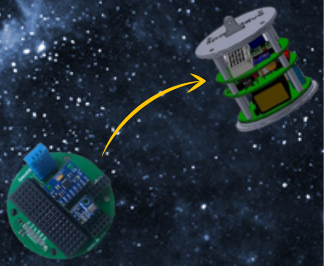
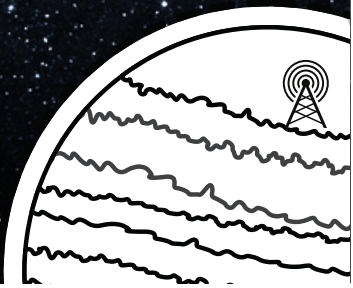
Prof. Dr. José Soares Batista Lopes
(Organizador)



Contribuições para o Desenvolvimento de Picossatélites didáticos

Com projetos inovadores desenvolvidos no curso técnico integrado de mecatrônica do IFRN - Campus Parnamirim



2021

JOSÉ SOARES BATISTA LOPES, DR. (ORGANIZADOR)

Contribuições para o Desenvolvimento de Picossatélites didáticos

Com projetos inovadores desenvolvidos no curso técnico integrado de mecatrônica do IFRN - Campus Parnamirim

2021

DIAGRAMAÇÃO
José Soares Batista Lopes
CAPA
José Soares Batista Lopes
ILUSTRAÇÕES
canva.com
ÍCONES DA CAPA
canva.com

FICHA CATALOGRÁFICA
Catalogação da publicação na fonte elaborada pelo Bibliotecário
Cícero Filho Tavares – CRB: 15/511

C764 Contribuições para o Desenvolvimento de Picossatélites didáticos: com projetos inovadores desenvolvidos no curso técnico integrado de mecatrônica do IFRN – Campus Parnamirim /José Soares Batista Lopes. – Parnamirim, 2021
109 p. : il. Collor

ISBN: 978-65-00-23874-7

1.Nanossatélite. 2. Cansat. 3. Cubesat. 4. Programação I Lopes, José Soares Batista. II. Gil, Pedro João Antônio Alvarez. III Soares, Árisla Bruna de Menezes. IV Luz, Maria Eduarda Lima da. V Santos, Natália Pereira dos. VI. Titulo.

CDU 629.7.05:629.78

Edição eletrônica: E-books
Linha Editorial: Tecnologias
(Ciências aplicadas)
Disponível para download em:
http://memoria.ifrn.edu.br

CONTATO
Prof. Dr. José Soares Batista Lopes
IFRN - Campus Parnamirim
Rua Antônia de Lima Paiva, 155 -
Bairro Nova Esperança, Parnamirim
CEP: 59143-455
jose.soares@ifrn.edu.br

CC BY NC SA

Sumário

APRESENTAÇÃO.....04

PROJETO E DESENVOLVIMENTO DE INTERFACES PARA O SPACEDUINO.....07
PEDRO JOÃO ANTONIO ALVAREZ GIL

ESTAÇÃO METEOROLÓGICA PARA CAPTAÇÃO DE DADOS CLIMÁTICOS DE UM PICO SATÉLITE..... 29
ÁRISLA BRUNA DE MENEZES SOARES

DESENVOLVIMENTO DE UM PROTÓTIPO DE SATÉLITE METEOROLÓGICO EM FORMA DE CUBO - CUBESAT.....49
MARIA EDUARDA LIMA DA LUZ

APLICAÇÃO DE FONTE DE ENERGIA ALTERNATIVA EM UM PROTÓTIPO DE SATÉLITE EM FORMA DE CUBO – CUBESAT..... 62
NATÁLIA PEREIRA DOS SANTOS

ESTUDO E IMPLEMENTAÇÃO DE UM REAL TIME OPERATING SYSTEMS- RTOS PARA O CUBESAT..... 79
HARTUR SOUZA DA SILVA

METODOLOGIA DE ENSINO APLICADA AO KIT EDUCACIONAL DE CANSAT.....96
MICHELLY DANTAS GUEDES

APRESENTAÇÃO

APRESENTAÇÃO

O livro **Contribuições para o Desenvolvimento de Picossatélites didáticos** é oriundo da orientação de alunos que cursaram o curso técnico integrado de mecatrônica do IFRN campus Parnamirim/RN, concluintes de 2019 e 2020. Este e-book aborda os conceitos iniciais para o desenvolvimento de picossatélites explorando suas funcionalidades e aplicações partindo das especificações de projeto até a sua construção e validação.

Os capítulos são sínteses dos relatórios finais de curso envolvendo o mesmo tema "picossatélites". Os projetos basearam-se no princípio do conceito STEM (acrônimo do inglês *Science, Technology, Engineering and Mathematics)* que significa Ciência, Tecnologia, Engenharia e Matemática.

O primeiro capítulo "Projeto e desenvolvimento de interfaces para o spaceduino" introduz o conceito do Cansat e mostra a circuitaria necessária para o projeto de um picossatélite. O capítulo apresenta o desenho dos circuitos no EasyEDA com os diagramas elétricos e eletrônicos. A vantagem e desvantagem na mudança de componentes PTH(Pin Througt Hole) para os componentes SMD(Surface Mount Device) para o Spaceduino. E por último, descreve testes de funcionamento para validar o protótipo.

O segundo capítulo "Estação meteorológica para captação de dados climáticos de um pico satélite" apresenta uma fundamentação teórica dos sensores, sistema eletrônico embarcado e a programação implementada. Faz um detalhamento dos sensores: BMP180, DHT11 e o MPU6050 para a captação de dados climáticos. Mostra as simulações dos circuitos utilizando o software de simulação Fritzing para a fase de teste dos subsistemas: computador de bordo, carga útil e o suprimento de potência. Por último, mostra a a confecção das placas e sua integração no Cansat.

APRESENTAÇÃO

O terceiro capítulo "Desenvolvimento de um protótipo de satélite meteorológico em forma de cubo - cubesat" apresenta o desenvolvimento de um protótipo de um satélite meteorológico em forma de cubo dividido em cinco etapas: a primeira compreende o funcionamento e as características dos sensores, na segunda ocorre a escolha de sensores de baixo custo, na terceira contempla as simulações, a quarta a montagem e a integração, e por último, a análise e resultados. Destaca-se as montagens de circuitos testes no protoboard e a modelagem 3D no software Fusion 360 mostrando a estrutura fixa e móvel do cubesat.

O quarto capítulo "Aplicação de uma fonte de energia alternativa em um protótipo de satélite em forma de cubo - Cubesat" explica a necessidade de uma fonte alternativa e a redundância dos sensores para o funcionamento dos satélites no espaço. Mostra simulações e a integração da placa solar descrevendo suas características em um pico satélite - cubesat na alimentação de energia para os sensores e os circuitos do sistema.

O quinto capítulo "Estudo e Implementação de um Real Time Operating Systems - RTOS para o Cubesat" explora a programação destacando as boas práticas: indentação, comentários e convenções de nomes para classes, métodos e variáveis por exemplo. Aplica os comandos e registradores específicos na programação dos sensores explorando as bibliotecas.

O sexto capítulo "Metodologia de ensino aplicado ao kit educacional de Cansat" constrói um manual de montagem e utilização para o kit educacional utilizando o Cansat para o ensino de tecnologia espacial, destacando os subsistemas do satélite aplicando a metodologia maker. Descreve os componentes do cansat, a sua estrutura física e seus subsistemas.

Por fim, agradeço aos autores pelo esforço, parceria e as contribuições no desenvolvimento desta obra. Desejo aos leitores uma boa e produtiva leitura dos capítulos, sobre os temas abordados, e que estas possam auxiliar no desenvolvimento de seus protótipos e contribuir para o seu aprendizado.

Prof. Dr. José Soares B. Lopes

PROJETO E DESENVOLVIMENTO DE INTERFACES PARA O SPACEDUINO

Pedro João Antonio Alvarez Gil

PROJETO E DESENVOLVIMENTO DE INTERFACES PARA O SPACEDUINO

Pedro João Antonio Alvarez Gil[1]

INTRODUÇÃO

O início da era espacial ocorreu devido ao lançamento do satélite realizado pela antiga União soviética (ERIKA, 2004), o Sputnik-1 consistia em um projeto esférico de cerca de 58 centímetro de diâmetro, com quatro antenas para a transmissão de sinais de rádio, que orbitou de forma funcional por cerca de 22 dias, mas devido a problemas técnicos, a comunicação entre o satélite e as estações terrestres se encerraram.

Com o avanço de pesquisas foi possível o desenvolvimento de novas categorias de satélites, como o nano e o pico satélite. Posteriormente, em 1998, o termo Cansat foi citado em um Simpósio Universitário de Sistemas Espaciais, criando-se no evento, uma meta de lançar um artefato do tamanho de uma lata de refrigerante ao espaço. O interesse nesse tipo de tecnologia cresceu exponencialmente, visto que tais satélites miniaturizados são mais fáceis e baratos de serem colocados em órbita

Com o aumento do interesse pelo desenvolvimento tecnológico espacial, começou a surgir novos projetos com custos de construção e teste mais baixos. Um exemplo dessa nova demanda, é o projeto desenvolvido pela equipe Miyazaki (2016), foi desenvolvido com intuito educacional. Pois, tem capacidade de verificar variáveis atmosféricas e obter a localização via GPS, além disso é possível utilizados em foguetes educacionais ou balões meteorológicos. O projeto é composto por 6 placas com funcionalidades especificas e cada uma com diâmetro de 53.4mm.

[1] *Técnico em Mecatrônica pelo Instituto Federal de Ciência e Tecnologia do Rio Grande do Norte – Campus Parnamirim – pedro.antonio.gil27@outlook.com*

O projeto originou-se com o intuito de facilitar os desenvolvimentos de projetos que tenham ênfase em engenharia espacial. Pois, de acordo ao projeto Arliss(1999), que fabrica foguetes para o lançamentos de pico-satélites em formato de lata de refrigerante, as dimensões do disponíveis para colocar na carga útil (é o volume do espaço total que é possível colocar no foguete (ANTAS, 1979)) do foguete são bem reduzidas.

Sendo assim, o projeto que for colocado nesse espaço requer um limite de dimensões para conseguir se colocado da forma correta. Além disso, no projeto é necessário que consiga enviar e receber comandos da central de controle, estrutura que estará na terra, sendo ela responsável pelo envio de comandos para o projeto que for lançado.

A Competição europeia de Cansat da ESA (2019) é um evento que ocorre todo ano, com intuito de incentivar jovens a seguirem uma carreira na ciência ou engenharia, visando garantir uma força de trabalho altamente qualificada na indústria espacial do futuro.

O evento se basear em desenvolver um pico-satélite em formato de lata de refrigerante, a placa apresentada neste trabalho seria mais que adequada para tal evento, pois iria facilitar a vida do projetista no evento, uma vez que uns dos trabalhos da equipe participante no evento é encontrar ou desenvolver uma placa de prototipação, que seja utilizada a central de controle do Cansat (Pico-satélite). Uma vez que já tenham uma placa para ser a central de comando, haveria mais tempo de dedicação nas outras etapas da competição.

METOLOGIA

Visando o melhor aproveitamento do tempo para a realização do projeto, foi divido em três etapas, primeiramente foi colocado os requisitos que a placa tinha que atender, depois foi necessário se aprofundar em técnicas de fabricação de placas e o manuseio de componentes eletrônicos. Ao obter os conhecimentos técnicos necessários para conseguir projetar a placa no EasyEDA, faltou somente reproduzir a placa através de técnicas de fabricação de pci’s e junto com a fabricação foram realizados os testes individuais das placas.

Devido o projeto ser utilizados com fins de desenvolvimento de projetos relacionados com o setor espacial, será utilizado os requisitos da Competição europeia de Cansat da ESA (2019). Evento que acontece anualmente, visando o desenvolvimento tecnológico e o incentivo de jovens pela engenharia espacial. A seguir alguns requisitos do projeto do spaceduino, para um bom funcionalmente e adaptabilidade em outros projetos:

- Diâmetro inferior 66mm;
- Placa Programável;
- Facilidade de comunicação de dados;
- Disponibilidade de mais de 18 entradas/saídas digitais;
- Disponibilidade de mais de 6 entradas/saídas analógicas;
- Pinos de alimentação de 5v e 3,3v;

Eletrônica

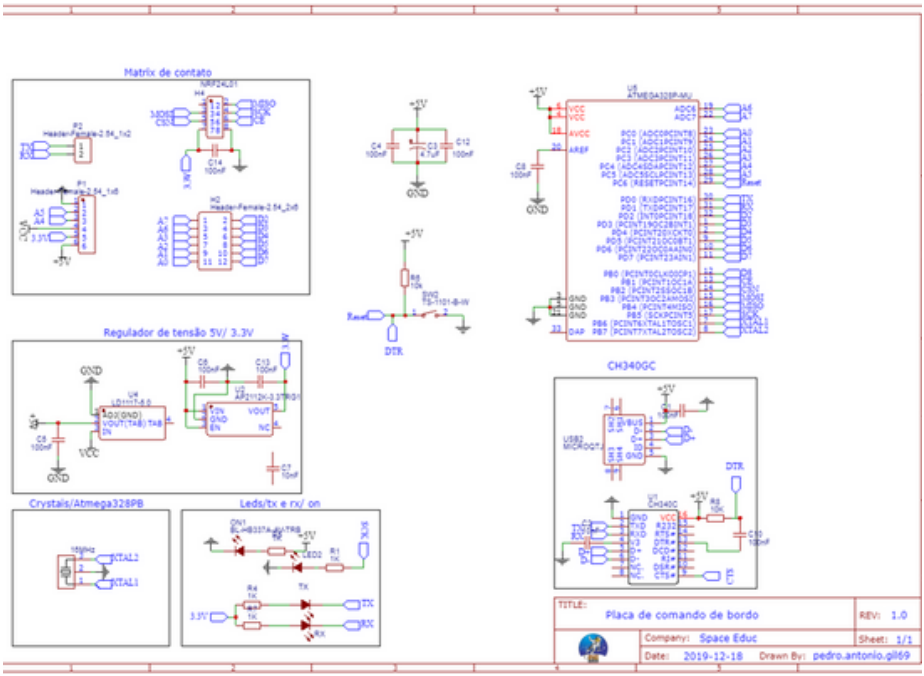
Foi necessário o estudo da eletrônica aprofundado de outras placas de prototipação, mas especificamente placas que utilizassem o microcontrolador Atmega328p na sua estrutura, um exemplo de placa que foi utilizada como referência do projeto é o Arduino Uno, pois o projeto dela é open-source, sendo assim é disponível ao público. Além disso, foi estudado formas de conectar um transmissor diretamente no CI, para pode utilizar a placa como um transmissor de sinais.

Dado o aumento da complexidade dos sistemas eletrônicos embarcados na placa do Spaceduino, foi necessário, o estudo das conexões dos componentes utilizados e a esquematização deles, para que os devidos posicionamentos não causem interferências e nem mal funcionamento nos outros componentes. Por exemplo, não fazendo ligações certas, existe a possibilidade de ocorrer curtos circuitos.na placa ou até mesmo ocorrer erros de posicionamentos dos componentes em cima da placa.

Após os estudos, dá-se início ao projeto do protótipo em uma plataforma conveniente. No caso, foi utilizado a plataforma Online EasyEDA, cujo apresenta uma interface simplificada e satisfaz os resultados esperados.

O diagrama elétrico finalizado pode ser visto na figura 1.

Figura 1 - Diagrama elétrico do projeto.

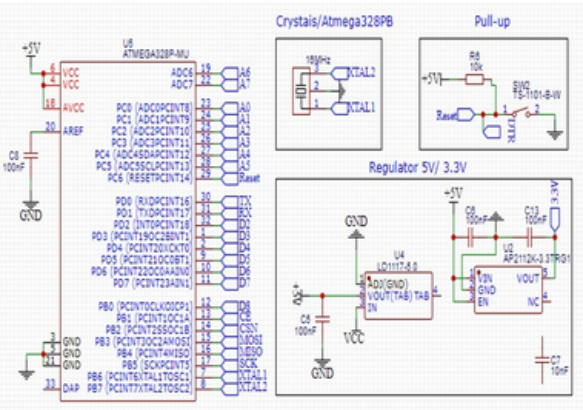


Fonte: Autoria própria.

PROTÓTIPO DO SPACEDUINO

Para a realização do projeto do Spaceduino, inicialmente foi realizada a divisão da eletrônica necessária, para o funcionamento correto do microcontrolador. Foi dividido em três partes, o Crystal que fica responsável pela geração do Clock, o regulador de tensão responsável pela alimentação constante do CI e o resistor ligado em pull-up para evitar o reset do microcontrolador. Após a divisão, as ligações das partes foram feitas, como demonstrado na figura 2.

Figura 2 - Eletronica essencial do Spaceeduino.



Fonte: Autoria própria.

Para fins de prototipação, inicialmente foi utilizado componentes do tipo Through-hole, pois são mais acessíveis e fáceis de trabalhar, após a finalização do projeto no software foi necessário reproduzir de forma caseira o protótipo do Spaceduino, através do método de transferência térmica, ao finalizar o tratamento da placa e de soldagem dos componentes, o resultado pode ser visto na figura 3.

A placa da figura 3 foi realizado alguns testes de funcionamento do tipo, medição de tensão nos pinos, envio de códigos e a comunicação com o transmissor.

Figura 3 - Diagrama elétrico do projeto.

Fonte: Autoria própria.

Testes realizados

Para a validação do protótipo, foram realizados os seguintes três testes:

- Teste de continuidade dos pinos;
- Teste de funcionamento do microcontrolador;
- Teste de comunicação via radio frequência;

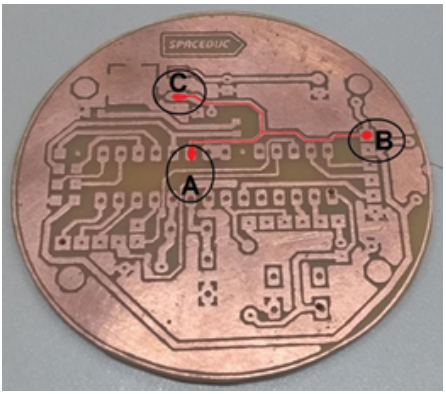
Primeiramente, para realizar o teste de continuidade é necessário ter um multímetro em mãos, para testar todas as trilhas da pci com as ponteiras do multímetro (pontas metálicas que estão conectadas diretamente com o equipamento).

O teste ocorre da seguinte maneira, para fins de explicação, veja a figura 8, tem linha vermelha que vai sinalizar uma trilha e seus pontos de conexões (A, B e C). Para a placa se aprovada no primeiro teste, é necessário que todas as ligações que foram feitas no diagrama esquemático estejam corretas na placa.

Por exemplo, o ponto A é para estar conectado com o ponto B e os dois pontos devem estar conectados com o ponto C. Então, a sua validação a verificação da conexão ocorre com o multímetro. Configura-se o multímetro para o modo de continuidade e, então coloca-se as suas ponteiras no ponto A e no ponto B, se o equipamento fizer um sinal sonoro, significa que a conexão está certa, o mesmo processo ira ocorrer para o ponto B e C.

Esse processo terá que ocorrer em todas as trilhas da placa, uma vez que todas tenham sido testadas e aprovadas, então já será possível soldar os componentes na pci, sem o perigo de ocorrer um curto-circuito ou qualquer outro problema.

Figura 4 - Placa do spaceduino.

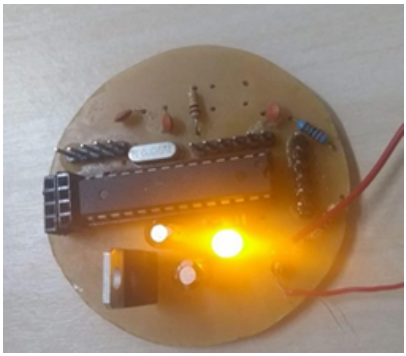


Fonte: Autoria própria.

Para a realização do segundo teste, foi necessário que a placa estivesse completamente soldada com todos os seus componentes, o intuito do teste foi validar se as conexões realizadas no Atmega328p estavam corretas. O teste ocorreu através do envio de um programa para a placa fazendo que a placa ativasse o led on bord na placa.

O protótipo foi aprovado, figura 5.

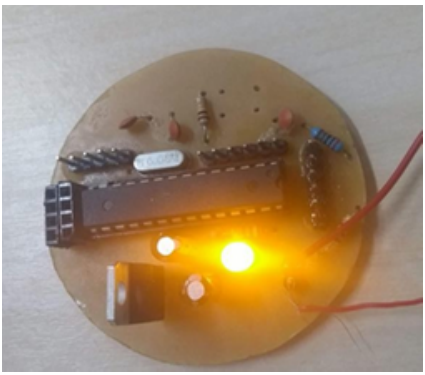
Figura 5 - Spaceduino com o led ligado .



Fonte: Autoria própria.

O último teste foi a validação da comunicação via rádio frequência. Para a sua realização foi necessário duas placas e dois rádios transmissores (figura 6).

Figura 6 - Modulo rádio frequência (NRF24L01).



Fonte: Autoria própria.

O teste ocorreu da seguinte maneira, uma das placas foi programada com o código do emissor e a outra com o código do receptor. Além de efetuar a comunicação observou-se também uma mensagem no Serial monitor da IDE do Arduino.

SPACEDUINO VERSÃO FINAL

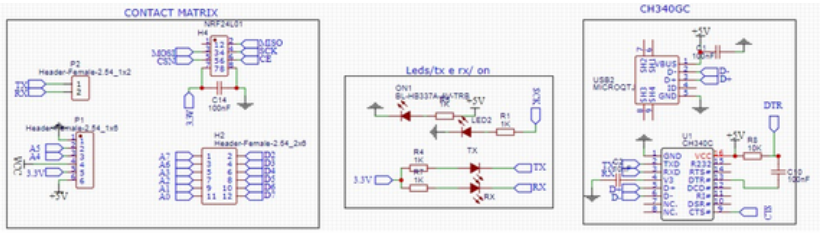
Após os testes prosseguimos com a etapa seguinte, que foi a de reduzir as dimensões da placa. Para isso foi necessário a troca de todos os componentes do tipo through-hole para o tipo SMD. Esta etapa foi realizada para facilitar o uso do spaceduino em projetos que requerem uma placa de pequenas dimensões. Os componentes utilizados na segunda etapa da placa e suas funções.

- Atmega328pb-AU - O Spaceduino foi baseado no microcontrolador Atmega328pb-AU, sendo responsável na placa pela execução de todos os comandos solicitados;
- Crystal 16Mhz - CSTCE16M0V53-R0 - O Crystal é responsável pela geração de sinais e Clocks para o microcontrolador;
- Capacitores cerâmicos 22pf - 0402 - Os capacitores cerâmicos fazem parte da eletrônica necessária para o funcionamento correto do Crystal, diminuem o ruído do sinal;
- Reguladores de tensão 5v - 3.3v – 1117v-5v e 1117v- 3.3v - O regulador de tensão é utilizado como forma de proteção do circuito, para ter uma voltagem constante de 5 volts e 3.3volts;
- Resistor de 10k - 0402 - a328

Além da eletrônica necessária, foi adicionado alguns componentes para um melhor aproveitamento da placa, por exemplo, a eletrônica do conversor USB-Serial, leds sinalizadores e barramentos ligados nos pinos do Atmega328pb.

O esquema elétrico desses novos componentes na figura 7.

Figura 7 - Barramentos, leds sinalizadores e conversor USB-Serial, esquerda para direita.



Fonte: Autoria própria.

A matriz de contato tem disponível 26 pinos de entrada/saída de sinal (4 podem ser utilizados como PWM), 19 entradas digitais, 7 entradas analógicas e os pinos de alimentação de 5v e 3.3 volts.

Além disso, já possui um barramento específico para utilizar na conexão do transmissor NRF240L1, como pode ser observado, na figura 10, com um rádio transmissor conectado no barramento. Os leds sinalizadores, como o próprio nome descreve, sinalizam eventos na placa, do tipo: momento que é enviado um programa novo, os leds sinalizam piscando, quando o Spaceduino é ligado em uma fonte de energia um led vermelho é acesso. Além disso, possuí um led para realizar testes sem a necessidade de adicionar outro componente na eletrônica, o teste que é normalmente realizado é com o programa “BLINK”.

Figura 8 - Placa do Spaceduino.

Fonte: Autoria própria.

Foi adicionado o circuito integrado CH340G na circuitaria. O CH340G é um conversor USB-Serial, de acordo ao datasheet fornecido por Jiangsu Haoheng (2012), o CI converte sinais USB em sinais seriais de nível TTL (Time To Live), permitindo a comunicação entre o computador e a plataforma do microcontrolador (Spaceduino). Além disso, foi conectado na placa uma entrada USB Micro-B para ser ligado diretamente com o conversor USB-Serial, dessa forma é possível usar um cabo micro USB-B na placa para o envio de códigos ou comandos, figura 9.

Figura 9 - A placa do Spaceduino conectada com o nrf24l01.

Fonte: Autoria própria.

Placas de suporte

Foram desenvolvidas outras placas para facilitar a realização de três testes no spaceduino. Cada placa tem sua função especifica e serão apresentadas a seguir, em ordem cronológica de prototipação.

A primeira placa foi projetada para validar se o spaceduino, funcionaria corretamente desconectado do notebook, com uma bateria externa, figura 10. A função dela é validar se o funcionamento do Spaceduino continuar constante mesmo com uma bateria externa,e se os reguladores de tensão do spaceduino, estão funcionando corretamente. Das três placas ela é a mais simples, pois toda a eletrônica dela se basearem alimentar o microcontrolador do Spaceduino com uma bateria externa.

Figura 10 - A Placa de suprimento de potência.

Fonte: Autoria própria.

A placa da figura 10 possuí a função de validar o funcionamento correto dos pinos de saída e entrada de sinal, foi projetada para que sua estrutura tenha três sensores fixos: o dht11(sensor de umidade), bmp180(sensor de altitude, pressão e temperatura) e o mpu6050(Giroscópio e acelerômetro), tais sensores facilitam os testes pois não é preciso utilizar jumpers, que geralmente causam erros de mal contato no circuito. Com essa placa foi possível testar todas as categorias de pinos existentes no microcontrolador de forma individual.

A placa com os sensores pode ser observada na figura 11.

Figura 11- A Placa com os sensores.

Fonte: Autoria própria.

A placa da figura 12 e 13 foi produzida para validar a comunicação de rádio frequência, funcionando da seguinte maneira, primeiramente é configurado para funcionar de acordo a lógica “Master/Slave” (OUDJIDA et al, 2010), sendo configurado o Spaceduino como master e a placa de recepção estará no modo Slave.

Figura 12- A Placa com os sensores.

Fonte: Autoria própria.

Para facilitar a realização de testes, o Spaceduino utilizou duas placas (figuras 9 e 10), uma fornece as variáveis e a outra energiza o circuito completo. Uma vez enviado o código pela IDE do Arduino podemos desconectar o cabo Micro-USB do Spaceduino e ligar a placa que alimentar o circuito completo.

Posteriormente essas ações, podemos enviar o código do “Slave” para a placa de recepção através a entrada USB no computador (figura 12). Caso a comunicação tenha sido efetuada com sucesso, as variáveis que a placa de sensores fornecem, aparecerão no serial monitor da IDE do Arduino.

Figura 13 - Spaceduino com as duas placas e a placa de recepção conectada no notebook.

Fonte: Autoria própria.

Na figura 14 mostra a tela do serial monitor com os dados que foram enviados pelos sensores da placa de testes.

Figura 14 - Serial monitor do Arduino.

```
-----------------------------
Testando o Spaceduino - Pj:
Temperatura do sensor MPU:26.25
Temperatura do sensor DHT11:28.2
Umidade: 39%
Pressao: 10025.34
Temperatura do sensor BMP:28.5
Altitude: 1
-----------------------------
Testando o Spaceduino - Pj:
Temperatura do sensor MPU:26.20
Temperatura do sensor DHT11:28.1
Umidade: 39%
Pressao: 10026.34
Temperatura do sensor BMP:28.0
Altitude: 1
-----------------------------
```

Fonte: Autoria própria.

Programação das Placas

A programação do Spaceduino foi implementada no ambiente de desenvolvimento integrado da plataforma Arduino (IDE), dessa forma a linguagem de programação utilizada foi baseada na linguagem ANSI C.

FUNDAMENTAÇÃO TEÓRICA

Nesta seção são abordados os conhecimentos que foram necessários para o desenvolvimento do projeto.

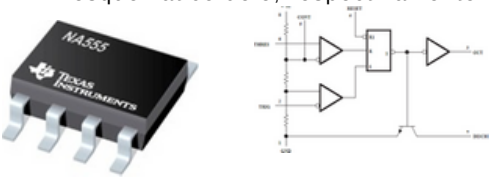
Circuitos integrados

Na realização das placas, existiu a necessidade de entendimento sobre o funcionamento e produção de Circuitos Integrados (CI), pois foram utilizados os seguintes componentes: Atmega328p e CH340H.

Um Circuito Integrado pode ser definido como um conjunto de componentes de elementos de circuito eletrônico (ALAN, 2008), como: resistores, diodos, capacitores e transistores, interligados de forma simultânea dentro da mesma estrutura, constituindo um dispositivo único que realiza determinada função.

Então, dentro de um CI, existe componentes elétricos que quando juntos realizam funções determinadas pelo projeto, no processo de produção, dos Chips, existem diversos tipos de encapsulamento, que determinam também a forma de fixação nas Placas de Circuito Impresso (PCI). Podem ser montados na superfície da placa, sem atravessá-la (Surface Mount Device, “SMD”), ou podem ser montados com seus terminais atravessando a placa (Pin Through Hole, PTH). Um exemplo de um CI do tipo SMD, figura 15.

Figura 15 - Circuito integrado NA555 e o Diagrama esquemático dele, respectivamente.



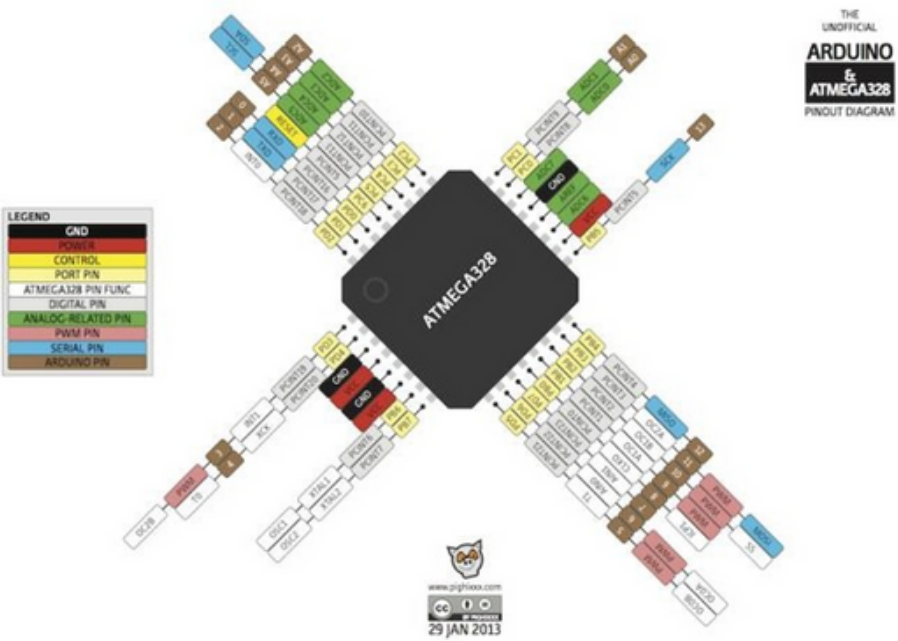
Fonte: Texas Instruments (2010).

O CI demostrado na figura 15 é o NA555, dentro dele existem os seguintes componentes eletrônicos: um flip-flop RS, dois comparadores simples, três resistores e um transistor (TEXAS INSTRUMENTS, 2010). Todos os componentes juntos no mesmo encapsulamento realizam a função de acordo com a configuração do temporizador, gerador de pulsos e/ou oscilador clock.

Atmega328p

O microcontrolador estudado no projeto foi o Atmega328p, encapsulamento SMD TQFP, faz parte da família de microcontroladores de 8 bits CMOS baseada na arquitetura AVR lançada pela ATMEL (2015). Foi utilizado na placa do Spaceduino, pois suas características técnicas favorecem para o desenvolvimento de placas de prototipação autorais.

Figura 15 - Circuito integrado NA555 e o Diagrama esquemático dele, respectivamente.



Fonte: Texas Instruments (2010).

Segundo o datasheet fornecido pela ATMEL (2015), o microcontrolador de 8 bits AVR RISC de alto desempenho combina memória flash ISP de 32 KB com recursos de leitura e gravação, EEPROM de 1024 Bytes, SRAM de 2 KB, 23 pinos de Entrada/Saída de sinal, 32 registradores disponíveis, três temporizadores timer/counters com modos de comparação, interrupções internas e externas, USART serial programável, timer de vigilância programável com oscilador interno e cinco modos de economia de energia selecionáveis por software.

O dispositivo opera entre 1,8-5,5 volts. É capaz de executar instruções poderosas em um único ciclo de clock, dispositivo alcança taxas de processamento próximas a 1 MIPS por MHz, equilibrando o consumo de energia e a velocidade de processamento.

EasyEDA

Foram testados diversos softwares de prototipação de placas de circuito impresso (PCI) ao recorrer da idealização do projeto. O EasyEDA acabou sanando todas as necessidades que tínhamos, sendo assim a mais adequada para projetar e fabricar o esquema elétrico da placa, figura 16.

Figura 16 - Area de trabalho do EasyEda.

Fonte: Autoria própria.

O Software permite a criação e edição de diagramas esquemáticos, a simulação Wikipedia (2021) de circuitos analógicos e digitais mistos e a criação e edição de layouts de placas de circuito impresso e, opcionalmente, a fabricação de placas de circuito impresso.

Uma grande vantagem do software, é sua biblioteca online de datasheets dos mais diversos componentes existentes no mercado, outra grande diferencial do EasyEDA é que todos os projetos são salvos em uma nuvem e tornando o acesso ao projeto fácil em outros dispositivos. Ao finalizar o trabalho, caso seja necessário é possível solicitar para a própria empresa do software a produção das placas de circuito impresso em um nível profissional.

As PCBs (Printed Circuit Board ou Placas de circuito impresso) têm como objetivo: minimizar o uso de fios, organizar os componentes no menor espaço possível e ter proteções contra curtos-circuitos. As placas de circuito impresso demostradas, neste trabalho, foram projetadas e produzidas no software EasyEda.

KIT EDUCACIONAL DE CANSAT

Neste tópico é apresentado o projeto que iniciou o desenvolvimento do spaceduino.

O kit educacional de cansat (GIL, 2019), é um projeto realizado com intuito de tornar-se uma ferramenta didática de ensino de engenharias aeroespaciais, mais especificamente no tema de satélites. O acrônimo Cansat vêm do termos em inglês: Can e Satellite – Lata e Satélite, como o próprio nome diz, é uma estrutura que tem o formato de uma lata de refrigerante com a função de um satélite. O projeto final pode se demonstrado na figura 17.

Figura 17- Kit Educacional de cansat completo.

Fonte: Autoria própria.

Com a apresentação deste protótipo, espera-se que haja um maior entendimento sobre a placa apresentada, e que possa facilitar o entendimento do aluno, que for desenvolver algum projeto com o Spaceduino. Por fim, espera-se que a placa torne-se conhecida em nível nacional, como uma placa compacta e simples de trabalhar, além disso, que possa corroborar para o estudo de ciências e tecnologias espaciais.

O kit foi baseado no cansat projetado pelo pesquisador Miyazaki (2016), visando obter resultados semelhantes aos apresentados por ele, mas sendo o nosso mais simples de construir, pela menor quantidade de componentes necessários, e público alvo ser estudantes concluintes do ensino fundamental. O kit foi projetado para ser utilizado como um quebra-cabeça “tecnológico”, pois todas as suas peças mecânicas sãos encaixáveis por conexões.

CONSIDERAÇOES FINAIS

Com a apresentação deste trabalho, espera-se que haja um maior entendimento sobre a placa apresentada, e que possa facilitar o entendimento do aluno, que for desenvolver algum projeto com o Spaceduino. Por fim, espera-se que a placa se torne conhecida em nível nacional, como uma placa compacta e simples de trabalhar, além disso, que possa corroborar para o estudo de ciências e tecnologias espaciais.

Ao decorrer do desenvolvimento do projeto, apareceram vários desafios, sendo o principal deles, foi aprender a desenvolver placas de circuito impresso, e trabalhar com componentes SMD. Com o decorrer da criação do projeto, tive que me aprofundar em conhecimentos da área de eletrônica, pois o projeto todo se base nessa área.

PESPECTIVAS FUTURAS

Os próximos passos do projeto, seria a troca do microcontrolador Atmega328p pelo CI do ESP8266, pois possibilitaria a conexão do spaceduino na internet e assim, abrindo um leque de possibilidades de melhoramentos no projeto.

Como a adição de técnicas de I.A(Inteligência artificial), no controle de atuadores de acordo com os sinais recebidos pelos sensores. Facilitando até no rastreamento após o lançamento via módulo de GPS. Posterior avanço do projeto, pretendesse também criar uma plataforma de desenvolvimento especifica para o Spaceduino, com intuito de torna o projeto independente de outros softwares. Um exemplo seria o projeto da LEGO Mindstorms NXT, que é um brinquedo lançado pela lego, voltado para educação tecnológica.

O equipamento possuí sua própria plataforma de desenvolvimento, chamada de Overmars (2002), e a linguagem de programação utilizada é o NXC. Por fim, existe diversas melhorias possíveis no projeto com intuito de tornar cada vez mais completo, a experiência no desenvolvimento de projetos com o spaceduino.

REFERÊNCIAS BIBLIOGRÁFICAS

ALLAN. **Circuito integrado**. 2008. Disponível em: <https://allancamposcurso.wordpress.com/2008/04/>. Acesso em: 28 maio 2021. Texas Instruments.

ARLISS. **A rocket launch for international student satellites**. 1999. Disponível em: www.arliss.org ANTAS, Luiz Mendes. Glossário de termos técnicos. São Paulo: Traço, 1979. 756 p. (Coleção Aeroespacial; t.1.). Disponível em : <ttps://www2.anac.gov.br/anacpedia/por_ing/tr1966.htm#:~:text=Diversas%20s%C3%A3o%20as%20defini%C3%A7%C3%B5es%20sobre,da%20carga%20bruta%20de%20uma>. Acesso em: 10 jan. 2021.

ATMEL. **ATmega328P: 8-bit AVR Microcontroller with 32K Bytes InSystemProgrammable Flash**. 2015. Disponível em: <http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive- Microcontrollers-ATmega328P_Datasheet.pdf>. Acesso em: 10jan. 2021.

AYDANO. **Uma Breve História da Conquista Espacial,** 2009. Disponível em:<http://seer.cgee.org.br/index.php/parcerias_estrategicas/article/viewFile/78/70 >. Acesso em: 10 jan. 2021.

Bricxcc. NXC. 2002. Disponível em: <http://bricxcc.sourceforge.net/nqc/doc/NQC_Tutorial.pdfAcesso em: 09 jan. 2021.

ESA. **ESA EUROPEAN CANSAT COMPETITION (ECS)**, 2019. Disponívelem: <https://esamultimedia.esa.int/docs/edu/CanSat_guidelines_201920.pdf>.Acesso em: 10 jan. 2021.

Haoheng. Ch340g. 2012. Disponível em: < https://html.alldatasheet.com/htmlpdf/1132603/WCH/CH340T/61/1/CH340T.html>. Acesso em: 13 Fev. 2021.

GIL, Pedro João A. A.. et al.. KIT EDUCACIONAL DE CANSAT.. In: **Anais do 2º Congresso Aeroespacial Brasileiro.** Anais...Santa Maria(RS)Universidade Federal de Santa Maria, 2019. Disponível em: <https//www.even3.com.br/anais/2cab2019/224825- KIT-EDUCACIONALDE-CANSAT>. Acesso em: 15 Fev. 2021.

OUDJIDA, A.k. et al. **Design and test of general-purpose SPIMaster/Slave IPs on OPB bus**. 2010 7th International Multi- Conference OnSystems, Signals And Devices, [s.l.], v. 0, n. 0, p.0-0. IEEE.http://dx.doi.org/10.1109/ssd.2010.5585592. Acesso em: 13 Fev. 2021.

OVERMARS, Mark. **Programming Lego Robots using NQC**. 2002.Disponível em: <http://bricxcc.sourceforge.net/nqc/doc/NQC_Tutorial.pdf.Acesso em: 09 jan. 2021.

Spice. **Simulation Program with Integrated Circuit Emphasis**, 1973. Disponível em: < https://en.wikipedia.org/wiki/SPICE>. Acesso em: 10 jan. 2021.

Yasuyuki Miyazaki. **Cansat pico size artificial satellite**, 2016. Disponível em: < http://unisec.jp/library/i-cansat/manual_CanSat_textbook_eng_v5.pdf>. Acesso em: 10 jan. 2021

WIKIPEDIA. **Spice - Simulation Program with Integrated CircuitEmphasis**. Disponível em: < https://en.wikipedia.org/wiki/SPICE>. Acessoem: 10 jan. 2021.

ESTAÇÃO METEOROLÓGICA PARA CAPTAÇÃO DE DADOS CLIMÁTICOS DE UM PICO SATÉLITE

Árisla Bruna de Menezes Soares

ESTAÇÃO METEOROLÓGICA PARA CAPTAÇÃO DE DADOS CLIMÁTICOS DE UM PICO SATÉLITE

Árisla Bruna de Menezes Soares[1]

INTRODUÇÃO

Em vários projetos desenvolvidos nas mais diversas áreas acadêmicas, principalmente em algumas esferas da engenharia e na robótica, o uso dos sensores se faz presente e suas aplicações são bem interessantes. Eles permitem leituras de determinadas características do ambiente, tais como a temperatura de um motor, o nível de água de um reservatório, um obstáculo no percurso de um robô ou ainda o fato de uma porta estar fechada ou não. Através dessas leituras se cria um sistema capaz de interagir com o ambiente. Por mais simples que o cenário aparente ser, o equipamento é imprescindível para que os melhores e mais otimizados detalhamentos a respeito dos procedimentos na situação problema sejam efetuados da melhor forma.

O termo *CanSat* se refere a um modelo de pico satélite (0,1 e 1kg) colocado no interior de um invólucro com dimensões de uma lata de refrigerante comum. A palavra advém da junção do termo *Can* que significa lata em inglês com o vocábulo *Sat* que é uma abreviação da palavra satélite. Essa proposta de modelo de satélite foi inicialmente posta em 1998 no simpósio University Space Systems Symposium (USS - Havaí) sugerida pelo professor Bob Twiggs da Universidade de Stanford. A primeira sugestão era construir uma estrutura que fosse do tamanho de uma lata de refrigerantes e lançá-la no espaço.

Atualmente já existem diversas competições em diversos países pelo mundo. As competições consistem na construção de um pequeno satélite, com tecnologia semelhante, em partes, à usada em satélites miniaturizados.

[1] *Técnica em Mecatrônica pelo Instituto Federal de Ciência e Tecnologia do Rio Grande do Norte – Campus Parnamirim – arislanauri@gmail.com*

Os estudantes que participam têm a oportunidade de projetar, construir e testar diversos sistemas, mecanismos e instrumentos para viabilizar um protótipo funcional de um *CanSat*. E o objetivo é exatamente esse, reproduzir um cenário real de operação que potencializa a aplicação prática de conhecimentos adquiridos nas áreas das Ciências Espaciais. Alguns dos países que desenvolvem periodicamente um evento de disputa de mano satélite são a França, Holanda, Estados Unidos, Espanha, Noruega e Indonésia.

METODOLOGIA

A metodologia adotada para a construção do projeto consistiu na divisão em três partes principais, são elas: projeto preliminar e conceitual, projeto detalhado e projeto final. Partes estas que são executadas de forma ordenada e constituem o esqueleto do trabalho. No projeto preliminar procurou focar na parte conceitual necessário para a construção, desta forma foi apresentado o funcionamento de cada sensor baseado no seu respectivo datasheet, mostrando o seu princípio e funcionamento e as suas características específicas (como dimensão, tensão de alimentação, amostragem, entre outras). Identificar os requisitos funcionais e não funcionais e apresentar um esboço da ideia de forma que atenda aos requisitos científicos faz parte da etapa também composta pelo desenvolvimento das linhas básicas.

Na etapa seguinte, a do detalhamento, ocorreu o incorporamento da ideia e a materialização do produto. São apresentados a montagem do circuito, tanto em plataforma online de simulação quanto fisicamente na protoboard e a sua versão final na placa de circuito impresso, a produção dos códigos com as suas explicações necessárias. Essa fase é intermediária entre as abstrações da fase anterior e a versão final da fase posterior.

Na versão final é apresentada a integração da placa com o pico satélite (CanSat), a fim validar a qualidade do projeto e certificar seu funcionamento. Esta parte representa a interface entre o projeto preliminar e a construção do produto final.

FUNDAMENTAÇÃO TEÓRICA

Sensores

Pode-se definir a palavra sensor, literalmente, como “ aquilo que pode sentir”. Dentro da esfera eletrônica, o sensor é um dispositivo ou circuito eletrônico capaz de aferir e fazer uma análise de determinadas características do ambiente, podendo estar abrangendo variáveis mais simples como temperatura, umidade ou luminosidade; uma condição com um pouco mais de dificuldade como a medição da rotação de um motor ou ainda situações que fogem um pouco do nosso cotidiano como a detecção de partículas subatômicas. Mesmo com a imensa variedade de sensores eletrônicos atualmente, estes podem ser divididos basicamente em dois tipos, são eles: sensores analógicos e sensores digitais. Esta classificação fundamenta-se na forma como o componente responde a variação da condição.

Os sensores analógicos se caracterizam como os mais comuns e são assim denominados em virtude de possuírem sinais analógicos que por sua vez são aqueles que mesmo limitados entre dois valores de tensão, podem assumir infinitos valores intermediários. Ou seja, para cada valor da condição medida haverá um valor de tensão que o corresponde. Um exemplo desse tipo de sensor seria o LDR, onde a sua resistência cai gradativamente à medida que a luminosidade sobre ele aumenta. Por meio de um divisor de tensão é possível perceber que através dessa variação de resistência ocorre uma variação na tensão.

No que se refere aos sensores digitais, estes são baseados em níveis de tensões bem definidos. Esses níveis de tensão podem se apresentar como high (alto) ou low (baixo) ou ainda “0” ou “1”. Isso permite constatar que esses sensores se utilizam da lógica binária que constituem a base do funcionamento dos sistemas digitais e que são o oposto dos sensores analógicos, tendo em vista que não possuem valores intermediários de tensão. Como exemplo, tem-se um emissor e receptor infravermelho que configuram um par óptico, neste caso apenas duas condições/estados são possíveis.

Apresenta o nível baixo quando o feixe atinge o receptor e o nível alto quando algo bloqueia o percurso do feixe infravermelho. O desenvolvimento e aperfeiçoamento dos sensores juntamente com suas aplicações concederam inúmeras vantagens para a sociedade moderna. Vantagens que vão desde realizar uma pesquisa científica com maior precisão e confiabilidade em um curto espaço de tempo até ter uma maior segurança na sua residência.

Sensores do pico satélite

O CanSat faz parte da categoria de satélites miniaturizados - mais especificamente, de acordo com a classificação, um pico satélite possui uma massa que varia entre 100g e um 1kg - que são satélites artificiais com dimensão e massa reduzidas. Esta é uma categoria recente que visa reduzir o custo de produção que é bastante elevado para satélites. Dentro do subsistema de carga útil do pico satélite se encontram os sensores responsáveis pela captação dos dados climáticos. São eles: BMP180, o DHT11 e o MPU 6050, responsáveis pela aferição das seguintes variáveis atmosféricas: temperatura, umidade, pressão e altitude.

O BMP180 é um sensor usado para detectar pressão atmosférica e temperatura e através da manipulação dessas variáveis é possível se obter uma terceira, a altitude (em metros). É um sensor preciso, de baixo custo e apresenta um baixo consumo também. Utiliza-se da comunicação serial I2C e pode ser alimentado com tensões entre 3.3V e 16V, tendo como recomendável 5V.

O DHT11 é responsável pela aferição de dados de temperatura e umidade. É um sensor simples de ser usado, de baixo custo e possui uma tensão de alimentação que varia entre 3V e 5V. Ideal para medição de temperaturas entre 0°C e 50°C -com uma precisão de mais ou menos 2°C- e para medições de umidade na faixa de 20% e 80% apontando uma precisão variando entre 5%. Apresenta uma taxa de amostragem de 1 Hz, ou seja, uma leitura por segundo.

O MPU 6050 possui um acelerômetro de 3 eixos e um giroscópio também de 3 eixos, além disso a placa ainda apresenta um sensor de temperatura que permite leituras que vão de -40°C a 85°C. Possui alta precisão, utiliza-se para comunicação a interface I2C-por meio dos pinos SCL e DAS- e sua alimentação varia entre 3V e 5V, mas para uma melhor precisão recomenda-se usar 5V.

Microcontrolador

Um microcontrolador pode ser definido como um compilado de computador e chip, ele desempenha funções de computador no tamanho reduzido de um chip. Neste mesmo chip estão integrados uma CPU e alguns circuitos auxiliares como memória de dados, circuitos de clock, interface de comunicação, portas I/O, temporizadores, contadores e outros periféricos. De forma geral, essas placas são fáceis de programar, possuem um custo acessível e consomem pouca energia. A forma como o microcontrolador atua com demais equipamentos para alguma finalidade específica é caracterizada como integração e é intitulada como circuitos embarcados, que nada mais é do que aplicações com um microcontrolador que seja usado de forma exclusiva por elas.

Comumente utilizado no conjunto do dispositivo Arduino UNO, “ ATmega328P é um microcontrolador com CMOS de 8 bits de baixa potência, baseado na versão aprimorada do AVR” (RUSSEL, 2010, p.95). O AVR é um microcontrolador cuja arquitetura é Harvard, seguindo a linha do *Reduced Instruction Set Computer* ou Computador com um Conjunto Reduzido de Instruções (*RISC*), ou seja, opera com um conjunto de poucas instruções e rápida resposta.

O Arduino se constitui de uma plataforma de prototipagem eletrônica, usado como ferramenta de controle de entrada e saída de dados. Seus componentes básicos são, além do próprio microcontrolador, um cristal ou oscilador, um regulador linear de cinco volts, alguns pinos de entrada/ saída para conectar circuitos externos. A programação é feita utilizando a IDE (Integrated Development Environment, ou Ambiente de Desenvolvimento Integrado) do arduino, sua linguagem de programação tem origem Wiring e é essencialmente C/C++. Diante disso, para controlar o sistema será utilizado a plataforma Arduino e seu microcontrolador.

Para a recepção dos dados é necessário ter uma estação base, que neste caso será computador. A linguagem escrita na IDE se encarrega do processo de transferência da estação base para o microcontrolador Atmega328p. Feito isso, a interface utilizada para a apresentação dos dados se resume à própria interface fornecida pelo software do Arduino.

Sistema eletrônico embarcado

Os sistemas embarcados são desenvolvidos integrando-se hardware (processadores e circuitos eletrônicos) e firmware (software de baixo nível) para realizar tarefas específicas. Visto que o projeto em questão se refere a um sistema de telemetria (captando e transmitindo dados de pressão, temperatura e umidade) com o envio de dados via comunicação serial e dentre as muitas possibilidades de sistemas eletrônicos a serem executadas para o funcionamento do projeto, a solução buscada é a que atenda de forma mais simples a construção e implementação do protótipo.

Sendo assim, para a escolha do “cérebro” do circuito a placa microcontrolada Arduino foi definida, uma que possui circuitos de entrada/saída e que pode ser facilmente conectada à um computador e programada via IDE utilizando uma linguagem baseada em C/C++, sem a necessidade de equipamentos extras além de um cabo USB, figura 1.

Figura 1 - Arduino uno.

Fonte: FilipeFlop.

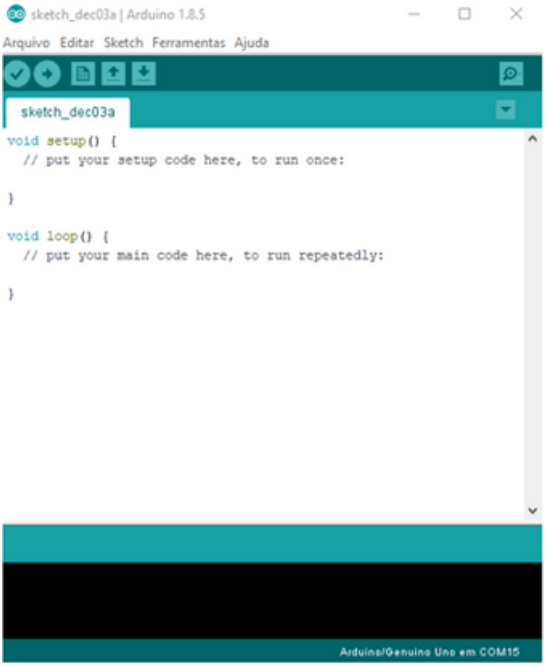
Software IDE Arduino

Para o Arduino ser carregado com a programação e consequentemente executada a ação, o IDE Arduino é utilizado.

Este software é um editor de texto que se conecta ao hardware do Arduino, verifica se a programação está correta e faz o upload para a placa correspondente.

Na figura 2 observa-se a interface do software sendo a parte branca onde a programação é desenvolvida e a parte preta as notificações a respeito da leitura do código.

Figura 2 - Interface de desenvolvimento IDE.



Fonte: Autoria própria.

A sintaxe da linguagem foi desenvolvida por Hernando Barragan, possui funções simples e específicas para trabalhar com as portas do Arduino. E basicamente são necessárias duas funções elementares para seu funcionamento: setup () e loop () (ARDUINO, 2016).

Projeto preliminar

Esta parte consiste na apresentação dos conceitos necessários para o desenvolvimento da parte posterior que se refere a montagem, confecção e programação do circuito. Sensor de pressão e temperatura BMP180.

Sensor de pressão e temperatura BMP180

O BMP180 consiste em um sensor piezo-resistivo, para robustez EMC, alta precisão e linearidade, como bem com estabilidade a longo prazo. Possui um conversor analógico-digital, uma unidade de controle com E^2PROM e uma interface I^2C serial, que permite fácil integração a sistemas microcontrolados, figura 3.

Figura 3 - Interface de desenvolvimento IDE.

Fonte: Filipeflop.

Os dados de pressão e temperatura são compensados pelos dados de calibração do E^2PROM[1] que armazena 176 bits de calibração individual, isso é usado para compensar deslocamento, dependência de temperatura e outros parâmetros do sensor. Os valores são guardados nas variáveis; UP = dados de pressão (16 a 19 bits) e UT = dados de temperatura (16 bits). O endereço do módulo é mostrado abaixo e LSB (least significant bit ou bit menos significativo) do endereço do dispositivo distingue entre a operação de leitura (1) e gravação (0), correspondente ao endereço 0xEF (leitura) e 0xEE (gravação).

Figura 4 - Endereço do módulo BMP180..

A7	A6	A5	A4	A3	A2	A1	W/R
1	1	1	0	1	1	1	0/1

Fonte:Datasheet sensor BMP180.

[1] É um tipo de memória não-volátil usada em computadores e outros dispositivos eletrônicos para armazenar pequenas quantidades de dados que precisam ser salvos quando a energia é removida.

Sensor de temperatura e umidade DHT11

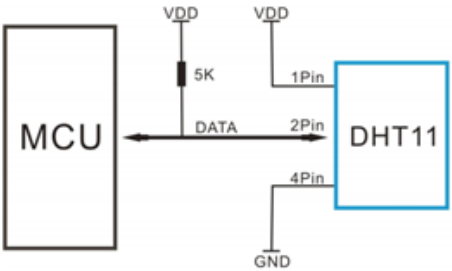
No sensor DHT11 cada elemento é estritamente calibrado em laboratório, extremamente preciso em calibração de umidade. Os coeficientes de calibração são armazenados como programas na memória OTP, que são usados pelo processo de detecção de sinal interno do sensor, figura 5.

Figura 5 - Sensor DHT11.

Fonte: Datasheet sensor DHT11.

A interface serial de fio único torna a integração do sistema rápida e fácil. Seu tamanho pequeno, baixo consumo de energia e até 20 transmissão de sinal do medidor, tornando-a a melhor escolha para várias aplicações, incluindo aquelas mais exigentes. O componente é um pacote de pinos de linha única de 4 pinos. É conveniente os pacotes connect e especiais podem ser fornecidos de acordo com a solicitação dos usuários, figura 6.

Figura 6 - Configuração de aplicação típica do sensor.



Fonte: Datasheet sensor DHT11.

Acelerômetros, giroscópio e sensor de temperatura MPU6050

Diferenciando-se dos demais equipamentos que possuem somente um sentido de medida, o MPU 6050 reúne duas funcionalidades em um único módulo, combinando acelerômetro e giroscópio, além de contar também com um sensor de temperatura incluso que permite leituras de -40 a 85°C, figura 7.

Figura 7 - Sensor MPU6050..

Fonte: Flip-flop..

De modo geral, um acelerômetro é um equipamento capaz de medir a aceleração de um corpo (como seu próprio nome já dá a entender) em relação à gravidade, ou seja, é um equipamento capaz de medir a aceleração exercida sobre determinados objetos, executando algum tipo de ação dependendo do movimento executado sobre o mesmo.

O giroscópio não possui uma funcionalidade radicalmente diferente do acelerômetro, ele é um sensor que utiliza da força da gravidade para indicar a posição de um determinado objeto no espaço, podendo identificar se você gira algo em seu próprio eixo ou saber se está apontando para cima ou para baixo. Dentro do protótipo, os módulos de acelerômetro e giroscópio ainda não foram explorados, utilizando-se apenas o módulo de temperatura.

DESENVOLVIMENTO

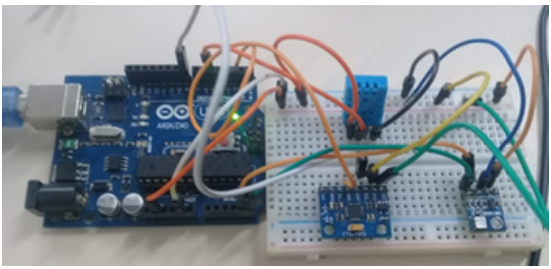
Elaboração do protótipo

A construção de protótipos propicia a verificação de detalhes bem como a avaliação de problemas em peças ou conjuntos, uma vez que podem ser analisados em instalações próprias para testes (PAHL et al., 2005). Sendo assim se faz relevante expressar que: “o protótipo é um modelo funcional que, embora seja construído por processos de fabricação artesanais, apresenta quase todas as características do produto final” (ROMEIRO FILHO et al., 2010, p. 301).

Oliveira Netto e Tavares (2006) citam que a construção do protótipo tem como principal finalidade propiciar o teste do produto em condições reais de operação; desta forma se faz necessário a construção de unidades que simulem uma fábrica piloto, com intuito de testar a interação entre produto e processo. O uso desta técnica auxilia na verificação de problemas, tornando possível resolvê-los antes que o produto seja lançado em larga escala.

Uma das etapas previstas neste trabalho foi a elaboração de um protótipo de uma estação meteorológica, que represente, dentro das limitações, um bom desempenho em diversas condições de operação. Para executar esta tarefa foi necessário aprender a usar os sensores e iniciar sua programação visando fazer as leituras das variáveis do ambiente, nesta etapa houve a montagem do circuito na matriz de contato (protoboard) e o início da programação na interface do Arduino. A montagem na protoboard pode ser vista na figura abaixo.

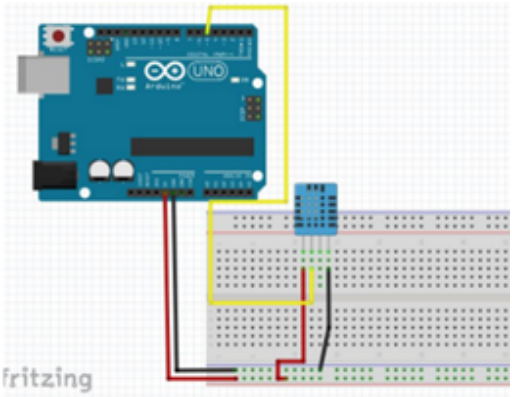
Figura 8 - Montagem física na protoboard.



Fonte: Autoria própria.

A etapa seguinte consiste na simulação do circuito, esta foi feita utilizando o software de simulação Fritzing, sua escolha se deu visto que já se tinha conhecimento do seu manuseio, figura 9 com o sensor DHT11.

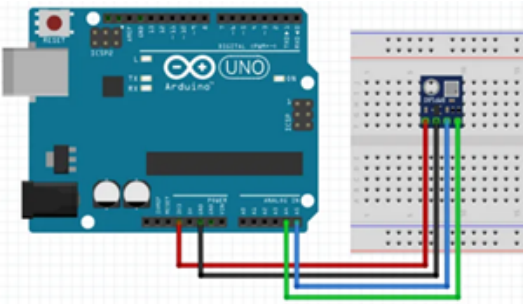
Figura 9 - Simulação DHT11.



Fonte: Autoria própria.

Na figura 10 com o sensor BMP180.

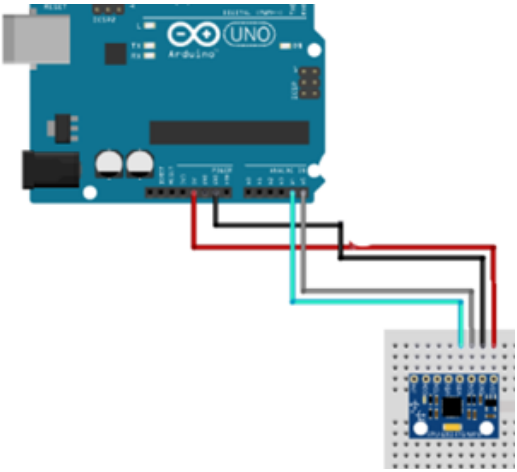
Figura 10 - Simulação BMP180.



Fonte: Autoria própria.

Na figura 11 com o sensor MPU6050.

Figura 11 - Simulação MPU6050.



Fonte: Autoria própria.

Posteriormente, com a necessidade de desenhar o circuito para confecção da placa de circuito impresso, utilizou-se o software comercial de simulação e módulos de projetos PROTEUS®, no qual foi desenhado o circuito eletrônico. Conforme pode ser visto na figura abaixo.

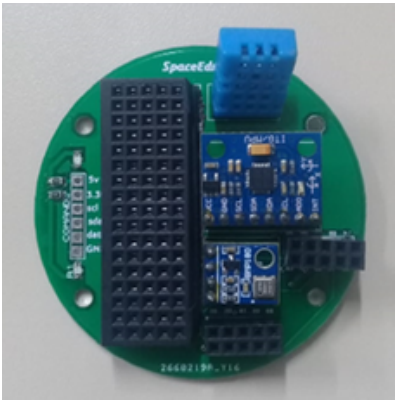
Figura 12 - Simulação BMP180.

Fonte: Autoria própria.

A etapa seguinte foi referente a confecção da placa de carga útil (estação meteorológica) e o teste de integração dela com as demais placas que compõem os subsistemas da plataforma Cansat. Uma vez que para o seu pleno funcionamento é necessária sua união a uma placa microcontrolada (que dentro do cansat é o subsistema referente ao computador de bordo que usa o microcontrolador Atmega328p) e a algo que forneça a tensão necessária, que neste caso é feita por meio da placa responsável pelo suprimento de potência que utiliza uma bateria de 9v.

A confecção da placa final se deu com uma empresa chinesa (LCSC Electronics) especializada nesse tipo de prototipagem, visto que pretendia-se atingir alta qualidade no material do protótipo. Após mais ou menos um mês e meio, as placas chegaram e a fim de finalizar o processo de confecção foi feita a soldagem dos sensores e da matriz de contatos expansível (que permite a inserção de outros componentes) na placa de circuito impresso, figura 13.

Figura 13 - Protótipo final.



Fonte: Autoria própria.

Programação

A programação do circuito é feita utilizando a linguagem do Arduino, que é baseada em C/C++. Através da exposição do código, percebe-se sua simplicidade visto que sua funcionalidade depende principalmente do uso de suas bibliotecas. Desse modo, sua aplicação é intuitiva dependendo de razoável conhecimento de programação básica. Os passos de resumem a inclusão das bibliotecas, declaração das variáveis, inicialização dos sensores e da comunicação serial, relação entre as leituras feitas pelo sensor e suas respectivas variáveis e por fim o comando para apresentação desses dados no monitor serial.

Para a programação dos sensores foi necessário utilizar as bibliotecas Adafruit_BMP085.h, DHT.h, e Wire.h(esta última para o sensor MPU6050). A primeira biblioteca citada em sequência foi utilizada valendo-se do sensor BMP180, visto esse ser totalmente compatível com a biblioteca supracitada. Na imagem abaixo tem-se o código utilizado para a captação dos dados da estação meteorológica.

Inicialmente, incluiu-se as bibliotecas (a maior parte do código utilizado em aplicações modernas é fornecido por estas bibliotecas) citadas anteriormente e define-se as variáveis do processo. Dentro da função void setup, inicializa-se os três sensores e a comunicação serial, para posteriormente fazer as correspondências entre as leituras dos sensores e as variáveis climáticas declaradas no início. Ademais, os valores lidos são exibidos na interface do Arduino. Abaixo tem-se o código completo

Figura 14 - Código final.

```
#include <Adafruit_BMP085.h>
#include <DHT.h>
#include<Wire.h>
#define pino_dht 5
#define DHTTYPE DHT11
DHT dht(pino_dht,DHTTYPE);
Adafruit_BMP085 bmp180;
float temp_dht= 0.0;
int umidade_dht = 0;
float temp_bmp = 0.0;
float pressao_bmp = 0.00;
float altitude_bmp;
//Endereco I2C do MPU6050
const int MPU=0x68;
int AcX,AcY,AcZ,Tmp,GyX,GyY,GyZ;
```

Figura 14 - Código final (continuação).

```
void setup()
{
  Serial.begin(9600);
  dht.begin();
  bmp180.begin();
  Wire.begin();
  Wire.beginTransmission(MPU);
  Wire.write(0x6B);

  //Inicializa o MPU-6050
  Wire.write(0);
  Wire.endTransmission(true);
}

void loop()
{
temp_dht = dht.readTemperature();
umidade_dht = dht.readHumidity();
temp_bmp = bmp180.readTemperature();
pressao_bmp = bmp180.readPressure();
altitude_bmp = bmp180.readAltitude(101500);
Wire.beginTransmission(MPU);
  Wire.write(0x3B);  // starting with register 0x3B (ACCEL_XOUT_H)
  Wire.endTransmission(false);
  //Solicita os dados do sensor
  Wire.requestFrom(MPU,14,true);
  //Armazena o valor dos sensores nas variaveis correspondentes
  AcX=Wire.read()<<8|Wire.read();  //0x3B (ACCEL_XOUT_H) & 0x3C (ACCEL_XOUT_L)
  AcY=Wire.read()<<8|Wire.read();  //0x3D (ACCEL_YOUT_H) & 0x3E (ACCEL_YOUT_L)
  AcZ=Wire.read()<<8|Wire.read();  //0x3F (ACCEL_ZOUT_H) & 0x40 (ACCEL_ZOUT_L)
  Tmp=Wire.read()<<8|Wire.read();  //0x41 (TEMP_OUT_H) & 0x42 (TEMP_OUT_L)
  GyX=Wire.read()<<8|Wire.read();  //0x43 (GYRO_XOUT_H) & 0x44 (GYRO_XOUT_L)
  GyY=Wire.read()<<8|Wire.read();  //0x45 (GYRO_YOUT_H) & 0x46 (GYRO_YOUT_L)
  GyZ=Wire.read()<<8|Wire.read();  //0x47 (GYRO_ZOUT_H) & 0x48 (GYRO_ZOUT_L)
delay(1000);
  Serial.print(" | AcY = ");
  Serial.print(AcY);
  Serial.print("Y=");
  Serial.print(AcY);
  Serial.print(" | AcZ = ");
  Serial.print(AcZ);
  Serial.print("Z=");
  Serial.print(AcZ);
  //Calcula a temperatura em graus Celsius
  Serial.print(" | Tmp = ");
  Serial.print(Tmp/340.00+36.53);
  Serial.print("T:");
  Serial.print(Tmp/340.00+36.53);
  Serial.print(" | GyX = ");
  Serial.print(GyX);
  Serial.print("X=");
  Serial.print(GyX);
  Serial.print(" | GyY = ");
  Serial.print(GyY);
  Serial.print("Y=");
  Serial.print(GyY);
  Serial.print(" | GyZ = ");
  Serial.println(GyZ);
  Serial.print("Z=");
  Serial.print(GyZ);
  Serial.println("Dados de temperatura:");
  Serial.print("Dht11 = ");
  Serial.println(temp_dht);
  Serial.print("Bmp180: ");
  Serial.println(temp_bmp);
```

Figura 14 - Código final (continuação).

```
  Serial.print("Dado de umidade: ");
  Serial.print(umidade_dht);
  Serial.println("%");
  Serial.print("Dado de pressao:");
  Serial.println(pressao_bmp);
  Serial.print("Dado de altitude:");
  Serial.println(altitude_bmp);
delay(10000);
}
```

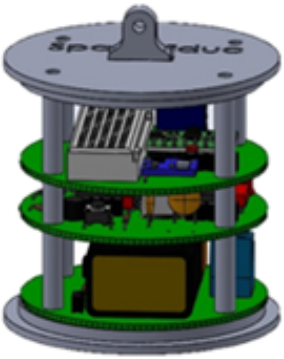
Fonte: Autoria própria.

PROJETO VERSÃO FINAL

Integração aos subsistemas do CanSat

O Cansat, como já foi apresentado, é um pico satélite, que dentro do volume reduzido de uma latinha de refrigerante comporta um circuito eletrônico que tem o intuito de representar os principais subsistemas de um satélite, são eles: computador de bordo, carga útil e suprimento de potência. A divisão desses subsistemas se dá em três estágios, a estação meteorológica do presente projeto corresponde a um deles e confere ao CanSat sua função meteorológica. Na primeira imagem abaixo é possível ver a integração dos três subsistemas juntamente com sua estrutura física, figura 15.

Figura 15 - Integração dos estágios.



Fonte: Autoria própria.

Na segunda, tem-se a versão final das três placas também de forma integrada, figura 16.

Figura 16 - Versão final do CanSat.

Fonte: Autoria própria.

CONSIDERAÇÕES FINAIS

Com seu desenvolvimento foi possível adquirir novos conhecimentos, correlacionar com muitos vistos no decorrer dos quatro anos de curso e concretizar a ideia proposta. Diante do exposto, a utilização do protótipo se aplica tanto na esfera acadêmica, proporcionando o ensino da tecnologia espacial, visto que a placa representa um importante subsistema do satélite e nesse sentido pode ser melhor explorada dentro do contexto espacial, com também pode ser uma ferramenta de análise do clima (em pequena escala) da região. Uma vez que os sensores apresentam boa acurácia e precisão e sua aquisição se caracteriza por ser de baixo custo.

Em suma, o projeto permite uma boa possibilidade de exploração em diversas áreas e seu desenvolvimento acarreta a interdisciplinaridade de áreas como Astronomia, Astronáutica, Robótica e Programação. Essa plataforma científica poderá ser utilizada por alunos posteriores na realização de atividades experimentais com ênfase na área aeroespacial.

REFERÊNCIAS BIBLIOGRÁFICAS

ARDUINO Uno. Disponível em: https://www.google.com/url?sa=t&source=web&rct=j&url=https://datasheet.octopart.com/A000066-Arduino-datasheet-38879526.pdf&ved=2ahUKEwi70JrJ_ezwAhXCFbkGHcyFDGUQFjACegQIBRAG&usg=AOvVaw2RtZx5wjCqa_FRBoTN0YM-. Acesso em: 10 maio 2021.

ARDUINO, Disponível em: https://www.google.com/url?sa=t&source=web&rct=j&url=https://datasheet.octopart.com/A000066-Arduino-datasheet-38879526.pdf&ved=2ahUKEwi70JrJ_ezwAhXCFbkGHcyFDGUQFjACegQIBRAG&usg=AOvVaw2RtZx5wjCqa_FRBoTN0YM-. Acesso em: 10 de Março de 2016.

BOSCH. BMP180: Digital pressure sensor. 2013. Disponível em: https://www.google.com/url?sa=t&source=web&rct=j&url=https://cdn-shop.adafruit.com/datasheets/BST-BMP180-DS000-09.pdf&ved=2ahUKEwjvleuo_ezwAhXSBtQKHVpsCqcQFnoECAYQAQ&usg=AOvVaw2GVPbzAj6_NxVb-N7lcGQe. Acesso em: 20 jul. 2019.

DHT11: Temperature & Humidity Sens. 2010. Disponível em:https://www.google.com/url?sa=t&source=web&rct=j&url=https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf&ved=2ahUKEwi62daQ_ezwAhVMIbkGHYoqDs0QFnoECDQQAQ&usg=AOvVaw2tdFgaeZgeQmJr3SkJiugP. Acesso em: 19 Set. 2019.

MPU6050: Product Specification. 2013. Disponível em: https://www.google.com/url?sa=t&source=web&rct=j&url=https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf&ved=2ahUKEwjl4vLa_OzwAhVnIbkGHf6OBeAQFnoECAkQAQ&usg=AOvVaw2WYsuQaLpbEJi8zeUd4k8E. Acesso em: 10 jul. 2019.

ROMEIRO FILHO, Eduardo (Coord.). **Projeto de produto**. Rio deJaneiro, RJ: Elsevier, 2010. (Coleção Campus - ABEPRO;Engenharia de produção).

RUSSELL, David. **Introduction to Embedded Systems**: UsingANSI C and the Arduino Development Environment. Nebraska:University Of Nebraska-lincoln, 2010.

DESENVOLVIMENTO DE UM PROTÓTIPO DE SATÉLITE METEOROLÓGICO EM FORMA DE CUBO - CUBESAT

Maria Eduarda Lima da Luz

DESENVOLVIMENTO DE UM PROTÓTIPO DE SATÉLITE METEOROLÓGICO EM FORMA DE CUBO - CUBESAT

Maria Eduarda Lima da Luz[1]

INTRODUÇÃO

No Brasil, dentro da estrutura organizacional da Agência Espacial Brasileira (AEB), foi idealizado e criado o primeiro Centro Espacial Brasileiro de Educação baseada em STEM, sigla em inglês para Science, Technology, Engineering and Mathematics (Ciência, Tecnologia, Engenharia e Matemática). O Centro Vocacional Tecnológico Espacial (CVT-Espacial), situado no Rio Grande do Norte (RN), dentro do Centro de Lançamento da Barreira do Inferno (CLBI), ele tem sido um veículo educacional importante para a inclusão social viabilizada pela aplicação das ciências espaciais e atividades STEM. No CVT-Espacial são desenvolvidas pesquisas e ministradas aulas dentro no espectro das ciências espaciais, destacando os temas de foguetes, satélites, sensoriamento remoto, veículos de exploração espacial e meteorologia, como afirma o professor de Mecatrônica José Soares (SOARES, 2018).

Levando em consideração que a comunidade escolar que é atendida pelo CVT-Espacial não realiza experimentos na área de satélites meteorológicos, foi observada a importância da criação e simulação de um protótipo de um satélite, utilizando sensores de baixo custo, para facilitar o ensino sobre dados atmosféricos, assim fazendo com que os alunos tenham a oportunidade de sair da teoria e possam ver na prática como a captação das variáveis funciona

[1]*Técnica em Mecatrônica pelo Instituto Federal de Ciência e Tecnologia do Rio Grande do Norte – Campus Parnamirim – marialluz1912@gmail.com*

A proposta é criar um instrumento simples de baixo custo para auxiliar no ensino sobre dados meteorológicos, no CVT-Espacial, englobando outras áreas como: eletrônica, comunicação e modelagem 3D. Dessa forma os alunos vão poder se aprofundar não apenas na meteorologia, como também no entendimento dos sensores, no funcionamento da transmissão e coleta de dados, e vão ter a oportunidade de conhecer softwares de simulação para a modelagem de protótipos. Consequentemente, o esperado é que os estudantes se sintam estimulados para criarem projetos na área aeroespacial, uma vez que tal setor vem ganhando destaque no âmbito nacional e propicia uma experiência científica para crianças e jovens.

METODOLOGIA

Este capítulo aborda sobre o desenvolvimento do protótipo de um satélite meteorológico em forma de cubo, o qual foi realizado da seguinte forma: 1a etapa - Compreender o funcionamento dos sensores primordiais, existentes em um satélite meteorológico; 2a etapa – Escolha dos sensores de baixo custo para medir temperatura, pressão atmosférica, umidade relativa do ar e altitude; 3a etapa - Simulação de cada sensor individualmente; 4a etapa - Montagem do projeto, com a integração dos sensores; 5a etapa - Análise e discussão sobre a precisão dos dados aferidos.

FUNDAMENTAÇÃO TEÓRICA

As alterações no clima de todo o mundo têm causado fatores como o aumento da temperatura da terra, aumento do nível do mar, escassez de água, favorecimento de eventos climáticos extremos como inundações, e por último, danos à saúde devido os eventos extremos, como as secas e ondas de calor, que podem favorecer a disseminação de doenças, devido ao aumento da poluição do ar, os mosquitos que crescem em locais quentes podem agravar as doenças respiratórias.

Ao entender como as mudanças climáticas ocorrem e o motivo de estarem ocorrendo, permite amenizar, reverter seus efeitos e até mesmo preparar a população caso venha ocorrer grandes eventos climáticos. Tudo isso só é possível graças à meteorologia.

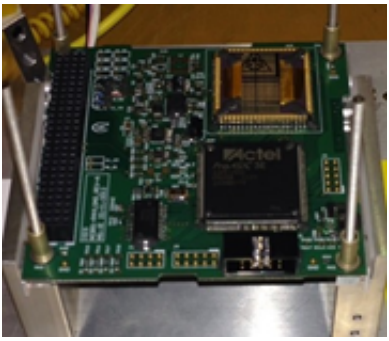
Um Satélite Meteorológico, segundo o INPE, é usado para monitorar o tempo e o clima da Terra.

Os satélites são dispositivos de alto custo devido a possuírem sistemas extremamente complexos (CHIN et al., 2016). Eles costumam ser divididos em alguns elementos, entre eles, computador de bordo, carga útil, suprimento de potência e telecomunicações. À vista disso, o principal desafio foi construir um protótipo que simulasse todos os seus subsistemas, sem elevar o custo dos equipamentos. O protótipo de um satélite meteorológico em forma de cubo, comparado com satélites reais, tem dimensões reduzidas com massa na faixa de 100g a 1kg.

DESENVOLVIMENTO

A carga útil de um satélite é responsável pelos subsistemas relacionados ao seu objetivo. O protótipo de um satélite meteorológico em forma de cubo, por ter objetivo de tornar a ciência acessível para todos, utiliza sensores de baixo custo que captam dados atmosféricos, como temperatura, pressão, umidade, altitude, entre outros. É importante ressaltar a importância da redundância na construção de um satélite, caso algum sensor venha a falhar ainda teremos os valores fornecidos pelos outros (CHIN et. al., 2017).

Figura 1 - Carga útil desenvolvida no INF do NanoSatC-BR1.



Fonte: Fonte: INF UFRGS (2020).

Para a carga útil do protótipo, foram feitas muitas pesquisas comparando precisão e custo dos sensores atmosféricos do mercado atual, chegou-se a 3 sensores: DHT11, BMP180 e MPU6050.

O DHT11 é um sensor que mede as variáveis de temperatura e umidade, que permite fazer leituras de temperaturas entre 0 a 50 Celsius e umidade entre 20 a 90%. Esse sensor possui um componente NTC para temperatura e componente medidor de umidade, ambos conectados a um controlador de 8-bits. Pode-se observar o sensor na figura à esquerda. Cada sensor tem pinos específicos para suas funções. Na figura à direita, pode-se notar a finalidade de três dos quatro pinos do DHT11, já que o pino 3 não é utilizado, figura 2.

O primeiro pino é o VCC, onde o sensor é conectado à tensão fornecida por uma fonte. Já o segundo pino, é onde acontece a troca de dados do sensor com o microcontrolador. O quarto pino é o pino GND, que tem o objetivo de garantir um valor comum a todos os pinos.

Figura 2 - Sensor DHT11.

Fonte: Filipeflop(2020).

O módulo BMP180 é um sensor que mede as variáveis de temperatura e pressão atmosférica, e baseado nesses dados é possível determinar a altitude e realizar previsões do tempo com grande precisão. Pode-se observar o sensor na figura à esquerda. Na figura à direita, pode-se notar a finalidade dos quatro pinos do BMP180. O primeiro pino é o VIN, onde o sensor recebe a tensão de entrada fornecida por uma fonte. Já o segundo pino é o GND, que garante um valor comum a todos os pinos.

Os pinos 3 e 4 (SCL e SDA) são analógicos, e são responsáveis pela troca de dados do sensor com o microcontrolador, figura 3.

Figura 3 - Sensor BMP180.

Fonte: Filipeflop(2020).

O sensor MPU6050 possui um acelerômetro e giroscópio, e contém os dois em um único chip do tipo MEMS. São 3 eixos para o acelerômetro e 3 eixos para o giroscópio, sendo ao todo 6 graus de liberdade (6DOF). Além disso, o sensor também capta a variável temperatura, permitindo medições entre -40 e +85 ºC. Pode-se observar o sensor na figura à esquerda. Na figura à direita, pode-se notar a finalidade dos oito pinos do MPU6050. O primeiro pino é o VCC, que fornece tensão para o módulo, entre 3 a 5 volts. Já o segundo pino é o GND, que garante um valor comum a todos os pinos.

Os pinos 3 e 4 (SCL e SDA) são analógicos, e são responsáveis pela troca de dados do sensor com o microcontrolador. Os pinos 5 e 6 (XDA e XDL) são opcionais e servem para fazer interface com outros módulos. O sétimo pino é o ADO, o qual serve para variar o endereço caso mais de um MPU seja utilizado em conjunto. O oitavo e último pino é o INT, pino de interrupção para indicar que os dados estão disponíveis para leitura, figura 4.

Figura 4 - Sensor MPU6050.

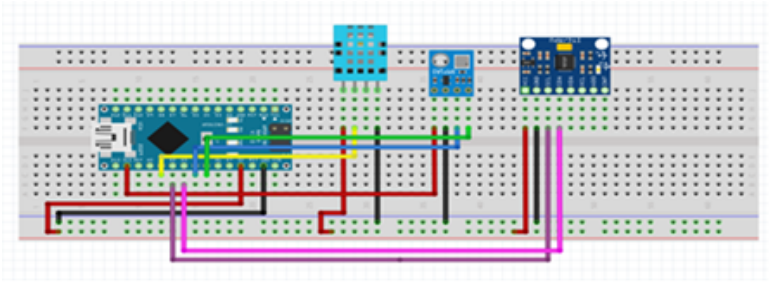
Fonte: Filipeflop(2020).

Como o protótipo do Pico Satélite utiliza os três sensores simultaneamente, foi preciso fazer a integração deles para verificar seus respectivos funcionamentos. Na figura abaixo, pode-se observar a montagem no software Fritzing.

Como existem pinos utilizados em comum pelos 3 sensores, o barramento de apoio da protoboard foi extremamente importante para garantir a conexão de todos os sensores com o Arduino Nano, tanto para o pino GND, como o 5 Volts, este último utilizado pelo DHT11 e MPU6050.

Os demais pinos dos sensores, de dados, foram conectados normalmente respeitando o máximo de uma ligação em cada pino analógico do Arduino, nesse caso os pinos utilizados foram A1, A2, A3, A4 e A5, figura 5.

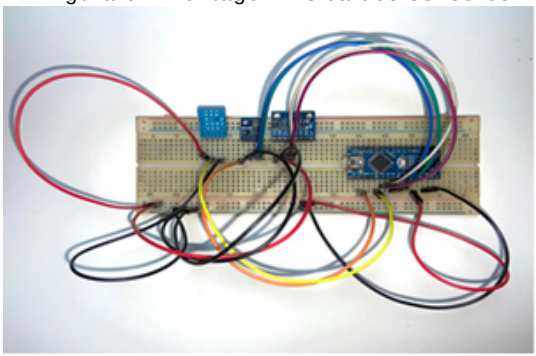
Figura 5 - Montagem no Fritzing.



Fonte: Elaborado pela autora (2020).

Na figura 6 é possível ver a montagem dos sensores reais, que foi realizada seguindo o mesmo esquema da montagem no Fritzing.

Figura 6 - Montagem física dos sensores.



Fonte: Elaborado pela autora (2020).

Todo satélite precisa de um componente que controle todos os demais, tal componente é chamado de computador de bordo, o qual tem uma extrema importância pois ele é responsável por gerenciar todos os subsistemas. Faz parte desse subsistema: microcontrolador, ambiente de programação e armazenamento.

No protótipo de um satélite meteorológico em forma de cubo, o computador de bordo utilizado é o microcontrolador Arduino Nano, figura 7, o qual foi escolhido devido ao seu tamanho reduzido. A linguagem de programação utilizada é a própria do Arduino, semelhante a linguagem C, capaz de produzir códigos compactos, como afirma Charles Borges (LIMA, 2012).

Figura 7 - Arduino Nano.

Fonte: Filipeflop(2020).

Para que todo o circuito funcione positivamente, foi necessário desenvolver um programa integrando todos os sensores e suas respectivas bibliotecas. O código foi desenvolvido na própria IDE do Arduino baseado na linguagem C++, com apenas algumas modificações.

A IDE Arduino suporta as linguagens C e C ++ usando rotas especiais de estruturação de código. Essa IDE fornece uma biblioteca de software, que fornece procedimentos de entrada e saída comuns. O código escrito pelo usuário requer apenas duas funções básicas, para iniciar o esboço e o loop do programa principal, que são compilados e vinculados ao programa principal.

A IDE do Arduino emprega o programa para converter o código executável em um arquivo de texto em codificação hexadecimal que é carregado na placa Arduino por um programa carregador no firmware da placa, por meio de um cabo USB conectado ao computador.

Na figura 8A observa-se as bibliotecas dos sensores sendo incluídas, como também os pinos sendo definidos. Já na figura 8B, estão presentes as variáveis sendo definidas.

Figura 8 - A - Biblioteca e B - Variáveis.

Cubesatt §

```
#include <DHT.h> // biblioteca do DHT11
#include <Wire.h> // biblioteca BMP180 e MPU6050
#include <Adafruit_BMP085.h>
#define DHTPIN A1 // pino do dht conectado ao Arduino
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);
Adafruit_BMP085 bmp;
```

Cubesatt §

```
const int MPU=0x68;
float umidade = 0;
float temperatura_dht = 0;
float temperatura_bmp = 0;
float altitude = 0;
float pressao = 0;
int Tmp;
```

Fonte: Elaborado pela autora (2020).

Na figura 9 observa-se o *void setup( )*, que é uma função obrigatória do Arduino, toda linha de código que estiver inclusa nesse bloco irá executar uma vez sempre no início do programa.

Figura 9 - Void setup( ).

Cubesatt §

```
void setup() {
  Serial.begin(9600); // Inicia o monitor serial
  //Inicia os sensores
  dht.begin();
  bmp.begin();
  Wire.begin();
  Wire.beginTransmission(MPU);
  Wire.write(0x6B);
  Wire.write(0); // Inicia o MPU6050
  Wire.endTransmission(true);
}
```

Fonte: Elaborado pela autora (2020).

Já na seguinte, observa-se o *void loop( )*, que é a outra função obrigatória do Arduino, assim que o *void setup()* é executado, o *void loop( )* é inicializado e executa da primeira linha até a última inserida nas chaves, assim que a última é executada, o programa volta para a primeira linha do loop e permanece dessa forma para sempre ou até que o usuário faça o upload de outra versão código.

A função *void loop()*, figura 10.

Figura 10 - Void loop( ).

```
Cubesat1§
void loop() {

  Wire.beginTransmission(MPU);
  Wire.write(0x3B);
  Wire.endTransmission(false);
  Wire.requestFrom(MPU,14,true);

  temperatura_dht = dht.readTemperature();
  Serial.print("Temperatura DHT11: ");
  Serial.print(temperatura_dht);
  Serial.println("°C");
  temperatra_bmp = bmp.readTemperature();
  Serial.print("Temperatura BMP180: ");
  Serial.print(temperatra_bmp);
  Serial.println("°C");
  Tmp = Wire.read()<<8|Wire.read();
  Serial.print("Temperatura MPU6050: ")
  Serial.print(Tmp/340.00+36.53);
  Serial.println("°C");
  umidade = dht.readHumidity();
  Serial.print("Umidade: ");
  Serial.print(umidade);
  Serial.println("%");
  pressao = bmp.readPressure();
  Serial.print("Pressão Atmosférica: ");
  Serial.print(pressao);
  Serial.println("Pa");
  altitude = bmp.readAltitude(101500);
  Serial.print("Altitude: ");
  Serial.print(altitude);
  Serial.println("m");
  delay(8000);
}
```

Fonte: Elaborado pela autora (2020).

Para o Arduino funcionar sem estar conectado via USB a um computador, é preciso conectá-lo a uma bateria externa, dessa forma o microcontrolador é alimentado e envia corrente elétrica para os demais sensores para que possam desempenhar suas respectivas funções.

Para isso, utiliza-se uma bateria comum, facilmente encontrada em supermercados, contendo 9 Volts, porque é uma tensão que o Arduino suporta e é aproximadamente a quantidade necessária para alimentar todo o circuito, figura 11.

Figura 11 - Bateria 9V.

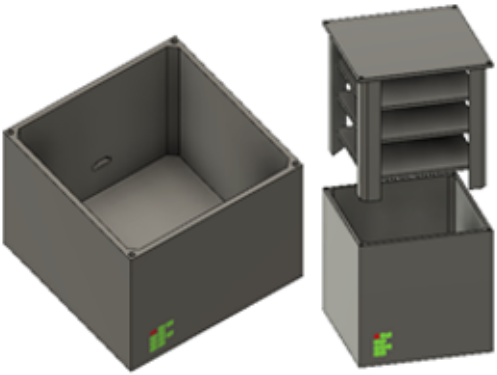
Fonte: Zenak Papelaria (2020).

A estrutura de um satélite é responsável por assegurar a integridade de todos os outros subsistemas, dessa forma é necessário que seja segura e resistente.

Utilizando como base os modelos de CubeSats já existentes, o protótipo foi idealizado em forma de cubo de 10 centímetros de aresta, o qual foi impresso por uma impressora 3D.

Para integrar todos os subsistemas do protótipo, utilizou-se o Fusion 360 para a modelagem 3D. A modelagem é em forma de cubo e possui duas estruturas: estrutura móvel, a qual possui três prateleiras para facilitar a divisão dos subsistemas (carga útil, computador de bordo e suprimento de potência), e a estrutura fixa, que serve como encaixe para a primeira, figura 12.

Figura 12 - Modelagem 3D do protótipo de satélite em forma de cubo.



Fonte: Elaborado pela autora (2020).

Para a impressão 3D, utilizou-se o material ABS, um polímero muito comum para confecção de estruturas nesse tipo de impressora, figura 13.

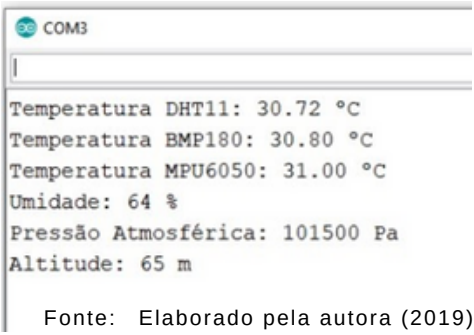
Figura 13 - Protótipo de satélite em forma de cubo impresso.

Fonte: Elaborado pela autora (2019).

ANÁLISE DOS RESULTADOS

A partir do circuito e da programação elaborada, obteve-se o resultado de todo o trabalho e dedicação: os valores captados por cada um dos sensores, como está apresentado na figura a seguir, com isso pode-se concluir que o projeto cumpriu seus objetivos e está em seu melhor funcionamento, figura 14.

Figura 14 - Porta serial da IDE do Arduino.



Fonte: Elaborado pela autora (2019).

Em relação aos objetivos gerais, o projeto teve como principal objetivo a construção de um Pico Satélite com funções meteorológicas, com a utilização de sensores de baixo custo e fácil acesso. Os resultados esperados eram o funcionamento de todos os seus subsistemas como carga útil, computador de bordo, suprimento de potência e sua estrutura mecânica.

Com o uso da programação em Arduino os sensores corresponderam aos resultados esperados captando dados como umidade, temperatura, pressão e realizando o cálculo da altitude. Pensando na próxima etapa do projeto, observa-se a necessidade de realizar testes com o Pico Satélite para verificar sua resistência, tais testes podem ser realizados no CVT-E, onde tem todos os equipamentos necessários para essas tarefas

CONSIDERAÇÕES FINAIS

O projeto da construção do satélite meteorológico em forma de cubo com a utilização de sensores de baixo custo propiciou a ampliação dos conhecimentos nas áreas: aeroespacial, meteorológica, eletrônica e modelagem 3D. Para utilizar os sensores empregados foi necessário um estudo da conexão elétrica e eletrônica específica de cada um deles, aprofundando o conhecimento de eletroeletrônica e aprimorando a confecção de placas de circuito impresso. Com parte de desenvolvimento da estrutura mecânica foi proporcionado o uso de softwares de modelagem 3D, como o Fusion 360. Referente à área da Meteorologia Aeroespacial foi despertado o interesse por esse setor e uma variedade de conhecimentos foram adquiridos como a importância da análise das condições atmosféricas antes de operações, conceitos relacionados a variáveis climáticas, sistema meteorológico brasileiro, entre outros.

REFERÊNCIAS BIBLIOGRÁFICAS

CHIN, J. et. al.. Basic Concepts and Processes for First-Time CubeSat Developers. NASA CubeSat Launch Initiative, California Polytechnic State University, San Luis Obispo (Cal Poly) CubeSat Systems Engineer Lab - Outuber, 2017.

LIMA, Charles Borges de. AVR e Arduino: técnicas de projeto. 2. ed. Florianópolis: Ed. dos autores, 2012.

PROJETO CVT-E. Disponível em: < https://parnamirim.rn.gov.br/newsItem.jsp?p=8477>Bem Paraná. Meteorologia: sua origem e evolução no tempo. Disponível em: https://www.bemparana.com.br/noticia/meteorologia-sua-origem-e-evolucao-no-tempo-#.XqtPkERv80M. Acesso em: 19 dez. 2020.

APLICAÇÃO DE FONTE DE ENERGIA ALTERNATIVA EM UM PROTÓTIPO DE SATÉLITE EM FORMA DE CUBO – CUBESAT

Natália Pereira dos Santos

APLICAÇÃO DE FONTE DE ENERGIA ALTERNATIVA EM UM PROTÓTIPO DE SATÉLITE EM FORMA DE CUBO – CUBESAT

Natália Pereira dos Santos[1]

INTRODUÇÃO

É perceptível que, após a Revolução Industrial iniciada no final do século XVIII, a busca por novas fontes de energia se tornou imprescindível para a humanidade visto que o impacto ambiental acarretado pelo contínuo uso das fontes convencionais (petróleo, carvão mineral, gás natural) vem aumentando significativamente através dos anos. O processo de combustão de derivados do petróleo bem como dos demais combustíveis fósseis, lança na atmosfera gases poluentes, como o dióxido de carbono, que implicam diretamente na destruição da camada de ozônio, no aquecimento do planeta em decorrência da intensificação do efeito estufa, na chuva ácida e na alteração climática em diversas regiões (GOLDENBERG e LUCON, 2006-2007).

Por isso o desenvolvimento de fontes alternativas de energia, como a energia eólica, nuclear, maremotriz e geotérmica, que utilizem recursos renováveis como forma de diminuir os danos ambientais é fundamental para a evolução da sociedade humana.

O campo aeroespacial foi um dos pioneiros a empregar a energia solar através de painéis fotovoltaicos, visto que eles se mostraram fundamentais e eficientes para recarregar baterias e suprir sistemas complexos como satélites. O Vanguard 1, satélite artificial norte-americano lançado em 1958, foi o primeiro a ser energizado por painéis solares, e, desde então, a energia fotovoltaica tem sido massivamente implementada no setor espacial.

[1] *Técnica em Mecatrônica pelo Instituto Federal de Ciência e Tecnologia do Rio Grande do Norte – Campus Parnamirim – nataliapereira150@gmail.com*

O CubeSat é um satélite de porte reduzido classificado como um pico satélite. O nome origina-se da união da palavra cube(cubo, em inglês) com o termo sat -iniciais da palavra satélite-. O formato cúbico deve apresentar 10 centímetros de aresta para configurá-lo como uma unidade chamada 1U, a qual possui massa de até 1,33 quilograma, podendo ser combinada para construir satélites de capacidade maior (2U, 3U…) (VILLELA, BRANDÃO, LEONARDI, 2016). A proposta do CubeSat foi iniciada em 1999 pelos professores Jordi Puig-Suari da California Polytechnic State University e Bob Twiggs da Universidade Stanford como um projeto acadêmico de cunho educacional que proporcionasse a estudantes uma experiência científica de desenvolvimento e simulação de uma missão espacial real. (MESSIER,2015).

METODOLOGIA

Este capítulo se debruça sobre a simulação e integração de placas solares em um pico satélite, de modo que funcionem como suprimento de potência utilizado para alimentar sensores eletrônicos de baixo custo que constituem a carga útil no interior do satélite. É exposto, brevemente, os referidos sensores e suas funções, assim como o computador de bordo, o modo de comunicação e a programação envolvida para a coleta e transmissão de dados atmosféricos.

Além disso é abordado a análise e estudo de três partes: montagem da carga útil do CubeSat, captação de dados meteorológicos dos sensores e por último o acoplamento das placas solares como forma de suprimento de potência. Na montagem da carga útil é possível compreender as especificidades de cada sensor bem como realizar a conexão eletrônica entre eles em um ambiente de simulação online.

Em seguida, é demonstrada a programação desenvolvida no IDE -Integrated Development Environment- do Arduino e executada a coleta dos dados atmosféricos fornecidos pelos sensores de modo a verificar a funcionalidade deles. Por fim, as placas solares são simuladas no circuito eletrônico como a fonte alternativa de energia para o funcionamento regular do protótipo de CubeSat.

FUNDAMENTAÇÃO TEÓRICA

Assim como os demais satélites, o CubeSat também é dividido em subsistemas que favorecem o arranjo espacial dos componentes em seu interior (BARBOSA, 2017). O subsistema da carga útil apresenta os sensores que configuram a função meteorológica do pico satélite. São eles: o BMP180, o MPU 6050 e o DHT11, responsáveis pela captação de dados atmosféricos e climáticos como temperatura, umidade, pressão e altitude.

Já o subsistema do computador de bordo conta com a placa de prototipagem Arduino UNO, uma vez que ela reúne hardware e software a fim de realizar tarefas pré-definidas e específicas. Dessa forma, é possível diminuir os recursos computacionais, as dimensões e o valor final da aplicação eletrônica, conferindo uma construção e implementação mais simples do projeto. Outra vantagem advinda da utilização do Arduino é a possibilidade de conexão a um computador por intermédio de um cabo USB e a programação é feita através de um ambiente de desenvolvimento integrado com uma linguagem embasada em C/C ++ (SLUZALA, 2017).

O Arduino UNO emprega o ATmega328P, “Um microcontrolador CMOS de 8 bits de baixa potência, baseado na versão aprimorada do AVR” (RUSSEL, 2010, p.95). Os microcontroladores são amplamente utilizados em equipamentos eletrônicos devido ao seu reduzido tamanho, fácil programação, baixo consumo energético e um ótimo desempenho que lhe confere um bom custo-benefício (KERSCHBAUMER, 2018). Eles são fabricados no formato de um chip, o qual possui, de forma integrada, uma CPU, memórias de programas e de dados, pinos I/O, circuitos de clock, pinos de 15 alimentação resistores, contadores, temporizadores entre outros componentes que auxiliarão nas execuções de tarefas.

O ATmega328P confere uma rápida velocidade de processamento dada sua microarquitetura baseada no modelo RISC -Reduced Instruction Set Computer-, reduzindo o tempo de resposta devido ao baixo número de instruções (SLUZALA, 2017).

A placa microcontrolada Arduino também com um oscilador, pinos I/O capazes de agregar circuitos externos, um regulador de tensão de 5V entre outras estruturas. Já a programação realizada por meio da IDE - Integrated Development Environment - utilizando uma linguagem de programação baseada em Wiring, sendo essencialmente C/C + + (RUSSEL, 2010).

O Arduino UNO emprega o ATmega328P, “Um microcontrolador CMOS de 8 bits de baixa potência, baseado na versão aprimorada do AVR” (RUSSEL, 2010, p.95). Os microcontroladores são amplamente utilizados em equipamentos eletrônicos devido ao seu reduzido tamanho, fácil programação, baixo consumo energético e um ótimo desempenho que lhe confere um bom custo-benefício (KERSCHBAUMER, 2018). Eles são fabricados no formato de um chip, o qual possui, de forma integrada, uma CPU, memórias de programas e de dados, pinos I/O, circuitos de clock, pinos de 15 alimentação resistores, contadores, temporizadores entre outros componentes que auxiliarão nas execuções de tarefas.

O ATmega328P confere uma rápida velocidade de processamento dada sua microarquitetura baseada no modelo RISC -Reduced Instruction Set Computer-, reduzindo o tempo de resposta devido ao baixo número de instruções (SLUZALA, 2017). A placa microcontrolada Arduino também com um oscilador, pinos I/O capazes de agregar circuitos externos, um regulador de tensão de 5V entre outras estruturas. Já a programação realizada por meio da IDE - Integrated Development Environment - utilizando uma linguagem de programação baseada em Wiring, sendo essencialmente C/C + + (RUSSEL, 2010).

DESENVOLVIMENTO

Tal como os demais satélites e pico satélites o CubeSat apresenta uma carga útil, a qual é composta por todos os equipamentos que configuram o objetivo da sua construção (BARBOSA, 2017). Os sensores empregados neste projeto estabelecem uma função de coleta de dados atmosféricos como umidade, pressão e temperatura, desse modo eles constituem a carga útil do protótipo.

A montagem eletrônica é simulada no software de código aberto Fritzing para posterior implementação física na protoboard de forma a assegurar a conexão elétrica correta. Tal software foi escolhido pelo fácil manuseio e ampla variedade de elementos disponíveis para a simulação de hardware eletrônico.

O BMP180, como mostra a figura 1, é um sensor utilizado para detectar pressão atmosférica e temperatura. Porém uma terceira variável, a altitude (em metros), pode ser obtida a partir da manipulação dessas outras duas, visto que a pressão e temperatura do ar são grandezas físicas inversamente proporcionais à altitude. Esse sensor funciona com uma tensão de alimentação entre 1,8V a 3,6V, mas ele apresenta um regulador de tensão embutido que permite aplicá-lo em um nível de sinal de 5V (Datasheet: BMP180).

Figura 1 - Sensor BMP180.

Fonte: Webtronico (2021).

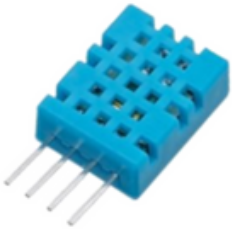
O MPU6050 mostrado na figura 2 possui um acelerômetro e um giroscópio ambos de 3 eixos, além disso ele possui um sensor de temperatura que permite medições entre -40°C e 85°C. Sua tensão de funcionamento varia entre 3V e 5V (Datasheet: MPU6050).

Figura 2 - Sensor MPU6050.

Fonte: Curto Circuito (2021).

Já o DHT11 como mostra a figura 3 é responsável pela captação de dados de temperatura e umidade. É um sensor que apresenta uma tensão de alimentação que varia entre 3V e 5V. Com ele é possível realizar a leitura de temperaturas entre 0°C e 50°C com uma precisão de ± 2°C e umidade de 20% a 90% com precisão de ± 5% (Datasheet: DHT11).

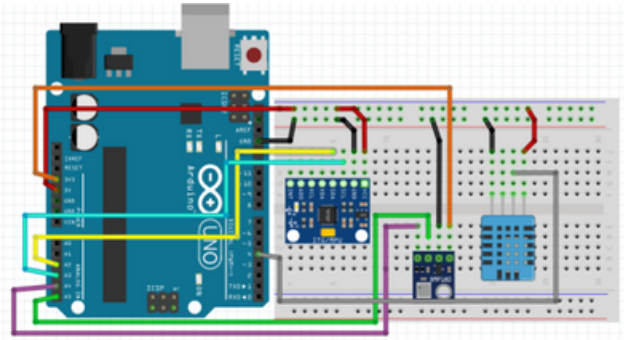
Figura 3 - Sensor DHT11.



Fonte: Baú da Eletrônica (2021).

O esquema mostrado na figura 4 refere-se a montagem dos sensores na protoboard e Arduino utilizando o software de simulação Fritizing e respeitando as especificidades de cada componentes, como a tensão de entrada máxima, pinos GND e analógicos.

Figura 4 - Sensor MPU6050.



Fonte: Autoria própria.

Com todos os sensores devidamente acoplados e a montagem da carga útil finalizada, constata-se que a função conferida ao CubeSat é a de captação de dados meteorológicos mediante o bom funcionamento apresentado pelos sensores. A etapa seguinte é escrever o código de modo a tornar possível a leitura das variáveis atmosféricas pelos sensores empregados.

Para a apuração dos dados climáticos dos sensores é necessário programá-los no IDE do Arduino. Para isso foram usadas bibliotecas que facilitam o desenvolvimento do código como a DHT.h, a Adafruit_BMP085.h e a Wire.h que serve tanto para o BMP180 quanto para o MPU6050, como é ilustrado na figura 5.

Figura 5 - Código para a captação dos dados.

Cubesatt | Arduino 1.8.5

Arquivo Editar Sketch Ferramentas Ajuda

Cubesatt §

```
#include <DHT.h> // biblioteca do DHT11
#include <Wire.h> // biblioteca BMP180 e MPU6050
#include <Adafruit_BMP085.h>
#define DHTPIN 5 // pino do dht conectado ao Arduino
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);
Adafruit_BMP085 bmp;

const int MPU=0x68; // Endereço da interface I2C MPU6050
float umidade = 0;
float temperatura_dht = 0;
float temperatura_bmp = 0;
float altitude = 0;
float pressao = 0;
int Tmp;

void setup() {
  Serial.begin(9600); // Inicia o monitor serial
  //Inicia os sensores
  dht.begin();
  bmp.begin();
  Wire.begin();
  Wire.beginTransmission(MPU);
  Wire.write(0x6B);
  Wire.write(0); // Inicia o MPU6050
  Wire.endTransmission(true);
}
```

Figura 5 - Código para a captação dos dados(continuação).

```
void loop() {

  Wire.beginTransmission(MPU);
  Wire.write(0x3B);
  Wire.endTransmission(false);
  Wire.requestFrom(MPU,14,true);


 temperatura_dht = dht.readTemperature();
 Serial.print("Temperatura DHT11: ");
 Serial.print(temperatura_dht);
 Serial.println("°C");

temperatra_bmp = bmp.readTemperature();
Serial.print("Temperatura BMP180: ");
Serial.print(temperatra_bmp);
Serial.println("°C");
Tmp = Wire.read()<<8|Wire.read();
Serial.print("Temperatura MPU6050: ")
Serial.print(Tmp/340.00+36.53);
Serial.println("°C");
umidade = dht.readHumidity();
Serial.print("Umidade: ");
Serial.print(umidade);
Serial.println("%");
pressao = bmp.readPressure();
Serial.print("Pressão Atmosférica: ");
Serial.print(pressao);
Serial.println("Pa");

  altitude = bmp.readAltitude(101500);
  Serial.print("Altitude: ");
  Serial.print(altitude);
  Serial.println("m");
  delay(8000);
}
```

Fonte: Autoria própria (2021).

Após a realização da programação e verificação do funcionamento dos sensores, é preciso implementar o sistema de suprimento de energia do protótipo, e isso é feito a partir da utilização de uma placa fotovoltaica da marca Canadian Solar, modelo CS6P-265, como ilustra a figura 6.

Figura 6 - Placa Canadian Solar.

Fonte: Autoria própria (2021).

Suas especificações que configuram um ótimo desempenho. Possui dimensões 1638x982x40 mm, tensão máxima de 37.7 V, corrente de 9.23 A e potência de 265W (figura 7).

Figura 7 - Placa Canadian Solar.

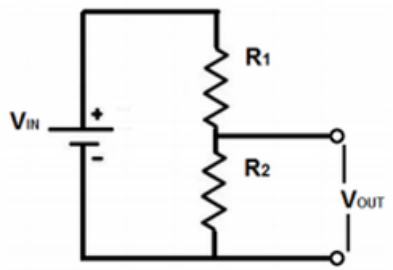
Fonte: Autoria própria (2021).

No entanto, o Arduino Uno possui limitações no que diz respeito à fonte de alimentação externa, visto que a tensão máxima de entrada deve variar entre 6 a 20V. Porém é recomendado utilizar uma fonte que varie entre 7 a 12V para evitar danos de superaquecimento do regulador de tensão interno e garantir o pleno funcionamento da placa de prototipagem (Datasheet: ATmega328P).

Dessa forma, como a tensão máxima fornecida pelo painel fotovoltaico é de 37.7 V, foi necessário o desenvolvimento de um circuito divisor de tensão para reduzir a tensão de saída da placa e possibilitar a alimentação externa do Arduino dentro dos limites permitidos.

Para a regulagem de tensão da placa trabalhou-se com a tensão de funcionamento ideal, a qual tem valor de 30,6V de acordo com os dados especificados pelo fabricante mostrados anteriormente na Figura XX. Dessa forma foi utilizada a plataforma gratuita e online Tinkercad para realizar a simulação do funcionamento do circuito divisor de tensão. Tal circuito foi implementado utilizando dois resistores associados em série para obter a tensão desejada (7V a 20V), visto que esse tipo de associação promove uma queda de tensão nos terminais dos resistores, como é ilustrado na figura 8.

Figura 8 - Associação de resistores em série.



Fonte: Learning about Electronics (2021).

Uma vez que o simulador Tinkercad não conta com o aparato das placas fotovoltaicas entre os componentes disponíveis, foi usada uma fonte de tensão configurada para fornecer 30V com o intuito de simular a tensão fornecida pela placa Canadian Solar além de dois voltímetros para verificar a tensão nos resistores.

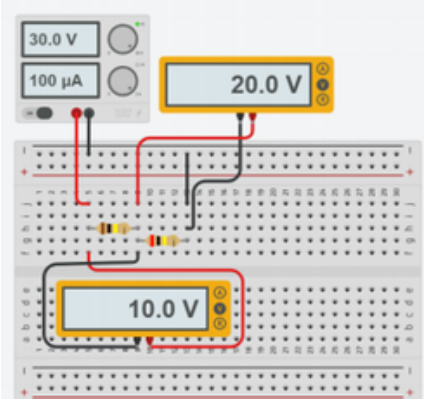
Primeiramente foi estipulado o valor de 100 kΩ para o 30 resistor R1 e com isso foi possível calcular o valor do R2 utilizando a Lei de Ohm para obter uma fórmula direta para o divisor de tensão:

$$U = R^* I \text{ (Lei de Ohm)}$$

$$Vout = \frac{R1 * (R1 * Vin)}{(R1 + R2)}$$

O Vout representa a tensão de saída desejada, a qual será aplicada no Arduino, já o Vin é a tensão fornecida ao circuito, no caso 30V. Para o protótipo foi estipulado 10V no R1 de modo a ficar dentro dos limites permitidos do Arduino. Logo, tem-se que o valor do resistor R2 é de 200 kΩ, o que promove uma tensão de 20V nos seus terminais como é mostrado na figura 9. Tal tensão também pode ser utilizada para alimentar o Arduino, uma vez que ele suporta uma tensão máxima de 20V, mas, como medida de segurança, o valor de escolhido foi de 10V.

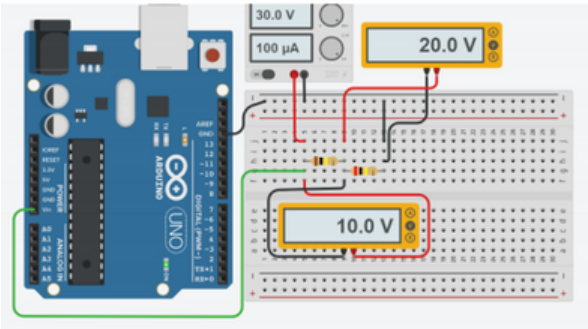
Figura 9 - Simulação do circuito divisor de tensão..



Fonte: Autoria própria (2021).

Com a regulagem da tensão garantida, o passo seguinte é conectar um adaptador na saída da placa Canadian Solar de modo que ele seja facilmente implementado no circuito divisor de tensão no lugar da fonte de tensão da simulação. O circuito, por sua vez, será responsável pela alimentação do Arduino pela porta Vin, como mostrado por meio da simulação no Tinkercad na figura 10.

Figura 10 - Alimentação do Arduino com a tensão da placa Canadian Solar regulada.



Fonte: Autoria própria (2021).

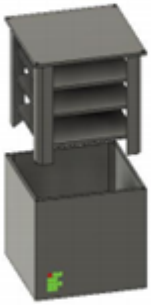
Com o sistema de suprimento de potência funcionando, fornecendo tensão à carga útil que está devidamente operando e fornecendo dados atmosféricos, resta o último passo na construção do protótipo que é o arranjo das partes do pico satélite no interior da estrutura do CubeSat.

A parte estrutural consiste em uma etapa crucial do protótipo, uma vez que ela comporta os subsistemas de carga útil e suprimento de potência, oferecendo certa proteção e uniformidade aos componentes.

O desenvolvimento da estrutura física foi idealizado no software de modelagem 3D Fusion 360 (figura 11) e conta com três prateleiras que compõem uma estrutura removível, promovendo uma melhor disposição espacial e divisão da parte dos sensores, computador de bordo (Arduino) e fonte de energia.

A figura 11 mostra o resultado do software de modelagem 3D Fusion 360.

Figura 11 - Estrutura modelada no Fusion 360.



Fonte: Autoria própria (2021).

A confecção foi realizada por meio de uma impressora 3D utilizando o material polimérico ABS. A estrutura das prateleiras encaixa na parte cúbica de 10 cm de aresta e é selada por quatro parafusos nas extremidades, como ilustrado na figura 12.

Figura 12 - Estrutura modelada no Fusion 360.

Fonte: Autoria própria (2021).

ANALISE DE RESULTADOS

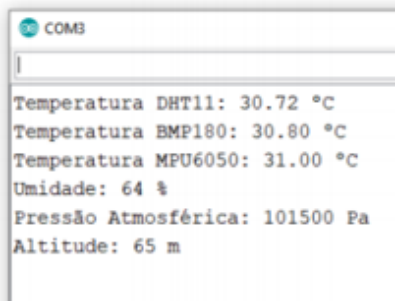
É possível captar a variável temperatura em três sensores distintos, visto que o DHT11, o BMP180 e o MPU6050 permitem a aferição dessa grandeza. Tal situação não confere um erro ou algo desnecessário, mas algo que é crucial no desenvolvimento de satélites.

A redundância, como é chamada essa aplicação, consiste em captar um mesmo tipo de dado por meio de equipamentos diferentes como uma forma de assegurar a coleta do dado analisado. Como satélites reais estão sujeitos a serem expostos aos demais tipos de ambientes e situações durante as missões espaciais, faz-se necessário ter uma garantia no que diz respeito a telemetria, ou seja, a captação de informações que podem ser fornecidas pelo satélite.

Além disso, essa técnica também serve para fins de comparação de modo a verificar se o dado recebido está condizente com a realidade e se algum sensor apresenta defeitos e necessidade de troca ou manutenção de modo a tornar o sistema tolerante a falhas. (PORTO, 2018).

Uma vez compilado e gravado o programa desenvolvido e realizada as devidas conexões entre o Arduino, PCB e computador por meio de um cabo USB, o monitor serial da IDE do Arduino é responsável por exibir os dados captados, como mostra a figura 13.

Figura 13 - Dados meteorológicos captados.



Fonte: Autoria própria (2021).

Com isso, percebe-se uma pequena diferença com relação aos valores de temperatura em cada um dos sensores, fazendo com que essa grandeza se encontre em uma faixa de 30º C a 31º C. Dessa forma, com a carga útil funcionando plenamente, enviando o pacote de dados atmosféricos, pode-se implementar o subsistema de suprimento de potência.

A aplicação de uma placa fotovoltaica no desenvolvimento do protótipo de um CubeSat apresentada neste trabalho foi apenas uma centelha para os propulsores de uma gama de melhorias, modificações e novas implementações em tal campo de pesquisa apesar dos resultados satisfatórios obtidos. Há diversos modelos de painéis solares com dimensões e eficiência variadas que podem ser utilizados em desenvolvimentos futuros de modo a trabalharem em conjunto com um sistema de baterias recarregáveis, se mostrando como uma forma promissora de recarga.

REFERÊNCIAS BIBLIOGRÁFICAS

BARBOSA EITERER, Bruno Vale. **Modelagem, simulação e testes de um sistema de energia aplicado a nanossatélites**. Florianópolis, SC: 2017.

BMP180. Disponível em: https://cdn-shop.adafruit.com/datasheets/BST-BMP180-DS000-09.pdf. Acesso em: 01/02/2021.

DHT11. Disponível em: https://img.filipeflop.com/files/download/Datasheet_DHT11.pdf Acesso em: 01 de fev. 2021.

GOLDEMBERG, José; LUCON, Oswaldo. Energias renováveis: um futuro sustentável. São Paulo, SP. Revista USP, 2006-2007. 6 - 15p.

KERSCHBAUMER, Ricardo. Microcontroladores. Luzerna, SC: 2018.

MESSIER, Douglas. **Tiny 'Cubesats' Gaining Bigger Role in Space**. Disponível em: https://www.space.com/29464-cubesats-space-science-missions.html. Acesso em:05 de fev. de 2021.

PLACA Solar Canadian Solar. Disponível em: https://s3.us-east-2.amazonaws.com/legacy.portalsolar.com.br/Content/EditorImages/files/Placa%20Solar%20Canadian%20Solar%20-%20Modelo%20CS6P-265P%20-%20265Watts.pdf. Acesso em: 08 de fev. de 2021.

PORTO, Roberta de Cássia Ferreira. **Dependabilidade em sistemas espaciais e análise de falhas reais em satélites**. São José dos Campos, SP. 2018.

RUSSELL, David. **Introduction to Embedded Systems: Using ANSI C and the Arduino Development Environment**. Nebraska: University Of Nebraska-lincoln, 2010. 276 p.

SLUZALA, Leonardo Bordignon. Estado da arte da utilização do Arduino em Física no ensino médio. Maringá, PR: 2017.

VILLELA, T.; BRANDÃO, A.; LEONARDI, R. **CubeSats e oportunidades para o setor espacial brasileiro**. Parc. Estrat., v. 21, n. 42, p. 91-114, 2016.

ESTUDO E IMPLEMENTAÇÃO DE UM REAL TIME OPERATING SYSTEMS- RTOS PARA O CUBESAT

Hartur Souza da Silva

ESTUDO E IMPLEMENTAÇÃO DE UM REAL TIME OPERATING SYSTEMS- RTOS PARA O CUBESAT

Hartur Souza da Silva[1]

INTRODUÇÃO

Os Sistemas Operacionais em Tempo Real para microcontroladores de 8 e 16 bits são ferramentas de programação que auxiliam na hora de criar um projeto e organizar o seu software para que funcione em tempo real e com segurança. Eles são voltados para aplicações onde é essencial a confiabilidade e a execução de tarefas em prazos compatíveis com a ocorrência de eventos externos e podem otimizar o tempo de processo dos dados recebidos via hardware e ter uma resposta rápida aos vários elementos que compõem o projeto (FRANK, 2014).

Com isso, a utilização desse sistema em um satélite de pequeno porte como um CubeSat é imprescindível pois procura-se a eficiência de captação dos dados e envio deles. O termo CubeSat refere-se a um pico satélite em forma de um cubo de aresta 10cm que compõe a unidade convencionada de 1U (CAL POLY, 2014).

A palavra tem origem do acrônimo das palavras cube que significa cubo em inglês e da utilização das três primeiras letras da palavra “satélite”. Esse modelo de satélite foi inicialmente proposto por ordi Puig-Suari, da California Polytechnic State University, e Bob Twiggs, da Stanford University, em 1999, para contornar o problema de os satélites terem o custo muito elevado e serem inacessíveis para os estudantes desenvolverem suas próprias experiências espaciais.

A partir disso, as oportunidades de criações de diversos satélites de pequeno porte surgiram para o setor educacional, principalmente para universidades.

[1] *Técnico em Mecatrônica pelo Instituto Federal de Ciência e Tecnologia do Rio Grande do Norte – Campus Parnamirim – hartursouza3@gmail.com*

Atualmente, existem diversos CubeSats já criados e lançados em órbita com diversas aplicações, além de competições que envolvem estudantes do ensino médio e superior. Por ser uma inovação recente da área espacial, os CubeSats hoje estão associados, também, a um modelo de negócios que fez o número de lançamentos de pico satélites como esse crescerem nos últimos.

As portas para uma programação mais eficiente se abrem quando tratamos do microcontrolador ATmega328p que está acoplado em um dos mais conhecidos sistemas eletrônicos embarcados, Arduino. Na programação convencional do Arduino, são utilizados diversos comandos que ocupam numerosas linhas para realizar algo que pode ser feito com apenas uma linha de código do microcontrolador. Isso, por sua vez, ocupa muito espaço na memória e tempo de processamento.

Ao destrinchar as funções do microcontrolador, observamos diversas funções e comandos que otimizam o código. Abaixando-se o nível de programação encontramos funções que manipulam registradores, como, por exemplo, interrupções que podem ser ativadas por meio do hardaware e, até mesmo, se utilizando do clock do sistema para fazer uma contagem com maior precisão, que são as funções “timer”. Esses comandos substituem códigos como a função delay(tempo) do Arduino que paralisa todo o programa a fim de esperar que passe o tempo determinado.

METODOLOGIA

A metodologia adotada para o desenvolvimento do projeto consiste na apresentação do Cubesat, como forma de estímulo ao desenvolvimento espacial de maneira educativa, passando assim, por boas práticas de programação, programa e detalhamento dos sensores, tomando por base a programação inicial que se refere ao timer, com o objetivo da integração de cada parte a fim de concluir a programação e chegar a um projeto final.

Procura-se, principalmente, destacar a parte conceitual necessária para o desenvolvimento do projeto final, dessa forma, será apresentado a programação do microcontrolador utilizada no código, detalhando cada conceito baseado no datasheet do AVR, tratando sobre os comandos e registradores utilizados bem como a utilização dos sensores no código e suas especificações, como dados e programação necessária para seu funcionamento.

FUNDAMENTAÇÃO TEÓRICA

CubeSat

O Cubesat (figura 1), foi desenvolvido com o objetivo de captar dados meteorológicos e transmiti-los a uma estação base. Para isso, foram utilizados 3 sensores: DHT11, BMP180 e MPU6050, que, juntamente, captam os dados de umidade, temperatura e pressão e a partir da manipulação dos dados é possível obter, também, a altitude.

Figura 1 - CubeSat.

Fonte: Autoria própria (2021).

Inicialmente, a programação do Cubesat se baseou na utilização de bibliotecas e na programação do próprio Arduino. Nas primeiras linhas do código, após a inclusão das bibliotecas, é feita a declaração das variáveis dos dados que serão lidos. No “void setup” é iniciado a comunicação serial e alguns sensores para caso haja algum problema com eles, a programação relatar o problema.

No “void loop”, ou seja, o que irá ficar se repetindo, as variáveis recebem os dados que foram lidos pelos sensores e, logo após, é realizado uma contagem de 10 segundos utilizando a função “delay()” e uma variável de contagem para que sempre que se passar 1 segundo, adicionar +1 ao valor da variável e envia-se um pacote de dados. A programação segue na figuras 2.

Figura 2 - Declaração das variáveis.

```
#include <Wire.h>
#include <Adafruit_BMP085.h>
#include <dht.h>
#include <MPU6050.h>
#define dht_pin 5

int temperatura_dht = 0;
int umidade = 0;
int temperatura_bmp;
int temperatura_mpu;
int pressao;
int Altitude;
int count=0;

dht dht_11;
Adafruit_BMP085 bmp;
MPU6050 mpu;

void setup(){
  Serial.begin(9600);
  if (!bmp.begin()) {
    while (1) {} }
  while(!mpu.begin(MPU6050_SCALE_2000DPS, MPU6050_RANGE_2G)) {}
}
void loop() {
   Serial.println("dentro loop");
   dht_11.read11(dht_pin);
   temperatura_dht = dht_11.temperature;
   umidade = dht_11.humidity;
```

Fonte: Autoria própria (2021).

Na Figura 3 é representado o início do código, com a declaração das bibliotecas referentes aos sensores, além de outras mais, das variáveis, métodos, void setup() e o início do void loop().

Figura 3 - Impressão dos dados.

```
    temperatura_bmp = bmp.readTemperature();
    pressao = bmp.readPressure();
    Altitude = bmp.readAltitude(101500);
    temperatura_mpu = mpu.readTemperature();
    if(count>=10){
     Serial.print("temp_DHT:");
     Serial.print(temperatura_dht);
     Serial.println("°C");
     Serial.print("Temp_bmp:");
     Serial.print(temperatura_bmp);
     Serial.println("°C");
     Serial.print("Temp_mpu:");
     Serial.print(temperatura_mpu);
     Serial.println("°C");
     Serial.print("umidade:");
     Serial.println(umidade);
     Serial.print("Pressao:");
     Serial.print(bmp.readPressure());
     Serial.println("Pa");
     Serial.print("Altitude:");
     Serial.print(Altitude);
     Serial.println("m");
     count=0;
     delay(1000);
     }
     Serial.println(count);
     count++;
     delay(1000);
}
```

Fonte: Autoria própria (2021).

PROGRAMAÇÃO

A programação inicial do cubesat, por ser simples e de fácil construção, apresentava dois problemas a serem resolvidos, um deles era a utilização da função “delay()” em que, ao utilizá-la, ela parava todo o processamento para esperar o tempo designado e, também, pelo grande uso de memória pela utilização das bibliotecas, trazendo funções inutilizadas pelo programa, problemas estes, indesejados na construção e desenvolvimento de um pico satélite.

A programação seguinte, segue uma diretriz diferente da inicial. A utilização de bibliotecas será descartada, bem como a função “delay()” do programa. O tempo será contado a partir da utilização do clock do microcontrolador e para a leitura dos dados, será extraído dos sensores apenas os códigos necessários a fim de otimizar tempo e espaço de memória.

BOAS PRÁTICAS DE PROGRAMAÇÃO

Inicialmente, a programação deve seguir alguns princípios que irão propiciar um melhor entendimento do código assim como sua posterior explicação. As boas práticas de programação é uma forma de desenvolver um código mais legível, fácil de compreender e manter. Essa técnica é passada de geração para geração, desta forma, qualquer pessoa saberá como proceder na manutenção do pomar e o resultado deverá ser sempre o mesmo. Podemos estender esse pensamento para o desenvolvimento de software, em que o uso de regras e boas práticas geram melhor legibilidade do código, contribuindo significativamente para que o ciclo de desenvolvimento de sistemas ocorra de maneira mais ágil, prática e de fácil manutenção.

Programadores mais experientes costumam dizer que “a pior coisa que existe é dar continuidade a um software desenvolvido por terceiros”. Provavelmente vocês já devem ter escutado um ditado que diz: “cada cabeça é uma sentença”, o que significa dizer que para resolver um mesmo problema, desenvolvedores podem usar caminhos completamente diferentes.

Apesar disso, se na criação do código o programador seguir regras básicas usando as boas práticas de programação que serão mostradas nesse artigo, essa continuidade e manutenção no desenvolvimento de sistemas será facilitado e menos árduo (NEITZKE, 2011). Algumas das mais importantes boas práticas são:

- Indentação - A indentação de código nada mais é que um espaço no começo de uma linha escrita e é empregada na maioria das linguagens de programação com o objetivo de organizar o algoritmo, facilitando a legibilidade do código e tornar a interpretação do código mais fácil a quem está olhando. Além disso, é usada para definir a hierarquia entre as partes do código sempre colocando as ramificações mais a frente (figura 4).

Figura 4 - Código indentado.

```
void loop(){

  digitalWrite( Led, estadoLed);
  unsigned long tempo_atual = millis();
  int leitura = digitalRead(botao);
  unsigned long diferenca;
  diferenca = (tempo_atual - tempo_anterior);
  if ( diferenca >= 300){
    if( leitura!=estadoAntBotao ){
      estadoBotao = leitura;
      tempo_anterior = tempo_atual;
      if( estadoBotao == LOW){
          estadoLed = !estadoLed;
      }
    }
  }
  estadoAntBotao=leitura;
}
```

Fonte: Autoria própria (2021).

• Comentários – Os comentários servem, principalmente, para ajudar o programador a se lembrar da função de cada parte do código. É recorrente, após algum tempo, o programador esquecer as funções do código, mesmo que tenha sido desenvolvido por ele próprio. Devido a isso, os comentários são uma ótima forma de organizar melhor o algoritmo, colocando cabeçalho (direitos, data e outras informações) e as demais observações. É um exemplo simples de um código com comentários o código Blink disponibilizado na plataforma Arduino (figura 5).

Figura 5 - Código Blink.

```
/*
  Blink

  modified 8 May 2014
  by Scott Fitzgerald
  modified 2 Sep 2016
  by Arturo Guadalupi
  modified 8 Sep 2016
  by Colby Newman

  This example code is in the public domain.

  http://www.arduino.cc/en/Tutorial/Blink
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```

Fonte: Exemplos IDE Arduino.

• Convenções de nomes para classes, métodos, variáveis – Usando as boas práticas já citadas e acrescentando a isso convenções de nomes, a leitura do programa ficará ainda melhor. Nomes adequados e padronizados, passarão informações que ajudarão na compreensão do código, indicando, por exemplo, o que uma variável irá armazenar ou o que um determinado método irá fazer. Agregando uma boa convenção de nomes, irá ajudar no ciclo de desenvolvimento do sistema. (NEITZKE, 2011).

Nas variáveis é importante evitar nomes de apenas um caractere (i, j, x, y), a não ser para aquelas usadas para índices em laços de repetição onde o algoritmo percorre vetores. Em variáveis constantes, isto é, que possuem um valor fixo, todas as letras devem estar em maiúsculo e com as palavras separadas por sublinhado ("_"), por exemplo: VALOR_TEMPERATURA. As classes sempre iniciam seu nome com a primeira letra em maiúsculo, já nos nomes compostos como: MenuPrincipal, as letras que iniciam as palavras começam com o caractere sempre maiúsculo. Nos métodos o que difere a convenção de nomes de classes para nomes de métodos é que para os métodos a primeira letra deve estar em minúsculo. Como os métodos são criados para executar algum procedimento, é preferível usar verbos, como: imprimeValor().

PROGRAMAÇÃO DOS SENSORES

Cada elemento do DHT11 é estritamente calibrado em laboratório, extremamente preciso em calibração de umidade. Os coeficientes de calibração são armazenados como programas na memória OTP, que são usados pelo processo de detecção de sinal interno do 15 sensor.

A interface serial de fio único torna a integração do sistema rápida e fácil, enviando 5 bytes, sendo o primeiro o inteiro da umidade, o segundo sendo o decimal da umidade, o terceiro o inteiro da temperatura, o quarto o decimal da temperatura e o quinto é a soma dos outros 4 bytes, realizando o teste de checksum. Seu tamanho pequeno, baixo consumo de energia e até 20 transmissão de sinal do medidor, tornando-a a melhor escolha para várias aplicações, incluindo aquelas mais exigentes.

O componente é um pacote de pinos de linha única de 4 pinos. É conveniente os pacotes connect e especiais podem ser fornecidos de acordo com a solicitação dos usuários. A programação utilizada no projeto baseia-se na biblioteca do DHT11, todavia, algumas alterações foram feitas para otimizar o código e não fazer a utilização da própria biblioteca, tornando possível o recebimento dos dados apenas utilizando a classe DHT11 e métodos essenciais. Alguns trechos do código demonstram com clareza o seu funcionamento, como na figura 6.

Figura 6 - Método readDht11.

```
int8_t Dht::readDht11(uint8_t pin)
{
    // Lendo os valores
    int8_t result = readSensor(pin, DHT_DHT11_WAKEUP, DHT_DHT11_LEADING_ZEROS);

    // O sensor irá informar apenas o decimal dos valores lidos.
    bits[0] &= 0x7F;
    bits[2] &= 0x7F;

    // Atribuindo o valor lido as variáveis
    umidade     = bits[0];  // bits[1] == 0;
    temperatura = bits[2];  // bits[3] == 0;

    // Teste CHECKSUM
    // bits[1] && bits[3] both 0
    // Se a soma dos bits for diferente do valor de bits[4], houve um erro.
    uint8_t sum = bits[0] + bits[2];
    if (bits[4] != sum)
    {
        return DHT_ERROR_CHECKSUM;
    }
    return result;
}
```

Fonte: Autoria própria.

O método readDht11 (figura 6), é iniciado quando decidimos o pino a ser utilizado para o envio de dados por parte do DHT. Nele está contido a atribuição do bit 0 e do bit 2 a, respectivamente, umidade e temperatura, que são os decimais transmitidos. A fim de não ter erro na leitura, o teste de checksum é realizado em cada método.

O BMP180 é baseado na tecnologia piezo-sensitiva para robustos EMC, alta precisão e linearidade, bem como estabilidade a longo prazo. Possui um conversor de analógico para digital e uma unidade de controle com E^2PROM[1] e interface serial I^2C.

O E^2PROM possui armazenado 176 bit de dados individuais de calibração. Isto é utilizado para compensar offset, dependência da temperatura e outros parâmetros do sensor. Podendo ser utilizado com diversos sistemas microcontrolados. Para tanto, um trecho do programa segue na figura 7.

Figura 7 - Pressão e altitude.

```
int32_t Bmp::readSealevelPressure(float altitude_meters) {
  float pressure = readPressure();
  return (int32_t)(pressure / pow(1.0 - altitude_meters / 44330, 5.255));
}

float Bmp::readAltitude(float sealevelPressure) {
  float altitude;
  float pressure = readPressure();
  altitude = 44330 * (1.0 - pow(pressure / sealevelPressure, 0.1903));
  return altitude;
}
```

Fonte: Autoria própria.

É interessante informar que na leitura do dado de altitude, faz-se o uso do dado de pressão, pois leva-se em conta a pressão estabelecida para fazer o cálculo aproximado da altitude, testes feitos em câmaras a vácuo foram feitos com esse sensor e é possível observar que conforme a pressão diminui o valor de altitude aumenta.

O MPU6050, diferente dos demais, reúne duas funções em um único módulo, combinando acelerômetro e giroscópio, tem embutido um recurso chamado DMP (Digital Motion Processor), responsável por fazer cálculos complexos com os sensores e cujos dados podem ser usados para sistemas como o de navegação (GPS). Além disso, tem embutido um sensor de temperatura que permite leituras de -40 a 85°C.

[1] É um tipo de memória não-volátil usada em computadores e outros dispositivos eletrônicos para armazenar pequenas quantidades de dados que precisam ser salvos quando a energia é removida.

Como o projeto inicial não visa a utilização do acelerômetro e giroscópio presente neste sensor, a programação é baseada apenas na coleta do dado de temperatura, conforme o método do programa (figura 8).

Figura 8 - readTemperature.

```
float MPU6050::readTemperature(void)
{
    int16_t T;
    T = readRegister16(MPU6050_REG_TEMP_OUT_H);
    return (float)T/340 + 36.53;
}
```

Fonte: Autoria própria.

PROGRAMAÇÃO DO TIMER

O timer, por sua parte, é uma interrupção gerada interna ou externamente no microcontrolador. As interrupções são muito importantes no trabalho com microcontroladores porque, assim, permitem simplificar o código em determinadas situações, tornando mais eficiente o desempenho do processamento.

Todas as interrupções no AVR também são “mascaráveis”, ou seja, elas podem ser individualmente habilitadas ou desabilitadas agindo-se sobre um bit específico. O sistema de interrupção do AVR possui, ainda, um bit que pode desabilitar ou habilitar todas as interrupções de uma só vez, o bit I do registrador de status (SREG).

Na Tabela 1 apresenta-se cada interrupção do ATmega328, existindo um total de 26 diferentes tipos de interrupções, sendo o timer, algumas delas (LIMA, 2012).

Figura 7 - Pressão e altitude.

Vetor	End.	Fonte	Definição da Interrupção	Prioridade
1	0x00	RESET	Pino externo, Power-on Reset, Brown-out Reset e Watchdog Reset	↑
2	0x01	INT0	interrupção externa 0	
3	0x02	INT1	interrupção externa 1	
4	0x03	PCINT0	interrupção 0 por mudança de pino	
5	0x04	PCINT1	interrupção 1 por mudança de pino	
6	0x05	PCINT2	interrupção 2 por mudança de pino	
7	0x06	WDT	estouro do temporizador *Watchdog*	
8	0x07	TIMER2 COMPA	igualdade de comparação A do TC2	
9	0x08	TIMER2 COMPB	igualdade de comparação B do TC2	
10	0x09	TIMER2 OVF	estouro do TC2	
11	0x0A	TIMER1 CAPT	evento de captura do TC1	
12	0x0B	TIMER1 COMPA	igualdade de comparação A do TC1	
13	0x0C	TIMER1 COMPB	igualdade de comparação B do TC1	
14	0x0D	TIMER1 OVF	estouro do TC1	
15	0x0E	TIMER0 COMPA	igualdade de comparação A do TC0	
16	0x0F	TIMER0 COMPB	igualdade de comparação B do TC0	
17	0x10	TIMER0 OVF	estouro do TC0	
18	0x11	SPI, STC	transferência serial completa - SPI	
19	0x12	USART, RX	USART, recepção completa	
20	0x13	USART, UDRE	USART, limpeza do registrador de dados	
21	0x14	USART, TX	USART, transmissão completa	
22	0x15	ADC	conversão do ADC completa	
23	0x16	EE_RDY	EEPROM pronta	
24	0x17	ANA_COMP	comparador analógico	
25	0x18	TWI	interface serial TWI – I2C	
26	0x19	SPM_RDY	armazenagem na memória de programa pronta	

Fonte: AVR e Arduino Técnicas de Projeto.

Na programação do timer para este projeto, conforme está ilustrado na figura 9, a interrupção escolhida foi o TC1 que inclui consigo o TIMER1 COMPA, pois suas especificações compreendiam melhor sua aplicação.

No ATmega328, existem dois temporizadores: os de 8 bits (TC0 e TC2) e o de 16 bits (TC1). Os temporizadores de 8 bits permitem contagens de 0 a 255, já o temporizador de 16 bits permite uma contagem de 0 até 65535. Com isso, o tempo para estourar o clock do microcontrolador é dado pela fórmula:

$$t_{estouro} = \frac{(TOP+1) \times prescaler}{f_{osc}}$$

onde TOP é o valor máximo de contagem, fosc é a frequência do clock empregado (oscilador) e o prescaler é o divisor dessa frequência (LIMA, 2012).

Figura 9 - Código Timer.

```
const uint16_t T1_init = 0;
const uint16_t T1_comp = 31250; //500ms

ISR(TIMER1_COMPA_vect)
{
 //Aqui fica o código que irá ser realizado após ter gerado a interrupção
}

void setup()
{
   //Modo de Comparação
   TCCR1A = 0;

   //Prescaler 1:1024
   TCCR1B |=  (1 << CS12);
   TCCR1B &= ~(1 << CS11);
   TCCR1B &= ~(1 << CS10);

   //Inicializa Registradores
   TCNT1 = T1_init;
   OCR1A = T1_comp;

   //Habilita Interrupção do Timer1
   TIMSK1 = (1 << OCIE1A);
}

void loop()
{
}
```

Fonte: Autoria própria.

DESENVOLVIMENTO

O uso do RTOS no desenvolvimento de um satélite de pequeno porte se aplica de diversas formas. Como já descrito anteriormente, a fluidez do trabalho com a aplicação do código se torna indispensável se comparado ao antigo, usado anteriormente com as bibliotecas, pois a melhor gestão do algoritmo permite ajustes que podem se adequar ao que o desenvolvedor de um projeto como este possa necessitar.

A melhor manipulação e controle dos dados que o RTOS oferece, ajuda na compreensão e na introdução a uma programação mais avançada, entrando na esfera aeroespacial, em que a administração de projetos como este é muito intensa. Visto isso, o conjunto de serviços oferecidos pelo RTOS permite que um aplicativo possa ser implementado, testado e mantido mais facilmente através da divisão do problema a ser resolvido em tarefas menores e de compreensão mais fácil (LIMA, 2012).

CONSIDERAÇÕES FINAIS

Com o desenvolvimento do projeto foi possível adquirir novos conhecimentos e correlacionar com outros vistos no decorrer dos quatro anos de curso de mecatrônica no IFRN campus Parnamirim.

Diante do que foi apresentado, a programação revela diversas áreas de conhecimento, relacionando-se com a eletrônica, propiciou uma aplicação para a esfera acadêmica, mas também, para o âmbito espacial, uma vez que foi aplicado a utilização de sensores com o objetivo de captar dados meteorológicos e a utilização de um microcontrolador de baixo custo e de fácil manuseio, visto que todo o material necessário para seu desenvolvimento é de fácil acesso a todos.

Contudo, dificuldades são habituais em projetos como este. Vários testes foram realizados na programação até chegar ao que era desejado por parte dos 3 sensores, a extensão dos códigos e suas complexidades dificultaram a leitura e a interpretação, ocasionando, portanto, uma menor abstração de conhecimento.

Todavia, a insistência no projeto se deu por diversas motivações, dentre elas, as principais seriam concluir o trabalho e poder abstrair o melhor dele. Em suma, o projeto permite uma boa possibilidade de exploração em diversas áreas e seu desenvolvimento acarreta a interdisciplinaridade de áreas como Astronomia, Astronáutica, Robótica e Programação. Essa plataforma científica poderá ser utilizada por alunos posteriores na realização de atividades e estudos com ênfase na área aeroespacial.

REFERÊNCIAS BIBLIOGRÁFICAS

CALIFORNIA POLYTECHNIC STATE UNIVERSITY - Cal Poly. **CubeSat Design Specification** (CDS Rev 13). Disponível em: httDocument Classification (squarespace.com). Acesso em: 07 de maio de 2021.

PUHLMANN, Henrique Frank Werner. **Sistemas operacionais de tempo real – introdução**. Disponível em: https://www.embarcados.com.br/sistemas-operacionais-de-tempo-real-rtos/. Acesso em: 07 de jul. de 2020.

LIMA, Charles Borges de. **AVR e Arduino: técnicas de projeto** / Charles Borges de Lima, Marco Valério Miorim Villaça. 2. ed. – Florianópolis: Ed. dos autores, 2012. 632 p.: il; 21,0 cm.

Neitzke, Neri Aldoir. **Boas Práticas de Programação**, 2011. Disponível em: https://www.devmedia.com.br/boas-praticas-de-programacao/21137. Acesso em: 22 fev. 2021.

METODOLOGIA DE ENSINO APLICADA AO KIT EDUCACIONAL DE CANSAT

Michelly Dantas Guedes

METODOLOGIA DE ENSINO APLICADA AO KIT EDUCACIONAL DE CANSAT

Michelly Dantas Guedes[1]

INTRODUÇÃO

A corrida espacial situada no período da guerra fria (1945-1991) foi um episódio de grandes avanços tecnológicos impulsionados pelos avanços na área aeroespacial. Nessa disputa pelo monopólio mundial- entre Estados Unidos da América (EUA) e União das Repúblicas Socialistas Soviéticas (URSS) - havia grande investimento no setor espacial, tanto no que diz respeito ao desenvolvimento físico das espaçonaves quanto nas áreas de conhecimento relacionadas à atividade (física engenharia, programação e matemática). Em concatenação à tais investimentos havia a grande empolgação da população pela exploração do espaço e, por consequência, a introdução destas tecnologias no ensino escolar.

Segundo Azevedo (1969): “Da caneta esferográfica aos transplantes de órgãos humanos passando pelos computadores, técnicas modernas de preservação e acondicionamento de alimentos, tudo hoje está em estreita relação com as pesquisas espaciais”. Uma vez notado o impacto do desenvolvimento dessa área no avanço de outras áreas, iniciou-se a inserção de conhecimentos da temática aeroespacial no ensino escolar. Nessa perspectiva em 1998 em um congresso sobre o tema, realizado no Havaí, o cientista e professor Bob Twiggs apresentou a ideia do que hoje se entende como nano satélites e, que culminou na ferramenta de ensino amplamente utilizada na atualidade: o cansat.

A palavra cansat vem do inglês e, é o acrônimo das palavras can e sattelite (HONDA, 2016). Sendo o equivalente no português a um satélite em forma de lata.

[1] *Técnica em Mecatrônica pelo Instituto Federal de Ciência e Tecnologia do Rio Grande do Norte – Campus Parnamirim – michelly.guedes.131@ufrn.edu.br*

METODOLOGIA

A metodologia utilizada neste trabalho consiste na experimentação da capacidade didática do kit educacional de cansat no ensino de tecnologia espacial, especificamente tratando-se dos subsistemas que compõem um satélite. Nesse viés trabalhou-se a capacidade de transmissão de conhecimento que é propiciado pelo referido kit educacional utilizando-se como método de ensino a metodologia maker, por meio da qual os estudantes aprendem sobre os subsistemas tendo um contato direto com estes e atuando de modo ativo na montagem do kit e nos testes posteriores a montagem. Imergindo-os em uma experiência cientifica de construção e teste de um satélite.

Com a aplicação prática do kit educacional de ensino à estudantes do 8° e 9° ano do ensino fundamental, busca-se atestar por meio da experimentação que a utilização do kit torna o processo de ensino-aprendizagem mais eficaz e que a modalidade prática embutida nele produz um maior engajamento nos estudantes e os tornam mais interessados na aquisição do conhecimento ofertado.

FUNDAMENTAÇÃO TEÓRICA

Para a realização desse trabalho fez-se necessária a consulta em diversos matérias de modo a embasar teoricamente os procedimentos e metodologias aplicados neste. Nessa perspectiva buscou-se descrever nesta seção estes materiais, apresentado a teoria que justifica a escolha da metodologia de ensino, os métodos adotados durante o desenvolvimento do projeto e os princípios educacionais que viabilizam a realização da proposta de ensino levantada por este trabalho.

Inicialmente conforme afirma Coutinho (2011) afirma: “a escola e os seus agentes têm de mudar os métodos e técnicas de ensino e pensar em formas eficientes e eficazes para preparar os estudantes para a sociedade do conhecimento”.

Nesse sentido a metodologia maker apresenta-se como uma nova prática pedagógica, por propor uma maior participação do estudante no processo de ensino aprendizagem, o que exerce grande impacto na absorção dos conteúdos ministrados. Acerca disto Santos (2002) afirma que, no que se refere ao desenvolvimento intelectual e à relação do sujeito com o saber, a participação exerce um papel fundamental e insubstituível.

A metodologia maker além de sua abordagem prática, também se debruça sobre o conceito de STEM (Science, Technology, Engineering and Mathematics) que trabalha dentro das práticas os conceitos das áreas e/ou disciplinas mencionadas acima, de tal modo a fazer uma relação entre o que se estuda teoricamente e a experiência pratica que se está executando, portanto, apresentando a aplicabilidade daquilo que é abordado em sala de aula. A essa relação entre teoria e pratica atribuise o título de tangibilidade, que nesse contexto educacional possui grande relevância, já que, desperta no estudante o interesse pelo estudo tão logo conhece sua aplicação.

Metodologia de ensino

A metodologia de ensino na qual se embasa esse trabalho é a maker, e é parte do objetivo deste verificar a eficácia dessa metodologia aplicada ao ensino da robótica, especificamente ao kit educacional de cansat sob o qual esse trabalho se deleita.

Segundo Martins (2007), as salas de aula, em vez de serem locais de rotinas enfadonhas pela eterna repetição do tradicional processo de ensino aprendizagem – ouvir, ler, escrever-, deviam ser verdadeiros laboratórios e oficinas vivas do “aprender fazendo”.

Desse modo, a metodologia maker apresenta-se como uma estratégia para criação de ambientes práticos e dinâmicos intitulados pelo autor de oficinas vivas. O termo maker pertence a língua inglesa e, se traduzido ao português significa criador, ou seja, o aluno submetido à essa metodologia de ensino assume papel de criador dentro do processo educacional, construindo e desenvolvendo ferramentas relacionadas ao tema que está estudando. O termo foi utilizado pela primeira vez, com o sentido que é empregado atualmente, pelo fundador da Maker Faire e da Make Magazine, o empresário Dale Dougherty.

A metodologia maker traz como slogan Do It Yourself (DIY), o que em tradução livre para o português significa Faça Você Mesmo (FVM), referindo-se ao papel do aluno dentro desse método, ressaltando a característica de criação e de experimentação, aonde aquele pode construir e/ou montar objetos/experimentos nos quais a pratica educativa encontra-se no próprio processo construtivo e não nos resultados obtidos, como observa-se nas práticas de ensino padrão.

Nessa perspectiva, observa-se a presença constante da robótica dentro dessas práticas, pois sendo o objetivo principal despertar o interesse do aluno pelo que está desenvolvendo, é conveniente que se aborde aspectos que já são do interesse em outros âmbitos, o que ocorre com a robótica e com o espaço, por exemplo.

Segundo Silva e Gil (2019) o maker vai à busca do seu próprio conhecimento, construindo, modificando, fabricando, prototipando e projetando. Nesse caso o termo maker refere-se ao aluno, protagonista nesse processo, tanto que em tradução livre ao português significa fazedor, o que ressalta novamente o protagonismo do aluno dentro dessa metodologia. Durante a experiência de maker os estudantes trabalham diferentes aspectos individuais e coletivos conforme visualiza-se no relatório Education for Life and Work: Developing Transferable Knowledge and Skills in the 21st Century, elaborado pela academia nacional de ciências (National Academy of Sciences) em 2018.

Nesse relatório são destacadas três habilidades necessárias à educação no século XXI, são elas: Domínio cognitivo, que se refere aos processos cognitivos e estratégias, conhecimento e criatividade, o domínio intrapessoal que se refere à abertura intelectual, trabalho ético e consciente e auto avaliação positiva, e o domínio interpessoal que se refere ao trabalho em equipe, colaboração e liderança.

Essas habilidades não são o foco principal do ensino padrão aplicado nas escolas, sendo este conteudista e pouco flexível, porém são trabalhadas constantemente dentro da cultura maker, conforme afirma Silva e Gil (2019, p. 134) os makers chamam atenção da sociedade pela forma como trazem o engajamento, aprendizagem e o protagonismo para o sujeito, e favorecem estes três grupos de domínio.

Resnick e Rosenbaum (2017) afirmam que só fazer coisas não é suficiente. Existem diferentes abordagens para se fazer coisas, e algumas proporcionam experiências de aprendizagem mais ricas que outras. Deste modo, pode-se observar a importância de desenvolver experiências maker que proporcionem aquisição de conhecimento, a essas experiências os autores intitulam de tinkering (em tradução livre para o português, pensante ou pensativo), nessa abordagem ressalta-se a importância de diversão, exploração e interatividade como estilo de engajamento dos alunos, no entanto atentando-se para a efetividade de transmissão de conhecimento por meio da experiência maker.

Quanto ao emprego dessa metodologia na educação, Brockveld (2017) afirma que:

> Neste sentido, a educação associada ao movimento maker é diferenciada em relação às aulas tradicionais porque o aluno adquire ferramentas para compreender e aprimorar os conhecimentos recebidos nas aulas expositivas, ou seja, o estudante aprende a aprender. (BROCKVELD.2017. p.6).

Logo, adotar a metodologia maker no ensino mostra-se efetivo para maior aquisição de conhecimento e também para o desenvolvimento de habilidades necessárias à sociedade do conhecimento. Desse modo, utilizar essa metodologia para o ensino das tecnologias espaciais mostra-se eficaz, pois soma-se às informações supracitadas o fato das tecnologias espaciais estarem em constante avanço e serem uma temática que desperta curiosidade nos estudantes pela sua estreita relação com meios consumidos por esse público, como por exemplo ficção cientifica.

Metodologia maker no ensino de tecnologia espacial

O cientista e filosofo alemão Galileu Galilei (1564-1642) afirmava já no século XVI que “Ciência sem experimentação não é ciência”. Atualmente, é igualmente importante o papel da experimentação na produção da ciência e do saber, de tal forma que iniciar essas práticas na escola apresenta-se como aspecto acelerador da produção e difusão do saber.

Nesse sentido a metodologia maker apresenta-se como a experimentação indispensável, ressaltada por Galileu, e propõem-se a levar a prática do experimento através de ambientes de criação, onde o estudante tem a oportunidade de imaginar, criar, testar e refazer, em um ciclo continuo e sem a presença dos métodos de avaliação normalmente aplicado dentro da sala de aula, tais como avaliações escritas.

Quanto a relação entre a experimentação e a metodologia maker Brockveld (2017) afirma que a base do movimento maker, então, encontra-se na experimentação. O objetivo nessa metodologia não é avaliar a execução do experimento, mas sim estimular o interesse do aluno na realização deste, sendo menos avaliativo e mais participativo, esse método tem conquistado adeptos em todo o mundo e tem sua efetividade atestada por relatórios educacionais, como o relatório da academia nacional de ciência “Education for Life and Work: Developing Transferable Knowledge and Skills in the 21st Century” divulgado em 2018.

Dessa maneira a aplicação da metodologia maker no ensino das tecnologias espaciais se mostra efetiva, já que o estudante terá a oportunidade de aprender acerca das ciências relacionadas ao espaço, testando e construindo experimentos. Assim sendo, a ferramenta apresentada nesse trabalho, o cansat, atende a esses aspectos, pois trabalha os conceitos de STEM (Science, Technology, Engineering and Mathematics) e seu processo construtivo embasa-se no protagonismo do aluno, tanto na montagem quanto na programação e testes. A área espacial por ser bastante retratada no cinema, e ser compreendida pelo senso comum como uma área enigmática, desperta naturalmente alguma curiosidade e interesse do estudante em estudá-la.

No entanto é preciso atribuir significado a pratica construtiva a ser realizada de modo a relacionar o que está sendo feito e o que se estuda, nesse aspecto o professor atua como um orientador, diferentemente dos métodos tradicionais aonde este é responsável por todo o processo de ensino aprendizagem. Quanto a isso Moran (2013) afirma: A aprendizagem se constrói num processo equilibrado entre três movimentos principais: a construção individual – em que cada aluno percorre seu caminho -; a grupal – em que aprendemos com os semelhantes, os pares, e a orientada, em que aprendemos com alguém mais experiente, com um especialista, um professor. Moran (2013,p.3).

O ambiente de experimentação e construção proporcionado pelo emprego da metodologia maker modifica a relação com o conhecimento, aproximando o aluno do seu objeto de estudo e, no que dista ao estudo das tecnologias espaciais tal aproximação é de grande relevância, pois torna palpável ao estudante o que autora imaginava ser inalcançável, considerando que para o senso comum tecnologias referentes ao espaço são de difícil acesso e alta complexidade.

O Kit Educacional de Cansat materializa essa experiência maker no ensino das tecnologias espaciais, pois seu processo construtivo aborda um importante elemento das dessas tecnologias, o satélite, e sua aplicação na educação propõe uma experiência construtiva na qual o aluno é protagonista. O processo de montagem do cansat é essencialmente uma experiência maker, na qual a integração dos subsistemas do satélite bem como seus testes, promovem a participação ativa do aluno, tendo contato com conceitos físicos quando analisados os dados dos sensores atmosféricos (pressão, temperatura, humidade e altitude) e matemáticos para a construção da lógica de programação e manipulação dos dados dos sensores (conversão de unidades). Além do contato com o trabalho individual e coletivo simultaneamente, aonde o aluno, aprende, compartilha, melhora sua solução e compartilha novamente, em um ciclo pedagógica e socialmente benéfico.

O manual de montagem que acompanha o Kit Educacional de Cansat apresenta-se como um intensificador dessa experiência construtiva autônoma, pois apresenta uma sequência básica de montagem que norteia o processo construtivo, mas que ao mesmo tempo permite, e solicita, a criatividade do aluno para que a montagem possa de fato se materializar. Dessa forma corroborando com o pensamento do pedagogo Paulo Freire que afirma ser a educação um processo no qual é necessário respeitar a autonomia do aluno, de modo que o professor que desrespeita essa autonomia transgrede os fundamentos éticos da própria existência, afogando assim a curiosidade do educando. (FREIRE, 1996. P.35).

DESENVOLVIMENTO

O kit educacional de cansat como ferramenta de ensino, é constituído por um circuito eletrônico materializados em placas de circuito impresso particionadas em três módulos acopláveis, estrutura mecânica desenhada em software de modelagem e desenvolvida em impressora 3D, programação do kit educacional executada no ambiente de desenvolvimento integrado da plataforma Arduino e material de suporte didático para auxiliar a execução das práticas.

O referido kit busca facilitar o processo de ensino das tecnologias, especificamente a subárea de satélite, utilizando como recurso didático a cultura maker almejando o maior envolvimento do aluno no processo de ensino, desta forma caracteriza-se como uma ferramenta colaborativa para a mudança nas estratégias de ensino que estão ocorrendo no mundo, as quais o Brasil também está engajado.

Tecnologia Espacial na educação

O kit educacional de cansat se materializa como uma ferramenta didática, com objetivo de ampliar os conhecimentos acerca da área espacial, de tal forma que o estudante possa vivenciar a experiência cientifica de desenvolvimento de um satélite e com isso possa despertar interesse pelo estudo nessa área do conhecimento.

O kit apresenta uma proposta de abordagem embasada na prática, e que além da área espacial trabalhará conteúdos correlatos a esta, como matemática, programação e a metodologia cientifica, trazendo-os de modo compatível ao nível de conhecimento da faixa com a qual se está trabalhando.

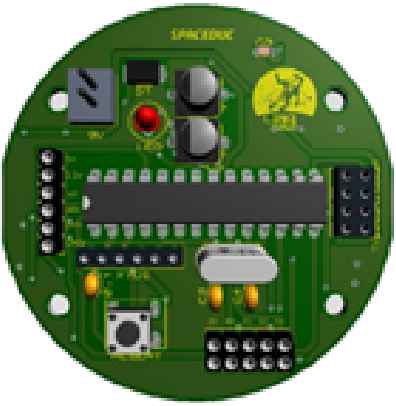
Visando a realização de uma experiência completa, dispõe-se das seguintes partes no kit didático: circuito eletrônicos dos principais sistemas de um satélite, estrutura física, manual de montagem e utilização e interface de programação e comunicação do kit.

1. Circuitos eletrônicos

A eletrônica do kit educacional organiza-se de maneira didática visando o melhor repasse do conhecimento, neste sentido dividiu-se o sistema em três níveis de modo que cada um apresente um subsistema que será analisado individualmente e em seguida em conjunto nas etapas de integração do kit. As placas de circuito impresso foram desenvolvidas no software Proteus®, ambiente que possibilita a construção do esquema elétrico, a simulação do funcionamento, o desenho das placas de circuito impresso e a simulação 3D dos sistemas desenvolvidos. Após essas etapas de desenvolvimento em software as placas foram enviadas para confecção objetivando-se a melhor apresentação dos sistemas, considerando o impacto visual como fator relevante no processo de ensino aprendizagem.

Os circuitos eletrônicos se dividem em três estágios, cada qual representando um subsistema. O primeiro estágio é responsável pelo controle do satélite, e comporta o micro controlador, um transmissor de rádio frequência, bem como um dispositivo luminoso que indica o funcionamento do kit. A organização dos componentes na placa de circuito impresso é como na figura 1.

Figura 1 - placa de controle.



Fonte: Autoria própria.

Neste subsistema são processadas todas as informações do satélite, através do micro controlador, o atmega328p, e transmitidas pelo transceiver, do modelo nrf24l01, cuja transmissão se dá na frequência de rádio especificamente na faixa de 2,4 GHz, estando configurado como transmissor, de modo que os dados enviados por ele são recebidos por um outro módulo, de mesmo modelo, instalado no computador.

Essa placa apresenta todos os componentes que compõem a plataforma arduino de tal modo que a linguagem e o ambiente de desenvolvimento são os mesmos utilizados pela plataforma. A segunda placa é a dos sensores, responsável pela coleta dos dados atmosféricos (Temperatura, pressão, umidade e altitude) cuja apresentação se dá como na figura 2.

Figura 2 - placa dos sensores.

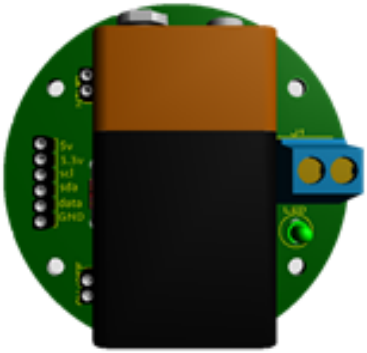
Fonte: Autoria própria.

Esta placa é o equivalente a carga útil do satélite, pois estão contidos os sensores responsáveis pela coleta de dado, de tal forma que as informações adquiridas por esse sistema possam ser processadas e enviadas através da placa de controle. Além dos sensores, esta placa contém uma matriz de contato (protoboard) que possibilita a inserção de novos componentes ao sistema, ampliando ou modificando a função inicial do satélite.

A terceira placa é responsável pelo suprimento de energia das demais placas, o que é feito através de uma bateria de lítio de 9V, de modo que represente o subsistema de suprimento de potência de um satélite.

Apresenta-se conforme ilustrado na figura abaixo.

Figura 3 - placa da bateria.



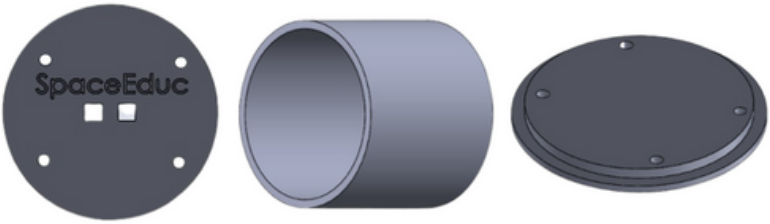
Fonte: Autoria própria.

O circuito eletrônico é a parte do kit didático que trabalha a interdisciplinaridade, pois são abordados conceitos de eletrônica durante a explanação e análise dos sistemas, conceitos matemáticos embutidos na lógica de programação.

2.Estrutura física

A estrutura mecânica é responsável por atribuir ao satélite o formato e volume de uma lata de refrigerante, sendo assim denominado de cansat. Esta estrutura foi desenvolvida utilizando o software de modelagem SolidWorks® de tal modo que a visualização da simulação da estrutura é como apresentada abaixo.

Figura 4 - involucro do cansat.



Fonte: Autoria própria.

3.Manual de montagem e utilização

A cultura maker propõe maior autonomia do aluno no processo construtivo, de modo que o manual de montagem se apresenta como um suporte nesse processo. Em um formato didático, descreve os principais passos da montagem e integração dos sistemas, apresentando-os com auxílio de imagens e símbolos que visam associar o processo descrito no manual com a prática maker executada pelos alunos.

O layout do manual também visa envolver o aluno na referida experiência cientifica, de modo a utilizar-se de recursos visuais para prender a atenção e despertar o interesse do aluno pela utilização do manual, desassociando a ideia de texto metódico e enfadonho. A capa do manual apresenta-se como na figura 5.

Figura 5 - Capa do manual de montagem.

Fonte: Autoria própria.

4. Interface de programação e de comunicação

A programação do kit educacional é desenvolvida no ambiente de desenvolvimento integrado da plataforma Arduino (IDE), dessa forma a linguagem de programação utilizada é a mesma utilizada pela plataforma, sendo baseada na linguagem C.

O código responsável pelo controle do cansat foi desenvolvido de maneira didática utilizando-se de recursos disponíveis à plataforma, tais como bibliotecas, visando a melhor compreensão do aluno que tenha contato com o kit educacional. A estrutura de programação pode ser particionada em duas etapas: a coleta dos dados atmosféricos (pressão, temperatura, umidade e pressão) e a transmissão destes, a primeira utiliza-se das bibliotecas dos sensores envolvidos, executando apenas a solicitação dos dados coletados e armazenando-os em variáveis. A segunda parte consiste na comunicação entre dois módulos transceiver, que configurados como transmissor e receptor são responsáveis pela transferência dos dados do cansat (coletados pelos sensores contidos nele) à estação base, o computador, que responsável por exibir os dados ao aluno.

A apresentação dos dados é feita utilizando uma das ferramentas da IDE do Arduino, o monitor serial. No entanto o desenvolvimento do projeto pretende desenvolver uma ferramenta mais didática e atrativa visualmente para a impressão dessas informações, de modo a corroborar com o propósito do kit educacional como um todo.

CONSIDERAÇÕES FINAIS

Com base nas informações apresentadas pelo presente trabalho pode-se concluir que a utilização do Kit Educacional de Cansat para o ensino das tecnologias espaciais e os conhecimentos correlatos a esta área, apresentam resultados positivos no que se refere à aquisição de conhecimento e ao melhoramento de habilidades essenciais à sociedade do conhecimento – Domínio intrapessoal, interpessoal e capacidade criativa.

Tais resultados são almejados por meio da aplicação da metodologia maker, sendo uma combinação que simultaneamente promove o ensino das tecnologias espaciais, a experimentação e incentivo da criatividade e, principalmente o protagonismo do aluno no processo de ensino aprendizagem.

Desse modo, pode-se considerar que a ferramenta e metodologia alvos desse trabalho corroboram para o aprimoramento da pratica educativa, sugerindo também uma melhoria na preparação do aluno para a sociedade e, por consequência um avanço para a sociedade como um todo, reforçando o papel transformador da educação, conforme afirmou o pedagogo Paulo Freire.

> "Se a educação sozinha não transforma a sociedade, sem ela tão pouco a sociedade muda".
> Paulo Freire.

REFERÊNCIAS BIBLIOGRÁFICAS

AZEVEDO, Rubens de. **Na era da Astronáutica**. São Paulo: Editora do Brasil, 1969.

BROCKVELD, Marcos Vinícius Vanderlinde et al. A cultura maker em prol da inovação:boas práticas voltadas a sistemas educacionais. P.6.2017.

COUTINHO, clara; LISBOA, Eliana. **Sociedade da informação, do conhecimento e da Aprendizagem: desafios para educação no século XXI**. Revista de Educação, Vol. XVIII, n.1, p 5-22, 2011

FREIRE, Paulo. **Pedagogia da autonomia: saberes necessários à prática educativa**. São Paulo: Paz e Terra, 1996

MARTINS, Jorge Santos. **PROJETOS DE PESQUISA: estratégias de ensino e aprendizagem em sala de aula**. 2. ed. Campinas: Autores Associados Ltda, 2007. 184 p.

MORAN, José. **Metodologias ativas para uma aprendizagem mais profunda**. p.15. 2013.

RESNICK, M., & ROSENBAUM, E. (2013). **Designing for Tinkerability**. In Honey, M., & Kanter, D. (eds.), Design, Make, Play: Growing the Next Generation of STEM Innovators, pp. 163-181. Routledge.

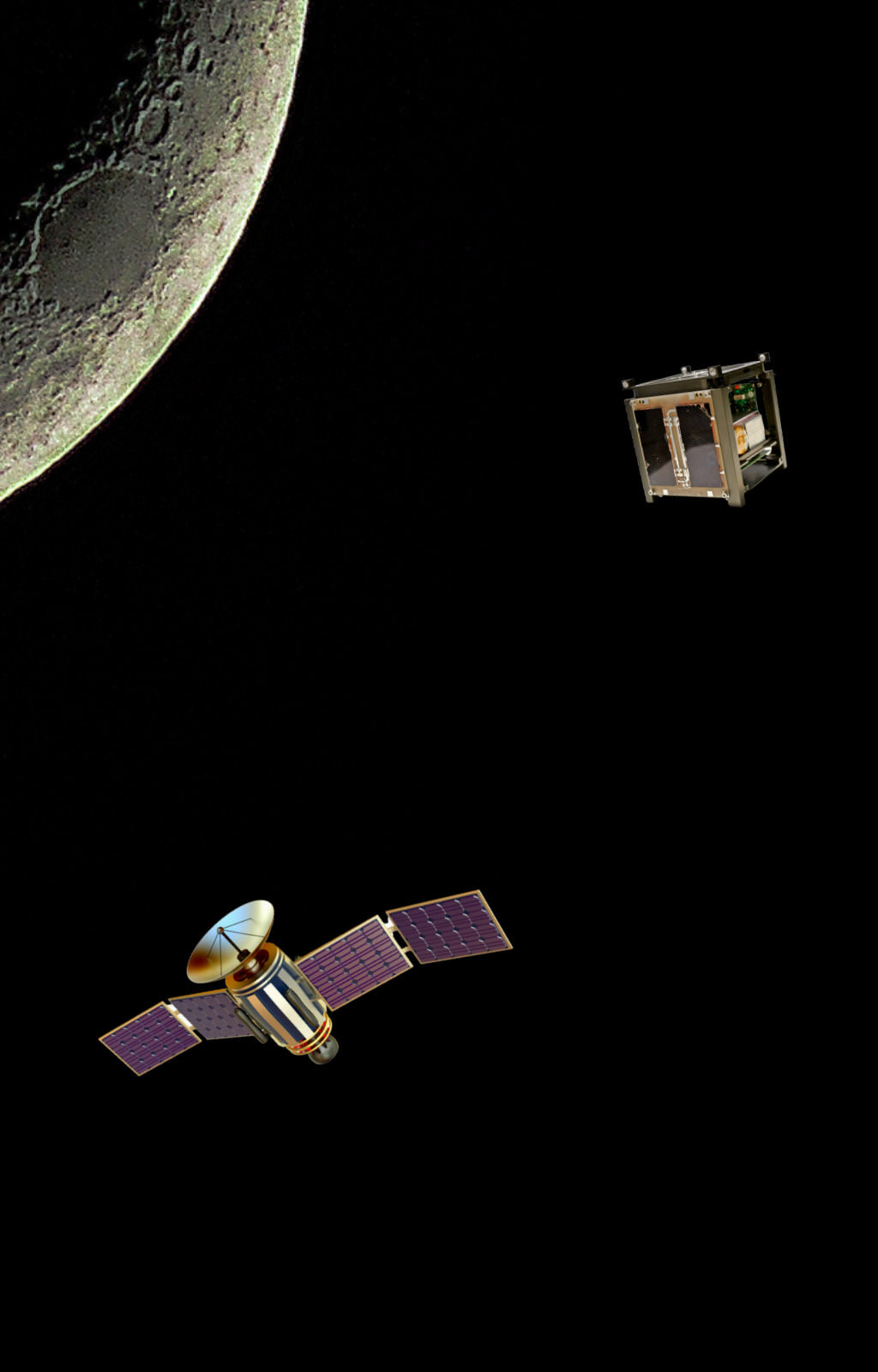
SANTOS, Josivaldo Constantino dos. **A participação ativa e efetiva do aluno no processo de ensino-aprendizagem como condição fundamental.** 2002. 163 f. Dissertação (Mestrado) - Curso de Educação, Ufrgs, Porto Alegre, 2002.

SILVA, Cristiane Samária Gomes da; GIL, María del Carmen Sforza. **Criando Material educacional: invoçao, arduino e movimento maker**. Ediciones Complutense, Madrid, p.129-144, nov. 2019.

SOBRE O ORGANIZADOR

JOSÉ SOARES BATISTA LOPES possui graduação em Engenharia da Computação pela Universidade Potiguar (2004) e mestrado em Engenharia Elétrica e de Computação pela Universidade Federal do Rio Grande do Norte (2011). Doutorado em Engenharia Elétrica e de Computação pela Universidade Federal do Rio Grande do Norte (2016). Já lecionou disciplinas nos cursos de graduação: Análise e Desenvolvimento de Sistemas, Tecnólogo em Automação Industrial e Engenharia Elétrica. Atualmente é professor do Curso Técnico Integrado e Subsequente de Mecatrônica do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte - Campus Parnamirim. Tem experiência na área de Engenharia Elétrica, com ênfase em Automação Eletrônica de Processos Elétricos e Industriais, atuando principalmente nos seguintes temas: Inteligência Artificial, DSPs, Controle, Robótica e Máquinas sem Mancais.

Lattes: http://lattes.cnpq.br/7539214391736004

ORCID: https://orcid.org/0000-0002-8083-1765

@prof_jose_soares