



TRABALHO DE CONCLUSÃO DE CURSO

**DESENVOLVIMENTO DE FIRMWARE EMBARCADO E
CARACTERIZAÇÃO DO MÓDULO DE GERENCIAMENTO
DE ENERGIA DE NANOSATÉLITES**

Thiago Caldas dos Santos

Brasília, Fevereiro de 2023

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia
Departamento de Engenharia Elétrica

TRABALHO DE CONCLUSÃO DE CURSO

**DESENVOLVIMENTO DE FIRMWARE EMBARCADO E
CARACTERIZAÇÃO DO MÓDULO DE GERENCIAMENTO
DE ENERGIA DE NANOSSATÉLITES**

Thiago Caldas dos Santos

*Trabalho de conclusão de curso submetido ao Departamento de Engenharia
Elétrica como requisito parcial para obtenção
do grau de Engenheiro em Engenharia Elétrica*

Banca Examinadora

Prof. Daniel Chaves Café, Ph.D, FT/UnB
Orientador

Prof. Eduardo Peixoto Fernandes da Silva, Ph.D,
FT/UnB
Examinador Interno

Prof. Ricardo Zelenovsky, Ph.D, FT/UnB
Examinador interno

FICHA CATALOGRÁFICA

SANTOS, THIAGO CALDAS DOS

DESENVOLVIMENTO DE FIRMWARE EMBARCADO E CARACTERIZAÇÃO DO MÓDULO DE GERENCIAMENTO DE ENERGIA DE NANOSSATÉLITES [Distrito Federal] 2023.

xvi, 57 p., 210 x 297 mm (ENE/FT/UnB, Engenheiro, Engenharia Elétrica, 2023).

Trabalho de conclusão de curso - Universidade de Brasília, Faculdade de Tecnologia.

Departamento de Engenharia Elétrica

- | | |
|------------------|-------------|
| 1. EPS | 2. Firmware |
| 3. Nanossatélite | 4. LODESTAR |
| I. ENE/FT/UnB | |

REFERÊNCIA BIBLIOGRÁFICA

SANTOS, T.C. (2023). *DESENVOLVIMENTO DE FIRMWARE EMBARCADO E CARACTERIZAÇÃO DO MÓDULO DE GERENCIAMENTO DE ENERGIA DE NANOSSATÉLITES*. Trabalho de conclusão de curso, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 57 p.

CESSÃO DE DIREITOS

AUTOR: Thiago Caldas dos Santos

TÍTULO: DESENVOLVIMENTO DE FIRMWARE EMBARCADO E CARACTERIZAÇÃO DO MÓDULO DE GERENCIAMENTO DE ENERGIA DE NANOSSATÉLITES .

GRAU: Engenheiro em Engenharia Elétrica ANO: 2023

É concedida à Universidade de Brasília permissão para reproduzir cópias deste Projeto Final 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 Projeto Final de Graduação pode ser reproduzida sem autorização por escrito do autor.

Thiago Caldas dos Santos

Depto. de Engenharia Elétrica (ENE) - FT

Universidade de Brasília (UnB)

Campus Darcy Ribeiro

CEP 70919-970 - Brasília - DF - Brasil

Dedicatória

Dedico este trabalho a Deus e a todos os meus familiares e amigos que sempre estiveram comigo ao longo de toda essa caminhada.

Thiago Caldas dos Santos

Agradecimentos

Agradeço, primeiramente, a Deus, pois toda a minha graduação, desde o início, assim como este projeto, só foram possíveis pelas graças que Ele me concedeu e pela intercessão de Nossa Senhora. Além disso, gostaria de agradecer também à minha família, em especial, a meus pais, Hilton e Dulce, e a minha irmã, Laura, que me acompanharam desde o início, antes mesmo que eu iniciasse a minha graduação e que, após essa conquista, me deram todo o suporte necessário para que eu pudesse concluir com êxito minha formação em Engenharia Elétrica, me incentivando a seguir em frente e buscar grandes desafios.

Sou grato também aos amigos que fiz ao longo desses anos de curso, Luquinhas, Vini, Li, Anne, Railson, Thiago, Gustavo, Julia, Dani e Helena, que foram muito importantes em diversos momentos e que me possibilitaram chegar tão longe, compartilhando comigo experiências que vou levar para toda a minha vida. Além do mais, agradeço à ENETEC e a todos que compartilharam comigo a minha jornada dentro da empresa júnior, onde tive a oportunidade de aprender e viver experiências incríveis.

Por fim, agradeço a todos os funcionários da Universidade de Brasília, em especial, ao professor Daniel Chaves Café, que foram fundamentais para o meu desenvolvimento profissional e pessoal.

Thiago Caldas dos Santos

RESUMO

O presente trabalho visa dar continuidade a projetos anteriores do Laboratório de Simulação e Controle de Sistemas Aeroespaciais (LODESTAR) da Universidade de Brasília (UnB), cujo intuito é estudar e desenvolver o módulo de gerenciamento de energia (EPS) de um nanossatélite. Nos trabalhos anteriores, foram propostas arquiteturas de *hardware* para o EPS, enquanto este trabalho tem como objetivo desenvolver o *firmware* para este subsistema, além de caracterizá-lo. Para isso, são apresentados modelos de painéis solares e baterias e tipos de conversores CC-CC para caracterização do EPS, além da modelagem e resultados obtidos. Ademais, são descritos os testes realizados para a validação do *firmware* desenvolvido, nos quais foram obtidos erros relativos inferiores a 5% para medições de corrente, tensão e temperatura, e também é apresentado o *hardware* adotado para implementação dos cenários de teste, composto por circuitos integrados, como o INA3221 e o CN3791. Por fim, o trabalho é concluído com o *firmware* do EPS desenvolvido para o microcontrolador MSP430FR5994, utilizando-se a linguagem C, e um programa de teste desenvolvido para o Arduino Uno, utilizado como microcontrolador auxiliar.

ABSTRACT

This work aims to continue previous projects of the Simulation and Control of Aerospace Systems Laboratory (LODESTAR) from the University of Brasilia (UnB), whose purpose is to study and develop the Electrical Power System (EPS) of a nanosatellite. In previous works, hardware architectures were proposed for the EPS, while this work looks for firmware development for this subsystem and the system characterization. For this, models of solar panels and batteries, types of DC-DC converters and simulation results using the models to characterize the EPS are presented. Furthermore, the tests carried out to validate the developed firmware are described, in which relative errors of less than 5% were obtained for current, voltage and temperature measurements, and also are presented the hardware adopted for the test scenarios implementation, composed of integrated circuits, like the INA3221 and the CN3791 . Finally, the work is concluded with the EPS's firmware developed for the microcontroller MSP430FR5994, in C language, and a test setup developed for the Arduino Uno, used as an auxiliary microcontroller.

SUMÁRIO

| | | |
|----------|---|----------|
| 1 | INTRODUÇÃO | 1 |
| 1.1 | CONTEXTUALIZAÇÃO | 1 |
| 1.2 | CUBESATS | 1 |
| 1.3 | PROJETO ALFACRUX | 4 |
| 1.4 | <i>Firmware</i> EMBARCADO | 4 |
| 1.4.1 | INTERFACE DE CONFIGURAÇÃO | 5 |
| 1.4.2 | DADOS DE TELEMETRIA | 5 |
| 1.4.3 | FORNECIMENTO DE ENERGIA AOS SUBSISTEMAS | 6 |
| 1.4.4 | CONTROLE DE TEMPERATURA DA BATERIA | 6 |
| 1.4.5 | COMUNICAÇÃO COM O COMPUTADOR DE BORDO | 7 |
| 1.4.6 | ARQUITETURA PROPOSTA | 8 |
| 2 | FUNDAMENTAÇÃO TEÓRICA | 9 |
| 2.1 | MODELOS DE PLACA SOLAR | 9 |
| 2.1.1 | MODELO IDEAL | 9 |
| 2.1.2 | MODELO COM INSERÇÃO DE UMA RESISTÊNCIA EM SÉRIE | 10 |
| 2.1.3 | MODELO COM UM DIODO E DUAS RESISTÊNCIAS | 11 |
| 2.1.4 | MODELO COM DOIS DIODOS E DUAS RESISTÊNCIAS | 11 |
| 2.2 | CONVERSORES CC-CC | 12 |
| 2.2.1 | CONVERSOR BUCK | 14 |
| 2.2.2 | CONVERSOR BOOST | 15 |
| 2.2.3 | CONVERSOR BUCK-BOOST | 16 |
| 2.2.4 | CONVERSOR SEPIC | 17 |
| 2.3 | MODELOS DE BATERIA | 18 |
| 2.3.1 | MODELO COM RESISTÊNCIA INTERNA | 19 |
| 2.3.2 | MODELO COM RESISTÊNCIA INTERNA E UM RAMO RC PARALELO | 19 |
| 2.4 | <i>Maximum Power Point Tracking (MPPT)</i> | 20 |
| 2.4.1 | PERTURBA E OBSERVA (P&O) | 21 |
| 2.4.2 | CONDUTÂNCIA INCREMENTAL (CI) | 23 |
| 2.5 | MICROCONTROLADOR | 25 |
| 2.6 | COMUNICAÇÃO SERIAL | 26 |
| 2.6.1 | <i>Universal Asynchronous Reception and Transmission (UART)</i> | 27 |
| 2.6.2 | <i>Inter-Integrated Circuit (I2C)</i> | 28 |
| 2.6.3 | <i>Serial Peripheral Interface (SPI)</i> | 28 |
| 2.7 | SENSOR DE CORRENTE - INA3221 | 29 |
| 2.7.1 | CONFIGURAÇÃO DO ENDEREÇO | 30 |
| 2.7.2 | COMUNICAÇÃO COM O DISPOSITIVO | 30 |

| | | |
|----------|---|-----------|
| 2.7.3 | INTERPRETAÇÃO DAS LEITURAS | 32 |
| 2.7.4 | FILTRO DE MÉDIA | 32 |
| 2.7.5 | MÓDULOS COM INA3221 | 33 |
| 3 | METODOLOGIA | 35 |
| 3.1 | SIMULAÇÃO DO CIRCUITO ANALISADO | 35 |
| 3.2 | TESTE DO <i>firmware</i> | 36 |
| 3.2.1 | TESTE DAS FUNCIONALIDADES | 36 |
| 3.3 | TESTE DO CIRCUITO COMPLETO | 37 |
| 4 | RESULTADOS | 39 |
| 4.1 | CARACTERIZAÇÃO DO CIRCUITO | 39 |
| 4.1.1 | PAINEL SOLAR | 39 |
| 4.1.2 | CONVERSOR CC-CC | 40 |
| 4.1.3 | CIRCUITO COMPLETO | 42 |
| 4.2 | ESCOLHA DO MICROCONTROLADOR | 43 |
| 4.3 | DESENVOLVIMENTO DO <i>firmware</i> | 45 |
| 4.3.1 | INTERFACE DE CONFIGURAÇÃO | 45 |
| 4.3.2 | DADOS DE TELEMETRIA | 47 |
| 4.3.3 | COMUNICAÇÃO COM O COMPUTADOR DE BORDO | 48 |
| 4.3.4 | CONTROLE DE TEMPERATURA DA BATERIA | 49 |
| 4.3.5 | FORNECIMENTO DE ENERGIA AOS SUBSISTEMAS | 50 |
| 4.4 | TESTE DO CIRCUITO COMPLETO | 50 |
| 5 | CONCLUSÃO | 54 |
| 5.1 | SUGESTÕES PARA TRABALHOS FUTUROS | 54 |
| | REFERÊNCIAS BIBLIOGRÁFICAS | 56 |

LISTA DE FIGURAS

| | | |
|------|--|----|
| 1.1 | Dimensões padrões de um Cubesat. | 2 |
| 1.2 | Exemplo de disposição dos subsistemas em um CubeSat. | 3 |
| 1.3 | Filme para aquecimento de baterias Li-ion. | 7 |
| 1.4 | Funcionalidades implementadas pelo <i>firmware</i> | 8 |
| 2.1 | Modelo ideal da placa solar. | 10 |
| 2.2 | Modelo da placa solar com a inserção de uma resistência série. | 10 |
| 2.3 | Modelo da placa solar com a inserção um diodo e duas resistências. | 11 |
| 2.4 | Modelo da placa solar com a inserção com dois diodos e duas resistências. | 12 |
| 2.5 | Regulador linear básico. | 13 |
| 2.6 | Conversor CC-CC chaveado básico. | 13 |
| 2.7 | Saída do conversor CC-CC chaveado básico. | 14 |
| 2.8 | Conversor Buck. | 15 |
| 2.9 | Conversor Boost. | 16 |
| 2.10 | Conversor Buck-Boost. | 17 |
| 2.11 | Conversor SEPIC. | 18 |
| 2.12 | Modelo com resistência interna. | 19 |
| 2.13 | Modelo com resistência interna e um ramo RC paralelo. | 20 |
| 2.14 | Curvas I-V do painel solar: (a) Curvas com irradiância constante e variação de temperatura; e (b) Curvas com temperatura constante e variação de irradiância. | 21 |
| 2.15 | Curva P-V de um painel solar. | 22 |
| 2.16 | Fluxograma que descreve o algoritmo Perturba e Observa. | 22 |
| 2.17 | Fluxograma que descreve o algoritmo da Condutância Incremental. | 24 |
| 2.18 | Microcontrolador MSP430. | 25 |
| 2.19 | Estrutura do pacote da comunicação UART. | 27 |
| 2.20 | Estrutura do pacote da comunicação I2C. | 28 |
| 2.21 | Diagrama da comunicação para escolha do registrador. | 30 |
| 2.22 | Diagrama de comunicação para escrita. | 31 |
| 2.23 | Diagrama de comunicação para leitura. | 31 |
| 2.24 | Estrutura dos registradores de leitura do INA. | 32 |
| 2.25 | Influência do filtro de média nas leituras. | 33 |
| 2.26 | Módulos INA3221. | 33 |
| 2.27 | Visão traseira dos módulos INA3221. | 34 |
| 3.1 | Painel solar CNC110x110-12 da Mazava. | 35 |
| 3.2 | Bateria NanoPower 18650 da <i>GOMSpace</i> | 36 |
| 4.1 | Circuito do painel solar. | 39 |

| | | |
|------|--|----|
| 4.2 | Curva da corrente em função da tensão do painel solar. | 40 |
| 4.3 | Circuito do conversor CC-CC. | 41 |
| 4.4 | Saída do conversor CC-CC para diferentes valores de <i>duty cycle</i> | 41 |
| 4.5 | Circuito completo..... | 42 |
| 4.6 | Tensões do painel solar e da bateria no circuito completo..... | 42 |
| 4.7 | Pinagem do MSP430FR5994. | 44 |
| 4.8 | Diagrama da máquina de estados da interface de configuração. | 46 |
| 4.9 | Interface de configuração do EPS. | 47 |
| 4.10 | Montagem do circuito de validação do <i>firmware</i> | 51 |
| 4.11 | Tensão do painel solar durante o teste realizado. | 51 |
| 4.12 | Corrente do painel solar durante o teste realizado. | 52 |
| 4.13 | Potência do painel solar durante o teste realizado. | 52 |

LISTA DE TABELAS

| | | |
|-----|---|----|
| 2.1 | Endereços configuráveis do INA3221. | 30 |
| 2.2 | Dados dos registradores do INA3221..... | 32 |
| 3.1 | Especificações do painel solar CNC110x110-12 para temperatura de 25°C e irradiância de 1000 W/m^2 | 35 |
| 3.2 | Especificações da bateria NanoPower 18650 da <i>GOMSpace</i> | 36 |
| 4.1 | Pinos do microcontrolador utilizados pelo firmware..... | 44 |
| 4.2 | Resultado do teste das medições..... | 48 |
| 4.3 | Erro relativo das medições | 48 |
| 4.4 | Instruções da comunicação entre EPS e computador de bordo..... | 49 |
| 4.5 | Resultado do teste de medições de temperatura..... | 50 |

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

Explorar e compreender o mundo ao seu redor é uma das motivações do ser humano ao longo da sua história. No século XV, os europeus, com o aprimoramento do conhecimento náutico, iniciaram a exploração do mundo com a utilização de grandes navegações visando o desenvolvimento econômico com a descoberta de novas rotas e o estabelecimento de novas relações comerciais.

Diversos anos depois, no século XX, a humanidade, que já havia explorado e mapeado a superfície do planeta Terra, não se deteve na exploração do próprio planeta onde habita e iniciou a exploração espacial. Em Outubro de 1957, a União das Repúblicas Socialistas Soviéticas (URSS) lançou o primeiro satélite artificial na órbita da Terra, o Sputnik 1. Tal acontecimento ocorreu durante o período conhecido como Guerra Fria, no qual a URSS e os Estados Unidos da América (EUA) disputavam a hegemonia da exploração do espaço.

Nesse contexto, o conhecimento acerca do espaço foi aprimorado, com o desenvolvimento de novas tecnologias, com destaque ao aperfeiçoamento dos satélites artificiais que possibilitaram o surgimento e aprimoramento de diversas outras tecnologias utilizadas até os dias de hoje. Dentre as aplicações mais notáveis, pode-se destacar a previsão do tempo, o sistema de posicionamento global via satélite (GPS), a transmissão de rádio e televisão, a comunicação celular e a internet, que evidenciam a importância dos satélites artificiais para a humanidade, para o progresso tecnológico dos últimos anos, sendo essencial compreender o seu funcionamento para torná-los mais eficientes e até mesmo utilizá-los na criação de novas tecnologias.

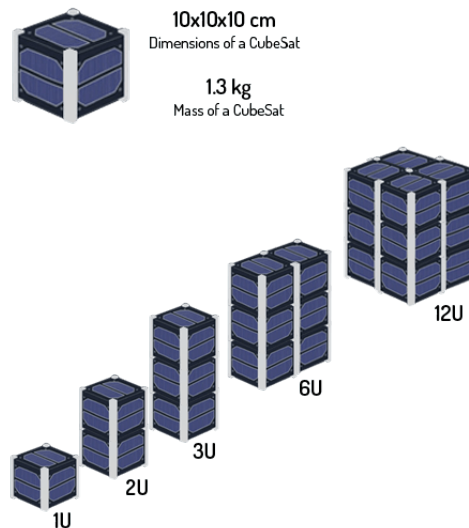
Desse modo, conforme a NASA, a agência dos EUA responsável pelo desenvolvimento de tecnologias e programas de exploração espacial, pode-se definir um satélite como “uma lua, planeta ou máquina que orbita um planeta ou estrela” (NASA, 2014). Portanto, os satélites artificiais são aqueles criados e lançados pelo ser humano e que passam a orbitar o planeta após o seu lançamento.

1.2 CUBESATS

Com o intuito de atender a inúmeras aplicações, os satélites artificiais possuem diversos formatos e tamanhos. Sendo assim, um tipo de satélite muito utilizado para pesquisas espaciais e comunicações radioamadoras é o CubeSat (ESA, 2022). Este faz parte de um grupo denominado de nanosatélites e é caracterizado por possuir um formato cúbico com lados de 10 cm e, conseqüentemente, volume de 1 litro e essa dimensão padrão é chamada de 1 unidade ou 1U. Desse modo, os CubeSats são constituídos por associações dessa unidade, tendo o tamanho de,

por exemplo, 1U, 2U, 3U ou 6U e normalmente, cada uma delas, possui menos que 1,33 kg de massa. Os padrões mais comuns podem ser visualizados na figura 1.1.

Figura 1.1: Dimensões padrões de um Cubesat.



Fonte: *A Review on Cube Satellites* (VINAY; SAINI, 2014).

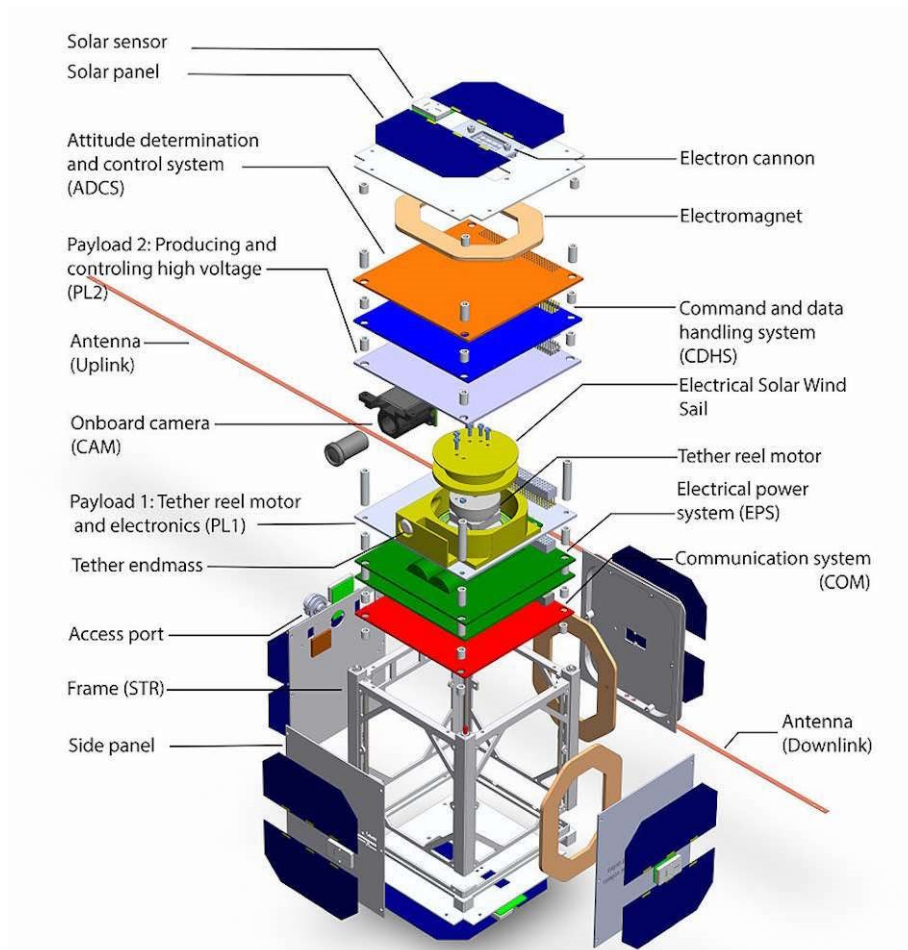
Ademais, apesar de possuírem uma pequena dimensão, os CubeSats contam com algumas partes importantes para o seu funcionamento, os seus subsistemas. Dentre estes, pode-se destacar:

- ***Communications system (COM)***: é o subsistema responsável por estabelecer a comunicação entre o satélite e a estação terrestre que o envia e também recebe informações advindas desse. Assim, a partir disso, é possível enviar comandos ao satélite e monitorar os dados de telemetria referentes ao seu estado, tais como, nível de bateria, temperatura, localização, dentre outros;
- ***Electrical power system (EPS)***: é o objeto de estudo desse trabalho. Por meio do EPS é realizado o fornecimento, o armazenamento e o controle da energia elétrica utilizada pelos demais subsistemas do satélite. Este subsistema é composto principalmente por painéis solares, baterias, reguladores de tensão e um circuito adicional responsável por otimizar a utilização da energia elétrica;
- ***Attitude determination and control system (ADCS)***: a orientação do satélite no espaço em relação a um referencial inercial é denominada atitude. A sua determinação e controle é responsabilidade do ADCS e para isso este subsistema utiliza atuadores, como rodas de reação, propulsores e hastes de torque que controlam a atitude a partir das informações obtidas pelos sensores, tais como rastreadores de estrelas, sensores de sol e Terra e sensores de taxa angular;

- **Command & data handling system (CDHS):** a interface entre os demais subsistemas é realizada pelo CDHS, possuindo um computador de bordo (OBC) responsável por desempenhar tal função. Desse modo, este subsistema gerencia a comunicação entre os demais, processando as informações e enviando comandos a estes;
- **Thermal Control System (TCS):** o objetivo do TCS é monitorar e controlar a temperatura do satélite de forma a mantê-la na faixa adequada para o seu bom funcionamento;
- **Payload:** está relacionado com o tipo de missão desempenhada pelo satélite, com a sua finalidade. Assim, este consiste nos equipamentos necessários para cumprir seu objetivo, como câmeras para fotos ou vídeos ou sensores específicos para algum tipo de experimento biológico.

Na figura 1.2, é possível observar um exemplo de disposição dos sistemas em um CubeSat.

Figura 1.2: Exemplo de disposição dos subsistemas em um CubeSat.



Fonte: *ESTCube-1* (LÄTT et al., 2014).

1.3 PROJETO ALFACRUX

Com o objetivo de desenvolver o conhecimento acerca dos satélites artificiais, a Universidade de Brasília (UnB) iniciou o projeto AlfaCruz no Laboratório de Simulação e Controle de Sistemas Aeroespaciais (LODESTAR) em 2019. Uma das motivações do projeto é possibilitar comunicação em regiões com condições adversas, que possuem obstáculos, como na região amazônica, na qual se encontra uma mata densa.

A primeira missão do projeto, o AlfaCruz I, teve como objeto de estudo um nanossatélite do tipo CubeSat que foi montado com peças compradas no mercado para compor os seus subsistemas. No dia 1º de Abril de 2022, o nanossatélite foi lançado pela missão *Falcon 9 Transporter-4* da SpaceX e está enviando dados de telemetria para a estação terrestre de comando e controle do laboratório (LODESTAR, 2021).

Com o envio do primeiro nanossatélite ao espaço, o próximo objetivo do projeto é desenvolver um satélite próprio da universidade, utilizando-se como base os estudos realizados a partir do primeiro satélite, tanto de seus subsistemas quanto dos dados que estão sendo enviados por este.

Nesse contexto, em projetos anteriores do laboratório LODESTAR ((FONSECA, 2020) e (MARTINS, 2021)), foram realizados estudos acerca de arquiteturas de *hardware* utilizadas para a implementação do EPS, nos quais foram propostas topologias de circuitos, analisadas por meio de simulações no simulador LTspice. Além disso, foi feito o projeto de *hardware* de uma placa de circuito impresso (PCB) para o EPS.

Tendo isto em vista, este projeto visa dar continuidade aos anteriores, com o desenvolvimento do *firmware* embarcado para o módulo de gerenciamento de energia, capaz de desempenhar funções importantes do EPS em conjunto com o uso do *hardware* adequado.

Dentre essas funcionalidades, o *firmware* desenvolvido tem como objetivo a realização de medições de corrente e tensão, bem como o fornecimento dos dados de telemetria. Ademais, objetiva-se possibilitar a configuração do EPS, definindo a temperatura adequada para operação das baterias e realizando o controle de uma malha para seu aquecimento, além de realizar o controle do fornecimento de energia para os demais subsistemas do satélite.

Portanto, com esse intuito, o *firmware* desenvolvido visa gerar a interface do EPS com os sensores e atuadores utilizados, além de fazer uso de diferentes protocolos de comunicação para estabelecer a comunicação com os demais subsistemas.

1.4 FIRMWARE EMBARCADO

O EPS, como mencionado anteriormente, é responsável pelo gerenciamento de energia do satélite. Sendo assim, para a construção deste subsistema é necessário o desenvolvimento de uma interface entre os dispositivos eletrônicos que o compõem, como o painel solar, a bateria e os

sensores, e um microcontrolador, responsável pelo processamento dos dados obtidos, bem como o controle de atuadores.

Sabendo disso, o *firmware* embarcado desenvolvido para aplicação do EPS deve ser capaz de desempenhar algumas funções básicas, tais como:

- Prover uma interface para configuração do subsistema na estação de solo, antes do lançamento do satélite;
- Obter e disponibilizar dados de telemetria do circuito, tais como corrente e tensão da bateria e do painel solar;
- Controlar o fornecimento de energia aos demais subsistemas do satélite;
- Realizar o controle da temperatura da bateria;
- Prover comunicação entre o EPS e o subsistema de controle, que é composto pelo computador de bordo.

1.4.1 Interface de configuração

Antes de ser enviado à órbita terrestre, na estação de solo, devem ser realizadas as configurações do subsistema de gerenciamento de energia. Para isso, é necessário o desenvolvimento de uma interface de configuração, na qual o usuário seja capaz de escolher os parâmetros utilizados nas tarefas desempenhadas pelo *firmware*.

Diante desse contexto, a comunicação UART se apresenta como uma ferramenta capaz de prover a interface entre o microcontrolador e um computador, o que possibilita o desenvolvimento de um ambiente de configuração simples para o usuário. Sendo assim, será desenvolvida uma interface que possibilite a configuração da temperatura na qual a bateria deverá ser mantida, bem como a realização da leitura dos dados de telemetria e também o controle do fornecimento de energia para os demais subsistemas do satélite utilizando a comunicação UART.

1.4.2 Dados de telemetria

O acesso aos dados de telemetria é imprescindível para a supervisão do funcionamento de qualquer dispositivo. No caso de um satélite em órbita é ainda mais fundamental, tendo em vista que somente é possível o controle do dispositivo de maneira remota e que os dados de telemetria são muito importantes para que sejam feitas análises e possíveis melhorias à tecnologia utilizada.

Desse modo, do ponto de vista do gerenciamento de energia, é fundamental o acesso às informações referentes às características elétricas do circuito, como tensão e corrente do painel solar, da bateria e dos subsistemas. Sendo assim, o *firmware* desenvolvido deve fazer uso de sensores que possibilitem a leitura dessas grandezas elétricas, tanto para controle do funcionamento interno do subsistema, quanto para envio de dados ao computador de bordo.

Nesse contexto, a medição das tensões do circuito pode ser realizada com a utilização de conversores A/D e divisores resistivos, caso necessário. Por outro lado, para medição de corrente, pode-se utilizar circuitos integrados (CI) que desempenhem tal função, como é o caso do INA3221 que possibilita a estimativa de corrente por meio da medição da tensão em resistores *shunt*. Ademais, tal sensor faz uso do protocolo I2C, sendo necessária a utilização de uma interface dedicada ou a sua implementação via *software* por parte do microcontrolador. Além disso, para este tipo de aplicação, o I2C se apresenta como uma alternativa eficaz, visto que possibilita a utilização de diversos sensores para somente uma interface do microcontrolador configurada como mestre, tendo em vista que o INA3221, por exemplo, possui a possibilidade de configuração para mais de um endereço distinto.

1.4.3 Fornecimento de energia aos subsistemas

Uma das funções mais importantes do sistema de gerenciamento de energia é o controle do fornecimento da energia elétrica aos demais componentes do satélite, tanto para economia, quanto para possíveis manutenções que possam ser necessárias. Com este intuito, podem ser utilizados dispositivos que funcionem como chaves eletrônicas, como transistores ou circuitos integrados que desempenhem tal função.

Em trabalhos anteriores (MARTINS, 2021), é apresentado como uma boa alternativa pra esta questão, o uso do circuito integrado TPS22976 (INSTRUMENTS, 2020c), que se trata de um dispositivo com dois canais independentes compostos por transistores MOS que suportam uma corrente superior a 5 A e podem ser controlados por tensões de 3,3 V.

Sendo assim, para interfaceamento deste CI com o *firmware* é necessária a configuração de pinos digitais (GPIO) como saídas e assim, já que o TPS22976 possui canais independentes, é possível o uso de tais pinos para controle individual do fornecimento de energia para cada subsistema do satélite.

1.4.4 Controle de temperatura da bateria

Em uma operação espacial existem diversos desafios impostos pelas condições do ambiente aos quais os satélites estão submetidos e, dentre estas, pode-se destacar a temperatura. Em órbita, os satélites enfrentam grandes variações, podendo atingir temperaturas superiores a 100 °C, quando expostos à radiação solar, e inferiores a -100 °C, no caso de sua ausência (POGHOSYAN; GOLKAR, 2017).

Baterias de íon-lítio, conforme seus *datasheets*, operam normalmente com temperaturas de até 0°C em seu carregamento e com temperaturas de até -20°C em caso de descarregamento (KNAP; VESTERGAARD; STROE, 2020). No caso de baterias comumente utilizadas em Cubesats, como a NanoPower 18650 da *GOMSpace*, a temperatura de operação é de 0°C a 45°C no carregamento e de -20°C a 60 °C no descarregamento (GOMSPACE, 2019).

Diante disso, para um correto funcionamento da bateria, deve haver um controle da temperatura na qual esta opera. Sensores de temperatura podem ser utilizados para monitoramento de tal grandeza, como o LM35 (INSTRUMENTS, 2017) da *Texas Instruments*, que realiza a medição da temperatura e a disponibiliza em sua saída analógica, sendo que um conversor A/D do microcontrolador deve ser empregado para a medição. Ademais, vale ressaltar que grande parte dos sensores disponíveis no mercado não realizam a medição de temperaturas abaixo de -55°C , como é o caso do LM35, porém como a faixa de operação das baterias de íon-lítio não chega a tal valor, isto não interfere na utilização do sensor, visto que a temperatura deverá ser mantida sempre acima deste número.

Além do mais, somente a medição da temperatura não é suficiente, é necessário que se utilizem mecanismos de controle desta grandeza. Em casos em que a bateria esteja com o valor de temperatura acima da sua faixa de operação podem ser utilizados sistemas de resfriamento (EGAB; OUDAH, 2022). Por outro lado, em caso de temperaturas baixas, pode-se utilizar filmes para aquecimento (YANG et al., 2022), como o evidenciado na Figura 1.3. Sendo assim, o *firmware* deve ser capaz de controlar o fornecimento de energia a estes dispositivos, a partir do uso de saídas digitais.

Figura 1.3: Filme para aquecimento de baterias Li-ion.



Fonte: ICStation.

1.4.5 Comunicação com o computador de bordo

O EPS por si só não é capaz de se comunicar com a estação de solo, receber comandos desta e enviá-la os seus dados de telemetria. O subsistema capaz de realizar estas tarefas é o subsistema de controle. Portanto, é necessária a comunicação entre ambos, para que os dados de telemetria sejam disponibilizados e para que o EPS seja configurável.

Sabendo disso, como o computador de bordo não se comunica somente com o subsistema de gerenciamento de energia, é necessário um protocolo de comunicação que permita a troca de

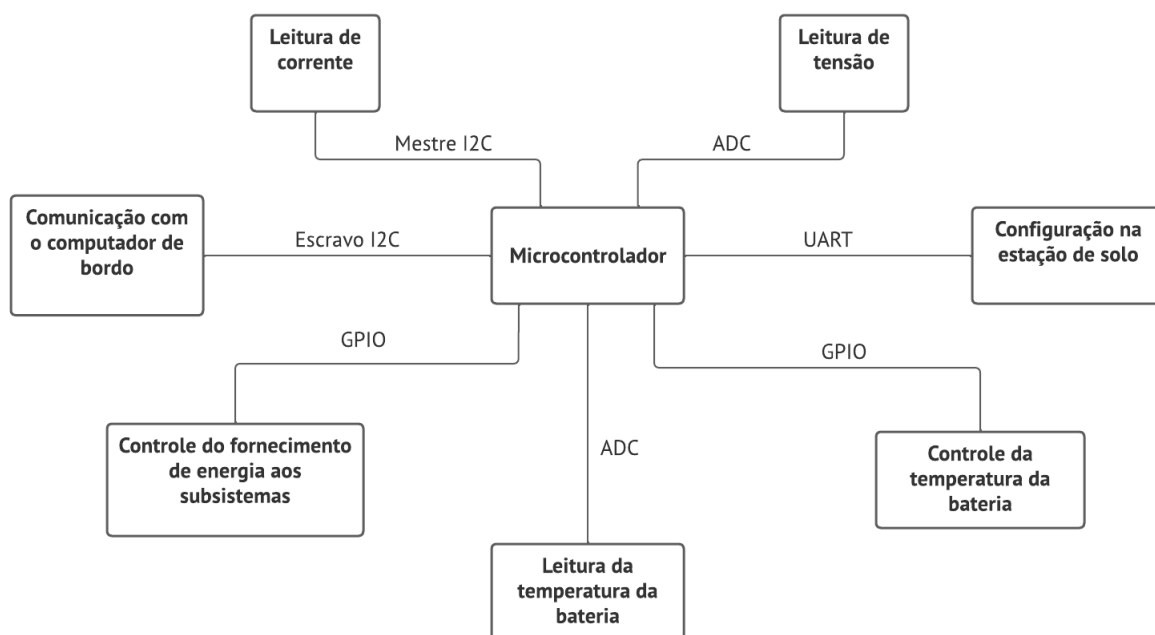
informações com mais de um dispositivo utilizando-se o mínimo de recursos possíveis. Nesse contexto, o I2C se apresenta como uma alternativa, em decorrência de possibilitar a comunicação com o computador de bordo sendo configurado como mestre, enquanto os demais subsistemas desempenham o papel de escravos.

Diante disso, o *firmware* deve ser capaz de prover uma interface de comunicação I2C, configurada como escravo, que receba as informações do subsistema de controle, responda com os dados requisitados e realize as configurações solicitadas.

1.4.6 Arquitetura proposta

Diante das funcionalidades essenciais que o *firmware* embarcado do subsistema de gerenciamento de energia deve estar apto a desempenhar, é proposta a arquitetura evidenciada na Figura 1.4. Nesta, sugere-se o uso de um microcontrolador que disponibilize as interfaces dedicadas para as tarefas pré-determinadas, além de estarem descritas as funções desempenhadas por tal dispositivo, bem como as interfaces utilizadas para cada uma delas.

Figura 1.4: Funcionalidades implementadas pelo *firmware*.



Fonte: Desenvolvida pelo próprio autor.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentados os conceitos necessários para o desenvolvimento e compreensão do trabalho. Sendo assim, primeiramente, serão apresentados modelos de painéis solares, baterias e os principais tipos de conversores CC-CC. Em seguida, serão abordadas a definição de MPPT e técnicas para sua implementação, a definição de microcontrolador e seus principais componentes e tipos de comunicação, bem como o entendimento do sensor de corrente utilizado no desenvolvimento do *firmware*.

2.1 MODELOS DE PLACA SOLAR

Um componente fundamental para o módulo de gerenciamento de energia do CubeSat é o gerador de energia, papel desempenhado pelas placas solares. Sendo assim, tendo em vista que o satélite encontra-se no espaço, toda a energia disponível para seu funcionamento provém destes módulos que utilizam a radiação solar para gerar energia elétrica.

Dessa maneira, para que seja possível o estudo destes componentes e do circuito de gerenciamento de energia como um todo, é necessário que sejam adotados modelos que representem o comportamento dos módulos fotovoltaicos. Assim, serão estudados modelos que têm como objetivo representar as características I-V do dispositivo, levando-se em conta a influência das condições ambientais ao se ampliar a complexidade da modelagem.

2.1.1 Modelo ideal

A célula fotovoltaica é constituída por duas camadas de material semicondutor dopadas individualmente, compondo uma junção P-N. Esta junção é exposta à irradiação incidente, absorvendo fótons com energia suficiente para fazer com que os elétrons passem da banda de valência para a banda de condução, gerando pares elétron-buraco, o que acarreta no surgimento de uma corrente elétrica (I_{pv}) (ARAÚJO; SOUSA; COSTA, 2020).

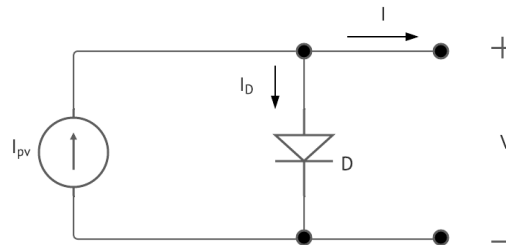
Sendo assim, observa-se que o comportamento de uma célula fotovoltaica é como a de um diodo de junção P-N comum e na ausência de irradiação solar, a corrente que passa pela célula pode ser expressa pela Equação 2.1.

$$I_D = I_S \cdot \left(e^{\frac{qV}{mkT}} - 1 \right) \quad (2.1)$$

em que I_S é a corrente de saturação reversa, T é a temperatura, em Kelvin, k é a constante de Boltzmann e q é a carga de um elétron, cujos valores são de, aproximadamente, $1,380 \cdot 10^{-23} J/K$

e $1,602 \cdot 10^{-19}C$, respectivamente. Ademais, tem-se o fator de idealidade (m) que varia entre 1 e 2, tendendo a 2 para tensões baixas, em que predomina o fenômeno de recombinação e a 1 para tensões altas, em que predomina a difusão. Sendo assim, o modelo mais simples para descrever a placa solar é evidenciado na Figura 2.1.

Figura 2.1: Modelo ideal da placa solar.



Fonte: Desenvolvida pelo próprio autor.

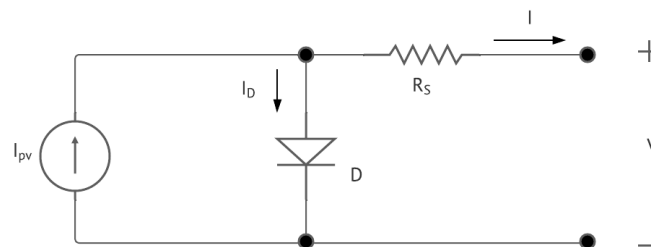
Desse modo, adicionando-se o termo da corrente I_{pv} gerada pela irradiação solar, tem-se que a corrente total do diodo neste modelo é dada pela Equação 2.2.

$$I_D = I_{pv} - I_S \cdot \left(e^{\frac{qV}{mkT}} - 1 \right) \quad (2.2)$$

2.1.2 Modelo com inserção de uma resistência em série

O modelo ideal da placa solar não reflete o comportamento real dos painéis solares, visto que não leva em conta não-idealidades as quais estes estão submetidos. Nesse contexto, com o intuito de melhorar a modelagem pode-se acrescentar um resistor em série ao modelo ideal para representar as resistências intrínsecas ao dispositivo, responsáveis por perdas, como resistências parasitas, resistências dos materiais que compõem o módulo, dentre outras (ARAÚJO; SOUSA; COSTA, 2020). Este modelo encontra-se evidenciado na Figura 2.2.

Figura 2.2: Modelo da placa solar com a inserção de uma resistência série.



Fonte: Desenvolvida pelo próprio autor.

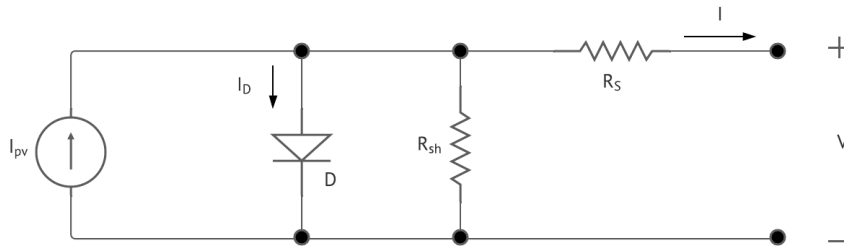
Dessa maneira, a relação que descreve a corrente do modelo é dada pela Equação 2.3.

$$I_D = I_{pv} - I_S \cdot \left(e^{\frac{q(V+R_S I)}{mkT}} - 1 \right) \quad (2.3)$$

2.1.3 Modelo com um diodo e duas resistências

A inserção de uma resistência série faz com que o modelo do painel solar leve em consideração os efeitos dissipativos e as resistências internas ao material que constituem o dispositivo. No entanto, além dessas não-idealidades, estão presentes algumas outras, cujo comportamento pode ser acrescentado à modelagem com a inserção de mais componentes. Sendo assim, para se levar em consideração o efeito da variação de temperatura e defeitos das células solares responsáveis por correntes de fuga, pode ser acrescentado ao circuito uma resistência em paralelo (R_{sh}). Desse modo, a resistência em série altera a tensão de saída, enquanto a em paralelo modifica a corrente, reduzindo-a (ARAÚJO; SOUSA; COSTA, 2020). Este modelo encontra-se evidenciado na Figura 2.3.

Figura 2.3: Modelo da placa solar com a inserção um diodo e duas resistências.



Fonte: Desenvolvida pelo próprio autor.

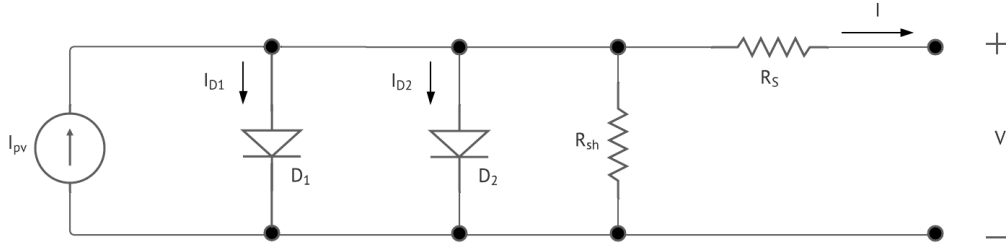
Dessa maneira, com a inserção da resistência em paralelo, a relação que descreve a corrente do modelo passa a ser dada pela Equação 2.4.

$$I_D = I_{pv} - I_S \cdot \left(e^{\frac{q(V+R_S I)}{mkT}} - 1 \right) - \frac{(V + R_S I)}{R_{sh}} \quad (2.4)$$

2.1.4 Modelo com dois diodos e duas resistências

Os modelos que contêm apenas um diodo não levam em consideração as perdas de recombinação na região de depleção, tornando a precisão da modelagem pior nas situações de baixas irradiâncias. Assim, para modelar tal efeito pode-se inserir um diodo adicional em paralelo ao primeiro (ARAÚJO; SOUSA; COSTA, 2020) como evidenciado na Figura 2.4.

Figura 2.4: Modelo da placa solar com a inserção com dois diodos e duas resistências.



Fonte: Desenvolvida pelo próprio autor.

Desse modo, a relação que descreve a corrente do modelo passa a ser dada pela Equação 2.5.

$$I_D = I_{pv} - I_{S1} \cdot \left(e^{\frac{q(V+R_S I)}{m_1 k T}} - 1 \right) - I_{S2} \cdot \left(e^{\frac{q(V+R_S I)}{m_2 k T}} - 1 \right) - \frac{(V + R_S I)}{R_{sh}} \quad (2.5)$$

Ademais, além dos modelos apresentados, existem alguns outros que têm como objetivo modelar ainda mais detalhadamente todos os fenômenos associados ao funcionamento da placa solar, como a inserção de resistências em série com os diodos para modelar as suas perdas, a inserção de um capacitor que representa a capacitância da junção P-N, dentre outros componentes que podem ser acrescentados.

Contudo, estudos realizados a respeito deste tema (ARAÚJO; SOUSA; COSTA, 2020), afirmam que os modelos contendo somente um diodo e uma ou duas resistências modelam adequadamente o funcionamento das placas solares, tendo em vista o efeito desprezível da recombinação sob condições normais de iluminação. Sabendo disso, para modelagem do painel solar neste trabalho será adotado o modelo com a fonte de corrente em paralelo com um diodo e com a adição de uma resistência série.

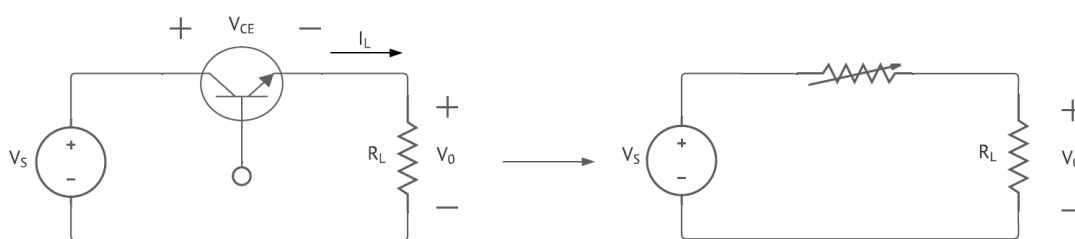
2.2 CONVERSORES CC-CC

Conversores CC-CC são circuitos eletrônicos utilizados para alterar níveis de tensão CC, elevando-os ou reduzindo-os, fornecendo uma saída regulada. Sabendo disso, este tipo de circuito é muito importante em projetos de eletrônica de potência e, para seu melhor entendimento, serão apresentadas as suas principais topologias de maneira simplificada. Por outro lado, na literatura (HART, 2016), podem ser encontradas análises mais detalhadas a respeito deste tipo de circuito.

Sendo assim, uma alternativa simples para implementação do conversor CC-CC é o regulador linear básico, descrito na Figura 2.5, no qual é alterado o nível de tensão da saída em relação à entrada com um divisor resistivo, sendo utilizado um transistor como um resistor variável. No

entanto, neste circuito, há uma perda considerável de potência no transistor, o que o torna ineficiente.

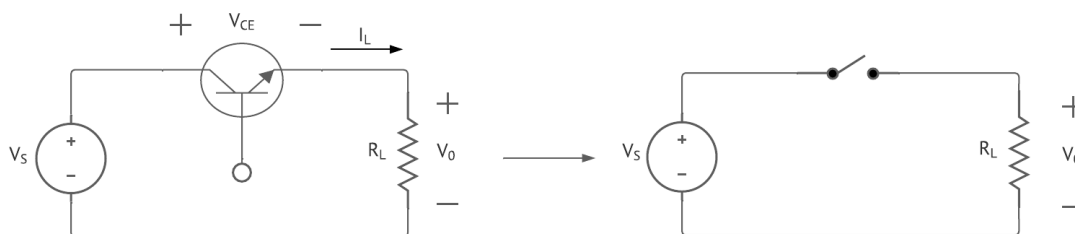
Figura 2.5: Regulador linear básico.



Fonte: Desenvolvida pelo próprio autor.

Nesse contexto, como alternativa mais eficiente, tem-se os conversores chaveados, como o da Figura 2.6. Neste tipo de conversor CC-CC, o transistor é utilizado como uma chave eletrônica que periodicamente é completamente ligada e desligada. Desse modo, no caso do transistor do tipo MOSFET, este opera nas regiões de triodo e corte, enquanto o TBJ opera nas regiões de saturação e corte.

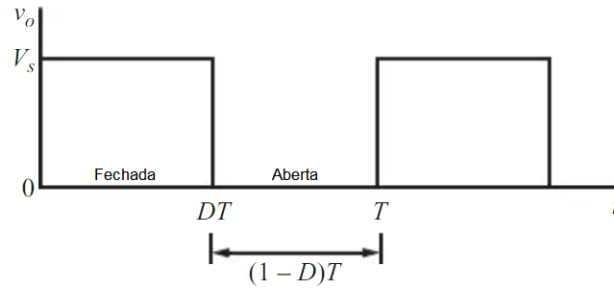
Figura 2.6: Conversor CC-CC chaveado básico.



Fonte: Desenvolvida pelo próprio autor.

Sabendo disso, a abertura e fechamento da chave eletrônica gera uma saída pulsada, descrita na Figura 2.7. Assim, o valor médio da tensão de saída passa a ser um produto entre a tensão de entrada e a fração do período de chaveamento em que a chave está fechada, denominada taxa de trabalho ou *duty cycle* (D).

Figura 2.7: Saída do conversor CC-CC chaveado básico.



Fonte: Desenvolvida pelo próprio autor.

Desse modo, é obtida a Equação 2.6 para a tensão de saída do circuito e a Equação 2.7 para o *duty cycle*.

$$V_o = \frac{1}{T} \int_0^T v_o(t) dt = \frac{1}{T} \int_0^{DT} V_S dt = V_S D \quad (2.6)$$

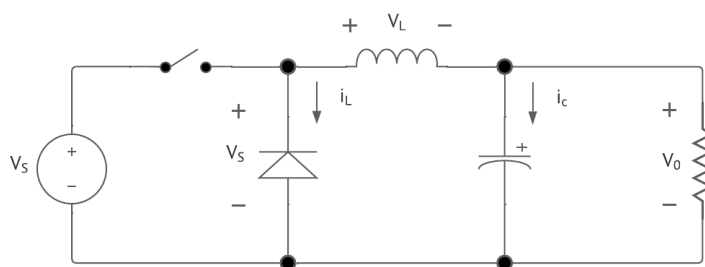
$$D = \frac{t_{ligado}}{t_{ligado} + t_{desligado}} = \frac{t_{ligado}}{T} = t_{ligado} \cdot f \quad (2.7)$$

Entretanto, devido ao fato da saída do circuito chaveado básico ser pulsada, não se tem uma saída contínua, apesar de seu valor médio possuir tal comportamento. Desse modo, visando uma saída CC, são inseridos filtros passa-baixas aos circuitos, dando origem às principais topologias de conversores CC-CC.

2.2.1 Conversor Buck

O conversor Buck é um circuito chaveado utilizado para converter uma corrente contínua na sua entrada para outra também contínua em sua saída, porém com nível de tensão igual ou inferior (HART, 2016). O circuito que implementa este tipo de conversor é descrito na Figura 2.8.

Figura 2.8: Conversor Buck.



Fonte: Desenvolvida pelo próprio autor.

Para se analisar o circuito em questão e se determinar as relações das grandezas elétricas, bem como os valores dos componentes, são necessárias algumas considerações.

1. Os componentes do circuito são ideais;
2. O circuito opera no estado estável;
3. O indutor opera no modo contínuo, em que a sua corrente é sempre positiva;
4. O capacitor tem valor muito alto e a tensão de saída é mantida constante em V_0 ;
5. O período total do chaveamento é T , o período fechado é de DT e o aberto é de $(1-D)T$.

Assumindo-se tais considerações e realizando-se a análise do circuito é obtida a Equação 2.8 para a tensão de saída.

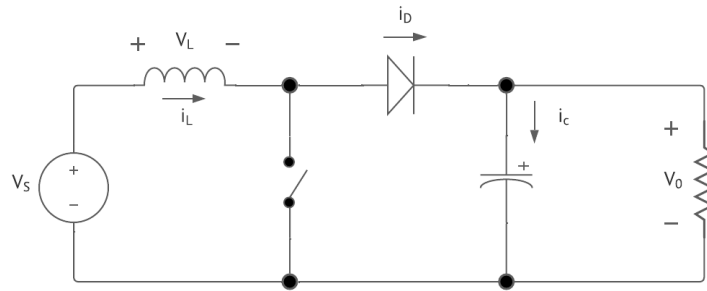
$$V_o = V_s D \quad (2.8)$$

Assim, observa-se que a tensão de saída é dependente apenas da tensão de entrada e do *duty cycle*, não tendo influência dos componentes do circuito. Assim, com a tensão de entrada constante, o nível de tensão da saída pode ser ajustado somente ao se alterar o tempo em que a chave permanece fechada e aberta.

2.2.2 Conversor Boost

O conversor Boost possui um comportamento similar ao conversor Buck, porém enquanto o Buck converte a tensão de entrada em outra na sua saída com nível de tensão igual ou inferior, o Boost eleva ou iguala a saída ao nível de tensão da entrada (HART, 2016). O circuito do conversor Boost está descrito na Figura 2.9.

Figura 2.9: Conversor Boost.



Fonte: Desenvolvida pelo próprio autor.

Para análise desse circuito são realizadas as mesmas considerações utilizadas para o conversor Buck. Dessa maneira, obtém-se a Equação 2.9, que descreve a relação entre a tensão de saída e entrada do circuito.

$$V_o = \frac{V_S}{1 - D} \quad (2.9)$$

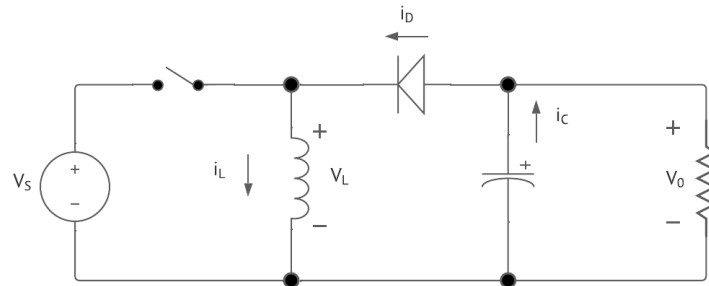
Segundo a literatura (HART, 2016), diferentemente do conversor Buck, o Boost possui uma alta influência da resistência do indutor, visto que, apesar de inicialmente ser realizada a consideração de que os componentes do circuito são ideais, é necessária a inserção dessa não-idealidade responsável por provocar perdas e alterar a eficiência do circuito. Sabendo disso, levando-se em consideração tal comportamento, obtém-se uma relação entre tensão de saída e entrada corrigida por um fator de correção, descrita pela Equação 2.10.

$$V_o = \left(\frac{V_S}{1 - D} \right) \cdot \left(\frac{1}{1 + r_L/[R(1 - D)^2]} \right) \quad (2.10)$$

2.2.3 Conversor Buck-Boost

Os conversores Buck e Boost possibilitam individualmente que o nível de tensão de saída seja reduzido ou aumentado em relação à entrada, respectivamente. No entanto, não é possível a utilização de nenhum destes dois circuitos para operarem tanto como elevadores de tensão quanto como abaixadores. Sabendo disso, uma alternativa de circuito que tanto eleva quanto reduz o valor da tensão é o conversor Buck-Boost (HART, 2016), cuja topologia é descrita na Figura 2.10.

Figura 2.10: Conversor Buck-Boost.



Fonte: Desenvolvida pelo próprio autor.

Para esta topologia de conversor CC-CC, assumindo-se as mesmas considerações dos circuitos anteriores e realizando-se a análise, obtém-se a relação entre tensão de entrada e saída descrita pela Equação 2.11.

$$V_o = -V_s \cdot \left(\frac{D}{1 - D} \right) \quad (2.11)$$

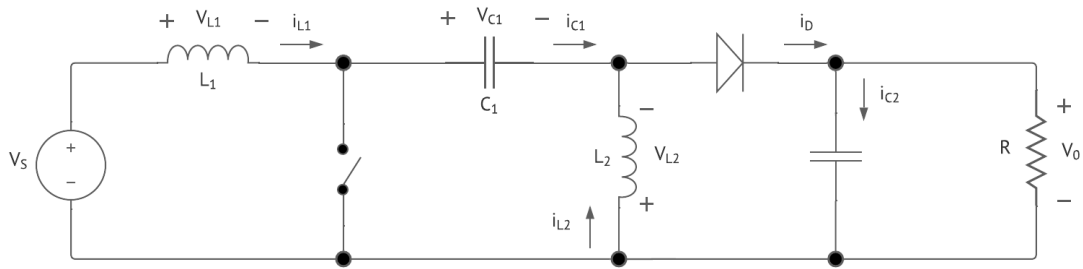
A partir dessa expressão, nota-se que a depender do valor do *duty cycle*, o conversor Buck-Boost eleva, reduz ou iguala à tensão de saída em relação à de entrada. Portanto, se $D > 0,5$, a tensão é elevada, se $D < 0,5$, esta é reduzida, já se $D = 0,5$, o nível de tensão se mantém.

Ademais, nota-se que neste tipo de circuito há uma inversão na polaridade, isto é, caso a tensão de entrada seja positiva, a tensão de saída será negativa ou, o contrário, se a entrada for negativa, a saída será positiva. Tal comportamento pode ser indesejado a depender da aplicação.

2.2.4 Conversor SEPIC

O conversor com indutância simples no primário (SEPIC) assim como o Buck-Boost possibilita que o nível de tensão seja elevado, reduzido ou igualado. Entretanto, o conversor SEPIC não possui a principal desvantagem do modelo de conversor anterior que é a inversão de polaridade da tensão de saída em relação à entrada (HART, 2016). O circuito deste tipo de conversor é evidenciado na Figura 2.11.

Figura 2.11: Conversor SEPIC.



Fonte: Desenvolvida pelo próprio autor.

Sabendo disso, assumindo-se as mesmas considerações utilizadas anteriormente e analisando-se o circuito do conversor SEPIC, tem-se a relação entre entrada e saída descrita na Equação 2.12.

$$V_o = V_s \cdot \left(\frac{D}{1 - D} \right) \quad (2.12)$$

Dessa forma, comparando-se a Equação 2.11 com a Equação 2.12, nota-se que estas são similares, o que indica que as duas topologias de conversores CC-CC possuem comportamentos análogos para as mesmas faixas de valores de *duty cycle*. Entretanto, por não gerar inversão de polaridade, o conversor SEPIC se mostra como uma alternativa para aplicações em que a polaridade da tensão é importante, possuindo tal vantagem em relação ao conversor Buck-Boost.

Diante disso, para caracterização do EPS, será adotado o conversor SEPIC, já que este pode ser usado para reduzir e elevar o valor de tensão sem inversão de polaridade, o que é importante, visto que a tensão do painel solar pode variar conforme as condições as quais tal dispositivo esteja submetido.

2.3 MODELOS DE BATERIA

As baterias são componentes fundamentais em um circuito elétrico, no qual há a necessidade de armazenamento de energia. No caso do objeto de estudo deste trabalho, estas se demonstram como indispensáveis, visto que a principal fonte de energia do satélite é proveniente do Sol e que a depender da posição de um em relação ao outro, a incidência solar pode não ser suficiente para suprir a demanda energética do satélite, fazendo-se necessário o armazenamento da energia gerada com o uso de baterias.

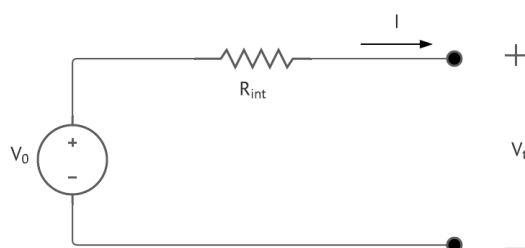
Neste contexto, as baterias são componentes utilizados no armazenamento de energia elétrica

que fazem uso de reações químicas para desempenhar tal função, possuindo alguns tipos distintos, dentre estes, as baterias de íon-lítio, de níquel-cádmio e de chumbo-ácido. Além disso, assim como os painéis solares, para o estudo da aplicação de baterias em circuitos é importante a utilização de modelos que representem o seu comportamento dentro de um circuito elétrico.

2.3.1 Modelo com resistência interna

O circuito mais simples para modelagem do funcionamento de uma bateria é o evidenciado na Figura 2.12, composto por uma fonte de tensão em série com uma resistência. Sabendo disso, neste modelo de circuito, três grandezas são importantes para a sua caracterização, a tensão de circuito aberto (V_0), a tensão terminal (V_t) e a resistência interna (R_{int}) (THAKKAR, 2021).

Figura 2.12: Modelo com resistência interna.



Fonte: Desenvolvida pelo próprio autor.

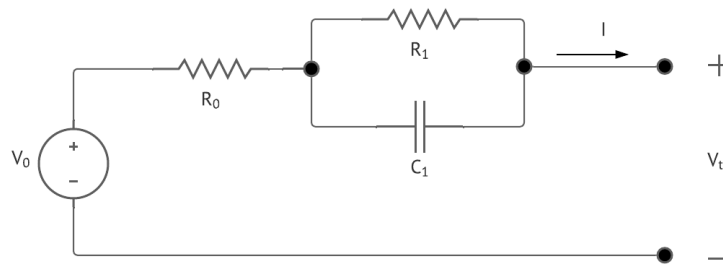
Sendo assim, a tensão terminal da bateria pode ser obtida ao se realizar a medição da tensão de circuito aberto com a bateria completamente carregada, enquanto a resistência interna pode ser determinada conectando uma carga ao circuito e medindo a tensão terminal e a corrente.

Além disso, por ser simplificada, a modelagem com somente uma resistência em série com a fonte de tensão apresenta algumas desvantagens. Dentre estas, o modelo não leva em consideração variações na resistência interna provocada por mudanças de temperatura e alterações no estado de carga da bateria e da concentração de eletrólitos.

2.3.2 Modelo com resistência interna e um ramo RC paralelo

Com o objetivo de tornar o modelo de bateria mais completo, aproximando-se de seu comportamento real, pode-se acrescentar um ramo RC paralelo em série à resistência interna da bateria, conforme é evidenciado na Figura 2.13. Este ramo RC tem como intuito descrever a resposta da bateria em comportamento transitório em determinado estado de carga (THAKKAR, 2021).

Figura 2.13: Modelo com resistência interna e um ramo RC paralelo.



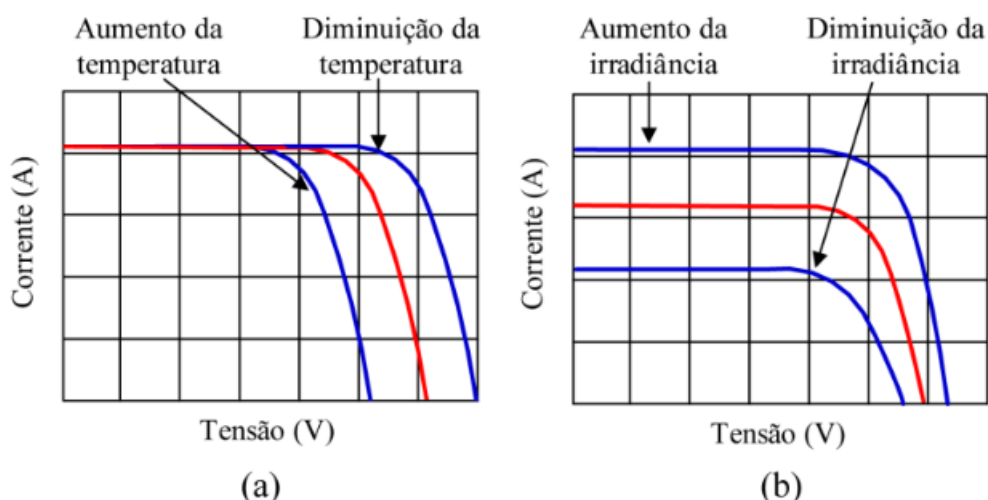
Fonte: Desenvolvida pelo próprio autor.

Ademais, assim como foi mencionado na modelagem descrita para os painéis fotovoltaicos, além dos modelos apresentados, existem alguns outros que têm como objetivo modelar de maneira mais detalhada todos os fenômenos associados ao funcionamento da bateria. Entretanto, o modelo da Figura 2.12 é suficiente para o estudo realizado neste trabalho e será o adotado para caracterização do EPS.

2.4 **MAXIMUM POWER POINT TRACKING (MPPT)**

Os painéis solares quando conectados diretamente a uma bateria, sem nenhum controle em relação à tensão e à corrente gerada por estes, apresentam baixa eficiência. Sabendo disso, é importante que sejam realizadas análises das curvas I-V dos módulos fotovoltaicos, descritas na Figura 2.14, de modo a manter a operação destes dispositivos em seu ponto de máxima potência. Ademais, vale destacar que tais curvas sofrem influências da temperatura e da irradiância solar às quais as placas solares são submetidas.

Figura 2.14: Curvas I-V do painel solar: (a) Curvas com irradiância constante e variação de temperatura; e (b) Curvas com temperatura constante e variação de irradiância.



Fonte: *As 7 principais técnicas de MPPT* (MORAES, 2020).

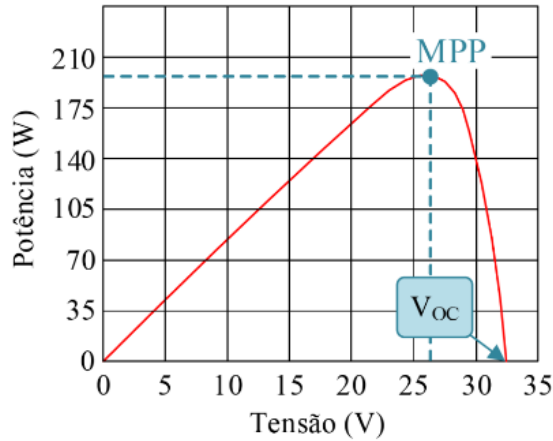
Sabendo disso, a técnica de MPPT (*Maximum Power Point Tracking*) é comumente utilizada para melhorar o desempenho dos painéis solares, fazendo com que estes operem nas condições em que forneçam máxima potência. Portanto, para atingir tal objetivo, esta técnica rastreia o ponto no qual a tensão e a corrente do dispositivo fornecem o máximo valor de potência e utiliza-se um sistema de controle para garantir que este seja o ponto de operação dos dispositivos.

Dessa maneira, para implementação do MPPT, utiliza-se um conversor CC-CC como interface entre o módulo fotovoltaico e a bateria, de modo a controlar a condição de operação. Este controle é implementado alterando-se o *duty cycle* do conversor e pode ser realizado utilizando-se dispositivos externos ao circuito, como microcontroladores, que executem os algoritmos de implementação do MPPT como os que serão descritos a seguir.

2.4.1 Perturba e Observa (P&O)

O algoritmo Perturba e Observa (P&O) é um método simples e um dos mais utilizados para implementação da técnica de MPPT. Trata-se de um algoritmo iterativo que constantemente introduz perturbações no sistema controlado e observa o seu comportamento (MORAES, 2020). Assim, tendo em vista que a variação do *duty cycle* do conversor CC-CC possibilita que seja variada a sua tensão de saída, é interessante que seja analisada a curva P-V do painel solar, descrita na Figura 2.15, mantendo-se sua corrente constante.

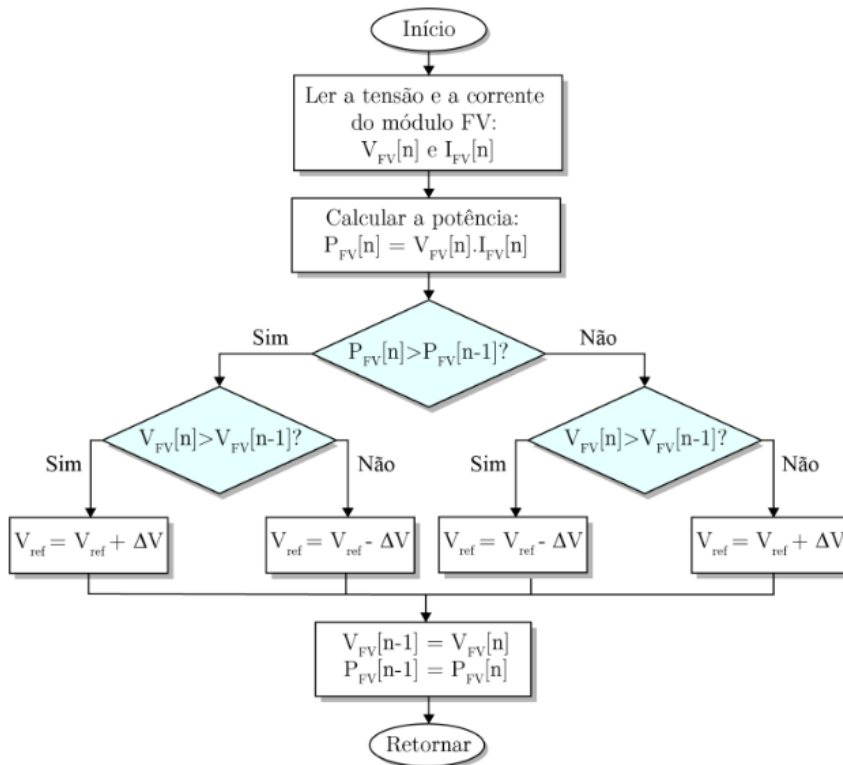
Figura 2.15: Curva P-V de um painel solar.



Fonte: *As 7 principais técnicas de MPPT* (MORAES, 2020).

Analisando-se a curva de Potência (W) x Tensão (V) de um painel solar com corrente constante, nota-se que uma variação na sua tensão provoca uma variação na potência. Assim, o algoritmo Perturba e Observa consiste em variar D do conversor e, conseqüentemente, a tensão, de forma a tentar manter o módulo fotovoltaico operando em seu ponto de máxima potência. Tal comportamento está descrito no fluxograma evidenciado na Figura 2.16.

Figura 2.16: Fluxograma que descreve o algoritmo Perturba e Observa.



Fonte: *As 7 principais técnicas de MPPT* (MORAES, 2020).

Neste contexto, o ΔV é o valor de tensão dependente do *duty cycle* do conversor e que pode ser controlado pelo sistema de controle, alterando-se a tensão V_{ref} na tentativa de manter o ponto de máxima potência. Portanto, observando-se o fluxograma, tem-se que inicialmente são obtidos os valores de tensão e corrente e, posteriormente, o de potência e verifica-se se a potência aumentou ou diminuiu em relação à iteração anterior, tendo-se os seguintes cenários:

1. Aumento da potência:

- (a) **Aumento da tensão em relação a iteração anterior:** aumenta-se a tensão com alteração de D;
- (b) **Redução da tensão em relação a iteração anterior:** reduz-se a tensão com alteração de D.

2. Redução da potência:

- (a) **Aumento da tensão em relação a iteração anterior:** reduz-se a tensão com alteração de D;
- (b) **Redução da tensão em relação a iteração anterior:** aumenta-se a tensão com alteração de D.

Apesar deste algoritmo ser simples e eficiente, caso haja uma variação abrupta tanto na temperatura quanto na irradiância fazendo com que as curvas analisadas sejam alteradas, o Perturba e Observa não funcionaria de maneira eficiente neste cenário. Isso ocorre visto que poderiam ser necessárias diversas iterações até que fosse alcançado novamente o ponto de máxima potência, a depender do deslocamento deste ponto e também do tamanho do passo de cada iteração (ΔV).

Ademais, este algoritmo não se estabiliza no ponto de máxima potência, já que em todas as iterações o valor de tensão é alterado, fazendo que o ponto de operação fique oscilando ao redor do ponto de máxima potência.

2.4.2 Condutância Incremental (CI)

Como visto anteriormente, o método Perturba e Observa não é capaz de alcançar estabilidade no ponto de máxima potência e acaba oscilando ao redor deste ponto. Uma alternativa para solucionar este problema é o método da Condutância Incremental (CI) que é capaz de verificar se o ponto desejado foi atingido e não oscilar (MORAES, 2020).

Sendo assim, o método da Condutância Incremental utiliza a derivada da potência em relação a tensão ($\frac{dP}{dV}$), visto que, no ponto de máxima potência na curva P-V, o seu valor é nulo. Além disso, na curva da Figura 2.15, a derivada à direita deste ponto é negativa e à esquerda é positiva e isto é utilizado pelo algoritmo.

$$\begin{cases} \frac{dP}{dV} = 0, & \text{no ponto de máxima potência} \\ \frac{dP}{dV} > 0, & \text{à esquerda do ponto} \\ \frac{dP}{dV} < 0, & \text{à direita do ponto} \end{cases}$$

Dessa maneira, a derivada da potência em relação a tensão pode ser reescrita conforme a Equação 2.13.

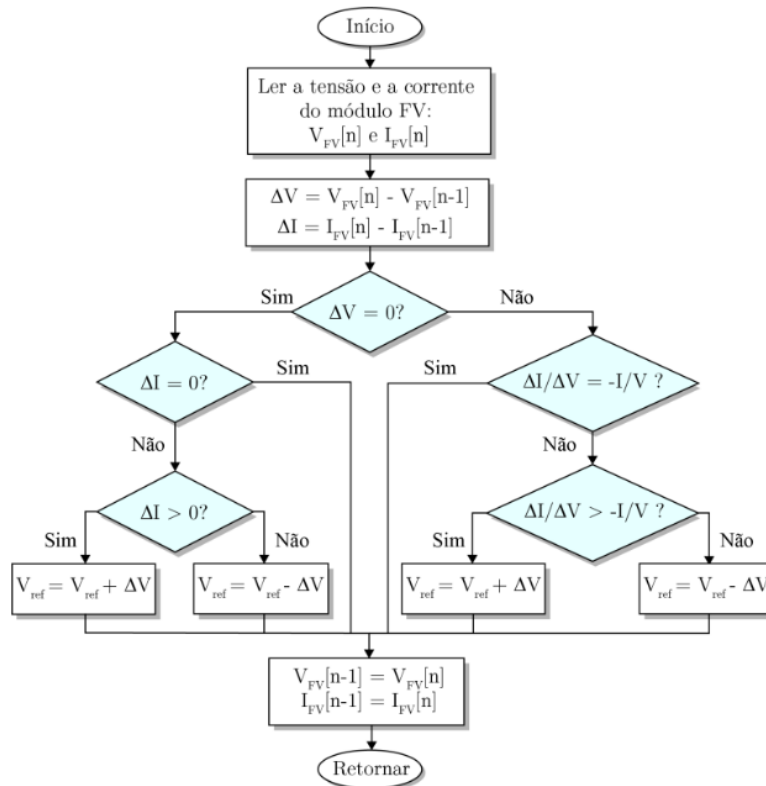
$$\frac{dP}{dV} = \frac{d(V \cdot I)}{dV} = I + V \cdot \left(\frac{dI}{dV}\right) \approx I + V \cdot \left(\frac{\Delta I}{\Delta V}\right) \quad (2.13)$$

Sendo assim, reescrevendo-se a relação em torno do ponto de máxima potência, tem-se:

$$\begin{cases} \frac{\Delta I}{\Delta V} = -\frac{I}{V}, & \text{no ponto de máxima potência} \\ \frac{\Delta I}{\Delta V} > -\frac{I}{V}, & \text{à esquerda do ponto} \\ \frac{\Delta I}{\Delta V} < -\frac{I}{V}, & \text{à direita do ponto} \end{cases}$$

O termo $\frac{\Delta I}{\Delta V}$ é denominado condutância incremental, enquanto o termo $\frac{I}{V}$ é denominado condutância instantânea. Sabendo disso, o algoritmo da Condutância Incremental funciona observando as variações de tensão e corrente em relação à iteração anterior e comparando as duas condutâncias mencionadas. O comportamento do algoritmo está descrito no fluxograma evidenciado na Figura 2.17.

Figura 2.17: Fluxograma que descreve o algoritmo da Condutância Incremental.



Fonte: *As 7 principais técnicas de MPPT* (MORAES, 2020).

Ademais, além da estabilidade que pode ser alcançada por esse método, comparando-o com o Perturba e Observa, nota-se uma vantagem para os casos em que há mudanças abruptas de irradiância e temperatura que modificam as curvas, visto que este reage melhor a estas situações.

2.5 MICROCONTROLADOR

Um microcontrolador (MCU), como o da Figura 2.18, é um circuito integrado projetado para realizar determinadas tarefas em um sistema embarcado. Este dispositivo tipicamente é composto por um processador, memórias voláteis e não voláteis, periféricos de entrada e de saída de dados e alguns outros componentes a depender da sua aplicação, como conversores A/D e D/A, interfaces de comunicação serial, dentre outros (IEEE, 2020).

Figura 2.18: Microcontrolador MSP430.



Fonte: *Texas Instruments*.

Sabendo disso, as aplicações nas quais os microcontroladores se encontram presentes são muito diversificadas. Estes dispositivos podem ser encontrados em veículos, eletrodomésticos, equipamentos médicos, máquinas de escritório, etc. No caso do objeto de estudo deste trabalho, os satélites, os microcontroladores também desempenham diversas funções importantes tanto no módulo de gerenciamento de energia, quanto em alguns outros, como o subsistema de controle e determinação de atitude e no computador de bordo.

Nesse contexto, um microcontrolador tem como base de seu funcionamento um programa de controle (*firmware*) gravado em sua memória não-volátil que implementa a função desempenhada pelo dispositivo, fazendo uso também da memória volátil para armazenar as variáveis do programa, bem como resultados intermediários. Ademais, utilizando-se os seus periféricos de entrada e saída, o microcontrolador faz uso dos dados de entrada, enviados por sensores ou outros dispositivos, e os processa com o intuito de desempenhar determinada tarefa e, se necessário, também envia informações pelos pinos de saída, fazendo uso de algum tipo de comunicação serial, como I2C, SPI e UART ou até mesmo gerando saídas que controlem outros dispositivos, como

uma modulação PWM.

Sendo assim, os principais componentes de um microcontrolador (IEEE, 2020) são :

- **Processador (CPU):** é a parte responsável por processar e implementar as instruções que caracterizam a função realizada pelo microcontrolador, executando operações aritméticas, lógicas e dando instruções aos demais componentes do dispositivo;
- **Memória:** é utilizada para armazenar os dados utilizados pelo processador para realizar às instruções para as quais foi programado. Para isso, o microcontrolador faz uso de dois tipos principais de memória:
 - **Memória de programa:** é a memória utilizada para armazenar o programa que contém as instruções executadas pelo processador. Por conta disso, se trata de uma memória não volátil, como a ROM, capaz de armazenar informações por um tempo sem a necessidade de uma fonte de alimentação;
 - **Memória de dados:** é a parte da memória que armazena os dados temporários do programa, como variáveis e resultados intermediários. Sendo assim, ao contrário da memória de programa, a de dados é uma memória volátil, como a RAM, que mantém os dados armazenados somente enquanto conectada a uma fonte de alimentação.
- **Periféricos de entrada e saída:** se trata da interface entre o processador e os circuitos externos. Assim, tais periféricos, também conhecidos como GPIO, são responsáveis por receber as informações nos pinos de entrada e enviar ao processador e também por transmitir os dados processados, enviando às instruções do processador aos dispositivos externos por meio dos pinos de saída.

2.6 COMUNICAÇÃO SERIAL

Uma das principais formas utilizada pelos dispositivos eletrônicos para trocarem informação é a comunicação serial. Neste tipo de comunicação, um bit é enviado de cada vez, de forma sequencial, sendo necessário algum tipo de sincronismo ou protocolos que organizem a comunicação, fazendo com que a mensagem enviada pelo transmissor seja possível de ser interpretada pelo receptor.

Ademais, por possuir apenas um bit enviado por vez, a comunicação serial quando comparada à paralela, que envia vários bits simultaneamente, é mais simples e econômica, visto que dispensa a utilização de vários canais de transmissão e recepção. Por conta disto, a comunicação serial é muito utilizada em dispositivos eletrônicos, como os microcontroladores.

Além disso, existem dois tipos básicos de comunicação serial, a comunicação síncrona e a assíncrona. A comunicação serial síncrona faz uso de um recurso de sincronismo, o *clock*, utilizado para que o transmissor e o receptor saibam o momento em que os dados serão enviados e

recebidos. Por outro lado, a assíncrona não faz uso desta ferramenta, porém para que a comunicação seja efetiva, o transmissor e o receptor utilizam a mesma taxa de transmissão (*baud rate*), sendo possível a previsão de quando a mensagem será enviada e recebida sem o uso de um sinal de sincronismo.

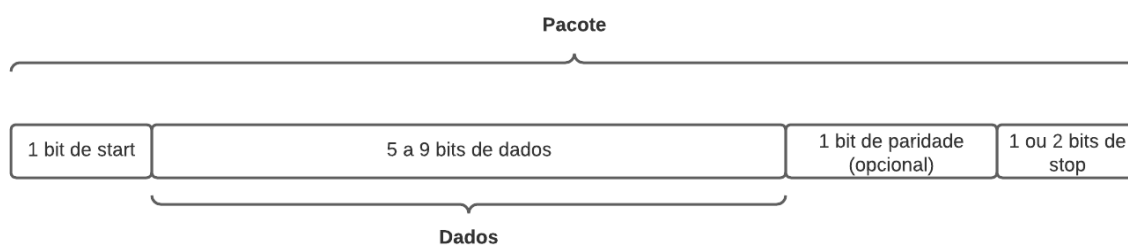
Além do mais, existem diversos tipos de comunicação serial que são utilizadas a depender da aplicação. No contexto dos microcontroladores, existem três tipos de comunicação que são mais utilizadas: UART, I2C e SPI.

2.6.1 *Universal Asynchronous Reception and Transmission (UART)*

A comunicação UART se trata de uma comunicação serial assíncrona, que não exige sincronismo do *clock* entre o receptor e o transmissor. Desse modo, esta utiliza duas linhas de dados, uma para transmitir (TX) e a outra para receber (RX), ou seja, o dispositivo que utiliza a comunicação UART envia os dados por uma linha e recebe por outra, o que faz com que somente seja possível que dois dispositivos se comuniquem. Além disso, como mencionado anteriormente, é fundamental que o transmissor e o receptor estejam com a mesma taxa de transmissão para que a comunicação seja efetiva (PEÑA; LEGASPI, 2020).

Ademais, na UART, a sequência de bits que são enviadas possuem uma estrutura, sendo o bit de *start* o primeiro a ser enviado, que indica o início da comunicação, seguido dos dados, do bit de paridade, utilizado para verificação da integridade dos dados, e do bit de *stop*, sinalizando o fim da mensagem. A estrutura do pacote da comunicação UART pode ser observada na Figura 2.19.

Figura 2.19: Estrutura do pacote da comunicação UART.



Fonte: Desenvolvida pelo próprio autor.

Desse modo, a comunicação UART se destaca por ser uma comunicação mais simples. No entanto, como mencionado, esta não permite que vários dispositivos se comuniquem simultaneamente, além de depender que a taxa de transmissão entre o receptor e o emissor seja a mesma.

2.6.2 Inter-Integrated Circuit (I2C)

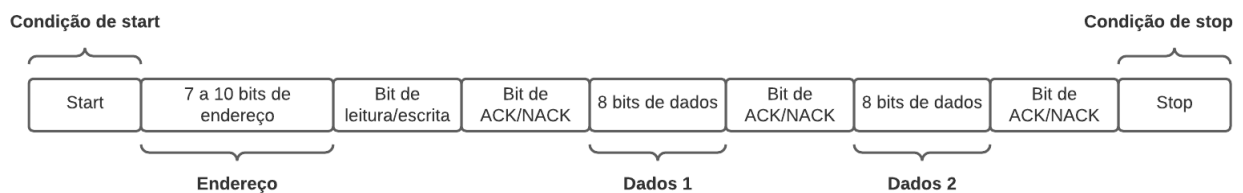
O I2C é um protocolo de comunicação serial síncrona que faz uso de dois fios para comunicação, o fio de dados (SDA) e o de clock (SCL). Diferentemente da UART, o I2C possibilita a comunicação entre mais de dois dispositivos pelo mesmo barramento de maneira bidirecional, sendo o dispositivo que controla o barramento I2C denominado de mestre e os que respondem ao mestre chamados de escravos. Sendo assim, tanto o mestre quanto os escravos assumem o papel de transmissor ou receptor em diferentes etapas da comunicação (LEENS, 2009).

Sabendo disso, para que a informação seja enviada, o mestre informa aos dispositivos escravos que a comunicação se iniciará, fazendo com que o SDA passe de nível lógico alto para baixo, enquanto o SCL está em nível alto. Analogamente, para finalizar a transmissão, o mestre passa o SDA de nível lógico baixo para alto, com o *clock* também em nível alto.

No I2C, o envio do dado é realizado conforme a velocidade do *clock*, assim para enviar um bit de dado, este deve permanecer estável entre a subida e a descida do *clock*. Ademais, é importante destacar que a comunicação é feita em *bytes*, ou seja, cada mensagem contém 8 bits de dados e ao término do envio desses bits, o transmissor libera a linha para que o receptor envie um sinal indicando que recebeu a mensagem, o *acknowledge* (ACK), levando a linha de dados para nível baixo. Se na nona batida do *clock*, o SDA permanecer em nível alto, significa que a mensagem não foi recebida com sucesso, essa sinalização é chamada de *not-acknowledge* (NACK).

Por fim, para se comunicar com mais de um dispositivo, o mestre envia antes de iniciar a comunicação um endereço, que pode ser de 7 ou 10 bits, que identifica o dispositivo para qual a mensagem está sendo enviada, permitindo a conexão de até 1024 dispositivos. A estrutura do pacote da comunicação I2C pode ser observada na Figura 2.20.

Figura 2.20: Estrutura do pacote da comunicação I2C.



Fonte: Desenvolvida pelo próprio autor.

2.6.3 Serial Peripheral Interface (SPI)

Assim como o I2C, o SPI é um protocolo de comunicação serial síncrona, porém ao invés de utilizar 2 fios, o SPI faz uso de 4 fios (LEENS, 2009):

- **SCLK:** sinal do *clock* gerado pelo dispositivo mestre;
- **MISO:** entrada de dados do mestre e saída de dados do escravo, no qual o mestre recebe os dados enviados pelos escravos;
- **MOSI:** entrada de dados do escravo e saída de dados do mestre, no qual o mestre envia os dados para os escravos;
- **SS:** linha utilizada para selecionar o escravo que se comunicará com o mestre.

Por possuir duas linhas de dados diferentes para enviar e receber dados, no protocolo SPI, o mestre e o escravo podem enviar e receber as mensagens simultaneamente. Desse modo, a comunicação funciona a partir de registradores de deslocamento, no qual são enviados bit por bit com referência no sincronismo do *clock*. Além do mais, por não possuir bits de *start* e *stop*, os dados podem ser enviados continuamente, sem limitação de tamanho, ao contrário do I2C em que a informação deve ser enviada *byte a byte*.

Ademais, o protocolo SPI não tem limitação de escravos, porém para cada escravo adicionado na comunicação é necessária a utilização de mais uma linha SS no mestre, o que aumenta o número de linhas de dados utilizadas por este. Entretanto, existe uma alternativa para esse problema, o uso de uma conexão em cadeia (*daisy chain*) dos escravos, conectando o MOSI de um escravo com o MISO do outro de forma que a mensagem que o mestre envia seja transmitida aos escravos com a utilização de somente um fio. No entanto, nesse caso, a comunicação acaba sendo mais lenta, visto que quanto mais escravos, mais demorado é para que a informação seja enviada do mestre ao escravo e do escravo ao mestre.

Por fim, dentre os tipos de comunicação apresentadas, o SPI é o que apresenta a maior taxa de transmissão de dados. Contudo, no SPI, ao contrário do I2C, é possível a presença de apenas um mestre e não há indicação de que as mensagens foram recebidas com sucesso pelo receptor.

2.7 SENSOR DE CORRENTE - INA3221

Em um circuito eletrônico, o monitoramento de corrente e tensão é fundamental para o seu bom funcionamento. Sabendo disso, são projetados circuitos integrados capazes de desempenhar tais funções. Dentre estes componentes de circuitos, pode-se destacar o INA3221 da *Texas Instruments*.

O INA3221 é um dispositivo utilizado para monitoramento de corrente elétrica, por meio da medida de queda de tensão em resistores *shunt*, além de ser capaz de realizar medições de tensões de alimentação do barramento. Para isso, este circuito integrado possui três canais com uma interface compatível com I2C e SMBUS e disponibiliza a configuração de tempos de conversão e o número de medidas feitas para realização do cálculo de média para redução de ruído. Além disso, este dispositivo é capaz de gerar alertas críticos e de advertência para o monitoramento de

condições programáveis para cada um dos canais (INSTRUMENTS, 2020a).

Sabendo disso, o INA3221 pode ser utilizado em circuitos com as mais diversas aplicações como, por exemplo, para estimar índices de estresse térmico em ambientes urbanos (SALGADO et al., 2019) ou para implementação do MPPT visando o fornecimento de máxima potência advindas de painéis solares (LOPEZ et al., 2016).

2.7.1 Configuração do endereço

Para comunicação com o INA3221 são disponibilizados quatro possibilidades de endereço configuráveis por meio do pino A0, a depender da conexão deste com os demais pinos. Os endereços possíveis encontram-se evidenciados na Tabela 2.1 obtida a partir do *datasheet* do dispositivo (INSTRUMENTS, 2020a).

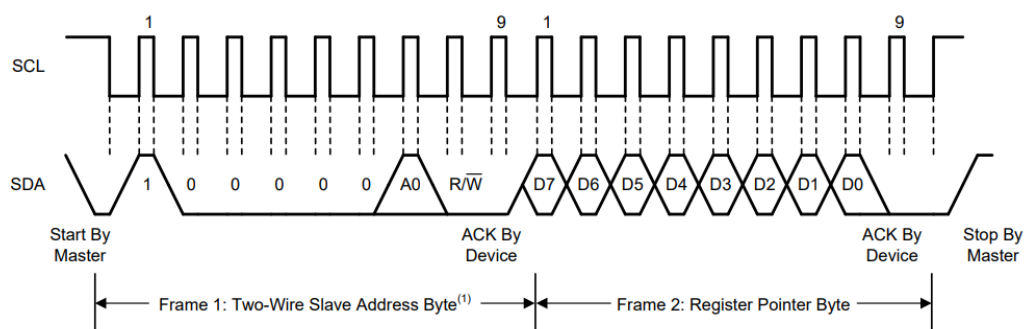
Tabela 2.1: Endereços configuráveis do INA3221.

| A0 | Endereço do escravo |
|-----|---------------------|
| GND | 1000000 |
| VS | 1000001 |
| SDA | 1000010 |
| SCL | 1000011 |

2.7.2 Comunicação com o dispositivo

Fazendo-se uso do protocolo I2C, o INA3221 opera como um escravo, o qual recebe comandos e envia respostas ao mestre. Sendo assim, para que a comunicação entre os dispositivos se inicie, o mestre deve iniciar com um *start*, seguido do endereço do dispositivo, do bit de $R\bar{W}$ em nível lógico baixo e de um *byte* correspondente ao ponteiro do registrador ao qual se deseja escrever ou ler. Na Figura 2.21, está evidenciada a estrutura da comunicação I2C para escolha do registrador do INA3221.

Figura 2.21: Diagrama da comunicação para escolha do registrador.



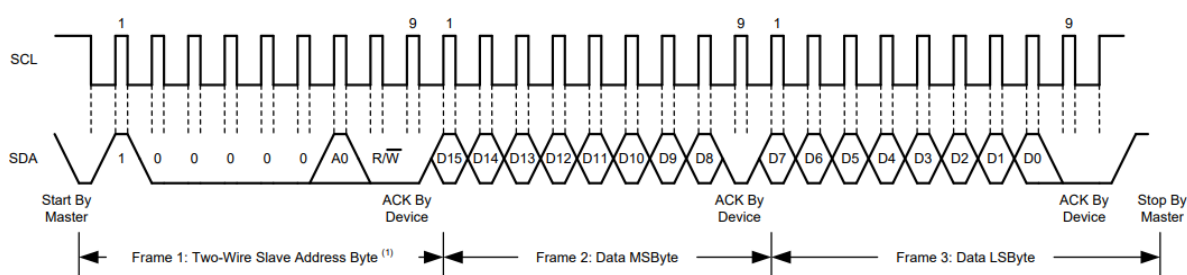
Fonte: *Datasheet* do INA3221 (INSTRUMENTS, 2020a).

Conforme a Figura 2.21, o *datasheet* recomenda que o envio do *byte* correspondente ao pon-

teiro seja seguido de um *stop*, porém durante o desenvolvimento do *firmware*, observou-se que para escrita em um registrador, os *bytes* a serem escritos devem ser enviados logo após o recebimento do *acknowledge*, sem o envio de um *stop*.

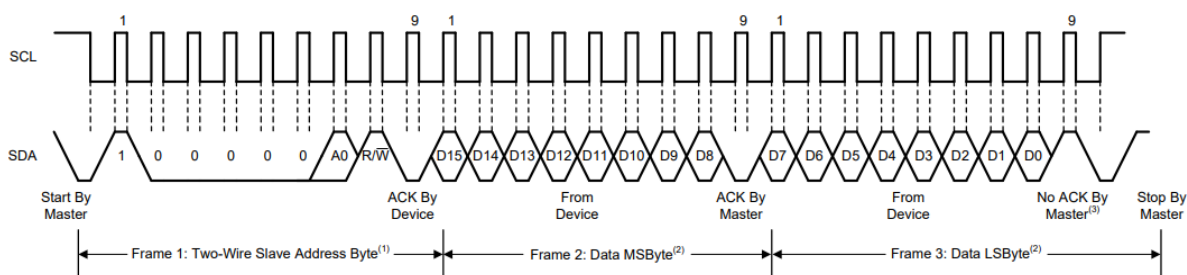
Sendo assim, após enviados os *bytes* anteriores, em caso de escrita, o mestre deve enviar mais dois *bytes* referentes ao dado que se deseja escrever no registrador e finalizar o envio com um *start* ou *stop*. Por outro lado, em caso de leitura, após enviado o *byte* referente ao registro que será lido, o mestre deve enviar um novo *start* com o bit de $R\bar{W}$ em nível lógico alto, aguardar o escravo responder com dois *bytes* de informação e finalizar a leitura com um *start* ou *stop*. Ademais, vale ressaltar que a comunicação com o INA3221 é feita com o primeiro *byte* sendo o mais significativo e o segundo, o menos significativo. Diante disso, na Figura 2.22, está evidenciada a comunicação para escrita no registrador, enquanto a Figura 2.23 descreve a comunicação para leitura.

Figura 2.22: Diagrama de comunicação para escrita.



Fonte: *Datasheet* do INA3221 (INSTRUMENTS, 2020a).

Figura 2.23: Diagrama de comunicação para leitura.



Fonte: *Datasheet* do INA3221 (INSTRUMENTS, 2020a).

Dentre os registradores disponibilizados pelo INA3221, os que serão utilizados no desenvolvimento do *firmware* são o registrador de configuração e os registradores que contêm as medições obtidas pelos três canais, tanto da tensão de barramento quanto da tensão nos resistores *shunt*. O endereço de cada registrador está contido na Tabela 2.2 com informações retiradas do seu *datasheet* (INSTRUMENTS, 2020a).

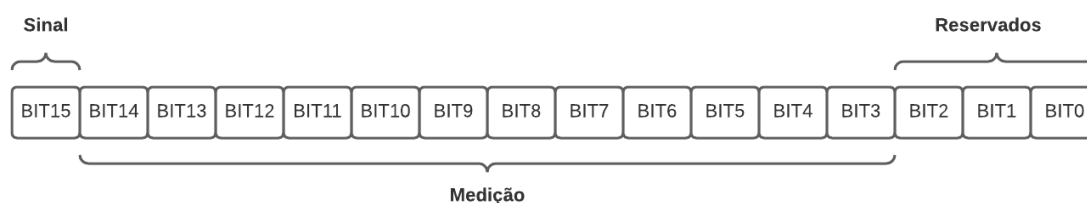
Tabela 2.2: Dados dos registradores do INA3221.

| Endereço do ponteiro (Hex) | Nome do registrador |
|----------------------------|--------------------------------|
| 0 | <i>Configuration</i> |
| 1 | <i>Channel-1 Shunt Voltage</i> |
| 2 | <i>Channel-1 Bus Voltage</i> |
| 3 | <i>Channel-2 Shunt Voltage</i> |
| 4 | <i>Channel-2 Bus Voltage</i> |
| 5 | <i>Channel-3 Shunt Voltage</i> |
| 6 | <i>Channel-3 Bus Voltage</i> |

2.7.3 Interpretação das leituras

Tanto o valor de queda de tensão lido em resistores *shunt* quanto as medições de tensões de alimentação do barramento são armazenados em registradores de 16 bits, nos quais o bit mais significativo representa o sinal da leitura, os três menos significativos estão reservados e não são utilizados e os demais correspondem ao valor da medição, conforme evidenciado na Figura 2.24.

Figura 2.24: Estrutura dos registradores de leitura do INA.



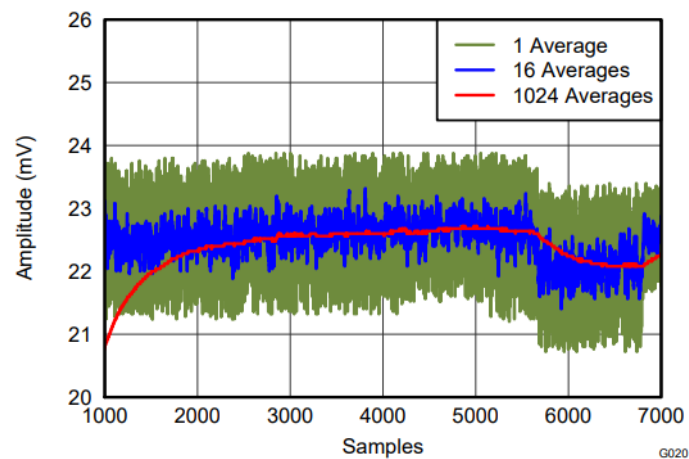
Fonte: Desenvolvida pelo próprio autor.

Além disso, vale destacar que para representação de números negativos é utilizada a notação em complemento de dois e que os bits menos significativos das leituras da tensão *shunt* e de barramento correspondem a $40 \mu V$ e $8 mV$, respectivamente. Sendo assim, os valores máximos representados pela notação utilizada pelo INA3221 são $163,8 mV$ para tensão *shunt* e $32,76 V$ para tensão de barramento.

2.7.4 Filtro de média

Tendo em vista que medições feitas por sensores estão susceptíveis a diversas fontes de ruído que podem comprometer a leitura realizada, o INA3221 disponibiliza um filtro de média. Este tipo de filtro consiste em realizar mais de uma leitura e, posteriormente, calcular a média dos valores obtidos. Na Figura 2.25, é possível se observar a influência do filtro de média no valor obtido pelo dispositivo, sendo utilizadas 1, 16 e 1024 medidas no cálculo da média.

Figura 2.25: Influência do filtro de média nas leituras.



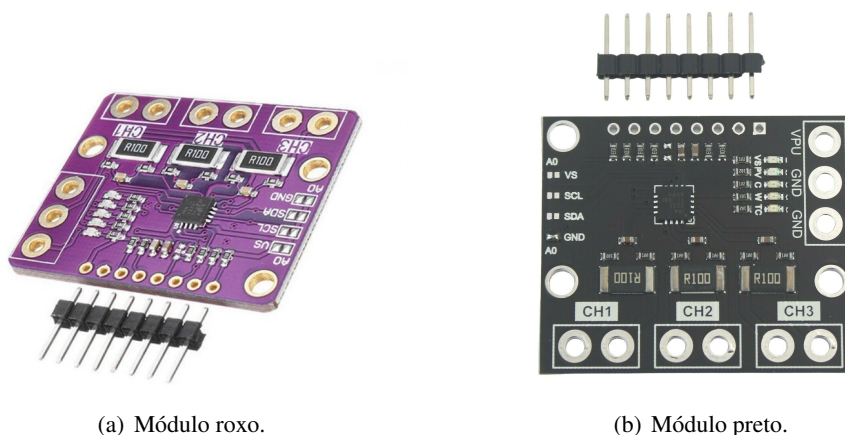
Fonte: *Datasheet* do INA3221 (INSTRUMENTS, 2020a).

Nesse contexto, a partir do registrador de configuração, é possível escolher a quantidade de amostras que serão utilizadas para o cálculo de média, bem como a escolha do tempo de conversão para as medidas. Tais parâmetros caracterizam o filtro e também possibilitam uma leitura mais precisa do sensor a depender do nível de ruído e da variação do valor lido.

2.7.5 Módulos com INA3221

Com o intuito de facilitar a aplicação de circuitos integrados, principalmente para prototipagem e realização de testes, são desenvolvidos módulos que contêm a *hardware* requisitado para utilização do CI, sem a necessidade da fabricação de uma placa de circuito impresso. Sabendo disso, no mercado, são encontrados dois módulos principais que utilizam o INA3221, um módulo roxo e um módulo preto evidenciados na Figura 2.26.

Figura 2.26: Módulos INA3221.

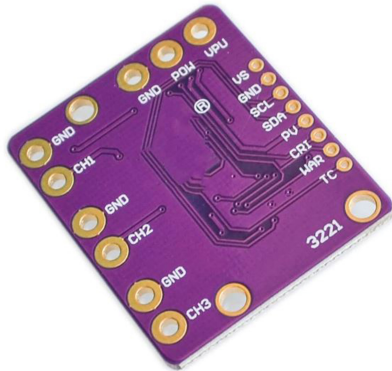


(a) Módulo roxo.

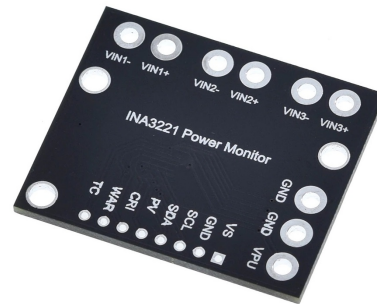
(b) Módulo preto.

Estes dois módulos apresentam uma diferença que vale a pena destacar. Os módulos pretos normalmente possuem canais conectados de maneira independente, sendo possível a realização de medidas de corrente que não se influenciam. Por outro lado, é comum se observar nos módulos roxos uma conexão série dos resistores *shunt*, o que faz com que as medições de corrente estejam correlacionadas. Além do mais, na Figura 2.27, é possível observar que os *pads* do módulo roxo disponíveis para leitura da corrente estão conectados a cada um dos canais e ao plano de terra, enquanto no módulo preto, os *pads* estão conectados somente aos canais.

Figura 2.27: Visão traseira dos módulos INA3221.



(a) Módulo roxo.



(b) Módulo preto.

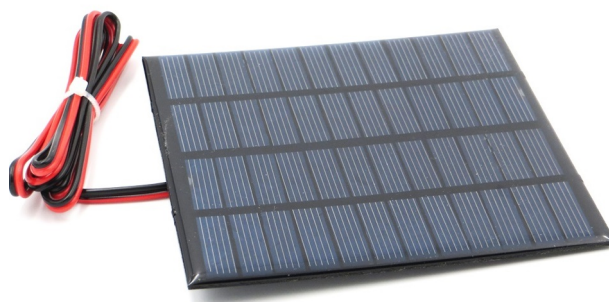
3 METODOLOGIA

3.1 SIMULAÇÃO DO CIRCUITO ANALISADO

Para iniciar o desenvolvimento do *firmware* embarcado para o módulo de gerenciamento de energia do nanossatélite é necessária a compreensão do circuito a ser monitorado por este. Dessa maneira, o circuito básico fundamental para análise do EPS trata-se de um painel solar, de um conversor CC-CC e de uma bateria, que podem ser modelados segundo o que foi apresentado na fundamentação teórica.

Sendo assim, para implementação do circuito a ser testado, será escolhido o modelo de painel solar CNC110x110-12 da Mazava, evidenciado na Figura 3.1, em decorrência da sua disponibilidade e viabilidade econômica para uso na validação do *firmware* a ser implementado. As suas especificações técnicas estão evidenciadas na Tabela 3.1.

Figura 3.1: Painel solar CNC110x110-12 da Mazava.



Fonte: Mazava.

Tabela 3.1: Especificações do painel solar CNC110x110-12 para temperatura de 25°C e irradiância de 1000 W/m²

| | |
|--|--------|
| Tensão de máxima potência (V_{mp}) | 12 V |
| Corrente de máxima potência (I_{mp}) | 150 mA |
| Tensão de circuito aberto (V_{oc}) | 14,6 V |
| Corrente de curto-circuito (I_{sc}) | 166 mA |
| Máxima potência | 1,8 W |

Além disso, o modelo que será adotado para simulação da placa solar trata-se de uma fonte de corrente em paralelo com um diodo e com uma resistência em série, visto que tal modelo é sufici-

ente para modelar o funcionamento do painel, conforme afirmam trabalhos anteriores (ARAÚJO; SOUSA; COSTA, 2020).

Ademais, para a aplicação de algoritmos que implementem o MPPT é necessária a conexão do conversor CC-CC com o painel solar, de forma a controlar a tensão em seus terminais e mantê-la em seu valor de máxima potência. Dessa forma, por possibilitar tanto o aumento quanto a redução da tensão sem inversão de fase, a topologia de conversor SEPIC se apresenta como a melhor escolha para a simulação.

Por fim, para a bateria, o modelo que será utilizado como referência trata-se da bateria íon-lítio NanoPower 18650 de 2600 mAh da *GOMSpace* (GOMSPACE, 2019), evidenciado na Figura 3.2. Sendo assim, para simulação, será adotado o modelo simplificado com a fonte de tensão em série com uma resistência conforme os parâmetros obtidos das especificações técnicas da bateria, descritas na Tabela 3.2.

Figura 3.2: Bateria NanoPower 18650 da *GOMSpace*.



Fonte: *NanoPower Battery 2600mAh* (GOMSPACE, 2019).

Tabela 3.2: Especificações da bateria NanoPower 18650 da *GOMSpace*

| | Min. | Tip. | Máx. | Unid. |
|---------------------------|------|------|------|-----------|
| Tensão | 2,75 | 3,6 | 4,2 | V |
| Impedância interna | - | - | 70 | $m\Omega$ |

3.2 TESTE DO *FIRMWARE*

3.2.1 Teste das funcionalidades

Conforme descrito pelo diagrama da Figura 1.4, o *firmware* desenvolvido deve desempenhar cinco tarefas principais:

1. Prover uma interface para configuração do EPS;
2. Obter e disponibilizar dados de telemetria;

3. Controlar o fornecimento de energia aos demais subsistemas;
4. Realizar o controle da temperatura da bateria;
5. Prover comunicação entre o EPS e o subsistema de controle.

Desse modo, para o desenvolvimento do *firmware* serão propostos testes específicos com intuito de validar cada funcionalidade de maneira individual. Para o teste da interface de configuração, a análise será feita com o auxílio de um *software*, o PuTTY, que permite a comunicação entre o microcontrolador e o computador, de modo a se avaliar os dados enviados a partir da comunicação UART e a configuração dos parâmetros do EPS.

Além disso, a análise dos dados de telemetria pode ser realizada por meio da medição de valores de corrente e tensão em circuitos simplificados, cujos valores das grandezas são facilmente obtidos de maneira teórica, sendo determinados os erros relativos de cada medição realizada. Ademais, o sensor de corrente INA3221 apresenta a possibilidade de utilização de um filtro de média, o que também será avaliado neste teste. Já o envio dos dados de telemetria pode ser verificado com a utilização de um microcontrolador auxiliar capaz de desempenhar o papel de mestre da comunicação I2C e realizar a leitura dos dados enviados pelo EPS.

Por outro lado, para teste do controle de temperatura da bateria, o sensor LM35 será exposto a diferentes cenários, com temperaturas distintas, sendo feita a comparação entre a medição realizada por este dispositivo e a obtida com o auxílio de um termômetro de referência, determinando-se também o erro associado a cada medida. Por fim, para teste de acionamento da malha de aquecimento e também do fornecimento de energia aos subsistemas, será feito o uso de um multímetro digital para verificação da tensão dos pinos digitais.

3.3 TESTE DO CIRCUITO COMPLETO

Para validação do *firmware* desenvolvido neste trabalho de maneira completa, é necessária a utilização de um *hardware* externo ao microcontrolador, composto por alguns dispositivos, dentre estes, os elementos utilizados na simulação: o painel solar e a bateria.

Nesse contexto, um dos principais papéis desempenhados pelo EPS é a implementação do algoritmo MPPT que pode ser realizada tanto via *hardware*, com uso de circuitos integrados dedicados a tal função, quanto via *firmware*, utilizando-se sensores para leitura de tensão e corrente e controlando-se o *duty cycle* de conversores CC-CC.

A primeira alternativa, com uso de um CI dedicado ao MPPT, apresenta vantagens em relação a implementação via *firmware* com uso de componentes discretos na composição do conversor CC-CC, como o evidenciado na Figura 2.11. Dentre estas vantagens, pode-se destacar a facilidade de implementação e a economia de recursos por parte do microcontrolador utilizado.

Diante disso, para composição do *hardware* para validação do trabalho, serão utilizados o

painel solar de 12 V, a bateria de 4,2 V, um módulo INA3221 e um CI responsável não só pela implementação do MPPT, mas também pelo gerenciamento dos ciclos de carregamento e descarregamento da bateria, o CN3791 (ELECTRONIC,) da *Consonance Electronics*. Além disso, vale destacar que para composição do circuito de teste, optou-se pela utilização da bateria 18650 da *HU*, tendo em vista a sua disponibilidade por parte do laboratório, esta que possui características similares ao modelo da *GOMSpace*, porém sua capacidade nominal é mais elevada, de 9800 mAh, mas que não altera os resultados da simulação.

Sendo assim, para validação do *firmware* desenvolvido será monitorado o funcionamento do circuito completo enquanto é avaliada a técnica de rastreamento do ponto de máxima potência implementada pelo CN3791.

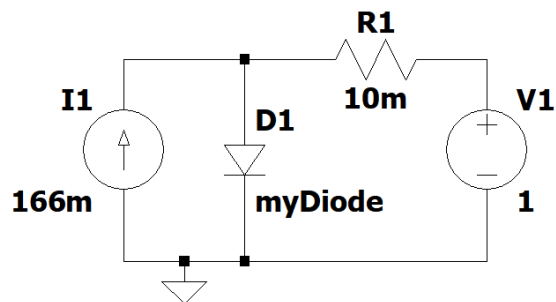
4 RESULTADOS

4.1 CARACTERIZAÇÃO DO CIRCUITO

4.1.1 PAINEL SOLAR

Conforme mencionado no capítulo de metodologia, antes do início do desenvolvimento do *firmware*, é importante a caracterização do *hardware* fundamental do subsistema de gerenciamento de energia. Com este intuito, inicialmente, foi realizada a simulação do circuito do painel solar evidenciado na Figura 4.1 através do simulador LTspice, obtendo-se a curva da Figura 4.2 como resultado. Para caracterização dos componentes do circuito, foram utilizadas as especificações do módulo fotovoltaico CNC110x110-12 descritas na Tabela 3.1.

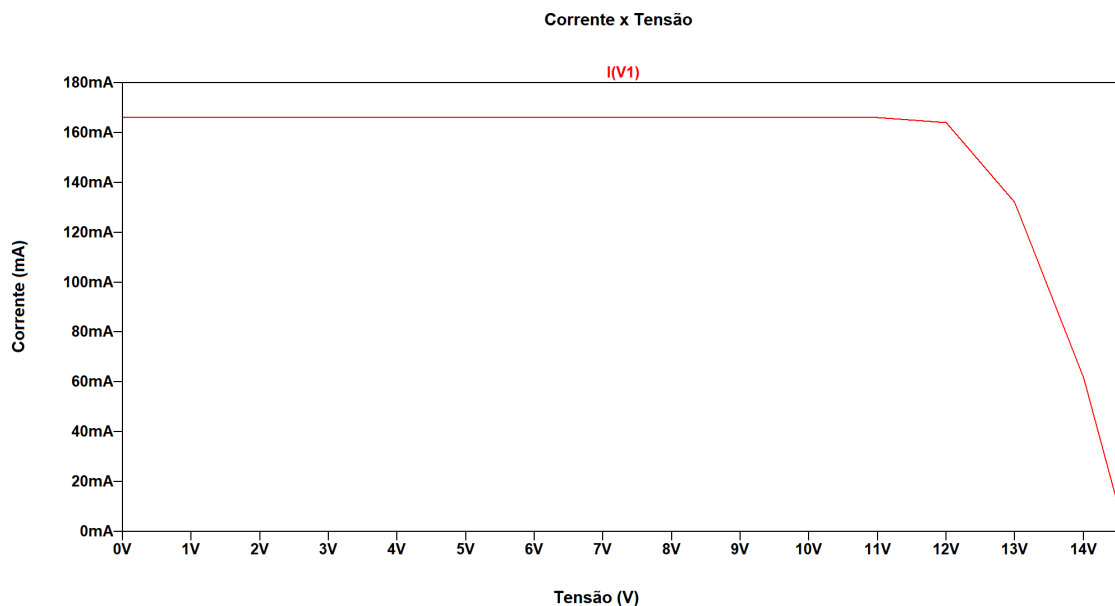
Figura 4.1: Circuito do painel solar.



```
.model myDiode D(Epsilon=258 Vfwd=11.68 Ron=.1 Roff=99Meg)  
.dc V1 0 14.6 1
```

Fonte: Desenvolvida pelo próprio autor.

Figura 4.2: Curva da corrente em função da tensão do painel solar.



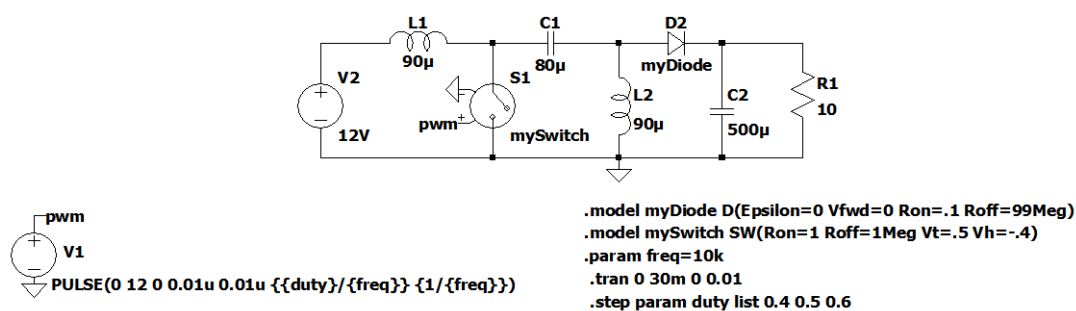
Fonte: Desenvolvida pelo próprio autor.

A partir desta simulação, é averiguado que o modelo composto pela fonte de corrente em paralelo com um diodo e com uma resistência em série é capaz de representar o comportamento do painel, visto que a tensão e a corrente do ponto de máxima potência se encontram de acordo com os dados presentes nas especificações da Tabela 3.1. Tais informações são importantes para validação de que o *hardware* utilizado para teste do *firmware* está em correto funcionamento.

4.1.2 Conversor CC-CC

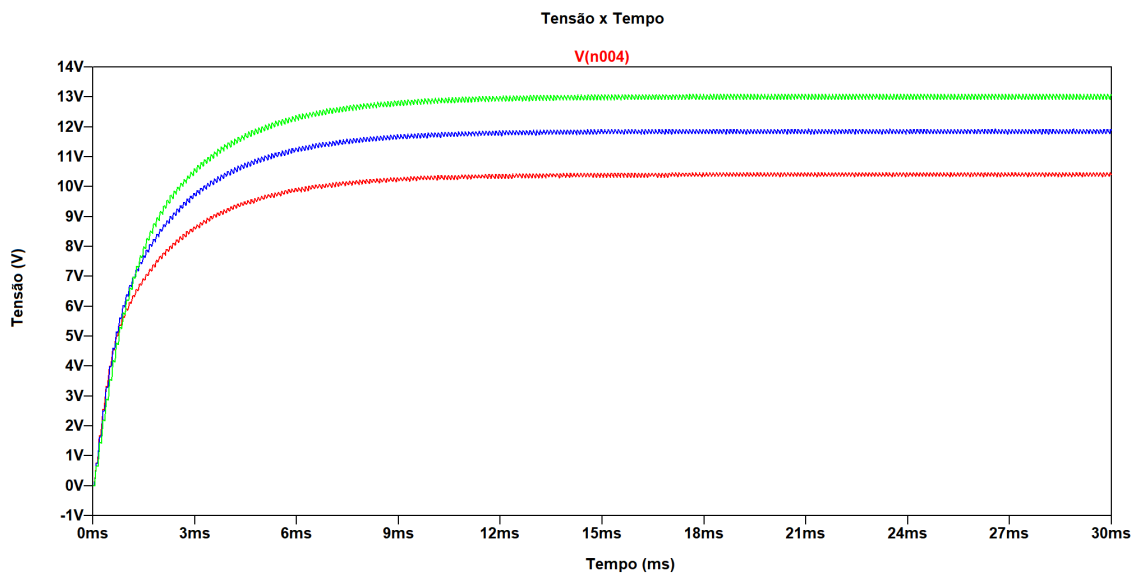
Além da simulação do painel solar, é interessante o estudo do conversor CC-CC, tendo em vista que, para aplicação de técnicas MPPT, este é um elemento primordial. Sendo assim, foi escolhida a topologia SEPIC para compor o circuito simulado, já que neste tipo de conversor não há inversão de fase da tensão e é possível tanto o aumento quanto a redução desta. O circuito simulado no LTspice encontra-se descrito na Figura 4.3 e a curva obtida como resultado da simulação está evidenciada na Figura 4.4.

Figura 4.3: Circuito do conversor CC-CC.



Fonte: Desenvolvida pelo próprio autor.

Figura 4.4: Saída do conversor CC-CC para diferentes valores de *duty cycle*.



Fonte: Desenvolvida pelo próprio autor.

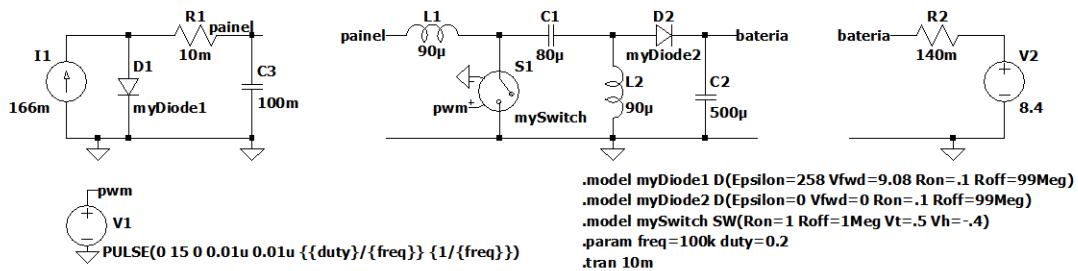
Na Figura 4.4, são utilizados três valores distintos de *duty cycle*, de 40% (vermelho), de 50% (azul) e de 60% (verde). Sendo assim, é possível observar que a tensão de entrada é a mesma da saída para o caso em que D é igual a 50%, enquanto que para 40%, a tensão de saída é inferior e para 60%, é superior, conforme descrito pela Equação 2.12.

Por outro lado, a partir da simulação do circuito com uso de componentes discretos na composição do conversor CC-CC, notou-se que seria mais vantajosa a utilização de um circuito integrado ao invés de se utilizar tais componentes, tendo em vista a maior facilidade de implementação.

4.1.3 Circuito completo

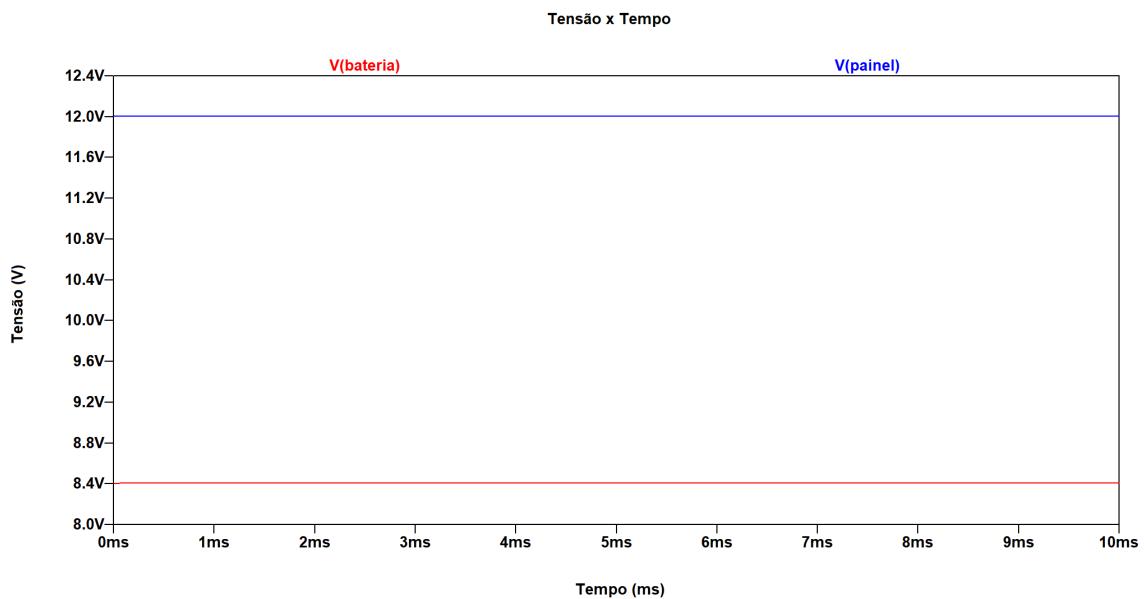
Por fim, para finalizar a caracterização do circuito com o intuito de melhor compreensão do cenário analisado antes da elaboração do *firmware*, realizou-se uma simulação com o circuito completo, conectando-se o circuito da Figura 4.1 e o da Figura 4.3 com o modelo de bateria constituído por uma fonte de tensão em série com a sua resistência interna.

Figura 4.5: Circuito completo.



Fonte: Desenvolvida pelo próprio autor.

Figura 4.6: Tensões do painel solar e da bateria no circuito completo.



Fonte: Desenvolvida pelo próprio autor.

Dessa maneira, obteve-se o resultado descrito na Figura 4.6 para a bateria com tensão de 8,4 V, que se trata do valor máximo correspondente a associação em série de duas baterias de 4,2 V. Dessa maneira, observa-se que a tensão do painel se manteve no valor de máxima potência, em 12 V, o que demonstra que as técnicas de MPPT podem ser implementadas ao se variar o *duty*

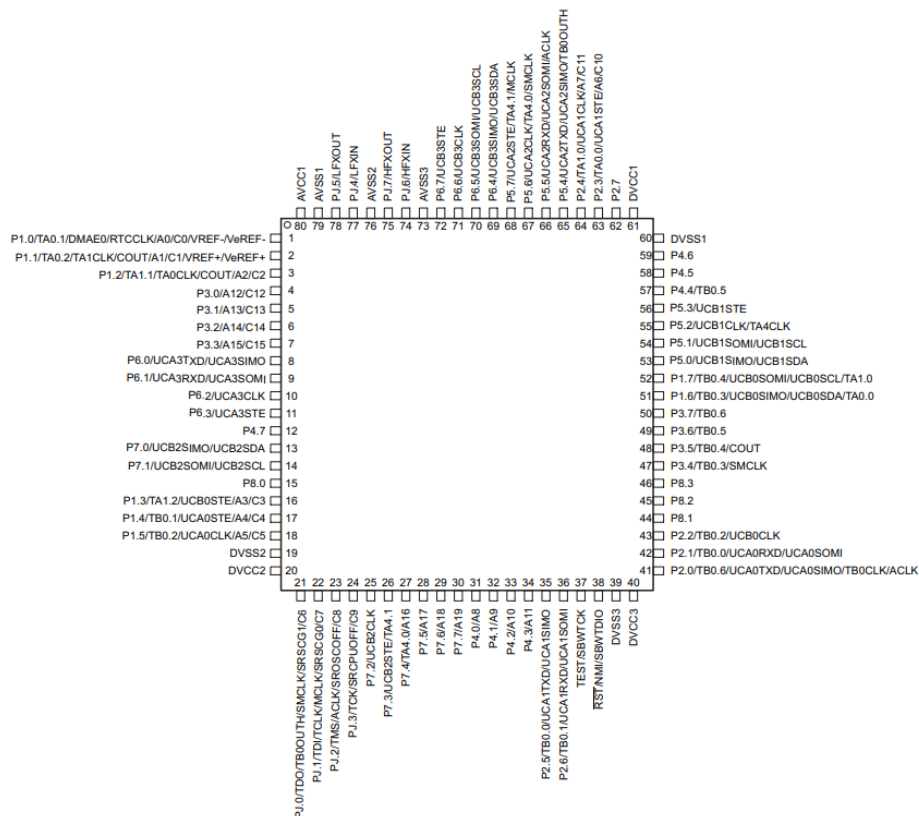
cycle do conversor, maximizando a potência fornecida pelo painel solar.

Assim sendo, por meio das simulações realizadas, é feita a caracterização do circuito do módulo de gerenciamento de energia do satélite. Contudo, como exposto anteriormente, para composição do *hardware* de teste, optou-se pela substituição do conversor CC-CC com componentes discretos por um circuito integrado dedicado à implementação do MPPT, bem como do gerenciamento dos ciclos de carregamento e descarregamento da bateria.

4.2 ESCOLHA DO MICROCONTROLADOR

Com base na arquitetura proposta, evidenciada na Figura 1.4, é necessária a escolha de um microcontrolador capaz de desempenhar as tarefas pré-estabelecidas, disponibilizando interfaces dedicadas para exercer tais funções. Diante disto, o microcontrolador escolhido para o projeto do *firmware* do subsistema de gerenciamento de energia é o MSP430FR5994 da *Texas Instruments* (INSTRUMENTS, 2021). Tal escolha se dá devido a este dispositivo fornecer todas as interfaces requisitadas para o desempenho das tarefas determinadas, além do fato de haver uma experiência prévia no desenvolvimento de *firmwares* para este modelo. Ademais, vale ressaltar que este microcontrolador possui memória ferro-elétrica (FRAM), um tipo de memória mais resistente a interferências que possam ocorrer no espaço e que possui um consumo de energia menor do que as memórias não voláteis convencionais. Por fim, este dispositivo possui funcionalidades equivalentes ao MSP430F5529, que é o utilizado nos laboratórios da UnB, o que também facilita o prosseguimento deste trabalho com a utilização do *firmware* desenvolvido em projetos futuros. Na Figura 4.7, encontra-se evidenciada a pinagem do MSP430FR5994.

Figura 4.7: Pinagem do MSP430FR5994.



Fonte: *MSP430FR5994 LaunchPad™ Development Kit* (INSTRUMENTS, 2022).

Sendo assim, com base nos pinos evidenciados na Figura 4.7 e nas interfaces destacadas na Figura 1.4, pode-se realizar o mapeamento dos pinos utilizados descrito na Tabela 4.1.

Tabela 4.1: Pinos do microcontrolador utilizados pelo firmware

| Pinos do microcontrolador | Funcionalidade do pino | Função desempenhada |
|---------------------------|------------------------|---|
| P1.3 | GPIO | Controlar o fornecimento de energia ao aquecedor da bateria |
| P2.0 | UART-TX | Enviar dados ao computador da estação de solo |
| P2.1 | UART-RX | Receber dados do computador da estação de solo |
| P3.0 | ADC | Ler a temperatura da bateria |
| P3.1 | ADC | Ler a tensão do painel |
| P3.2 | ADC | Ler a tensão da bateria |
| P3.3 | ADC | Ler a tensão da carga (subsistemas) |
| P3.4 | GPIO | Controlar o fornecimento de energia aos subsistemas |
| P3.5 | GPIO | Controlar o fornecimento de energia aos subsistemas |
| P3.6 | GPIO | Controlar o fornecimento de energia aos subsistemas |
| P3.7 | GPIO | Controlar o fornecimento de energia aos subsistemas |
| P5.0 | I2C-SDA | Enviar e receber dados do sensor de corrente - INA3221 |
| P5.1 | I2C-SCL | Clock da comunicação com o sensor de corrente - INA3221 |
| P7.0 | I2C-SDA | Enviar e receber dados do computador de bordo |
| P7.1 | I2C-SCL | Clock da comunicação com o computador de bordo |

4.3 DESENVOLVIMENTO DO *FIRMWARE*

Para desenvolvimento do *firmware* embarcado do EPS foi utilizada a linguagem C na programação do microcontrolador MSP430FR5994 por meio do ambiente de desenvolvimento fornecido pela empresa fabricante, o *Code Composer Studio*.

Tendo em vista que o EPS necessita de realizar diversas tarefas de maneira simultânea, foi escolhida a construção do programa fazendo-se uso de máquinas de estado (WAGNER, 2006). Este tipo de implementação é interessante, visto que permite que as tarefas a serem executadas sejam divididas em subtarefas, o que simula a execução dos processos de maneira paralela, apesar do microcontrolador executar as instruções de forma sequencial, já que o *clock* do sistema é extremamente rápido.

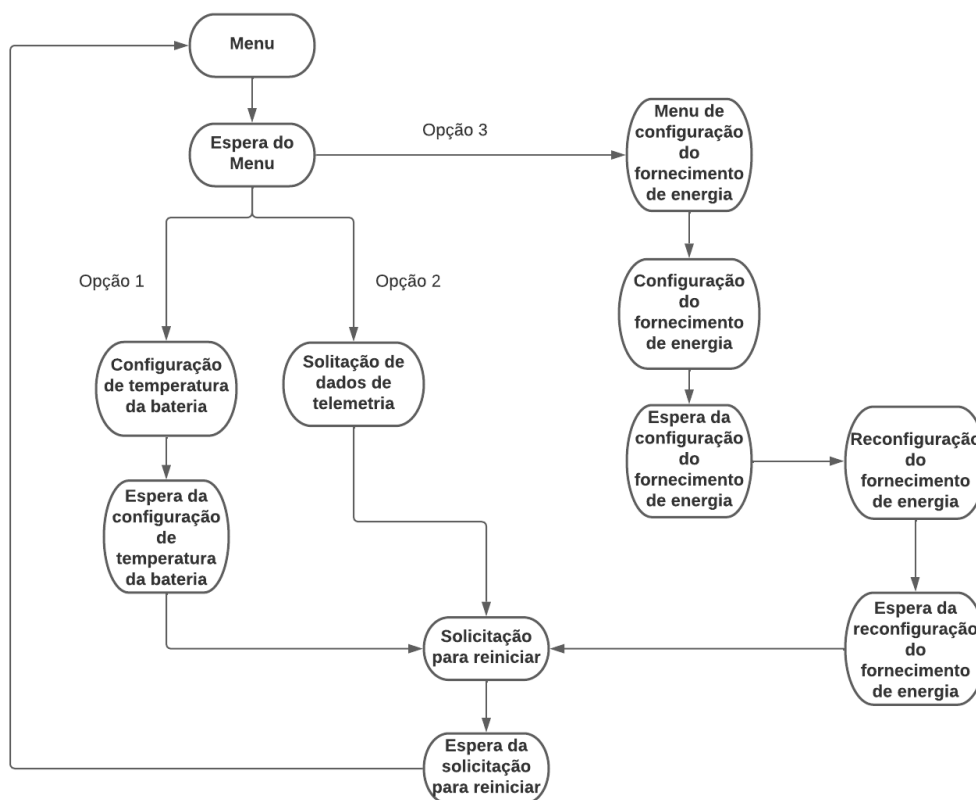
Além disso, outro recurso importante para a elaboração do *firmware* multitarefas é o uso de rotinas de interrupção, disponibilizadas pelo microcontrolador adotado. Tal ferramenta possibilita que o programa não permaneça interrompido em travas de execução enquanto aguarda o acontecimento de algum evento, como é o caso da técnica de *pooling*, mas sim, permite que determinadas tarefas sejam executadas a partir de eventos de *hardware*, o que contribui na simulação do paralelismo de execução do programa.

4.3.1 Interface de configuração

A fim de criar a interface de configuração do EPS, adotou-se o tipo de comunicação assíncrona por meio da interface UART dedicada do microcontrolador configurada para uma taxa de comunicação de 9600 bps. Assim sendo, por meio de tal recurso dedicado tanto o envio quanto o recebimento de dados foram configurados para serem realizados a partir de interrupções, com a escrita no *buffer* de envio e leitura do *buffer* de resposta, respectivamente.

Além disso, a lógica computacional da comunicação do EPS com o usuário foi construída com o uso de uma máquina de estados. Nesta parte do *firmware*, as máquinas possibilitam que este não fique interrompido enquanto as respostas do usuário estão sendo aguardadas, com estados que esperam tal interação e verificam se ocorreram para prosseguimento da máquina. Os estados da interface de configuração encontram-se evidenciados no diagrama da Figura 4.8.

Figura 4.8: Diagrama da máquina de estados da interface de configuração.

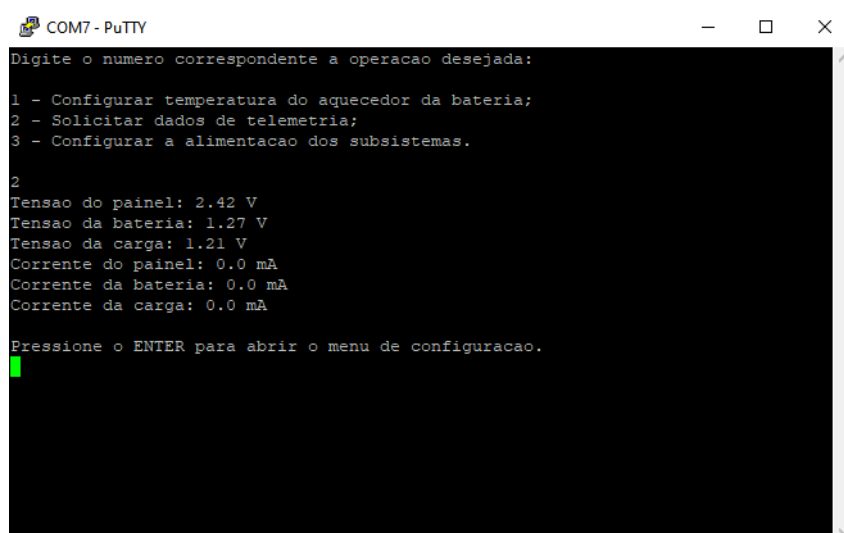


Fonte: Desenvolvida pelo próprio autor.

A partir do diagrama, nota-se que foram definidas três funcionalidades principais com base nas configurações necessárias na estação de solo, antes do envio do satélite para a órbita terrestre, sendo estas: configuração da temperatura de operação da bateria, solicitação dos dados de telemetria e configuração do fornecimento de energia aos subsistemas do satélite. Vale destacar que para inserção de novas configurações em tal interface, basta que sejam acrescentados novos estados na máquina.

Por fim, para validação desta etapa do *firmware*, foi utilizado um *software* de emulação de terminal em um computador, o PuTTY, de modo a estabelecer uma comunicação serial com o microcontrolador, com o intuito de se realizar a leitura do menu de configuração da interface e enviar os comandos para a escolha dos parâmetros configuráveis. A interface de configuração pode ser observada na Figura 4.9.

Figura 4.9: Interface de configuração do EPS.



```
COM7 - PuTTY
Digite o numero correspondente a operacao desejada:
1 - Configurar temperatura do aquecedor da bateria;
2 - Solicitar dados de telemetria;
3 - Configurar a alimentacao dos subsistemas.
2
Tensao do painel: 2.42 V
Tensao da bateria: 1.27 V
Tensao da carga: 1.21 V
Corrente do painel: 0.0 mA
Corrente da bateria: 0.0 mA
Corrente da carga: 0.0 mA
Pressione o ENTER para abrir o menu de configuracao.
```

Fonte: Desenvolvida pelo próprio autor.

4.3.2 Dados de telemetria

Para supervisão do funcionamento adequado do módulo de gerenciamento de energia, é preciso que sejam realizadas as medidas de corrente e tensão nos três componentes principais do circuito: o painel solar, a bateria e a carga alimentada, isto é, os subsistemas.

Diante disso, com o objetivo de medir a tensão, fez-se uso de conversores A/D disponibilizados pelo microcontrolador, sem necessidade do acréscimo de outro circuito integrado que desempenhasse este papel. No entanto, conforme a *datasheet* do dispositivo (INSTRUMENTS, 2021), os pinos suportam tensões de até 4,1 V, logo, o uso de divisores resistivos se faz necessário, visto que a tensão do painel pode ser superior a 14 V, enquanto a da bateria chega a 4,2 V.

Por outro lado, para medição de corrente, foi necessária a utilização de um CI dedicado para esta função, o INA3221. Como este dispositivo se comunica por meio do protocolo I2C, foi feito o uso da interface I2C do MSP430, configurada como mestre. Além disso, o INA3221 oferece endereços configuráveis para a comunicação a depender da conexão realizada no seu pino de endereçamento, assim, o pino A0 do CI foi conectado ao GND, definindo o seu endereço como 0x40, como descrito na Tabela 2.1.

Para validação desta parte do *firmware* e dos sensores utilizados, realizaram-se medidas de tensão e corrente em três cenários de teste distintos:

- **Cenário A:** Fonte de tensão CC de 3,3 V em série com um resistor de 1 k Ω ;
- **Cenário B:** Fonte de tensão CC de 3,3 V em série com dois resistores de 1 k Ω ;
- **Cenário C:** Fonte de tensão CC de 5 V em série com dois resistores de 1 k Ω .

Para os cenários com dois resistores em série, foi realizada a medição da queda de tensão em um deles. Além disso, para avaliar o efeito do filtro de média aplicado pelo INA3221 nas medidas, foram feitas medições configurando-o para realizar somente 1 medida (sem filtro) e 64 medidas (com filtro). Os resultados dos testes encontram-se na Tabela 4.2.

Tabela 4.2: Resultado do teste das medições

| | Cenário A | Cenário B | Cenário C |
|----------------------------|-----------|-----------|-----------|
| Tensão | 3,29 V | 1,64 V | 2,52 V |
| Corrente sem filtro | 3,2 mA | 1,60 mA | 2,39 mA |
| Corrente com filtro | 3,2 mA | 1,60 mA | 2,40 mA |

A partir das medições, observa-se que os seus valores estão em conformidade com os teóricos, com pequenos erros relativos evidenciados na Tabela 4.3. No entanto, vale ressaltar que era esperado uma diferença considerável entre as medições de corrente com e sem a utilização do filtro de média, conforme o *datasheet* do INA3221 (INSTRUMENTS, 2020a), porém, na prática, as leituras realizadas se apresentaram como valores muito próximos.

Tabela 4.3: Erro relativo das medições

| | Cenário A | Cenário B | Cenário C |
|----------------------------|-----------|-----------|-----------|
| Tensão | 0,30 % | 0,60 % | 0,80 % |
| Corrente sem filtro | 3,03 % | 3,03 % | 4,40 % |
| Corrente com filtro | 3,03 % | 3,03 % | 4,00 % |

4.3.3 Comunicação com o computador de bordo

Os dados de telemetria obtidos pelos sensores devem ser disponibilizados para consulta na estação terrestre. Entretanto, a comunicação entre o EPS e a estação não é feita diretamente quando o satélite está em órbita, sendo necessário que o *firmware* estabeleça uma comunicação entre o EPS e o computador de bordo, responsável por se comunicar também com o subsistema de comunicação, para envio de informações do subsistema de gerenciamento de energia à estação de solo.

Com esse objetivo, fez-se uso de uma interface I2C do microcontrolador como escravo, que pudesse enviar e receber informações do computador de bordo. Para isso, foi configurada uma comunicação em que o subsistema de controle deve enviar um *byte* ao EPS indicando a operação desejada e, em seguida, solicitar a leitura de dois *bytes* para recebimento dos dados de telemetria obtido pelos sensores. As operações definidas no *firmware* encontram-se na Tabela 4.4.

Tabela 4.4: Instruções da comunicação entre EPS e computador de bordo

| Código da instrução (Hex) | Instrução |
|---------------------------|---------------------------|
| 0x00 | Ler a tensão do painel |
| 0x01 | Ler a tensão da bateria |
| 0x02 | Ler a tensão da carga |
| 0x03 | Ler a corrente do painel |
| 0x04 | Ler a corrente da bateria |
| 0x05 | Ler a corrente da carga |

Além disso, vale destacar que para envio das medidas realizadas pelo EPS para o computador de bordo, o valor da leitura que é tratado como *float* de 32 bits pelo *firmware* do EPS é convertido para um valor inteiro sem sinal de 16 bits antes de seu envio. Essa conversão é feita multiplicando-se o valor da medida por 100, com intuito de transformar o número decimal em um número inteiro, e interpretando o dado como um inteiro sem sinal de 16 bits. Ademais, para representação de leituras negativas, é utilizada a notação do complemento de dois, de forma que estes números também possam ser representados com 2 *bytes*.

Sendo assim, para validação desta etapa do *firmware*, foi utilizado um microcontrolador auxiliar para desempenhar o papel de mestre da comunicação, um Arduino Uno (ARDUINO, 2023). Dessa maneira, este dispositivo foi programado para enviar um *byte* ao MSP430 e depois solicitar a leitura dos dados, realizando a interpretação dos dados após o recebimento, convertendo-os novamente para *float* e, dessa forma, foi observado que os dados recebidos pelo mestre eram os mesmos lidos pelos sensores, validando esta funcionalidade do *firmware* embarcado.

4.3.4 Controle de temperatura da bateria

O primeiro passo para controle da temperatura da bateria é medir o seu valor com a utilização de um sensor. Para esse propósito, foi utilizado o LM35, cuja saída é analógica, o que faz com que seja necessária a utilização de um conversor A/D do microcontrolador para realização da leitura.

Sabendo disso, assim que o valor de temperatura é obtido, é feita uma comparação entre tal resultado e a temperatura de operação configurada via interface serial. Assim sendo, caso a leitura esteja abaixo do parâmetro configurado, é acionada a saída digital que controla a operação dos aquecedores, como o da Figura 1.3, para que a temperatura fique dentro da faixa adequada de operação da bateria.

Para validação desta parte do *firmware*, foram montados três cenários de teste para se obter temperaturas distintas. São estes:

- **Cenário A:** Medição de temperatura em um recipiente com água gelada;
- **Cenário B:** Medição de temperatura em um recipiente com água à temperatura ambiente;
- **Cenário C:** Medição de temperatura em um recipiente com água quente.

Em cada um dos testes, a temperatura de operação da bateria foi ajustada para um limiar próximo do valor a ser medido de modo que foi observado o acionamento da saída digital. Os valores de temperatura obtidos pelo sensor encontram-se na Tabela 4.5, assim como o valor obtido por um termômetro auxiliar e o erro relativo entre as medidas para comparação.

Tabela 4.5: Resultado do teste de medições de temperatura

| | Cenário A | Cenário B | Cenário C |
|---------------------------------|------------------|------------------|------------------|
| Temperatura - LM35 | 5,7 °C | 26,5 °C | 69,0 °C |
| Temperatura - Termômetro | 6,0 °C | 27,0 °C | 70,0 °C |
| Erro relativo | 5,0 % | 1,85 % | 1,43 % |

4.3.5 Fornecimento de energia aos subsistemas

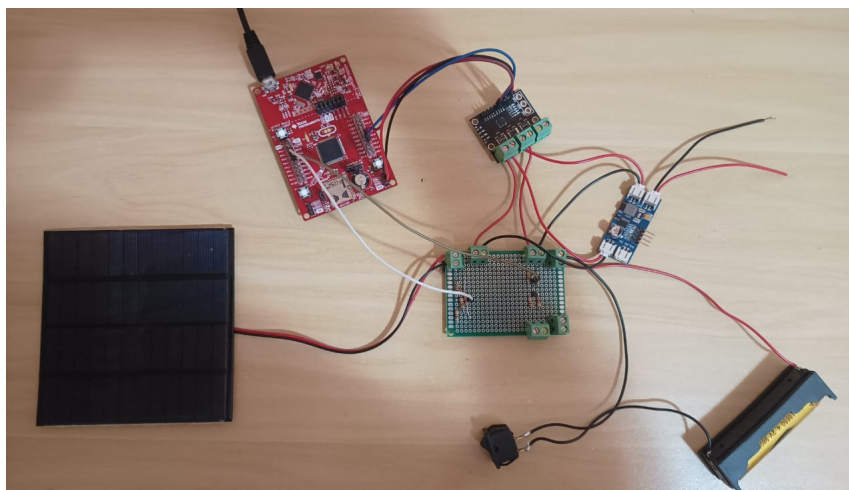
Por fim, para finalizar o desenvolvimento do *firmware* embarcado do EPS é importante a implementação do gerenciamento da alimentação dos demais subsistemas do satélite. Assim sendo, utilizou-se o circuito integrado TPS22976 como referência, logo, foram configuradas saídas digitais do microcontrolador para realização do controle do abastecimento de energia, visto que o TPS22976 opera como uma matriz de chaves acionadas por pinos digitais.

Dessa maneira, para validação desta funcionalidades, foram escolhidos, a partir da interface serial, quais seriam os subsistemas a serem alimentados e, com um auxílio de um multímetro, verificou-se a tensão das saídas digitais do microcontrolador correspondentes a cada subsistema, comparando-se com a configuração previamente realizada.

4.4 TESTE DO CIRCUITO COMPLETO

Com a elaboração do *firmware* concluída, o próximo passo é a montagem do circuito completo com o painel solar, a bateria, o sensor de corrente (INA3221) e o circuito integrado dedicado às funções de implementação do MPPT e gerenciamento da bateria (CN3791). O circuito montado encontra-se evidenciado na Figura 4.10.

Figura 4.10: Montagem do circuito de validação do *firmware*.

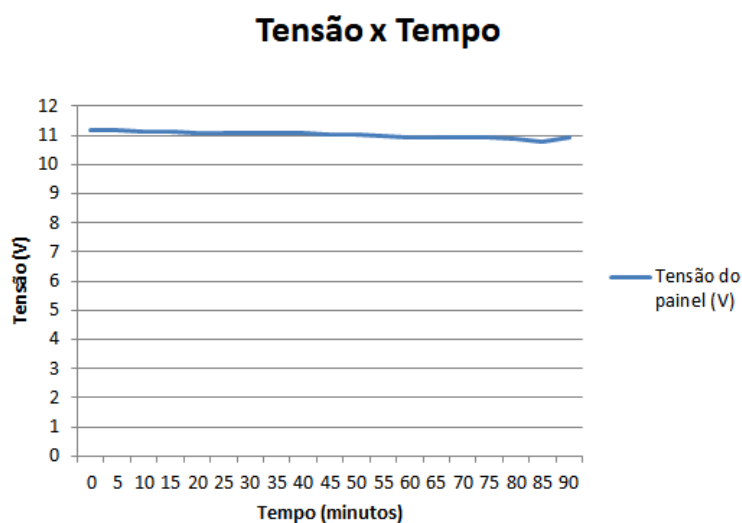


Fonte: Desenvolvida pelo próprio autor.

Um aspecto que pode ser avaliado com a montagem do circuito completo é a operação do *firmware* em um teste de longa duração. Com este intuito, foi escolhido como objeto de estudo, o comportamento do MPPT implementado pelo CN3791, avaliando-se de maneira conjunta se o *firmware* desempenhava suas funções adequadamente durante o teste.

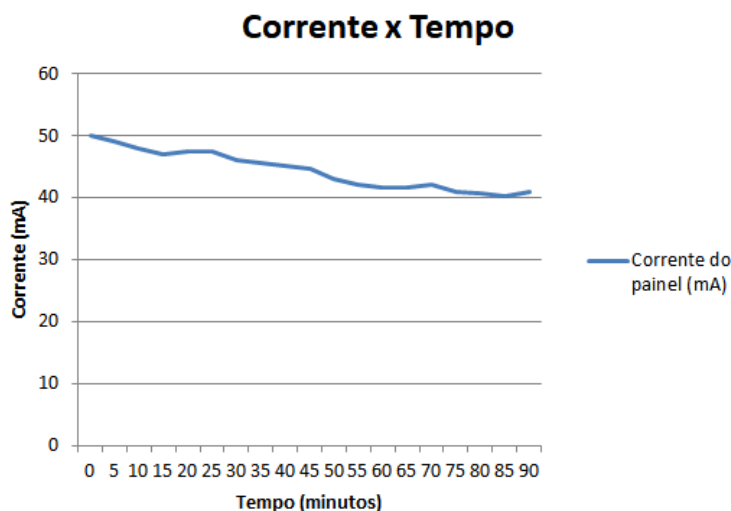
Dessa maneira, o circuito foi exposto à radiação solar durante o período de uma hora e meia e, durante este tempo, foram realizadas medições de tensão e corrente do painel solar. Para melhor visualização dos dados obtidos, foram esboçados gráficos com as leituras realizadas, evidenciados na Figura 4.11 e na Figura 4.12.

Figura 4.11: Tensão do painel solar durante o teste realizado.



Fonte: Desenvolvida pelo próprio autor.

Figura 4.12: Corrente do painel solar durante o teste realizado.

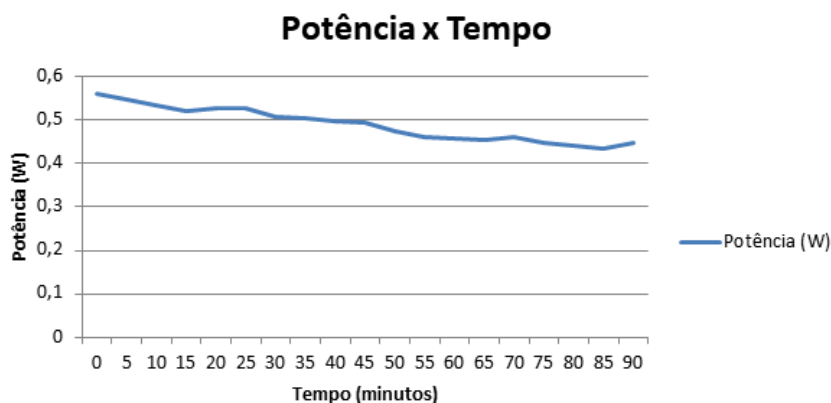


Fonte: Desenvolvida pelo próprio autor.

A partir dos gráficos traçados, é possível a observação de que tanto a tensão quanto a corrente do painel sofreram pequenas variações e possuem valores distintos dos da Tabela 3.1. A diferença entre os valores medidos e tabelados pode ser explicada pelo fato do painel ter sido submetido a condições diferentes nos dois casos, o que ocasiona um deslocamento no ponto de máxima potência, como é expresso na Figura 2.14.

Por outro lado, as pequenas flutuações sofridas são decorrentes de pequenas variações na irradiância solar, visto que por se tratar de uma fonte natural, não foi possível o controle deste parâmetro durante a realização dos testes. Contudo, ao se calcular a potência fornecida pelo painel durante o experimento, como descrito no gráfico da Figura 4.13, é possível observar uma variação pequena, de aproximadamente 0,1 W.

Figura 4.13: Potência do painel solar durante o teste realizado.



Fonte: Desenvolvida pelo próprio autor.

Logo, este resultado pode indicar uma implementação bem sucedida da técnica de MPPT por parte do circuito integrado utilizado. Isto é importante, pois sugere que a utilização de um CI para desenvolvimento deste papel determinante do EPS é uma opção viável, o que economiza recursos por parte do microcontrolador utilizado.

Ademais, durante o período de análise também foi possível a validação do correto funcionamento do *firmware*, sendo realizadas medições com um multímetro digital para comparação com os dados medidos pelos sensores e utilização das demais funcionalidades de maneira simultânea, tais como comunicação serial com o uso de um computador e envio dos dados para o mestre I2C.

Por fim, o *firmware* desenvolvido para o microcontrolador MSP430FR5994 e o *firmware* auxiliar criado para o Arduino Uno, com o intuito de testar a comunicação entre os dois dispositivos, encontram-se nos seguintes links:

- **Firmware principal:** <<https://github.com/thiago-caldas/EPS-Firmware>>
- **Firmware auxiliar:** <<https://github.com/thiago-caldas/Arduino-master>>

5 CONCLUSÃO

Neste trabalho foi apresentado o desenvolvimento de um *firmware* embarcado para o sistema de gerenciamento de energia de um nanossatélite (EPS) para o laboratório LODESTAR da Universidade de Brasília (UnB), além da caracterização deste subsistema.

Primeiramente, foi feita uma contextualização histórica do desenvolvimento dos satélites, bem como a apresentação de diversas aplicações desta tecnologia e a importância de serem realizados estudos em relação ao tema. Em seguida, apresentou-se os CubeSats, os subsistemas que o constituem e as funções desempenhadas por cada um deles, com destaque ao subsistema estudado ao longo deste trabalho, o EPS. Além disso, também foi apresentado o projeto AlfaCruz da UnB, que é o motivador deste trabalho, bem como uma arquitetura proposta para o *firmware* desenvolvido.

Na fundamentação teórica, como forma de caracterizar o módulo de gerenciamento de energia, foram apresentados modelos para modelagem dos principais componentes do circuito do EPS, o painel solar e a bateria, além de diferentes topologias de conversores CC-CC. Ainda neste capítulo, foram demonstradas técnicas para rastreamento do ponto de máxima potência do painel solar, de modo a maximizar a energia fornecida ao satélite, além da exposição dos principais componentes de um microcontrolador, dos tipos de comunicação serial mais utilizadas neste tipo de dispositivo e a apresentação do funcionamento do sensor de corrente INA3221.

Além disso, para um desenvolvimento estruturado do *firmware*, no capítulo de metodologia, foram definidas as métricas utilizadas na sua avaliação, iniciando-se com as simulações realizadas para melhor entendimento do cenário avaliado e seguindo-se com a definição dos testes que avaliariam as tarefas que o *firmware* deveria ser capaz de desempenhar.

Por fim, no capítulo de resultados, foi apresentado o desempenho do *firmware* em cada um dos testes propostos para avaliação de todas as funcionalidades implementadas, assim como o resultado das simulações para caracterização do módulo de gerenciamento de energia. Além disso, nesta seção, foi apresentado o microcontrolador escolhido para o projeto, bem como os recursos utilizados. Em seguida, foi possível observar a partir do teste com o circuito completo, o funcionamento correto da técnica de MPPT implementada via *hardware*.

Sendo assim, os objetivos definidos no começo do trabalho foram concluídos de maneira satisfatória com a caracterização do cenário de teste e criação do *firmware* que desempenha as principais funções do subsistema de gerenciamento de energia.

5.1 SUGESTÕES PARA TRABALHOS FUTUROS

Dentro do objeto de estudo deste trabalho, foram observadas algumas vertentes que não fazem parte do escopo deste projeto, mas que poderiam melhorar o funcionamento do *firmware*

desenvolvido e do subsistema de gerenciamento de energia do satélite como um todo.

Dentre estes aspectos, pode-se destacar como sugestões para trabalhos futuros, o estudo mais aprofundado das baterias, de seus ciclos de carregamento e descarregamento, com o objetivo de se descobrir as suas condições ideais de operação e a implementação de um controle adaptativo que a mantenha em tais condições.

Além disso, para o sistema de configuração dos parâmetros do EPS via *firmware*, pode ser desenvolvida uma interface gráfica que facilite o uso do usuário e que permita a economia de recursos por parte do microcontrolador.

Por fim, sugere-se a integração do *firmware* desenvolvido com os *hardwares* projetados em trabalhos anteriores para uma construção completa do subsistema de gerenciamento de energia, a fim de utilizá-lo em satélites futuramente desenvolvidos pelo laboratório LODESTAR da Universidade de Brasília.

REFERÊNCIAS BIBLIOGRÁFICAS

- ARAÚJO, N.; SOUSA, F.; COSTA, F. Equivalent models for photovoltaic cell—a review. *Revista de Engenharia Térmica*, v. 19, n. 2, p. 77–98, 2020.
- ARDUINO. *Arduino® UNO R3*. 2023. Disponível em: <<https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>>.
- EGAB, K.; OUDAH, S. K. Thermal management analysis of li-ion battery-based on cooling system using dimples with air fins and perforated fins. *International Journal of Thermal Sciences*, Elsevier, v. 171, p. 107200, 2022.
- ELECTRONIC, C. *4A, Standalone Li-ion Battery Charger IC With Photovoltaic Cell MPPT FunctionCN3791*. Disponível em: <<http://www.consonance-elec.com/en/72.html>>.
- ESA. *CubeSats*. 2022. Disponível em: <https://www.esa.int/Enabling_Support/Preparing_for_the_Future/Discovery_and_Preparation/CubeSats>.
- FONSECA, H. N. Estudo de viabilidade e proposta de topologia de um sistema de gerenciamento de energia para nanosatélites da constelação alfa crux. 2020.
- GOMSPACE. *NanoPower 2600mAh*. 2019. Disponível em: <https://gomspace.com/UserFiles/Subsystems/datasheet/gs-ds-nanopower-battery_2600mAh.pdf>.
- HART, D. W. *Eletrônica de potência: análise e projetos de circuitos*. [S.l.]: McGraw Hill Brasil, 2016.
- IEEE. *O Que É Um Microcontrolador?* 2020. Disponível em: <<https://edu.ieee.org/br-ufcgras/o-que-e-um-microcontrolador/>>.
- INSTRUMENTS, T. *LM35 Precision Centigrade Temperature Sensors*. 2017. Disponível em: <<https://www.ti.com/lit/ds/symlink/lm35.pdf>>.
- INSTRUMENTS, T. *INA3221 Triple-Channel, High-Side Measurement, Shunt and Bus Voltage Monitor with I2C- and SMBUS-Compatible Interface*. 2020. Disponível em: <<https://www.ti.com/lit/ds/symlink/ina3221.pdf>>.
- INSTRUMENTS, T. *MSP430FR58xx, MSP430FR59xx, and MSP430FR6xx Family User's Guide*. 2020. Disponível em: <<https://www.ti.com/lit/ug/slau367p/slau367p.pdf>>.
- INSTRUMENTS, T. *TPS22976 5.7-V, 6-A, 14-m On-Resistance Dual-Channel Load Switch*. 2020. Disponível em: <<https://www.ti.com/lit/ds/symlink/tps22976.pdf>>.
- INSTRUMENTS, T. *MSP430FR599x, MSP430FR596x Mixed-Signal Microcontrollers*. 2021. Disponível em: <<https://www.ti.com/lit/ds/symlink/msp430fr5994.pdf>>.
- INSTRUMENTS, T. *MSP430FR5994 LaunchPad™ Development Kit (MSP-EXP430FR5994)*. 2022. Disponível em: <<https://www.ti.com/lit/ug/slau678c/slau678c.pdf>>.
- KNAP, V.; VESTERGAARD, L. K.; STROE, D.-I. A review of battery technology in cubesats and small satellite solutions. *Energies*, MDPI, v. 13, n. 16, p. 4097, 2020.
- LÄTT, S. et al. Estcube-1 nanosatellite for electric solar wind sail in-orbit technology demonstration. *Proceedings of the Estonian Academy of Sciences*, Teaduste Akadeemia Kirjastus (Estonian Academy Publishers), n. 2, p. 200, 2014.

- LEENS, F. An introduction to i²c and spi protocols. *IEEE Instrumentation & Measurement Magazine*, IEEE, v. 12, n. 1, p. 8–13, 2009.
- LODESTAR. *The AlfaCrux Mission*. 2021. Disponível em: <<https://lodestar.aerospace.unb.br/projects/alfacruX>>.
- LOPEZ, A. et al. Max power point tracking system for cubesats. 2016.
- MARTINS, L. A. B. Desenvolvimento de um sistema de gerenciamento de energia para o cubesat alfacruX. 2021.
- MORAES, C. *As 7 principais técnicas de MPPT*. 2020. Disponível em: <<https://eletronicadepotencia.com/as-7-principais-tecnicas-de-mppt/>>.
- MOSTACCIUOLO, E. et al. Modeling and power management of a leo small satellite electrical power system. In: *2018 European Control Conference (ECC)*. [S.l.: s.n.], 2018. p. 2738–2743.
- NASA. *What Is a Satellite?* 2014. Disponível em: <<https://www.nasa.gov/audience/forstudents/5-8/features/nasa-knows/what-is-a-satellite-58.html>>.
- PEÑA, E.; LEGASPI, M. G. Uart: A hardware communication protocol understanding universal asynchronous receiver/transmitter. *Visit Analog*, v. 54, n. 4, 2020.
- POGHOSYAN, A.; GOLKAR, A. Cubesat evolution: Analyzing cubesat capabilities for conducting science missions. *Progress in Aerospace Sciences*, Elsevier, v. 88, p. 59–83, 2017.
- SALGADO, C. A. V. et al. Arduino-based prototype to estimate heat stress indices in urban environments. In: EDITORIAL UNIVERSITAT POLITÈCNICA DE VALÈNCIA. *Proceedings 5th CARPE Conference: Horizon Europe and beyond*. [S.l.], 2019. p. 173–182.
- TAMILSELVI, S. et al. A review on battery modelling techniques. *Sustainability*, MDPI, v. 13, n. 18, p. 10042, 2021.
- THAKKAR, R. R. Electrical equivalent circuit models of lithium-ion battery. *Management and Applications of Energy Storage Devices*, IntechOpen London, UK, 2021.
- VINAY, G.; SAINI, S. A review on cube satellites. 2014.
- WAGNER, F. *Modeling software with finite state machines: a practical approach*. [S.l.]: Auerbach Publications, 2006.
- YANG, W. et al. Preheating performance by heating film for the safe application of cylindrical lithium-ion battery at low temperature. *Fire Technology*, Springer, p. 1–21, 2022.