir o mesmo endereço do sensor de pressão BMP180.

Na entrada SDA do dispositivo (porta 7) é nomeado um label de I2C\_SDA. Ou seja, todas as conexões que tiverem esse label I2C\_SDA serão conectadas entre si. Nesta trilha é conectado um Test point (TP6), este test point é um pequeno circulo da placa impressa onde é possível conectar a ponta de prova de um osciloscópio e checar a comunicação. Para completar o circuito é conectado um resistor pull-up R1.

Na entrada SCLK (porta 8) possui as mesmas características da SDA entretanto com a net label nomeada de I2C\_SCL, o test point de TP7 e o resistor pull-up R2.

A grande diferença entre elas é que na porta SCLK é transmitido o clock gerado pelo microcontrolador e o SDA é o canal que transmite dados.

#### 4.1.3.2 Sensor Angular de Três Eixos

O sensor L3GD20H é um giroscópio de três eixos produzido pela ST Microelectronics. Uma de suas aplicações é em sistemas embarcados com GPS, aplicações com robôs, videogames e realidade virtual [40].

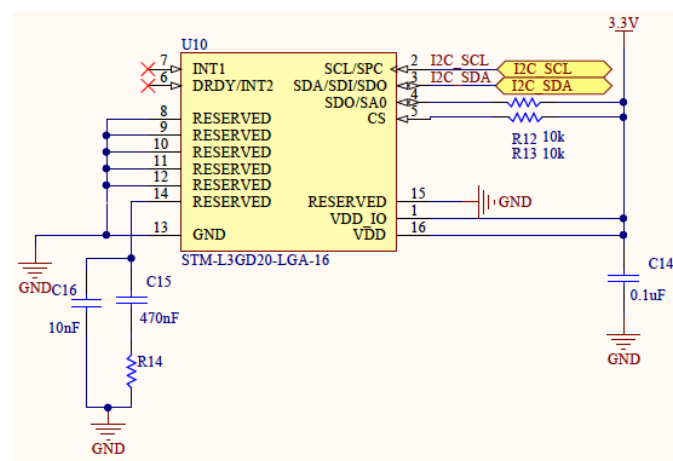


Figura 25 – Sensor de torção e giro L3GD20H

O esquemático do sensor L3GD20H encontra-se na Figura 25. Este dispositivo é alimentado por uma tensão 3,3 Volts pela porta 16. É conectado um capacitor de desacoplamento C14 de 0,1uF. Também conectado a fonte de alimentação de 3,3V encontra-se a porta VDD\_IO na porta 1. Essa porta usa essa tensão como fornecedor de carga para as portas de comunicação do sensor.

Utiliza-se um resistor de pull-up na porta 5 (CS) para habilitar a comunicação I2C no sensor. Outro resistor de pull-up é utilizado na porta 4 (SDO/SA) para indicar qual o ultimo número do endereço do sensor no protocolo I2C.

A porta 14 é reservada ao fabricante do sensor e ela exige de acordo com o datasheet um capacitor conectado em paralelo com um capacitor e um resistor em série.

As portas de 8 a 12 também são reservadas ao fabricante e exigem que sejam conectadas ao terra (GND). As portas 13 e 15 também são conectadas ao GND.

A porta 2 (SCL) é conectada ao label I2C\_SCL e a porta I2C\_SCL. Todas as conexões que possuem o mesmo label são conectadas entre si. Ou seja, a porta 2 do sensor L3GD20H está conectada com a porta 8 do sensor MS5611 que, por sua vez, esta conectada a um resistor de pull-up. Esta porta também está conectada a outra porta I2C\_SCL que conecta essas conexões com outras conexões de diferentes esquemáticos do mesmo projeto.

O mesmo ocorre para a porta 3 só que com o label e porta I2C\_SDA aonde ocorre a troca de pacote de dados.

### 4.1.3.3 Acelerômetro 3D e Magnetômetro 3D

O sensor LSM303DLHC é um pacote de sistemas que possui um sensor acelerômetro 3D e um sensor Magnetômetro 3D. Utiliza-se esse sensor para detecção de posição, detecção de queda livre, videogames e realidade virtual [41].

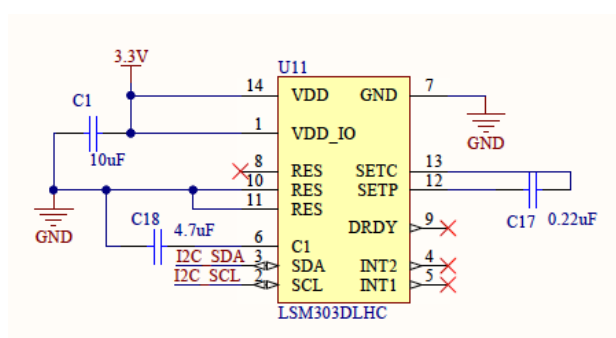


Figura 26 – Sensor acelerômetro e magnetômetro LSM303DLHC

O esquemático do sensor LSM303DLHC encontra-se na Figura 26. Este dispositivo é alimentado por uma tensão 3,3 Volts pela porta 14. É conectado um capacitor de



desacoplamento C1 de 10uF. Também conectado a fonte de alimentação de 3,3V encontra-se a porta VDD\_IO na porta 1. Essa porta usa essa tensão como fornecedor de carga para as portas de comunicação do sensor.

As portas 10 e 11 são reservadas para o fabricante e exigem pelo datasheet serem conectadas ao GND. A porta 6 exige, pelo datasheet, a conexão de um capacitor de C18 de 4,7uF. O datasheet do componente também exige que seja conectado um capacitor C17 de 0,22uF entre as portas 13 e 12.

A porta 3 (SDA) é conectada ao label I2C\_SDA e a porta 2 (SCL) conectada ao label I2C\_SCL.

#### 4.1.3.4 Sensor de Pressão e Temperatura

O BMP180 é um sensor de temperatura e pressão produzido pela Bosch. O seu baixo consumo de energia o faz perfeito para utilização em telefones celulares e Navegações em GPS [43].

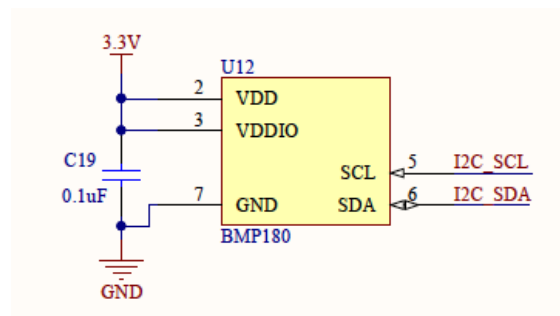


Figura 27 - Sensor de pressão e temperatura BMP180

O esquemático do sensor BMP180 encontra-se na Figura 26. Este dispositivo é alimentado por uma tensão 3,3 Volts pela porta 2. É conectado um capacitor de desacoplamento C19 de 0,1uF. Também conectado a fonte de alimentação de 3,3V encontra-se a porta VDD\_IO na porta 1. Essa porta usa essa tensão como fornecedor de carga para as portas de comunicação do sensor.

A porta 5 (SCL) é conectada ao label I2C\_SCL e a porta 6 (SDA) é conectada ao label I2C\_SDA.

#### 4.1.3.5 Sensor de Umidade e Temperatura

O componente SHT15 é um sensor de temperatura e umidade produzido pela empresa Sensirion. Este dispositivo pode ser usado para equipamentos médicos, automação e controle de processos [21].

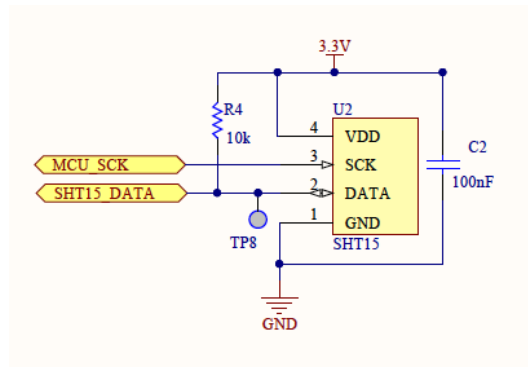


Figura 28 - Sensor de humidade e temperatura SHT15

Este sensor é alimentado pela alimentação 3,3V pela porta 4 junto com um capacitor de desacoplamento C2 100nF. Este é o único sensor que se comunica pelo protocolo 2-wire-bus. Por isso temos que a porta 3 recebe o sinal de clock de uma porta diferente do microcontrolador. A porta 3 está conectada a uma porta MCU\_SCK, mesma porta de clock que o cartão micro SD utiliza. Para o canal de dados o sensor utiliza um resistor pull-up. Este canal de comunicação, utilizado pela porta 2 do sensor, é conectado a porta SHT15\_DATA. Observe o esquemático do sensor na Figura 28.

#### 4.1.4 POWER SUPPLY

Toda placa eletrônica precisa ser alimentada por uma fonte de tensão. Entretanto essa fonte pode apresentar uma tensão superior ao necessário para os componentes. Para evitar a queima de componentes devido a esse excesso de tensão, utiliza-se um *step-down-converter* para diminuir a tensão e, então, alimentar a placa na tensão correta. O componente escolhido para reduzir a tensão foi o TPS62142 desenvolvido pela *Texas Instrument* [43].

No esquemático da Figura 29 encontra-se o regulador de tensão utilizado na placa LaicanSat. Foi utilizado o circuito integrado TPS62142 da *Texas Instrument*.

Esse dispositivo é um *step-down converter*. Este componente converte tensões entre 5 e 17 volts para uma tensão estável de 3,3 volts podendo manter em até 2 amperes a corrente de carga.

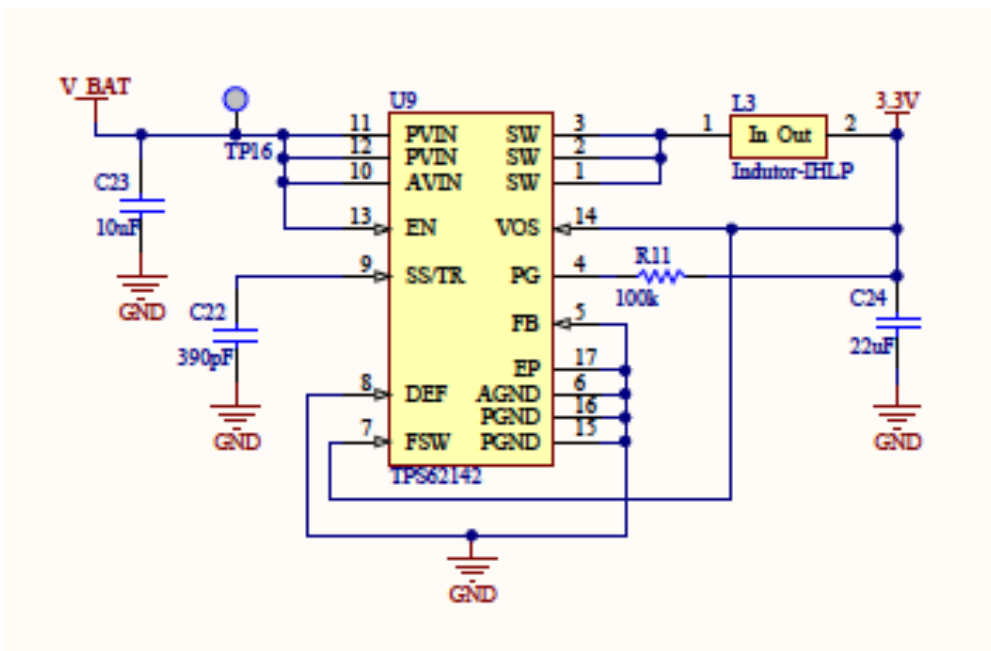


Figura 29 - Regulador de tensão TPS62142

Veja na Figura 30 uma exemplificação de como funciona um *step-down converter*.

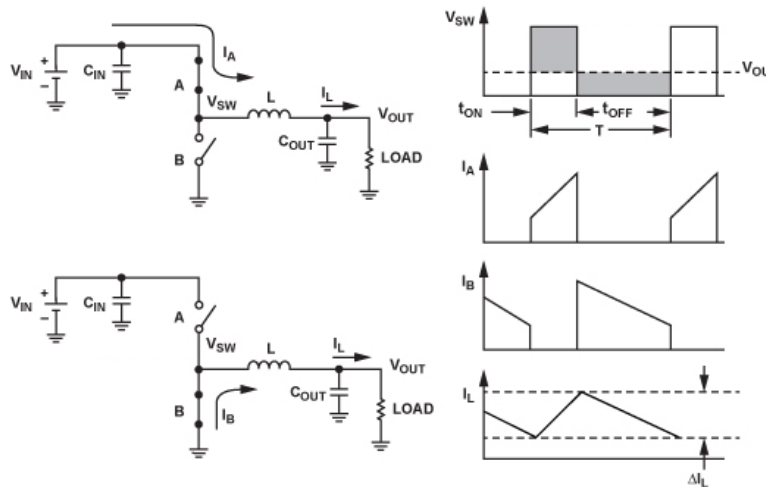


Figura 30 - Step Down Converter

Quando a chave A esta ligada e a B desligada, percebemos que o circuito carrega o capacitor  $C_{out}$  por meio de uma corrente  $I_A$  mas de forma devagar por causa do indutor L que impede uma corrente alta. Quando  $V_{out}$  chega a uma tensão determinada a chave A é desligada e a chave B é conectada. Por causa do indutor a corrente  $I_B$  do capacitor  $C_{out}$  não é descarregada rapidamente, fazendo com que a tensão decaia devagar. Quando essa tensão em  $V_{out}$  chega a um limiar inferior a chave B é desconectada e a chave é conectada

repetindo todo o processo. Dessa forma, é fácil perceber no último gráfico que a tensão fica variando na tensão determinada pelo Circuito integrado.

Na Figura 29 O nosso  $V_{in}$  seria o  $V_{Bat}$ , o  $C_{in}$  seria o C23. O indutor L é no esquemático do projeto o indutor-IHLP e o C24 seria o capacitor  $C_{out}$ . O TPS62142 é o responsável por esse chaveamento entre as chaves A e B.

O esquemático do *step-down converter* possui mais algumas conexões importantes para o seu funcionamento. A porta 14 (VOS) é uma porta de feedback para o sistema de controle do componente, por isso é conectada na saída aonde deseja-se estabilizar 3,3 Volts.

A entrada 4 (PG) conectada a saída por meio de um resistor *pull-up* serve para saber se o dispositivo esta funcionando. Caso esteja em uma tensão alta informa que a tensão adequada, caso esteja em baixa tensão o dispositivo não está funcionando de forma adequada.

A entrada 7(FSW) é uma opção para selecionar qual a frequência de switch do dispositivo. A entrada conectada em alta tensão seleciona frequência de 1,25 MHz de *switching frequency*.

Este circuito foi simulado na plataforma Webench Design. Veja Figura 31.

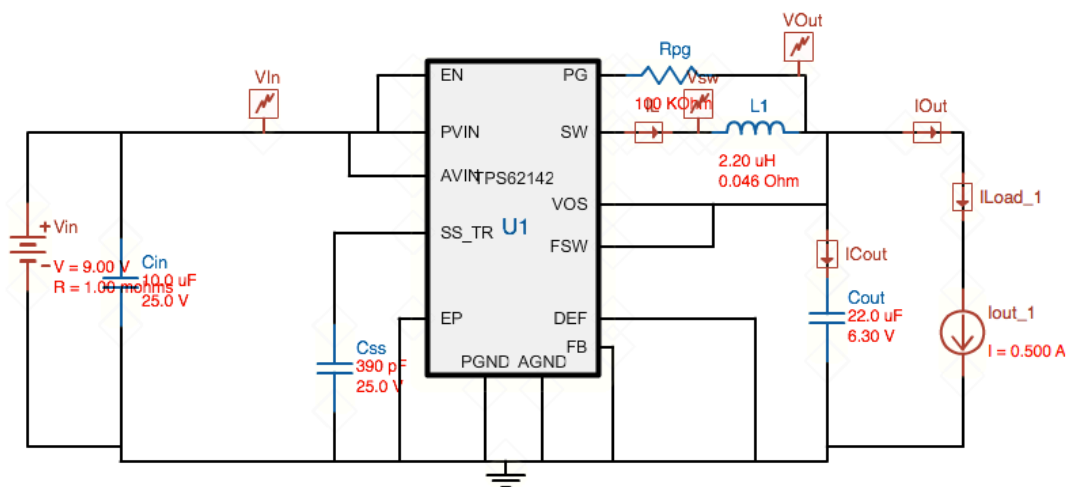


Figura 31 - Simulação TPS62142

Após efetuar a simulação foi obtido as tensões em VOut e Vsw. VOut representa a tensão 3,3V da Figura 29 e o Vsw a tensão nas portas 1, 2 e 3 do TPS62142 da Figura 29.

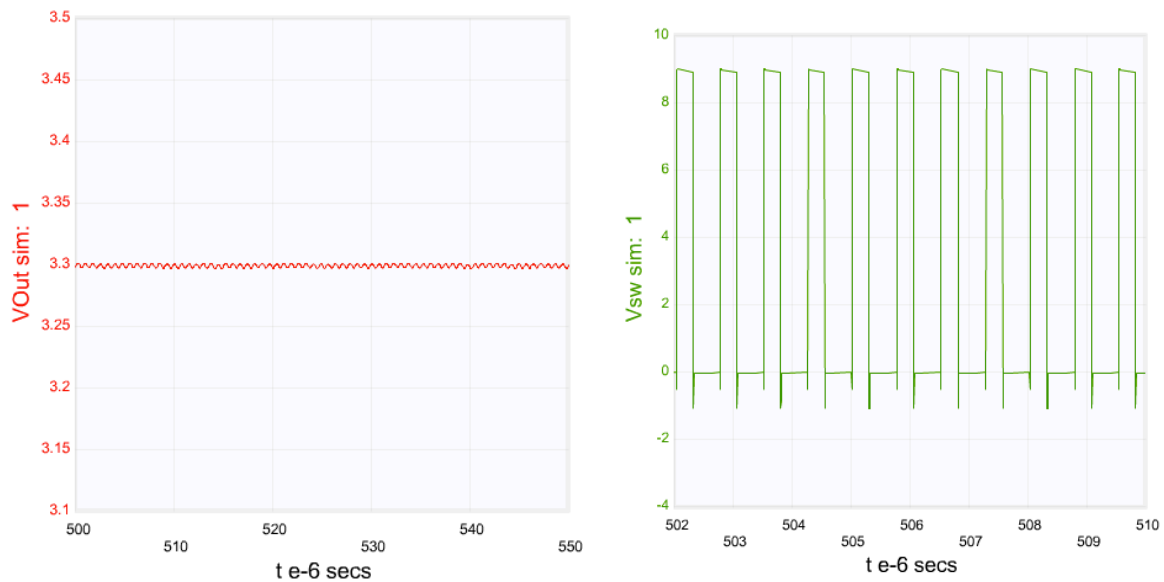


Figura 32 - Gráfico de tensão Vout e Vsw

Na Figura 32, encontra-se a esquerda o gráfico de Vout e a direita o gráfico de Vsw. Observando o gráfico de Vout percebemos que a tensão varia muito próximo da tensão desejada de 3,3 V. Ao observar o gráfico Vsw percebemos o chaveamento realizado pelo dispositivo para manter a tensão em 3,3V.

#### 4.1.5 MCU

O projeto do microcontrolador foi feito baseado no esquemático na plataforma Teensy 3.1 da PJRC Eletronic Projects [18], veja Figura 33. Esta plataforma possui um microcontrolador ARM Cortex M4 de 32 bits com um *clock* de 16 MHz.

O conector J2 da Figura 33 é o conector de micro USB. É por meio do cabo USB que o microcontrolador é energizado por 5 volts pela porta 1 (VBUS) do conector. A comunicação do microcontrolador, utilizando o protocolo de par diferencial da USB, ocorre por meio das entradas 2 (D-) e 3 (D+). Conectada em cada porta existe um resistor de 33 ohm exigidos pelo protocolos para limitar a corrente trafegada na trilha. O conector também se conecta ao terra (GND) da placa e a sua casca metálica é aterrada através de um indutor de ferrite.

Um capacitor de desacoplamento C4 de 22uF é conectado próximo ao conector micro USB. Esses 5 volts passam por dois diodos em paralelo D1 para depois ser conectado ao microcontrolador na entrada 8 (VREGIN).

A entrada VREGIN possui a função de receber 5 volts da conexão USB e reduzir para 3,3 V. Apos a redução ele devolve essa tensão reduzida de volta para o circuito através da porta 7 (VOUT33). Bem próximo a saída VOUT33 é utilizado um capacitor de desacoplamento C6

de 2,2uF. As entradas VDD 3, 30 e 48 são todas alimentadas em 3,3 Volts e possuem um capacitor de desacoplamento C9,C8 ou C7.

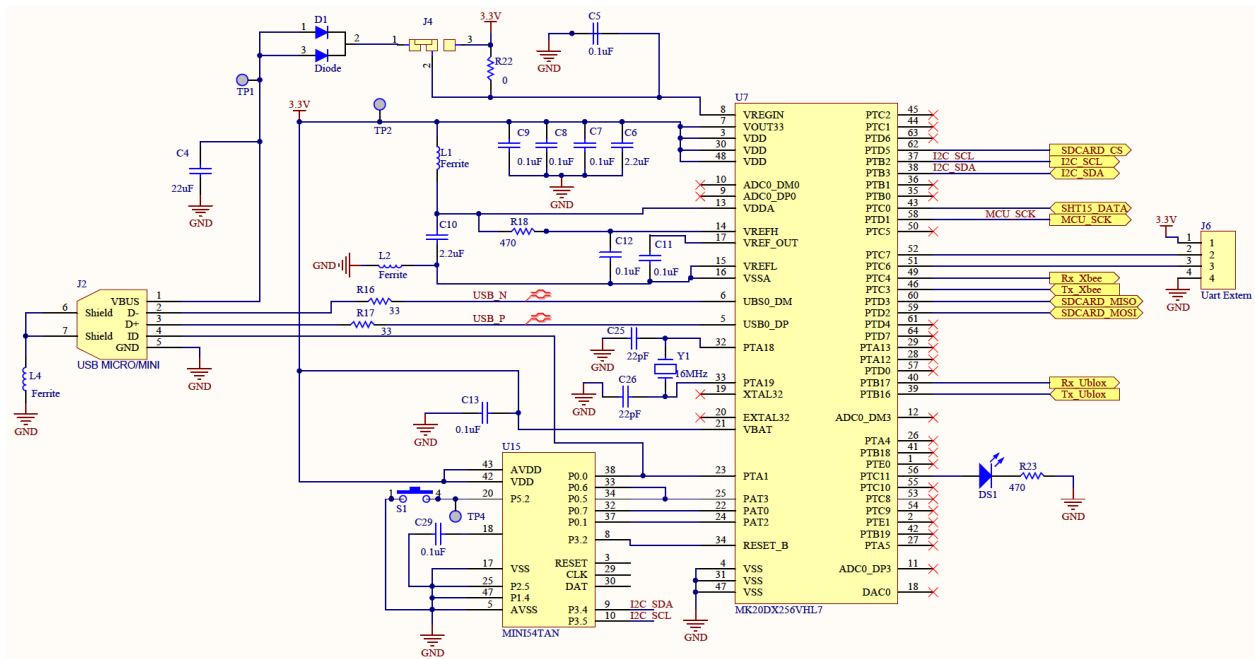


Figura 33 - Esquemático do Microcontrolador

O microcontrolador pode tanto ser alimentado pelo micro USB ou pela fonte de tensão da própria placa. Uma forma de seleccionar isso é por meio da chave J4 implementada no circuito.

As entradas de 13 a 17 do MK20DX256VHL7 são utilizadas para a conversão de sinais analógicos em digitais. VDDA é a entrada para a alimentação do conversor. A referência de alta tensão do conversor é encontrada na entrada 14 (VREFH) e deve ser conectada na mesma tensão que o VDDA. Utiliza-se um resistor R18 para fazer essa conexão e para limitar a corrente.

A entrada VSSA é a conexão terra do conversor. Conectada junto com ela tem-se a referência de baixa tensão VREFL na entrada 15. Para evitar ruídos na conversão do sinal utiliza-se um capacitor C12 de 0,1uF conectado entre as entradas 14 e 15. Utiliza-se dois indutores de ferrite L1 e L2 para filtra ruídos que possam existir no circuito. E para melhorar a filtragem o circuito de ruídos utiliza-se um capacitor C10 de 2,2uF entre as entradas 13 e 16.

O cristal de 16MHz é conectado nas entradas 32 e 33. O cristal que é o responsável pelo clock do microcontrolador. Todo cristal exige para o seu funcionamento um capacitor de 22pF em cada uma das entradas ligadas ao terra.

É necessário alimentar a entrada 21 (VBAT) com 3,3 volts e um capacitor de desacoplamento C13 de 0,1uF. Essa porta é sempre alimentada quando uma conexão utilizando bateria é utilizada.

No esquemático do microcontrolador existe um outro microcontrolador, só que com a função de *bootloader*. O componente U15, MINI54TAN, é o microcontrolador responsável pela programação do microcontrolador principal, o MK20DX256VHL7. Esse *bootloader* é um componente crítico e único no projeto. É necessário comprar este componente do site do projetador do teensy, a empresa PJRC. Este dispositivo já vem com a programação de *bootloader* necessária para o funcionamento do projeto.

O MINI54TAN já programado tem a função de *bootloader*, ou seja, ele possui um código dentro dele que torna possível a programação utilizando a linguagem de Arduino. Ao realizar a programação no computador, realizamos o processo de gravação no microcontrolador por meio do cabo USB. Como o *bootloader* não é conectado diretamente na usb é necessário, na primeira gravação, apertar o botão de reset S1. Dessa forma o *bootloader* impede o funcionamento do *watchdog* do microcontrolador e, por conseguinte, consegue comunicar com o outro microcontrolador através das entradas 22 a 25 do microcontrolador. Na hora da compilação do programa em Arduino é necessário ter instalado no computador o programa teensy loader para conseguir gravar o código.

Na entrada 56 do microcontrolador foi implementado um LED seguido de uma resistência R23 conectada ao terra. Esse LED é utilizado para a indicação de sinal GPS. É programado uma sequência de sinais longos e curtos que possibilita determinar se o microcontrolador está ou não recebendo os sinais de forma correta. Ele também posteriormente pode indicar outras funcionalidades de acordo com a programação desenvolvida.

É possível analisar no esquemático que o microcontrolador se liga por meio de portas com as outras partes do esquemático como o protocolo I2C, SPI, serial e 2-wire-bus.

Como muitas ferramentas eletrônicas se comunicam pelo protocolo UART, foi disponibilizado no projeto uma conexão serial externa pelo conector J6. Nesse conector é possível conectar um dispositivo para receber e transmitir informação, e também foi disponibilizado uma alimentação de 3,3 volts.

## **4.2 Placa de Circuito Impressa**

Toda placa de circuito integrado é dividida em camadas. Essas camadas podem ser camadas físicas ou ilustrativas. Cada camada tem suas características para serem desenvolvidas. As camadas foram divididas da seguinte forma:

- Cotas
- Toplayer
- Bottomlayer
- Topoverlay
- Fabinfo

Veja na Figura 34 onde se encontram todas as camadas do projeto. Como existe uma sobreposição das camadas da placa é impossível entender muita coisa. Por isso o desenvolvimento das placas precisam ser separados e depois vistos sobrepostos para checar se o desenvolvimento foi correto.

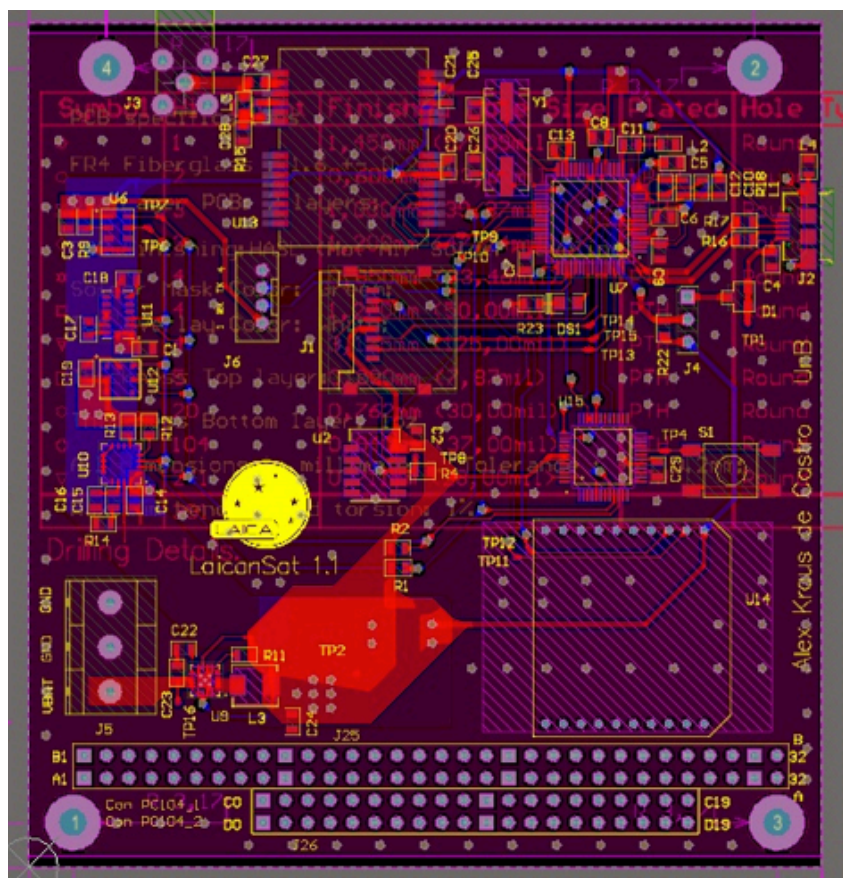


Figura 34 – LaicanSat PCB

A placa é dividida em setores para seu melhor funcionamento, Veja Figura 35. O setor MCU se localiza no canto superior a direita, os sensores no canto esquerdo, a fonte de alimentação no canto inferior a esquerda. O GPS e o Xbee trabalham com alta frequência por isso foram separados localizando o primeiro setor GNSS no canto superior a esquerda e o segundo XBEE no canto inferior a direita. No meio se encontra o setor cartão micro SD e o sensor SHT15. Optou-se de deixar o sensor SHT15 próximo ao cartão micro SD porque o sensor se comunica utilizando o mesmo clock utilizado no micro SD, facilitando assim o desenvolvimento do layout da placa.



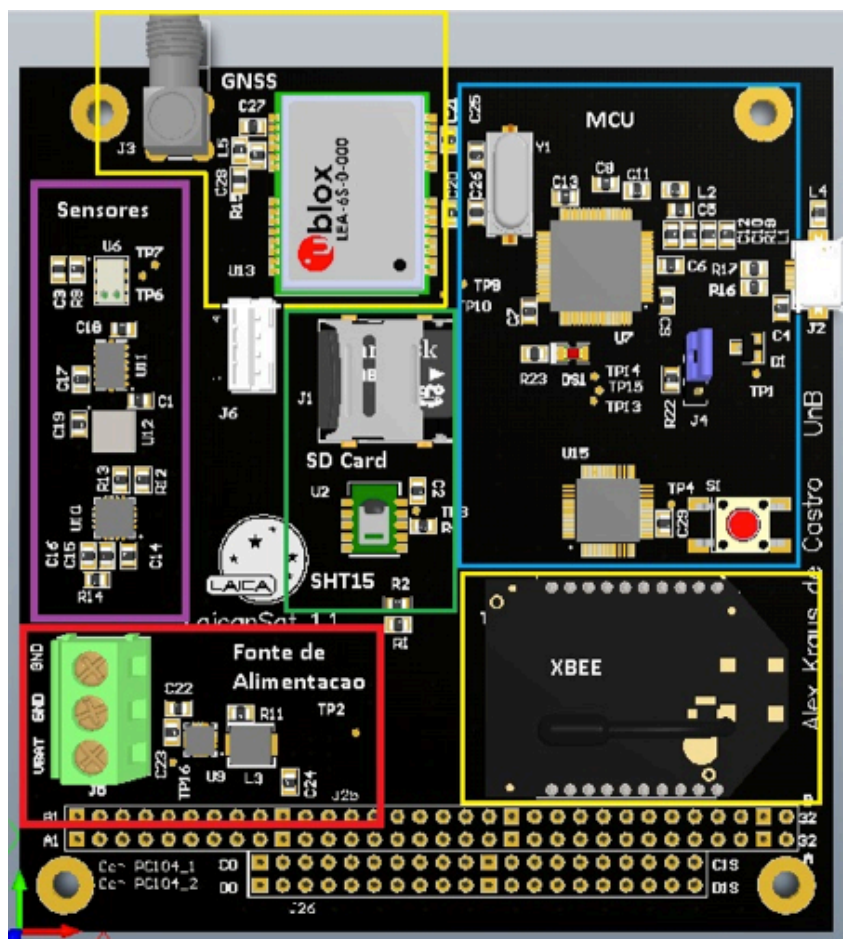


Figura 35 - Divisões da placa

### 4.2.1 COTAS

Na camada de cotas deve-se estabelecer qual é a dimensão da placa junto com furos e restrições. Para o desenvolvimento da placa utilizou-se o padrão PC104.

Na Figura 36 encontra-se o padrão PC104. O padrão estabelece que o tamanho da placa seja de 90cm de largura e 95cm de altura. Exige que existam 4 furos de 3,17 cm de raio. A indicação de aonde deve-se os furos podem ser encontradas na Figura 36. Outra exigência do padrão PC104 são furos para um conector de 16-bit ISA. A indicação desse conector será melhor detalhada na camada de Topoverlay.

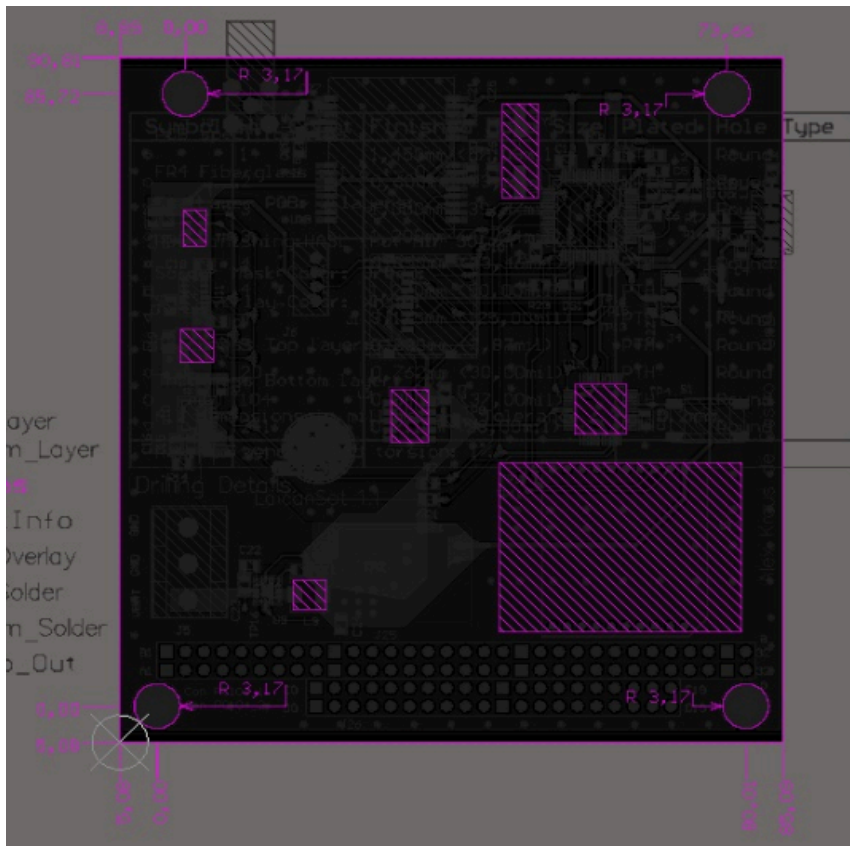


Figura 36 – Cotas

#### 4.2.2 TOPLAYER

Na camada Toplayer encontra-se a maior parte de conexões do projeto e aonde se encontram todos os componentes. Veja Figura 37.

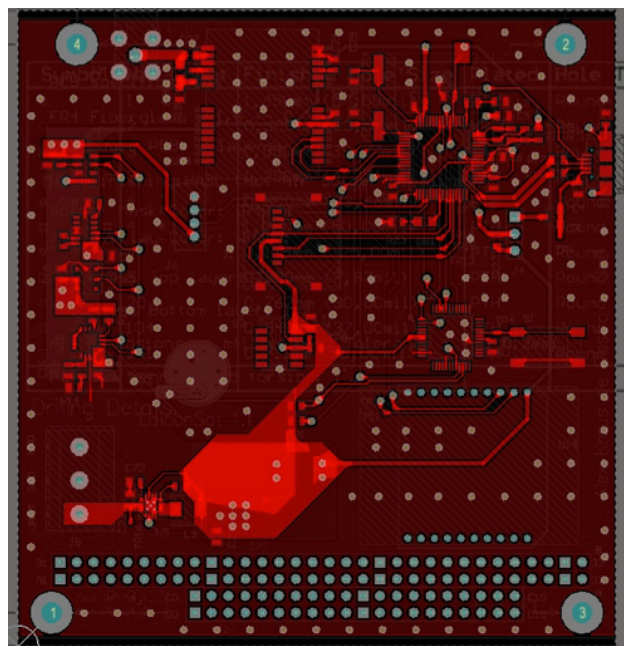


Figura 37 - Toplayer PCB

Analisando a Figura 37 percebemos varias conexões e um grande plano vermelho claro. Este plano vermelho mais claro é chamado de plano Terra. A tensão terra é distribuída por toda a placa. É possível também ver que existem vários pontos brancos no meio desse plano. Esses pontos brancos são chamados de via. Ele faz a conexão dessa camada com a camada da Bottomlayer. Existem vários dessas conexões para obter uma boa distribuição dessa tensão por toda placa evitando ruídos que podem acontecer.

Para explicar de forma mais eficiente foi removido esse plano de terra para ficar mais fácil a visualização das conexões.

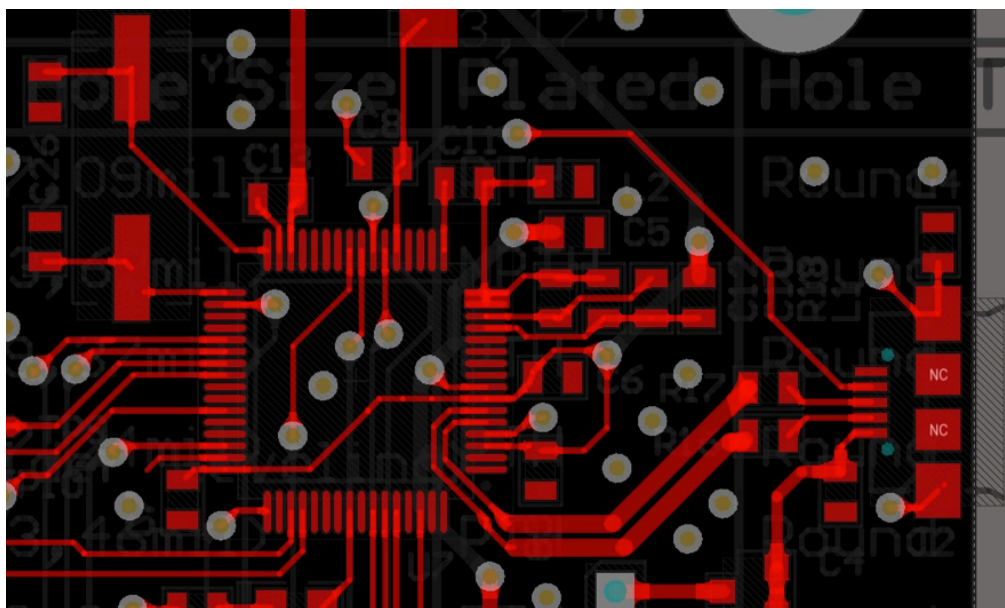


Figura 38 - Toplayer MCU

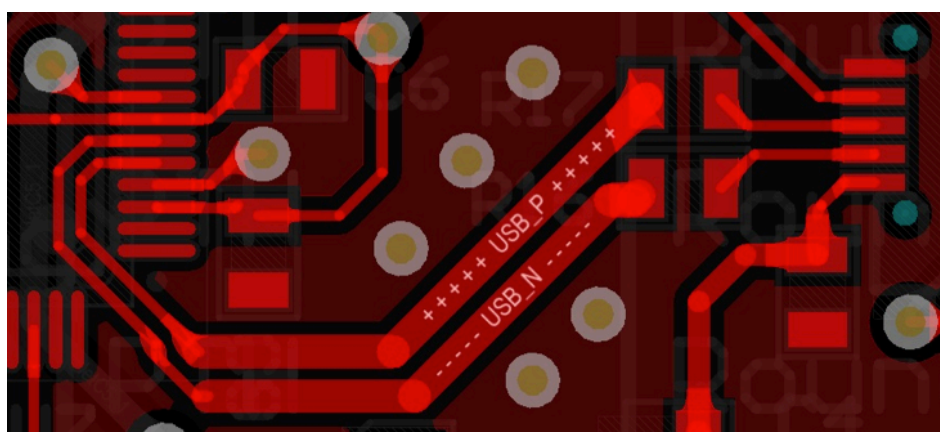


Figura 39 - Trilha USB

Na Figura 38 encontra-se as conexões ligadas no microcontrolador MK20DX256VHL7. As vias também foram utilizadas para transmitir informações. Como o plano de terra foi removido é possível perceber que existem vias de comunicação. Para identificar essas vias basta observar as vias que se conectam com alguma trilha. Essas vias são utilizadas em

ocasiões aonde não é possível traçar diretamente as trilhas. Por isso utilizamos a Bottomlayer para traçar a trilha e depois subimos a trilha novamente por meio de vias.

É de grande importância que toda trilha traçada seja feita utilizando um ângulo de 45 graus. Deve-se evitar o máximo traçar trilha com ângulos de 90 graus. Trilhas com 45 graus otimizam comunicações e evita a ocorrência de explosão de elétrons no momento da curva. Essa característica ocorre geralmente quando é utilizado uma comunicação aonde a frequência do sinal pode chegar na casa dos MHz.

Existem componentes que devem ser conectados o mais próximo possível do microcontrolador. Todo capacitor de desacoplamento, para ser mais eficaz, deve seguir essa regra, como um exemplo observe o capacitor C7 no canto inferior da esquerda do microcontrolador. Localize esse capacitor na Figura 41. Este capacitor se encontra o mais perto possível do microcontrolador.

Para a comunicação utilizando o protocolo USB deve-se atentar as trilhas de comunicação de par diferencial. Observe as trilhas de par diferencial USB\_P e USB\_N na Figura 39. Essa comunicação tem que seguir regras rígidas como ambas as duas trilhas devem possuir o mesmo comprimento e devem possuir uma impedância de 90 ohms. Deve-se evitar passar outras trilhas em paralelo. É exigido também que embaixo da trilha, na camada na Bottomlayer não passe nenhuma trilha, que seja apenas um plano de terra. Para o cálculo da impedância da trilha utilizou-se o programa PCB Saturn que indicou que para a impedância da trilha ser de 90 ohm era necessário que a largura da trilha seja de 0,726 mm.

O cristal utilizado para a frequência do clock também deve ser localizado o mais próximo possível. Ele se encontra no canto esquerdo superior do microcontrolador. Como o cristal utiliza uma frequência de 16 MHz procura-se traçar trilhas o mais reto possível para o microcontrolador. Deve-se evitar que qualquer trilha seja traçada na camada da Bottomlayer embaixo do cristal.

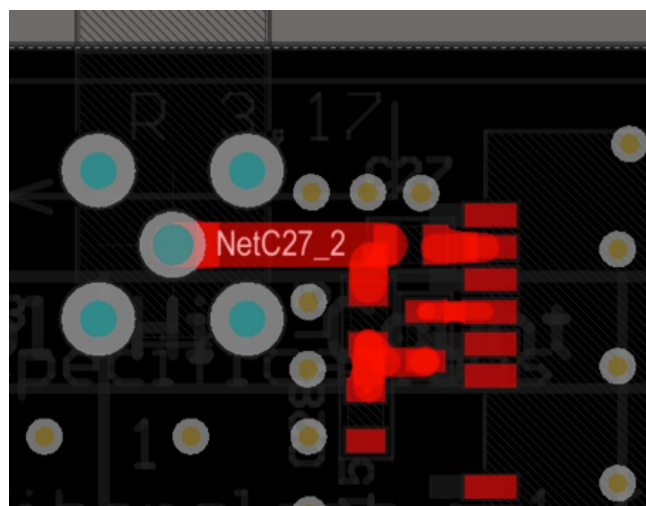


Figura 40 – Toplayer Antena

A trilha de conexão com a antena externa também tem regras rígidas. A impedância da trilha deve ser de exatamente 50 ohm. Esse cálculo também foi feito utilizando o software PCB Saturn. O software calculou que a trilha deve ter uma largura de exatamente 1,524 mm nomeada de NETC27\_2, veja Figura 40. Essa trilha deve ser traçada em linha reta. Também deve ser isolada, ou seja, foi utilizado vias conectada a tensão terra e todo o seu trajeto em volta deve ser o plano de terra. Na camada da Bottomlayer também não deve ser traçada nenhuma trilha, existindo apenas um plano de terra consistente.

Observe a Figura 41 onde se encontra a visão do microcontrolador com todas as camadas sobrepostas.

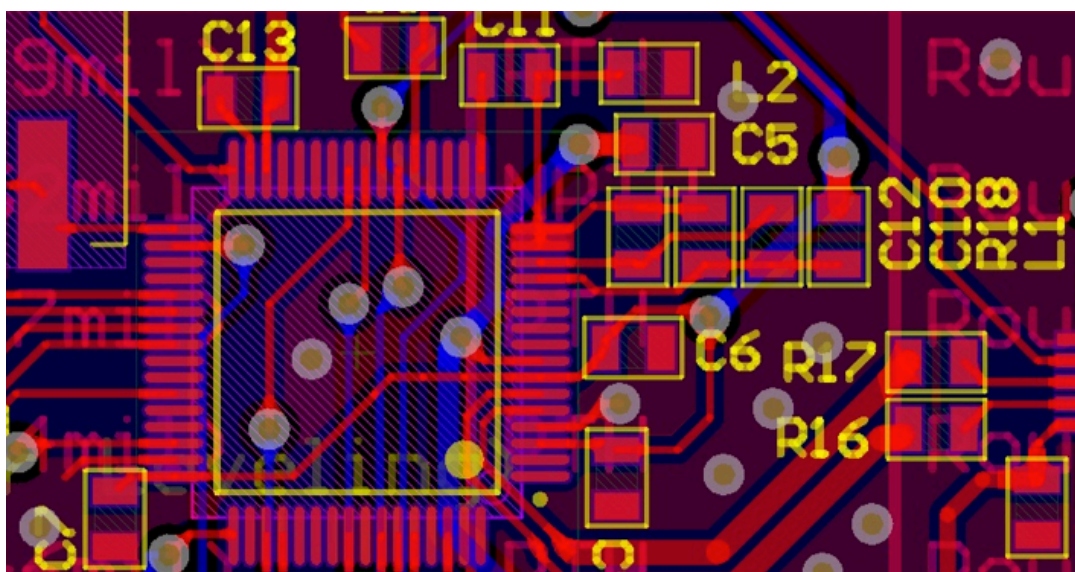


Figura 41 - MCU com camadas sobrepostas

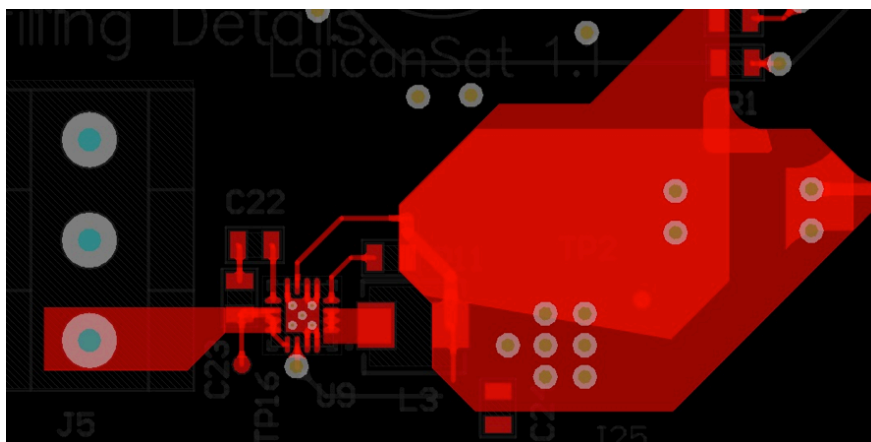


Figura 42 - Toplayer fonte de tensão

Na Figura 42 encontra-se o layout da fonte de tensão. A esquerda encontra-se o conector aonde será conectada uma bateria de 9 volts aproximadamente. Essa tensão é transmitida

para o conversor de tensão. É possível perceber um grande polígono vermelho a direita. Como a tensão de 3,3 Volts é transmitida para todo o circuito é importante ter uma grande área para essa tensão pois ela será distribuída para todo o circuito. Existem 7 vias dentro desse polígono localizadas próximas ao indutor. É necessário utilizar essas 7 vias para que o plano de 3,3 volts seja transmitido para a Bottomlayer de forma mais eficiente, uma via não seria muito eficiente no caso.

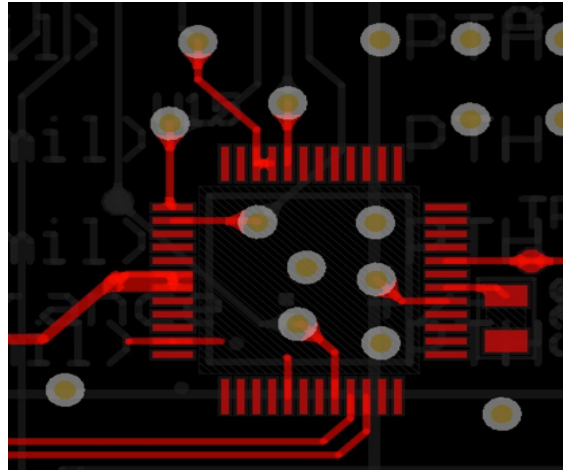


Figura 43 - Toplayer Mini54Tan

Na Figura 43 encontra-se o layout do bootloader MINI54TAN. As vias que se encontram dentro do microcontrolador são utilizadas para passa informação. Observe que todas as trilhas respeitam a regra de 45 graus. O bootloader possui um capacitor de desacoplamento mais próximo possível do componente. Acima desse capacitor você encontra um círculo, este círculo é um teste point. Test point são círculos na placa aonde é possível conectar uma ponta de teste de um osciloscópio e testar a tensão naquele ponto.

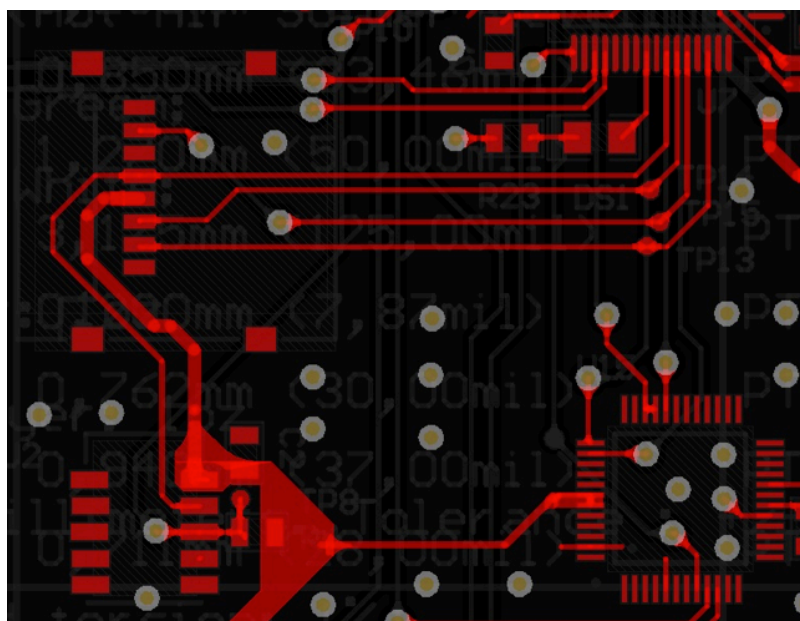


Figura 44 - Toplayer SD Card + SHT15

Na Figura 44 encontra-se o layout do SD card e do sensor SHT15. Na imagem observa-se as conexões do SD Card direto no microcontrolador. É possível observar a trilha do clock do MCU entrar no pad do SD Card e ir até o sensor SHT15. Percebemos um grande plano vermelho ao lado direito do sensor. Este plano é um plano consistente de tensão 3,3 volts que é transmitido para o sensor, o SD card e ainda vai para o bootloader.

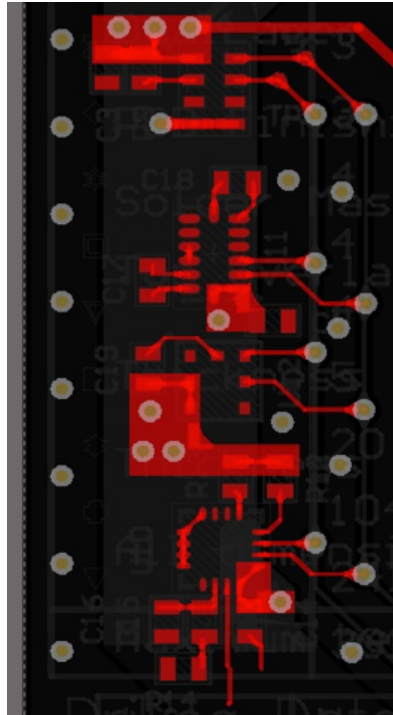


Figura 45 – Sensores

O layout dos sensores é encontrado na Figura 45. Todos os 4 sensores deste setor se comunicam por meio do protocolo I2C. Então para se comunicarem com o barramento é possível observar que todos os sensores deslocam duas trilhas para a direita se conectando cada trilha em uma via. Como será explicado na camada Bottomlayer, essas vias se conectam no barramento do protocolo I2C. Para energizar os sensores, é possível analisar que perto de cada componente tem um polígono com uma ou mais dentro dele. Como será explicado na Bottomlayer, um plano de 3,3 volts é transmitido até essa zona por meio da Bottomlayer. Por isso utiliza-se vias para energizar os sensores.

### **4.2.3 BOTTOMLAYER**

Bottomlayer é a outra camada física que pode ser utilizada para a transmissão de corrente ou tensão do projeto. Veja a bottomlayer na Figura 46.

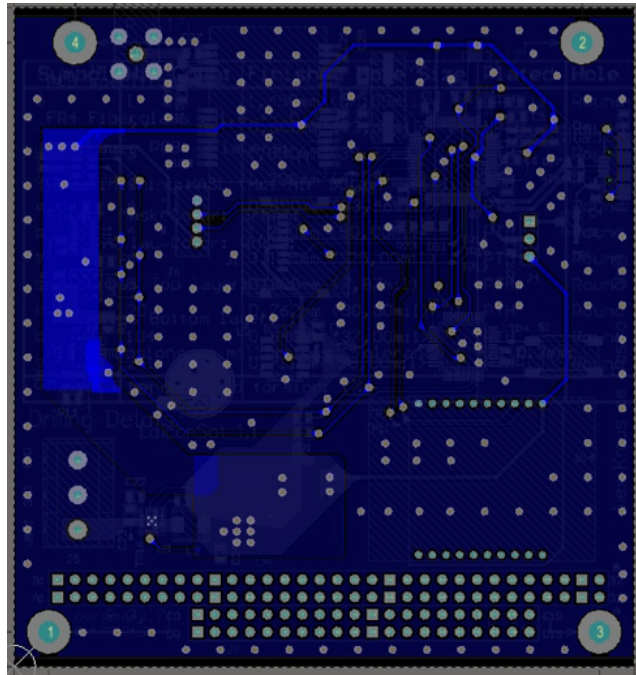


Figura 46 – Bottomlayer

Observe que a Bottomlayer possui um grande plano de terra azul claro em volta de toda a placa. Esta camada é utilizada para a conexão de trilhas aonde foi não foi possível conectar diretamente na toplayer.

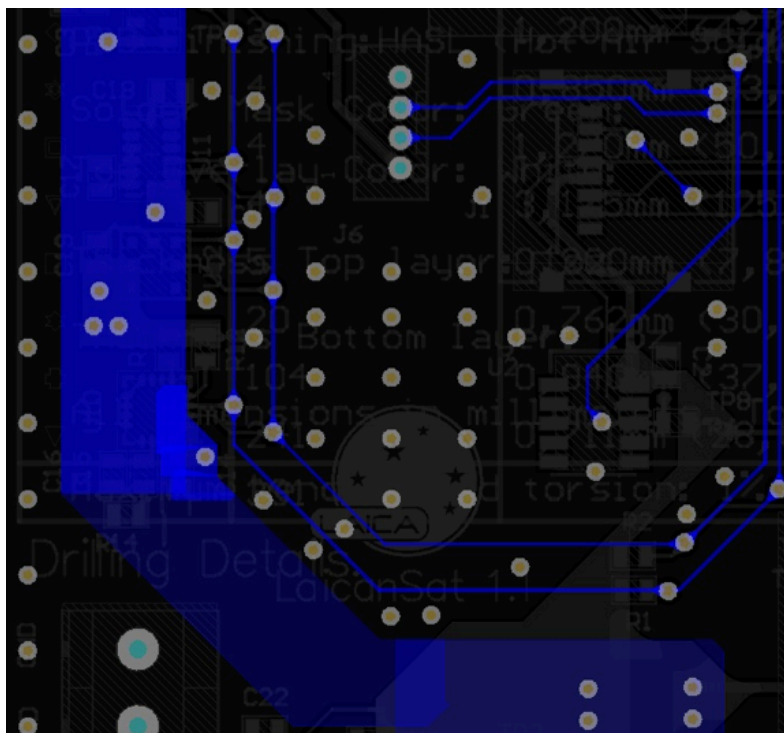


Figura 47 – Bottomlayer sensores

Na Figura 47 encontra-se uma imagem da Bottomlayer sem o plano de terra para facilitar a análise das trilhas.



No local aonde se encontram os sensores, foi desenhado um polígono na Bottomlayer. Esse polígono tem como tensão 3,3 volts. Isso foi feito para melhor transmitir a tensão para os sensores. Para conectar o sensor a esse polígono basta traçar uma trilha que se conecta em uma via conectada no plano de 3,3 volts da Bottomlayer.

Para melhor desenvolvimento do protocolo I2C, foi realizada uma trajetória das trilhas mostrada na Figura 47. Essa trajetória se assimila a o formato de U. Foi planejado essa trajetória para que ela se conecta-se ao MINI54TAN e aos sensores sem fazer uma ramificação. Ou seja, para efeito de performance, utilizando o protocolo I2C deve-se evitar ramificações neste meio físico de comunicação. Os sensores se conectam neste ramo tranchando as trilhas das entradas ate uma via que esteja conectada no ramo do protocolo I2C.

#### **4.2.4 TOPOVERLAY**

A camada Topoverlay é aonde se encontram toda a topográfica da placa. Ou seja, é nela que se encontraram os desenhos dos componentes junto com os seus designadores. Isso é necessário para que uma pessoa consiga identificar os componente sem a necessidade de precisar do software.

Nesta camada é possível encontrar qual o versionamento da placa, junto com a logo da empresa.

Como foi dito anteriormente, a posição dos conectores do formato PC104 podem ser encontrados na Figura 48. Os conectores possuem o nome de Con\_PC104\_1 e Con\_PC104\_2 e se encontram na parte de baixo da placa.

No conector da bateria, na camada Topoverlay indica qual é a correta tensão para aplicar vindo da bateria.

Algumas boas praticas de disposição de componentes também são bastante uteis para o processo de produção da placa. Veja Figura 49.

É importante que todos os componentes tenham seus designadores próximos. Como exemplo temos o indutor L2 que possui seu designador L2 ao lado direito de aonde deve ser posicionado o componente. Em alguns casos não existe a possibilidade de colocar o designador ao lado do componente, por isso tenta-se arranjar uma logica parecida com a dos componentes C12,C10,R18 e L1. O componente mais a esquerda é o capacitor C12 seguido do capacitor C10,R18 e L1.

Outro ponto importante é sempre ter um indicador indicando aonde se encontra o pino de número 1. Vide o microcontrolador U7 na Figura 49. Dentro do quadro que representa o seu

componente existe um circulo amarelo indicando que a porta 1 do componente se localiza na primeira perna do circuito integrado da lateral direita.

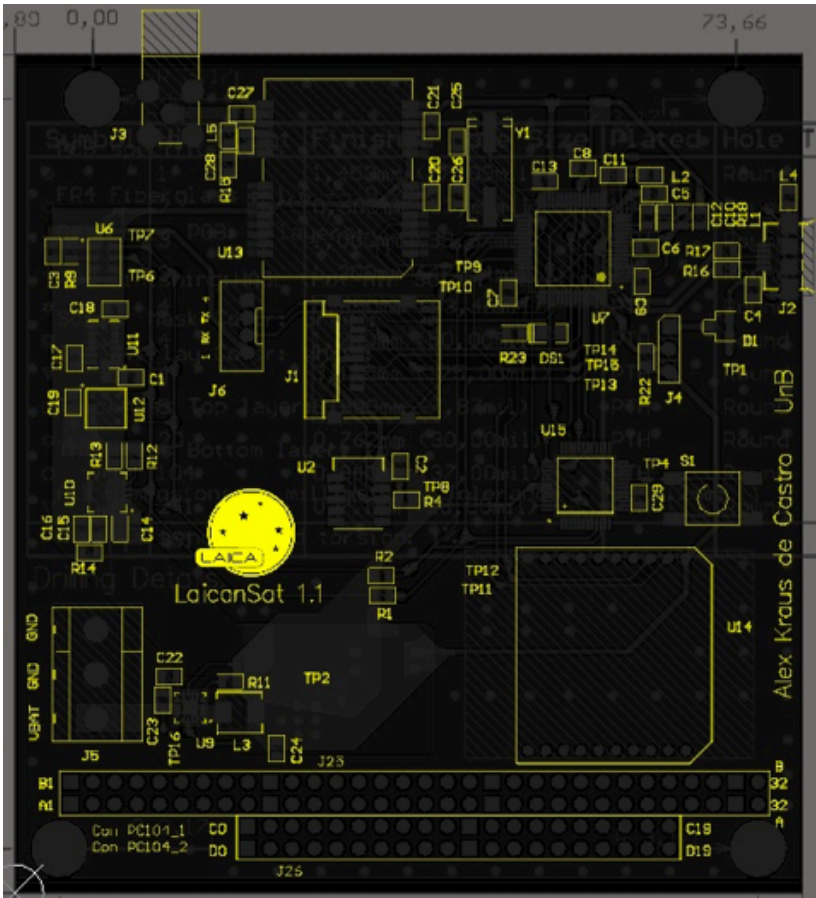


Figura 48 - Topoverlay

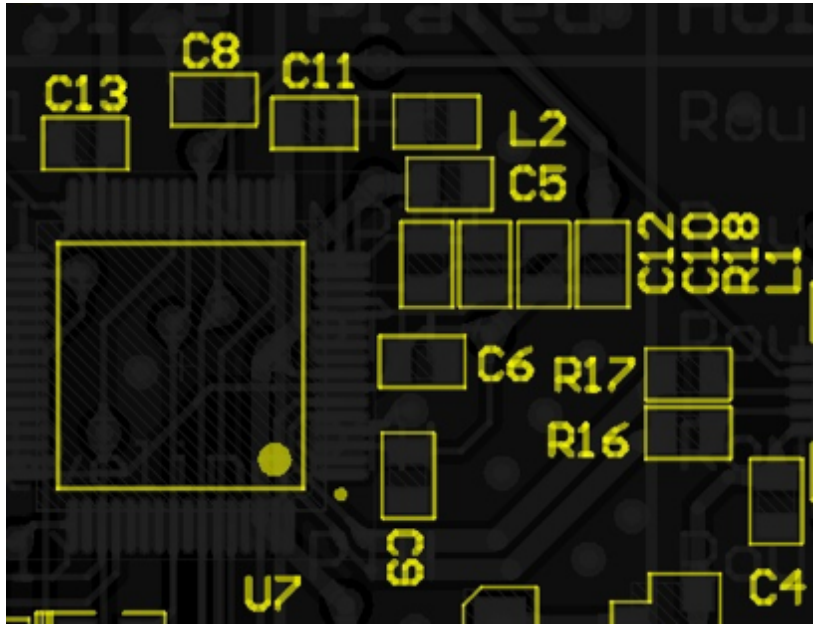


Figura 49 - Topoverlay MCU

### 4.2.5 FABINFO

É na camada Fabinfo aonde serão colocada informações necessárias para a produção da placa. Veja Figura 50.

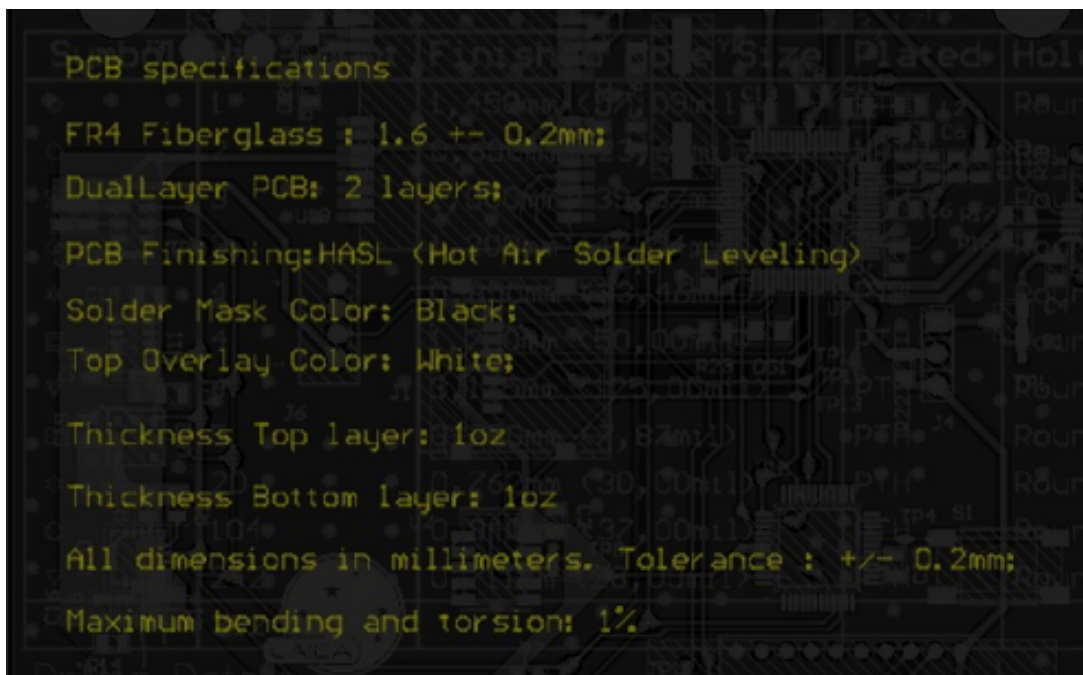


Figura 50 – Fabinfo

Nesta camada especificamos qual o tipo de dielétrico utilizado entre as camadas Toplayer e Bottomlayer e sua espessura. Informamos que será utilizado apenas 2 camadas e como devesse ser o processo de fabricação. No processo foi pedido o HASL por ser o mais barato. Indicamos a espessura do cobre da Toplayer e Bottomlayer em 1 oz. Nesta parte também informa-se as cor da placa junto com a cor das letras nela utilizadas na Topoverlay. Foi requerido uma placa preta com a cor das letras de branco.

### 4.3 TESTES E ANÁLISE DOS RESULTADOS

Após todo o desenvolvimento do projeto, foi mandando o arquivo Gerber para a empresa GRIFFUS para sua fabricação. O arquivo Gerber contém todas as informações importantes para a fabricação da peça, como dimensões, trilhas, furos, vias, quantas camadas entre outras informações. A Figura 51 apresenta a imagem da placa confeccionada.

Utilizando os equipamentos necessários para a soldagem, a primeira soldagem a ser feita é a da fonte de tensão. É importante soldar primeiro a fonte de tensão para averiguar que está funcionando pois, uma fonte mal soldada ou mal projetada pode gerar uma tensão oscilante indesejada que pode queimar os componentes da placa inteira. Após a soldagem dos componentes testou-se a fonte. Primeiramente as pontas de prova de um multímetro são

conectadas no terra e a outra ponta no Test Point 2. Test Point 2 foi feito justamente para medir a tensão 3,3 volts. Liga-se uma fonte DC no circuito em 5 volts. Varia a fonte de 5 volts ate 15 volts. A tensão no Test Point 2 permaneceu estável em 3,3 volts provando que a fonte funciona da forma esperada.

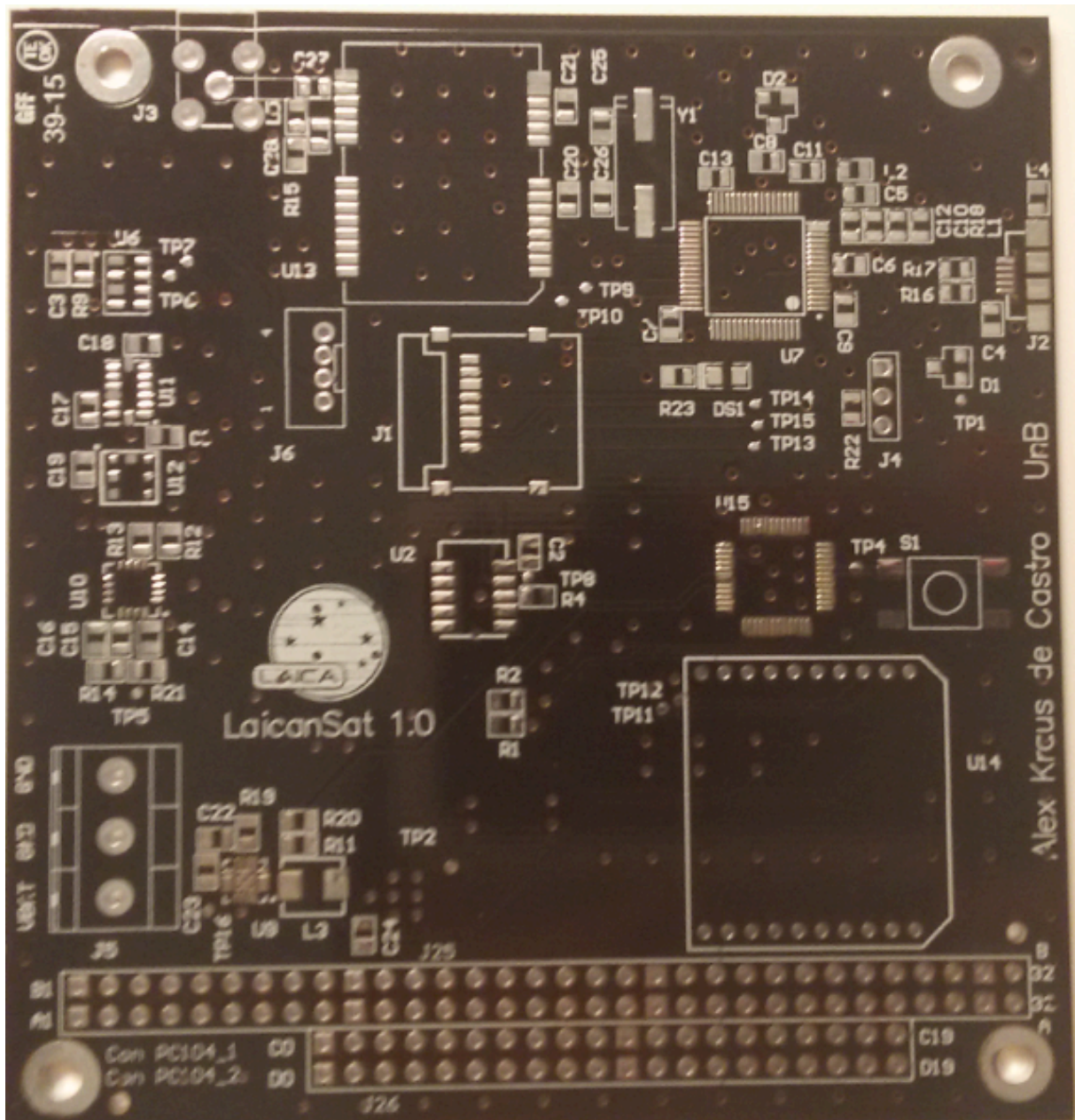


Figura 51 - Placa LAICAnSat Produzida pela GRIFFUS

O setor do microcontrolador foi soldado logo após. Foi feito um teste utilizando um código para testar o LED da placa. O LED piscou da forma correta informando que todo o setor do MCU estava funcionando. Depois foram soldados todos os outros componentes do projeto.

É necessário uma atenção na hora da soldagem de dois dispositivos da placa. O mais crítico é o módulo GNSS. Este dispositivo só pode ser soldado por um soprador de ar quente se este for esquentar a placa por baixo do dispositivo. Nunca deve-se jogar ar quente em cima do módulo. Ou deve-se soldar com um ferro de solda perna por perna evitando qualquer

curto. O sensor SHT15 é o outro componente crítico durante o processo de solda. Após soldar o sensor, deve-se esperar 12 horas para testar o componente. O datasheet informa que caso essa regra não seja atendida, o sensor pode perder a sua calibração.

Após a solda de todos os componentes deve-se testar a conectividade de todas as entradas da placa antes de ligá-la. Deve-se principalmente fazer o teste de conectividade entre o plano de 3,3 volts com o terra. Esses dois planos nunca podem estar em curto.

Concluído todo esse processo foi ligada a placa e alguns códigos de teste foram utilizados para testar o funcionamento de todos os componentes.

O primeiro teste feito foi o do canal I2C dos sensores. Foi feito um programa onde o microcontrolador pergunta aos dispositivos do barramento o seu endereço. Todos os componentes responderam os seus respectivos endereços de acordo com a Figura 52.

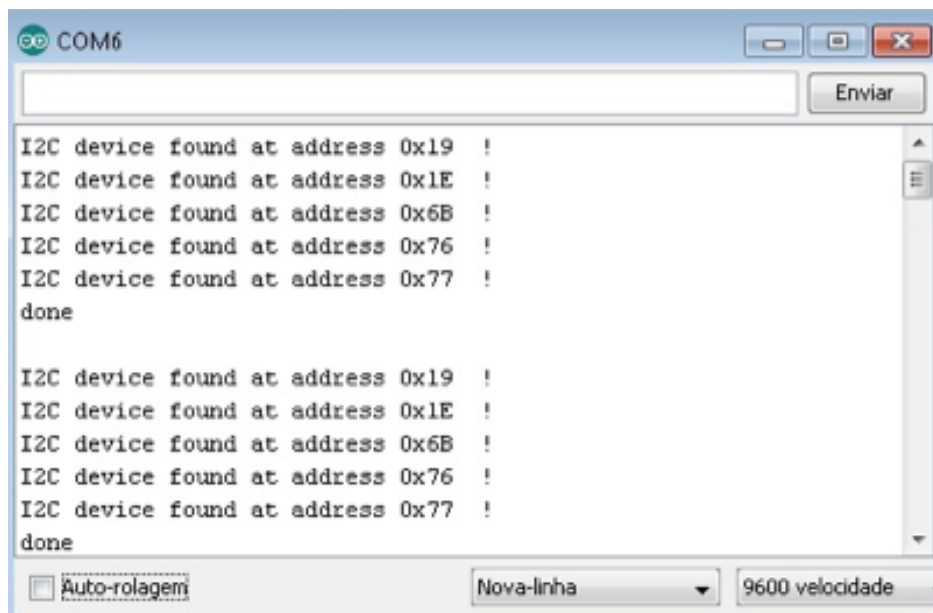


Figura 52 - Teste barramento I2C

O segundo teste foi utilizar um código para verificar o funcionamento de um cartão micro SD. O código simplesmente checa a existência do dispositivo, caso ele exista verifica se existe um arquivo com o nome de teste.txt. Caso o arquivo de teste não exista ele gera um arquivo com o nome de teste.txt. O código grava o arquivo no cartão micro SD. Novamente ele verifica se arquivo existe. Como o arquivo existe ele apaga esse arquivo e checa novamente se ele existe.

Esse teste testa quase todas as funcionalidades do cartão micro SD, como a checagem do arquivo, criação e remoção. Veja Figura 53.

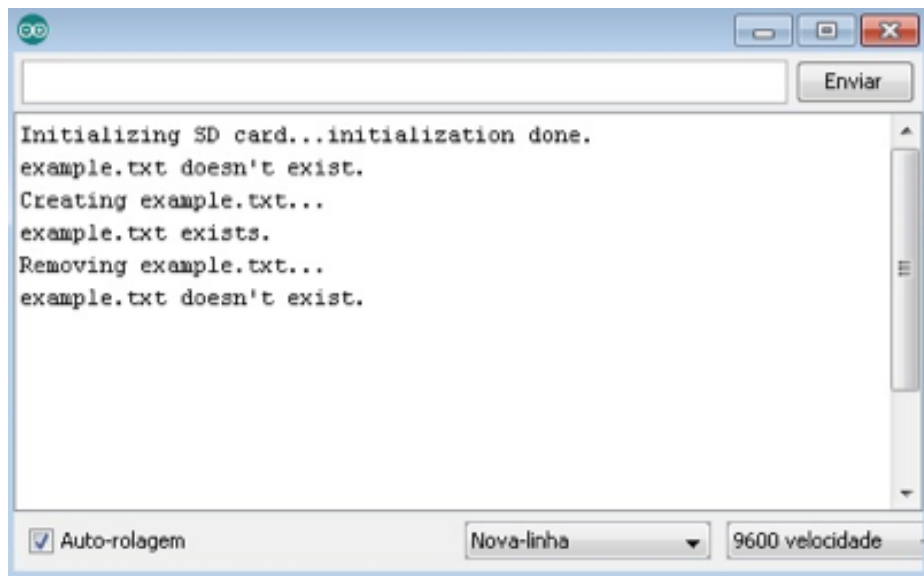


Figura 53 - Teste Cartão Micro SD

O terceiro teste foi criar um código para receber os dados enviados pelo módulo GNSS. Este código consiste apenas em mandar todos os caracteres recebidos pelo GNSS para o computador. Com este teste pode-se perceber que o GNSS estava funcionando e que ele estava recebendo informações da antena. Quando a antena não está funcionando recebe-se as coordenadas 99.99,99.99,99.99\*2E. Veja a Figura 54.

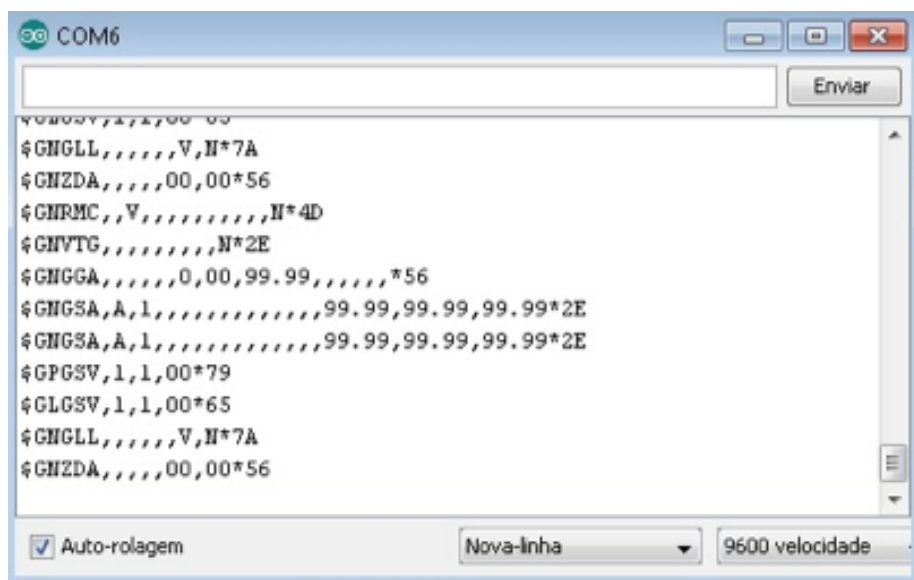
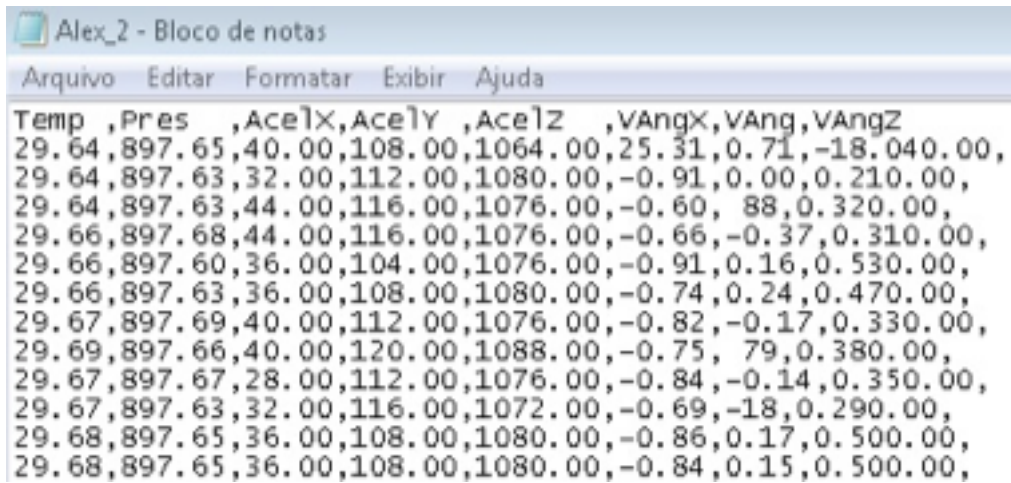


Figura 54 - Teste GPS

O quarto teste foi produzir um código que testa os sensores da IMU. O microcontrolador recebe os dados recebidos pelos sensores e armazena no cartão de memória. Na Figura 55 encontra-se os dados coletados pelo microcontrolador e armazenados no cartão de

memória no formato txt. No arquivo encontra-se a temperatura, pressão, aceleração nos eixos x,y,z e as velocidades nas direções x,y e z.



```
Alex_2 - Bloco de notas
Arquivo  Editar  Formatar  Exibir  Ajuda
Temp ,Pres ,AceX,AceY ,AceZ ,VAngX,VAng,VAngZ
29.64,897.65,40.00,108.00,1064.00,25.31,0.71,-18.040.00,
29.64,897.63,32.00,112.00,1080.00,-0.91,0.00,0.210.00,
29.64,897.63,44.00,116.00,1076.00,-0.60, 88,0.320.00,
29.66,897.68,44.00,116.00,1076.00,-0.66,-0.37,0.310.00,
29.66,897.60,36.00,104.00,1076.00,-0.91,0.16,0.530.00,
29.66,897.63,36.00,108.00,1080.00,-0.74,0.24,0.470.00,
29.67,897.69,40.00,112.00,1076.00,-0.82,-0.17,0.330.00,
29.69,897.66,40.00,120.00,1088.00,-0.75, 79,0.380.00,
29.67,897.67,28.00,112.00,1076.00,-0.84,-0.14,0.350.00,
29.67,897.63,32.00,116.00,1072.00,-0.69,-18,0.290.00,
29.68,897.65,36.00,108.00,1080.00,-0.86,0.17,0.500.00,
29.68,897.65,36.00,108.00,1080.00,-0.84,0.15,0.500.00,
```

Figura 55 – Teste dos sensores armazenando no cartão de memória

## 5. CONCLUSÃO

Este projeto tinha como finalidade desenvolver uma placa eletrônica para o sistema mínimo da plataforma LAICAnSat. Inicialmente foi apresentado um breve histórico do projeto, resumindo os principais acontecimentos nos primeiros experimentos realizados pela plataforma LAICAnSat. Tendo em vista uma melhoria no desenvolvimento do sistema embarcado, buscou-se uma padronização do sistema embarcado. Os padrões adotados para a placa eletrônica foi o PC104 que se encaixa em estruturas do tipo CubeSat.

Realizou-se um estudo sobre a finalidade dos componentes do protótipo realizado nos experimentos. Feita a análise sobre os componentes, criou-se um esquemático mais eficiente e por fim foi desenvolvido um layout mais robusto da placa eletrônica. Após a fabricação da placa e a soldagens dos elementos, utilizou-se o *firmware* de testes baseados no utilizado no experimento passado e foi obtido uma resposta satisfatória.

Uma dificuldade encontrada no projeto foi a soldagem dos elementos e o seus testes elétricos. A soldagem dos componentes na placa apresentaram uma certa dificuldade de execução em virtude das ferramentas e habilidades necessárias. A soldagem só se tornou viável pois foi utilizado todo o equipamento destinado a solda da empresa AUTOTRAC. Deve-se dar atenção ao soldar a placa, pois na primeira soldagem da placa o MINI54TAN parou de funcionar por motivos não detectados. Por isso é importante a soldagem dos componentes serem testados eletricamente antes se serem ligados.

Por fim, foi elaborado os detalhes executados nesse trabalho para que no futuro outros pesquisadores possam desenvolver e implementar novos projetos.

Tendo em vista o que foi exposto neste manuscrito pode-se concluir que o projeto atingiu os objetivos, tais como: reconfiguração dos elementos da placa em um esquemático; desenho do layout da placa eletrônica com a implementação do padrão PC104 e Cubesat; realização de testes funcionais.

Desta forma, conclui-se que os objetivos foram alcançados. A Figura 56 apresenta a imagem final da placa eletrônica LAICAnSat.



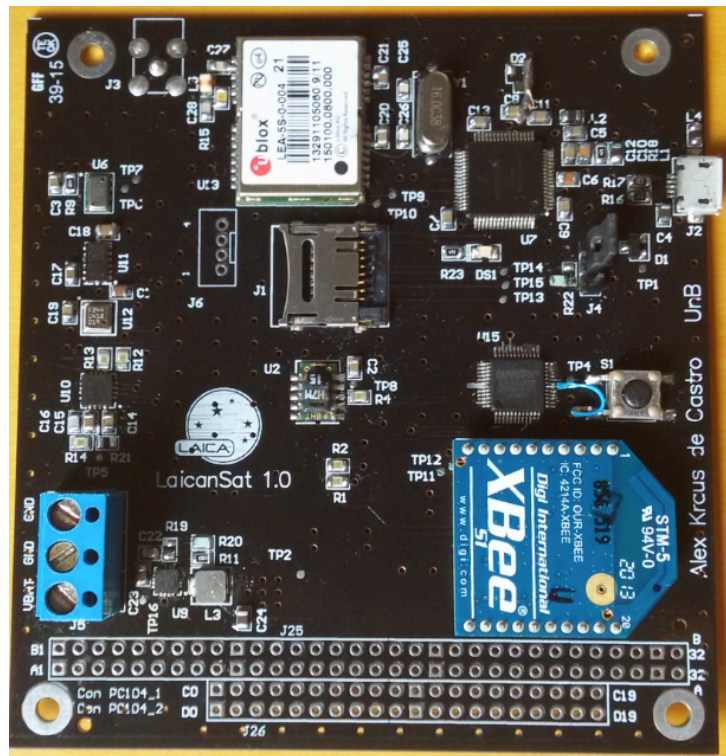


Figura 56 - Placa eletrônica LaicanSat

## Sugestões para trabalhos futuros

Uma forma em que o projeto poderia ser implementado seria colocar na entrada da bateria uma proteção contra curto circuito e inversão de polaridade. Isso tornaria a placa mais segura e protegeria os componentes caso algum erro seja cometido. Outra forma seria alterar a placa na intuito de implementar alguns motores para o controle do paraquedas.

## Referência Bibliográfica

- [1] YAKIMENKO, O. A.; SLEGGERS, N. J.; TIADEN, R. Development and testing of the miniature aerial delivery system snowflake. Proceedings of the 20th AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar. AIAA, May 2009.
- [2] BENTON, J. E.; YAKIMENKO, O. A. On development of autonomous haho parafoil system for targeted payload return. Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, v. 225, n. 3, p. 336-346, 2011.
- [3] N. Yajima, T. Imamura e N. Izutsu, Scientific Ballooning: Technology and Applications of Exploration Balloons Floating in the Stratosphere and the Atmosphere of Other Planets, Springer, 2009.
- [4] M. S. Smith e G. Allison, "The Return of the Balloon as an Aerospace Test Platform," em AIAA International Balloon Technology Conference, Norfolk, 1999.
- [5] NEHME, P. H. D. et al. Development of a meteorology and remote sensing experimental platform: The laicansat-1. Aerospace Conference, 2014 IEEE, p. 1-7, 2014.
- [6] THE AMERICAN INSTITUTE OF AERONAUTICS AND ASTRONAUTICS (AIAA). Junho 2013. Disponível em: <<https://www.aiaa.org/SecondaryTwoColumn.aspx?id=18574>>.
- [7] Cansats in Europe, "Requirements," [Online]. Disponível em: [http://esamultimedia.esa.int/docs/edu/2016\\_European\\_CanSat\\_Competition\\_Guidelines.pdf](http://esamultimedia.esa.int/docs/edu/2016_European_CanSat_Competition_Guidelines.pdf) [Acesso em 04 Dezembro 2015].
- [8] ANTARIKSHA LABS PVT. LTD. Disponível em: <<http://www.antarikshalabs.in/cansatleadership-training-program>>.
- [9] W. V. Jones, Evolution of scientific ballooning and its impact on astrophysics research. In: Advances in Space Research, Elsevier, 2014, pp. 1405 - 1414.
- [10] Cubesat, [Online]. Disponível em: <http://www.gaussteam.com/services/satellite-subsystem/onboard-computer/> [Acesso em 04 Dezembro 2015].
- [11] Cubesat, [Online]. Disponível em: <https://pt.wikipedia.org/wiki/CubeSat> [Acesso em 04 Dezembro 2015].
- [12] PC104, [Online]. Disponível em: <https://en.wikipedia.org/wiki/PC/104> [Acesso em 04 Dezembro 2015].
- [13] Heidt, H.; A new Generation of Picosatellite for Education and Industry Low-Cost Space Experimentation. 14 Annual/USU Conference on Small Satellites

- [14] The CubeSat: The picosatellite Standard for Research and Education em AIAA SPACE 2008 Conference and Exposition September,2008, San Diego, California.
- [15] PC104, Consortium, History [Online]. Disponível em: <http://pc104.org/consortium/history/> [Acesso em 05 Dezembro 2015].
- [16] Zimmermann, A. L.; PC/104 : Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos.
- [17] Projeto do Nanossatélite Serpens, [Online]. Disponível em: <http://www.aeb.gov.br/projeto-do-nanossatelites-serpens-e-aprovado-em-revisao-final> [Acesso em 06 dezembro de 2015]
- [18] Teensy, [Online]. Disponível em: <https://www.pjrc.com/teensy/> [Acesso em 07 Dezembro 2015].
- [19] IMU breakout, [Online]. Disponível em: <http://www.adafruit.com/product/1604/> [Acesso em 07 Dezembro 2015].
- [20] MS5611, [Online]. Disponível em: <http://www.dx.com/> [Acesso em 07 Dezembro 2015].
- [21] SHT15, [Online]. Disponível em: <https://www.sparkfun.com/products/13683/> [Acesso em 07 Dezembro 2015].
- [22] DS18B20, [Online]. Disponível em: <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf> [Acesso em 07 Dezembro 2015].
- [23] INA219, [Online]. Disponível em: <https://www.adafruit.com/products/904> [Acesso em 07 Dezembro 2015].
- [24] MIC2940, [Online]. Disponível em: [http://www.micrel.com/\\_PDF/mic2940.pdf](http://www.micrel.com/_PDF/mic2940.pdf) [Acesso em 07 Dezembro 2015].
- [25] TSL2561, [Online]. Disponível em: <https://www.adafruit.com/products/439> [Acesso em 07 Dezembro 2015].
- [26] UVM-30A, [Online]. Disponível em: <http://www.dx.com/pt/p/uvm-30a-uv-sensor-detector-module-black-165105#.VmXYhMaRKT8> [Acesso em 07 Dezembro 2015].
- [27] TPS62142, [Online]. Disponível em: <http://www.ti.com/product/tps62142> [Acesso em 07 Dezembro 2015].
- [28] Ublox LEA-6, [Online]. Disponível em: <https://www.u-blox.com/en/product/lea-6-series> [Acesso em 07 Dezembro 2015].

- [29] Ublox M8T, [Online]. Disponível em: <https://www.u-blox.com/en/product/neolea-m8t>[Acesso em 07 Dezembro 2015].
- [30] UART, Serial communication [Online]. Disponível em: <https://learn.sparkfun.com/tutorials/serial-communication> [Acesso em 07 Dezembro 2015].
- [31] I2C, Protocol [Online]. Disponível em: <https://learn.sparkfun.com/tutorials/i2c> [Acesso em 07 Dezembro 2015].
- [32] SPI, Serial Peripheral Interfacel [Online]. Disponível em: <https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi> [Acesso em 07 Dezembro 2015].
- [33] 2-Wire-Bus, [Online]. Disponível em: <https://www.sparkfun.com/products/13683> [Acesso em 07 Dezembro 2015].
- [34] Altium [Online]. <http://www.altium.com> [Acesso em 07 Dezembro 2015].
- [35] Altium,schematics and PCB [Online].  
<http://techdocs.altium.com/display/ADOH/Tutorial+-+Getting+Started+with+PCB+Design> [Acesso em 07 Dezembro 2015].
- [36] Arduino [Online]. <https://www.arduino.cc> [Acesso em 07 Dezembro 2015].
- [37] WEBENCH Design [Online].  
[http://www.ti.com/lscds/ti/analog/webench/overview.page?DCMP=sva\\_web\\_webdesigncntr\\_en&HQS=sva-web-webdesigncntr-vanity-lp-en](http://www.ti.com/lscds/ti/analog/webench/overview.page?DCMP=sva_web_webdesigncntr_en&HQS=sva-web-webdesigncntr-vanity-lp-en) [Acesso em 07 Dezembro 2015].
- [38] SaturnPCB [Online]. <http://saturnpcb.com/#page=page-1> [Acesso em 07 Dezembro 2015].
- [39] Xbee[Online]. <http://www.digi.com/lp/xbee> [Acesso em 07 Dezembro 2015].
- [40] L3GD20H[Online].  
[http://www.st.com/web/catalog/sense\\_power/FM89/SC1288/PF254039](http://www.st.com/web/catalog/sense_power/FM89/SC1288/PF254039) [Acesso em 07 Dezembro 2015].
- [41] LSM303DLHC [Online].  
<http://www.st.com/web/en/resource/technical/document/datasheet/DM00027543.pdf> [Acesso em 07 Dezembro 2015].
- [42] BMP180 [Online]. <https://www.adafruit.com/datasheets/BST-BMP180-DS000-09.pdf> [Acesso em 07 Dezembro 2015].

[43] TPS62142 [Online]. <http://www.ti.com/product/tps62142> [Acesso em 07 Dezembro 2015].

# ANEXO1 – Lista de Materiais

## Bill of Materials

Placa Eletronica LAICAnSat

Source Data From:

Alex Kraus de Castro

Project:

LAICAnSat

Variant:

None

Creation Date:

09/12/2015

03:45:07

Print Date:

42347

42347,06963

Footprint	Comment	Designator	Description	Quantity
DISC-0805	10uF	C1, C23	Capacitor	2
DISC-0805	100nF	C2, C3, C20, C21	Capacitor	4
DISC-0805	22uF	C4, C24	Capacitor	2
DISC-0805	0.1uF	C5, C7, C8, C9, C11, C12, C13, C14, C19, C29	Capacitor	10
DISC-0805	2.2uF	C6, C10	Capacitor	2
DISC-0805	470nF	C15	Capacitor	1
DISC-0805	10nF	C16	Capacitor	1
DISC-0805	0.22uF	C17	Capacitor	1
DISC-0805	4.7uF	C18	Capacitor	1
DISC-0805	390pF	C22	Capacitor	1
DISC-0805	22pF	C25, C26	Capacitor	2
DISC-0805	47pF	C27, C28	Capacitor	2
DISC-SOT-23	Diode	D1	Diode Dual Commom Cathode	1
DISC-LED-0805	LED	DS1	Led	1
CON-MICROSD-HINGE-9P	MICRO-SD	J1	MICRO-SD	1
CON-USB-MICRO-AB	USB MICRO/MINI	J2	CON USB MICRO/MINI	1
CON-RP-SMA-PTH-RA	CONNECTOR	J3	COAXIAL CONECTOR	1
CON-HEADER-1X3-PTH	Jumper	J4	Jumper Double	1
CON-BORNE-3	Conector Bateria	J5	HEADER 1X3	1
CON-89400-4P-PTH	Uart Extern	J6	HEADER 1X4	1
Footprint_ConectorPC104_1	Con PC104_1	J25		1
Footprint_ConectorPC104_2	Con PC104_2	J26		1
DISC-0805	Ferrite	L1, L2, L4	Inductor	3
Footprint_Indutor-IHLP	Indutor-IHLP	L3		1
DISC-0805	27nH	L5	Inductor	1

DISC-0805	10k	R1, R2, R4, R12, R13, R14	Resistor	6
DISC-0805	4k7	R9	Resistor	1
DISC-0805	100k	R11	Resistor	1
DISC-0805	10	R15	Resistor	1
DISC-0805	33	R16, R17	Resistor	2
DISC-0805	470	R18, R23	Resistor	2
DISC-0805	0	R22	Resistor	1
DISC-PUSHBUTTON-4P-SMD	SWITCH	S1	Push Button NA	1
MISC-TP	TESTPOINT	TP1, TP2, TP4, TP6, TP7, TP8, TP9, TP10, TP11, TP12, TP13, TP14, TP15, TP16	Test Point	14
footprint_SHT15	SHT15	U2	Humidy and temperature sensor	1
footprint_MS5611	MS5611	U6	Altimeter Sensor	1
footprint_MK20DX128VLHS	MK20DX256VHL7	U7	Microcontroller	1
footprint_TPS62142	TPS62142	U9		1
LGA16-4X4X1.1_N	STM-L3GD20-LGA-16	U10		1
LGA14-3X5X1_N	LSM303DLHC	U11		1
FOOTPRINT_BMP180	BMP180	U12		1
MOD-UBLOXLEASMD28	LEA-M8T	U13	UBLOX LEA-6A/LEA-6S	1
footprint_Xbee	Xbee	U14		1
footprint_Mini54Tan	MINI54TAN	U15	Microcontroller	1
DISC-CRYSTAL-HC49-US	16MHz	Y1	Crystal	1

## Anexo 2 -Códigos de Testes

### 1-Scan

```
#include <Wire.h>
void setup() {
  Wire.begin();
  Serial.begin(9600);
  Serial.println("\nI2C Scanner");
}

void loop() {
  byte error, address;
  int nDevices;
  nDevices = 0;
  for(address = 1; address < 127; address++ )
  {

    Wire.beginTransmission(address);
    error = Wire.endTransmission();
    if (error == 0)
    {
      Serial.print("I2C device found at address 0x");
      if (address<16)
        Serial.print("0");
      Serial.print(address,HEX);
      Serial.println(" !");
      nDevices++;
    }
    else if (error==4)
    {
      Serial.print("Unknow error at address 0x");
      if (address<16)
        Serial.print("0");
      Serial.println(address,HEX);
    }
  }
  if (nDevices == 0)
    Serial.println("No I2C devices found\n");
  else
    Serial.println("done\n");
  delay(1000);
}
```

### 2-TESTE GPS

```
char a;

void setup() {
  Serial.begin(9600);
  Serial1.begin(9600);
}
void loop() {
  if(Serial1.available(>0)){
    a=Serial1.read();
    Serial.print(a);
  }
}
```



### 3- TESTE SDCARD

```
#include <SPI.h>
#include <SD.h>
File myFile;
void setup()
{
  const int slaveSelectPin = 20;
  pinMode (slaveSelectPin, OUTPUT);
  SPI.setMOSI(7); SPI.setMISO(8);SPI.setSCK(14);Serial.begin(9600);
  while (!Serial) {
    ;
  }
  Serial.print("Initializing SD card...");
  if (!SD.begin(20)) {
    Serial.println("initialization failed!");
    return;
  }
  Serial.println("initialization done.");
  if (SD.exists("example.txt")) {
    Serial.println("example.txt exists.");
  }
  else {
    Serial.println("example.txt doesn't exist.");
  }
  Serial.println("Creating example.txt...");
  myFile = SD.open("example.txt", FILE_WRITE);|
  myFile.close();
  if (SD.exists("example.txt")) {
    Serial.println("example.txt exists.");
  }
  else {
    Serial.println("example.txt doesn't exist.");
  }
  Serial.println("Removing example.txt...");
  SD.remove("example.txt");
  if (SD.exists("example.txt")) {
    Serial.println("example.txt exists.");
  }
  else {
    Serial.println("example.txt doesn't exist.");}}}
```

## **Anexo 3 –Esquemático**