

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

BRUNO ALEXANDRE ZANARDI

**DESENVOLVIMENTO DE UMA BANCADA DE TESTE DE TRAÇÃO
PARA MOTORES *BRUSHLESS* CC APLICADO AO AERODESIGN**

CURITIBA

2021

BRUNO ALEXANDRE ZANARDI

**DESENVOLVIMENTO DE UMA BANCADA DE TESTE DE
TRAÇÃO PARA MOTORES *BRUSHLESS* CC APLICADO AO
AERODESIGN**

**Development of a traction test bench for Brushless DC motors
applied to AeroDesign**

Trabalho de conclusão do curso de Engenharia Elétrica apresentado como requisito para obtenção do título de Engenheiro Eletricista da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador: Prof. Dr. Amauri Amorin Assef

CURITIBA

2021



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es).

Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

BRUNO ALEXANDRE ZANARDI

**DESENVOLVIMENTO DE UMA BANCADA DE TESTE DE TRAÇÃO PARA
MOTORES *BRUSHLESS* CC APLICADO AO AERODESIGN**

Trabalho de Conclusão de Curso de Graduação apresentado
como requisito para obtenção do título de Bacharel em
Engenharia Elétrica da Universidade Tecnológica Federal do
Paraná (UTFPR).

Data de aprovação: 30/novembro/2021

Amauri Amorin Assef
Doutorado
Universidade Tecnológica Federal do Paraná – Campus Curitiba

Winderson Eugenio dos Santos
Doutorado
Universidade Tecnológica Federal do Paraná – Campus Curitiba

Mariana Antonia Aguiar Furucho
Doutorado
Universidade Tecnológica Federal do Paraná – Campus Curitiba

CURITIBA

2021

Dedico este trabalho à minha mãe Simone, meu pai Alberto e meu irmão Gustavo. Sem o apoio de vocês nada seria possível. Dedico também ao meu avô Reginato e avó Marli por todo o auxílio durante meu período de graduação. Agradeço à equipe do AeroDesign UTFPR Curitiba pelo imenso apoio durante o desenvolvimento do trabalho.

RESUMO

Nas etapas preliminares de desenvolvimento do projeto de uma aeronave, é necessário testar o desempenho das hélices na força de tração estática do avião, assim como a tração e a potência do motor em função de uma faixa de velocidade de operação durante o voo. Para se obter uma aeronave com excelentes qualidades de desempenho na competição SAE Brasil AeroDesign, é de fundamental importância que a tração disponível fornecida pela hélice seja determinada com a maior precisão possível e uma das técnicas é pela utilização de uma bancada para testes do motor. No ano de 2020, foi introduzido na categoria regular da competição SAE AeroDesign Brasil a possibilidade da motorização elétrica devido as recentes restrições aos motores a combustão do tipo *glow*. Este trabalho teve como objetivo o desenvolvimento de uma bancada de testes de tração de motor *Brushless* CC, capaz de performar todos os testes necessários a fim de auxiliar a escolha do grupo motopropulsor no projeto desenvolvido pela equipe AeroDesign UTFPR Curitiba. O projeto do sistema de controle e potência inclui um Raspberry Pi +3B como unidade controladora principal e uma placa Arduino Uno para interface com sensores e circuitos periféricos. A unidade controladora é responsável pelo tratamento dos valores dos sensores e interface *Front-End*, além do sinal de acionamento do motor *Brushless* CC. A placa controladora auxiliar possui interface para uma célula de carga de 5 kg, Tubo de Pitot para medição da velocidade do ar, sensor de efeito Hall para leitura de rotação de hélice do motor, sensores para monitoramento de tensão, corrente e temperatura. O software de controle é baseado na ferramenta Node-RED para escolha de parâmetros, identificação de falhas, e armazenamento de configurações e resultados dos testes. Foram realizados testes estáticos para as hélices APC 11x6" e WINDSOR S-1380 em Open Loop e modos Turbo, Degrau e Pico, e dinâmicos para o segundo modelo com modulação PWM até 50%, sendo avaliados os parâmetros de tração, rotação e potência. O teste em Modo *Open Loop* proporcionou, ao usuário, controlabilidade sob o conjunto moto-propulsor, com os dados de tração do motor sendo condizentes com o PWM aplicado. Os testes em Modo Rotina se mostraram uma ótima ferramenta de comparação entre conjuntos moto-propulsores, podendo-se avaliar as diferenças curvas de tração, potência e rotação entre hélices. A realização dos testes dinâmicos foi possível com a validação do sistema de fixação da bancada e também da utilização do Tubo de Pitot para medição da velocidade do ar. Todos os resultados comprovam que o objetivo principal do trabalho foi alcançado com sucesso, sendo disponibilizada ao grupo de Elétrica e Desempenho para a equipe AeroDesign UTFPR Curitiba.

Palavras-chave: *brushless* CC motor; bancada de testes; AeroDesign; sistema embarcado; Node-RED.

ABSTRACT

In the preliminary stages on an aircraft project development, it is necessary to test the propeller performance on the plane's static traction force as well as the traction and power of the engine as a function of an operating speed range during flight. To obtain an aircraft with excellent performance qualities in SAE AeroDesign Brasil competition, it is fundamental that the available traction provided by the propeller be determined with the highest precision as possible and one of the techniques is by using a test bench for the motor. In 2020, it was introduced the possibility of electrical motorization in the regular category of the SAE AeroDesign Brasil competition due to recent restriction with glow-type combustion engines. Then, it is observed the possibility to develop a Brushless DC Motor test bench, capable of performing all tests needed to assist the choice of the powertrain group in the project developed by AeroDesign UTFPR Curitiba. The control and power system design includes a Raspberry Pi +3B as the main controller unit and an Arduino Uno board for interfacing with sensors and peripheral circuits. The controller unit is responsible for processing the values of the sensors and Front-End interface, in addition to the Brushless DC motor drive signal. The auxiliary controller board has an interface for a 5 kg load cell, Pitot tube for measuring air speed, Hall effect sensor for reading engine propeller speed, sensors for monitoring voltage, current and temperature. The control software is based on the Node-RED tool for choosing parameters, identifying faults, and storing test settings and results. Static tests were performed for the APC 11x6" and WINDSOR S-1380 propellers in Open Loop and Turbo, Step and Peak modes, and dynamic tests for the second model with PWM modulation up to 50%, with the parameters of traction, rotation and power. The test in Open Loop Mode provided the user with controllability under the powertrain, with the engine traction data being consistent with the distributed PWM. Testing in Routine Mode is a great tool for comparing powertrain sets, which can be evaluated as differences in traction curves, power and rotation between propellers. The performance of dynamic tests was possible with the validation of the bench support system and also the use of the Pitot Tube to capture the air velocity. All the results prove that the main objective of the work was successfully achieved, being made available to the Electrical and Performance group for the AeroDesign UTFPR Curitiba team.

Keywords: *brushless* CC motor; test bench; AeroDesign; embedded system; Node-RED.

LISTA DE ILUSTRAÇÕES

Figura 1 – Esboços de aeronaves feitos por Leonardo Da Vinci (1485-1490)	16
Figura 2 – Dois esboços marcantes feitos por George Cayley	17
Figura 3 – Principais componentes de uma aeronave	19
Figura 4 – Motor OS .61 tipo <i>glow</i>	19
Figura 5 – Bancada de Tração desenvolvida pela equipe AeroDesign UTFPR Curitiba .	20
Figura 6 – Evolução do Relatórios de Desempenho entre os anos de 2014 à 2019 . . .	21
Figura 7 – Diagrama da metodologia proposta por Marwedel (2010)	23
Figura 8 – Metodologia de trabalho baseado em Marwedel (2010)	25
Figura 9 – Modelo de um sistema mecatrônico	30
Figura 10 – Circuito Ponte de Wheatstone	32
Figura 11 – Representação esquemática do sensor de efeito Hall	33
Figura 12 – Representação esquemática do Tubo de Pitot	34
Figura 13 – Representação esquemática do motor <i>Brushless CC</i>	38
Figura 14 – BEMF trapezoidal	38
Figura 15 – BEMF senoidal	38
Figura 16 – Circuito controlador de motores <i>Brushless CC</i>	39
Figura 17 – Motor BLDC trifásico vs tempo de chaveamento	39
Figura 18 – Diodo <i>Fly-Back</i>	41
Figura 19 – Componente na Programação baseada em Fluxo	43
Figura 20 – Representação gráfica de um nó	44
Figura 21 – Representação do fluxo no Editor	44
Figura 22 – Representação gráfica de um fluxo composto de nós	44
Figura 23 – Representação gráfica de um Subfluxo	45
Figura 24 – Paleta	46
Figura 25 – Visão geral do editor	46
Figura 26 – Raspberry Pi 3B+	48
Figura 27 – Arduino Uno	48
Figura 28 – Célula de Carga e HX711	49
Figura 29 – Tubo de Pitot e módulo MPX7002DP	50
Figura 30 – Módulo Sensor de Efeito Hall 3144E	51
Figura 31 – Sensor de Corrente ACS712 5A	51
Figura 32 – Sensor de Tensão 0-25V CC	51
Figura 33 – Sensor de Temperatura LM35	52
Figura 34 – Bateria Zippy 11,1 V 2500 mAh	52
Figura 35 – Motor Turnigy D3548 / 4 1100KV <i>Brushless Outrunner</i>	53
Figura 36 – Hobbyking 80A ESC 4A SBEC	53
Figura 37 – Sensor Allegro ACS758	54
Figura 38 – Divisão do software do sistema	55
Figura 39 – Fluxograma das escolhas de teste	56
Figura 40 – Etapa 1 - Parâmetros de Teste	57
Figura 41 – Etapa 2	57
Figura 42 – Visualizar Testes Anteriores	58
Figura 43 – Fluxograma da etapa 4	59
Figura 44 – Fluxograma da Etapa 4a	60

Figura 45 – Fluxograma da etapa 4b	61
Figura 46 – Etapa de Implementação	62
Figura 47 – Visão em blocos do projeto	63
Figura 48 – Diagrama dos pinos da placa controladora principal	64
Figura 49 – Sensor ligado à Placa controladora principal	64
Figura 50 – Diagrama dos pinos da placa controladora auxiliar	65
Figura 51 – Sensores próximos ao motor	65
Figura 52 – Sensores de monitoramento	66
Figura 53 – Conversor Buck	66
Figura 54 – Circuito para gravação do <i>bootloader</i> do Arduino	67
Figura 55 – Circuito de Alimentação do Sistema de Potência	67
Figura 56 – Circuito de acionamento do relé	68
Figura 57 – ESC e motor BLDC	68
Figura 58 – Estrutura de fixação do motor e célula de carga	69
Figura 59 – Acoplamento do motor	69
Figura 60 – Fixação do Tubo de Pitot	70
Figura 61 – Bancada montada na parte superior de um veículo	70
Figura 62 – Estrutura de fixação da bancada por meio de ventosas	71
Figura 63 – Fluxo de Entrada dos Parâmetros	72
Figura 64 – Fluxo de tratamento dos valores advindos da placa controladora auxiliar	73
Figura 65 – Interface da etapa de Verificação	74
Figura 66 – Fluxo de verificação do sensor de bateria do motor	74
Figura 67 – Fluxo de realização do teste	75
Figura 68 – Fluxo de gravação do teste	76
Figura 69 – Suporte de fixação previsto anteriormente no projeto	77
Figura 70 – Estrutura de fixação	77
Figura 71 – Sistema de ventosas	78
Figura 72 – Barras estabilizadoras para diminuir movimento de flambagem da bancada	79
Figura 73 – Estrutura de fixação do motor, célula de carga e sensor de rotação	79
Figura 74 – Sistema de coleta de rotações do motor	80
Figura 75 – Bancada montada para testes dinâmicos	80
Figura 76 – Chapa de acrílico protegendo o circuito eletrônico	81
Figura 77 – Vista superior do circuito elétrico	81
Figura 78 – Chave utilizada para desligamento instantâneo do motor	82
Figura 79 – Conexão via cabo Ethernet com o Raspberry Pi 3B+	83
Figura 80 – Passos para ter o acesso remoto do Raspberry Pi 3B+	83
Figura 81 – Abas do Projeto	84
Figura 82 – Acesso ao <i>Front-End</i> para realização dos Testes	85
Figura 83 – Aviso ao usuário pela falta de informação na entrada	85
Figura 84 – Tipos de avisos para a verificação dos sensores	86
Figura 85 – Interface de realização dos testes - Modo <i>Open Loop</i>	86
Figura 86 – Interface de realização dos testes - Modo Rotina	87
Figura 87 – Arquivos CSV contendo os parâmetros dos testes	87
Figura 88 – Arquivos CSV contendo os valores dos sensores durante o teste	87
Figura 89 – <i>Dashboard</i> de Visualização dos Dados	88
Figura 90 – Resultados de (a) Tração e (b) PWM durante o teste de <i>Open Loop</i>	89
Figura 91 – Resultados de (a) Rotação e (b) Potência durante o teste de <i>Open Loop</i>	89
Figura 92 – Resultados de (a) Tensão e (b) Corrente durante o teste de <i>Open Loop</i>	90

Figura 93 – Resultados de (a) Tração e (b) PWM durante o teste de Rotina - Modo Turbo	90
Figura 94 – Resultados de (a) Rotação e (b) Potência durante o teste de Rotina - Modo Turbo	91
Figura 95 – Resultados de (a) Tensão e (b) Corrente durante o teste de Rotina - Modo Turbo	91
Figura 96 – Resultados de (a) Tração e (b) PWM durante o teste de Rotina - Modo Degrau	92
Figura 97 – Resultados de (a) Rotação e (b) Potência durante o teste de Rotina - Modo Degrau	92
Figura 98 – Resultados de (a) Tensão e (b) Corrente durante o teste de Rotina - Modo Degrau	92
Figura 99 – Resultados de (a) Tração e (b) PWM durante o teste de Rotina - Modo Pico .	93
Figura 100 – Resultados de (a) Rotação e (b) Potência durante o teste de Rotina - Modo Pico	93
Figura 101 – Resultados de (a) Tensão e (b) Corrente durante o teste de Rotina - Modo Pico	94
Figura 102 – Resultados de (a) Tração e (b) PWM durante o teste de <i>Open Loop</i>	94
Figura 103 – Resultados de (a) Rotação e (b) Potência durante o teste de <i>Open Loop</i> . . .	95
Figura 104 – Resultados de (a) Tensão e (b) Corrente durante o teste de <i>Open Loop</i>	95
Figura 105 – Resultados de (a) Tração e (b) PWM durante o teste de Rotina - Modo Turbo	96
Figura 106 – Resultados de (a) Rotação e (b) Potência durante o teste de Rotina - Modo Turbo	96
Figura 107 – Resultados de (a) Tensão e (b) Corrente durante o teste de Rotina - Modo Turbo	96
Figura 108 – Resultados de (a) Tração e (b) PWM durante o teste de Rotina - Modo Degrau	97
Figura 109 – Resultados de (a) Rotação e (b) Potência durante o teste de Rotina - Modo Degrau	97
Figura 110 – Resultados de (a) Tensão e (b) Corrente durante o teste de Rotina - Modo Degrau	97
Figura 111 – Resultados de (a) Tração e (b) PWM durante o teste de Rotina - Modo Pico .	98
Figura 112 – Resultados de (a) Rotação e (b) Potência durante o teste de Rotina - Modo Pico	98
Figura 113 – Resultados de (a) Tensão e (b) Corrente durante o teste de Rotina - Modo Pico	99
Figura 114 – Montagem da bancada no topo do veículo para testes dinâmicos	99
Figura 115 – Vista lateral da montagem da bancada no topo do veículo para testes dinâmicos	100
Figura 116 – Conexão com a bancada e leitura dos dados	100
Figura 117 – Resultados da (a) Corrente do Sistema de controle e (b) Velocidade do ar durante o teste de <i>Open Loop</i> - Teste Dinâmico	101
Figura 118 – Resultados de (a) Tração e (b) PWM durante o teste de <i>Open Loop</i> - Teste Dinâmico	101
Figura 119 – Resultados de (a) Temperatura da Bateria e (b) Tensão do Sistema de Controle durante o teste de <i>Open Loop</i> - Teste Dinâmico	102
Figura 120 – Resultados de (a) Rotação e (b) Potência durante o teste de <i>Open Loop</i> - Teste Dinâmico	102
Figura 121 – Resultados de (a) Tensão e (b) Corrente durante o teste de <i>Open Loop</i> - Teste Dinâmico	102
Figura 122 – Utilização da CPU e Memória - Pré-Teste	103
Figura 123 – Utilização da CPU e Memória - Iniciando Node-RED	104
Figura 124 – Utilização da CPU e Memória - Durante o teste	104
Figura 125 – Pesquisa objetiva	114
Figura 126 – Pesquisa objetiva	115
Figura 127 – Fluxo de Entrada dos Parâmetros	117
Figura 128 – Fluxo para a escolha do Tipo de Teste	117
Figura 129 – Fluxo de leitura dos parâmetros de testes anteriores	118

Figura 130	–Interface para a escolha dos parâmetros do teste	118
Figura 131	–Extração dos parâmetros elétricos do motor, ESC e bateria	119
Figura 132	–Avaliação dos parâmetros	119
Figura 133	–Código para envio dos dados ao Microcontrolador Principal	120
Figura 134	–Nó "serial in"	121
Figura 135	–Fluxo de tratamento dos valores advindos do microcontrolador auxiliar . . .	121
Figura 136	–Nó de separação da mensagem serial	122
Figura 137	–Nó de suavização dos valores dos sensores	122
Figura 138	–Fluxo de leitura do sensor BMP180	123
Figura 139	–Nó do tipo <i>exec</i>	123
Figura 140	–Código do sensor BMP180 executado em Python	123
Figura 141	–Nó de separação dos valores do sensor BMP180	124
Figura 142	–Fluxo de realização do teste	125
Figura 143	–Calibração do ESC 80A	126
Figura 144	–Fluxo de gravação do teste	127
Figura 145	–Código construtor do nome do Arquivo CSV	128
Figura 146	–Código para gravar os parâmetros do teste	128
Figura 147	–Código para gravar os dados em CSV	129
Figura 148	–Fluxo de Visualização dos Dados	129

LISTA DE TABELAS

Tabela 1 – Motor BLDC versus Motor <i>Brushed</i> CC	36
Tabela 2 – Motor BLDC versus Motor de Indução CA	37
Tabela 3 – Comparação entre baterias permitidas pela competição SAE AeroDesign Brasil	42
Tabela 4 – Custo do Sistema de Controle	130
Tabela 5 – Custo do Sistema de Potência	130
Tabela 6 – Custos da parte mecânica e chicote elétrico	130

LISTA DE ABREVIATURAS, SIGLAS E ACRÔNIMOS

ABREVIATURAS

m/s metros por segundo

SIGLAS

°C	Graus Celsius
A	Ampère
a.C	Antes de Cristo
A/D	Analógico-Digital
API	<i>Application Programming Interface</i>
BEMF	<i>Back electromotive force</i> (Força contra eletromotriz)
BLDC	<i>Brushless DC Motor</i>
BJT	<i>Bipolar Junction Transistor</i>
CA	<i>Corrente Alternada</i>
CC	<i>Corrente Contínua</i>
CI	<i>Circuito Integrado</i>
CMOS	<i>Complementary metal–oxide–semiconductor</i>
CPU	Central Processing Unit
CSV	<i>Comma-separated values</i>
d.C	Depois de Cristo
dB	Decibel
DRAM	<i>Dynamic Random Access Memory</i>
EPROM	<i>Erasable Programmable Read Only Memory</i>
ESC	<i>Electronic Speed Control</i>
<i>ft</i>	<i>feet</i> (pés)
GHz	Gigahertz
GND	<i>Ground</i>
GPIO	<i>General Purpose Input/Output</i>
HDMI	<i>High-Definition Multimedia Interface</i>
<i>hp</i>	<i>horse power</i>
HTTP	<i>Hypertext Transfer Protocol</i>
HW	Hardware
I/O	<i>Input/Output</i>
I ² C	<i>Inter-Integrated Circuit</i>
ICSP	<i>In-circuit Serial Programming</i>
IDE	<i>Integrated Development Environment</i>

IEEE	<i>Institute of Electrical and Electronic Engineers</i>
IGBT	<i>Insulated Gate Bipolar Transistor</i>
IoT	<i>Internet of Things</i>
IP	<i>Internet Protocol</i>
JTAG	<i>Joint Test Action Group</i>
Kbps	<i>Kilobit por segundo</i>
LAN	<i>Local Area Network</i>
lb	<i>Libra</i>
LED	<i>Light Emitting Diode</i>
m	<i>metros</i>
mAh	<i>miliampére-hora</i>
Mbps	<i>Megabit por segundo</i>
MOSFET	<i>Metal-Oxide-Semiconductor Field Effect Transistor</i>
mV	<i>miliVolt</i>
N	<i>Newton</i>
OFMD	<i>Orthogonal frequency-division multiplexing (multiplexação por divisão de frequências ortogonais)</i>
Pa	<i>Pascal</i>
PBF	<i>Programação baseada em fluxo</i>
PI	<i>Pacotes de Informação</i>
PWM	<i>Pulse Width Modulation</i>
RPM	<i>Rotações por minuto</i>
SAE	<i>Society of Automotive Engineers</i>
SBEC	<i>Switching Battery Eliminator Circuit</i>
SCL	<i>Serial Clock</i>
SD	<i>Secure Digital Card</i>
SDA	<i>Serial Data</i>
SRAM	<i>Static Random Access Memory</i>
UML	<i>Unified Modeling Language</i>
UTFPR	<i>Universidade Tecnológica Federal do Paraná</i>
V	<i>Volt</i>
VDC	<i>Voltage Direct Current</i>
VNC	<i>Virtual Network Computing</i>
W	<i>Watts</i>
WLAN	<i>Wireless Local Area Network</i>

SUMÁRIO

1	INTRODUÇÃO	16
1.1	TEMA	16
1.1.1	Delimitação do Tema	18
1.2	PROBLEMAS E PREMISSAS	18
1.3	OBJETIVOS	22
1.3.1	Objetivo Geral	22
1.3.2	Objetivo Específico	22
1.4	JUSTIFICATIVA	22
1.5	PROCEDIMENTOS METODOLÓGICOS	23
1.6	ESTRUTURA DO TRABALHO	24
2	REVISÃO DA LITERATURA	27
2.1	PROJETO AERODESIGN	27
2.2	CONTROLE E COMUNICAÇÃO DO SISTEMA EMBARCADO	29
2.3	SENSORES E ATUADORES	30
2.3.1	Sensores	30
2.3.1.1	Medição de Tração do Motor	31
2.3.1.2	Medição de Temperatura	32
2.3.1.3	Medição de Corrente do Sistema	32
2.3.1.4	Medição da Pressão	33
2.3.1.5	Medição do Escoamento de Ar	34
2.3.1.6	Medição de Rotação	35
2.3.2	Atuadores	35
2.3.2.1	Motores <i>Brushless</i> CC	35
2.3.2.2	Relé Eletromagnético	40
2.3.2.3	Bateria	40
2.4	PROGRAMAÇÃO BASEADA EM FLUXO	43
2.4.1	Programação baseada em fluxo com Node-RED	43
2.4.1.1	Nós	43
2.4.1.2	Fluxo	44
2.4.1.3	Contexto	44
2.4.1.4	Mensagem	45
2.4.1.5	Subfluxo	45
2.4.1.6	Fiação ou Conexões	45
2.4.1.7	Paleta	45
2.4.1.8	Área de Trabalho	46
3	LEVANTAMENTO DE ESPECIFICAÇÃO DE COMPONENTES DO SISTEMA DE CONTROLE E POTÊNCIA	47
3.1	LEVANTAMENTO DE HARDWARE	47
3.1.1	Sistema de Controle	47
3.1.1.1	Unidade Controladora Principal	47
3.1.1.2	Unidade Controladora Auxiliar	48
3.1.1.3	Sensores e Periféricos	49
3.1.1.4	Bateria	52

3.1.2	Sistema de Potência	52
3.1.2.1	Motor <i>Brushless</i> CC	52
3.1.2.2	Controlador de Velocidade (ESC)	53
3.1.2.3	Bateria	53
3.1.2.4	Relé Eletromagnético	54
3.1.2.5	Sensor de Corrente do Motor	54
4	ESPECIFICAÇÃO DO SISTEMA DE SOFTWARE	55
4.1	VISÃO GERAL	55
4.2	ESCOLHA DOS PARÂMETROS	56
4.3	ETAPA 2 - IDENTIFICAÇÃO DE FALHAS	57
4.4	ETAPA 3 - VISUALIZAR TESTES ANTERIORES	58
4.5	ETAPA 4 - INICIAR TESTE	58
5	IMPLEMENTAÇÃO	62
5.1	HARDWARE DESIGN	62
5.1.1	Diagrama Elétrico - Sistema de Controle	62
5.1.2	Diagrama Elétrico - Sistema de Potência	65
5.1.3	Bancada Mecânica	68
5.2	SOFTWARE DESIGN	71
5.2.1	Entrada de Parâmetros	71
5.2.2	Coleta de dados dos sensores	72
5.2.3	Etapa de Verificação	73
5.2.4	Modos de teste	74
5.2.5	Etapa de Visualização de Dados	74
6	RESULTADOS	77
6.1	MONTAGEM FÍSICA DA BANCADA	77
6.1.1	Montagem mecânica	77
6.1.2	Montagem elétrica	80
6.2	COMUNICAÇÃO E INTERFACE GRÁFICA	82
6.3	TESTES	88
6.3.1	Teste Estático com a Hélice APC 11x6	89
6.3.1.1	<i>Open Loop</i>	89
6.3.1.2	Modo Turbo	90
6.3.1.3	Modo Degrau	91
6.3.1.4	Modo Pico	93
6.3.2	Teste Estático com a Hélice WINDSOR S-1380	94
6.3.2.1	<i>Open Loop</i>	94
6.3.2.2	Modo Turbo	95
6.3.2.3	Modo Degrau	96
6.3.2.4	Modo Pico	98
6.4	TESTE DINÂMICO COM A HÉLICE WINDSOR S-1380	99
6.5	DESEMPENHO DO SOFTWARE	103
7	CONCLUSÕES E CONSIDERAÇÕES FINAIS	105
	REFERÊNCIAS	108

APÊNDICES	113
APÊNDICE A – PESQUISA DE CAMPO	114
APÊNDICE B – ENTRADA DE PARÂMETROS - DETALHAMENTO DE SOFTWARE	116
APÊNDICE C – COLETA DE DADOS DOS SENSORES - DETALHA- MENTO DE SOFTWARE	120
APÊNDICE D – MODOS DE TESTE - DETALHAMENTO DE SOFT- WARE	125
APÊNDICE E – ETAPA DE VISUALIZAÇÃO DE DADOS - DETA- LHAMENTO DE SOFTWARE	127
APÊNDICE F – ORÇAMENTO DO PROJETO	130

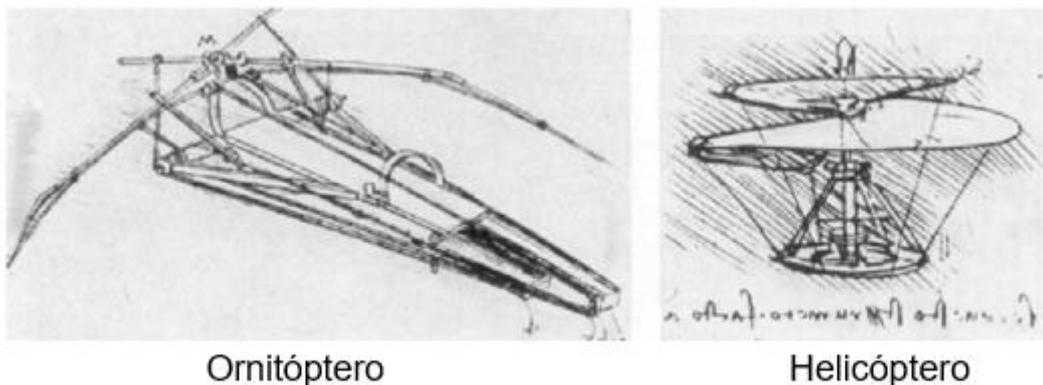
1 INTRODUÇÃO

Sistemas embarcados têm se tornado cada vez mais essenciais no nosso dia a dia. Eles controlam funções importantes em nossos carros, como em *airbags*, sistemas de freio, fechamento das portas, entre outras funções. O desenvolvimento de sistemas embarcados modernos vem se tornando cada vez mais difícil e desafiante. Esses sistemas têm papel fundamental em muitas empresas do setor *high-tech*; são essenciais nos sistemas de transporte modernos - desde carros, veículos ferroviários e aeronaves. Mais e mais os softwares serão cruciais para a funcionalidade e integração de componentes mecânicos e elétricos. Essa integração irá finalmente solucionar o problema do pouco contato entre disciplinas de engenharia mecânica, engenharia elétrica e ciência da computação (K. *et al.*, 2012).

1.1 TEMA

A habilidade de voar é de interesse do ser humano desde as histórias mitológicas da antiguidade, quando foi percebida a inveja humana na capacidade de voo das aves, criando um desejo de repetir o mesmo feito (TORENBEEK; WITTENBERG, 2009). De acordo com Petrescu *et al.* (2017), Archytas de Taranto, filósofo da Grécia Antiga, foi considerado o primeiro indivíduo a desenvolver um dispositivo artificial de voos com autopropulsão, com um modelo em formato de pássaro, no ano 400 a.C. Por volta de 1500 d.C., Leonardo da Vinci propôs uma série de ideias de "máquinas voadoras", baseadas no conceito de asas *swinging* (PETRESCU *et al.*, 2017). Os esboços podem ser vistos na Figura 1.

Figura 1 – Esboços de aeronaves feitos por Leonardo Da Vinci (1485-1490)



Ornitóptero

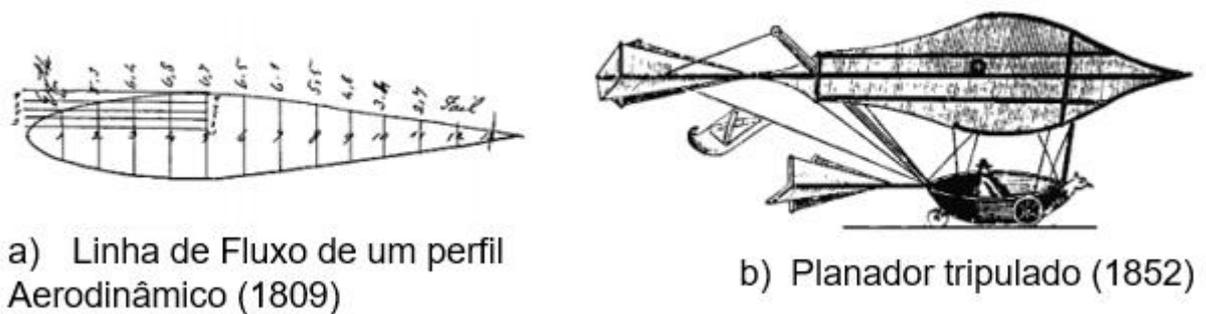
Helicóptero

Fonte: Torenbeek e Wittenberg (2009).

O design atual de aeronaves teve sua origem em 1799 pelo britânico George Cayley, que

determinou a mudança do modelo de asas "batendo", que simulam um pássaro, para um modelo de asas fixas (TORENBEEK; WITTENBERG, 2009). Além disso, como meio de estabilização da aeronave, propôs uma estrutura vertical e horizontal em forma de cauda, tornando-se o primeiro conceito de asa fixa, que futuramente possibilitou o princípio do voo mecânico (TORENBEEK; WITTENBERG, 2009). Na Figura 2 é possível analisar os esboços feitos por George Cayley.

Figura 2 – Dois esboços marcantes feitos por George Cayley



a) Linha de Fluxo de um perfil Aerodinâmico (1809)

b) Planador tripulado (1852)

Fonte: Torenbeek e Wittenberg (2009).

Até cerca de 1910, a aviação se manteve limitada ao sucesso de experimentos pioneiros e, posteriormente, cientistas e engenheiros tomaram a liderança no desenvolvimento de futuros projetos (TORENBEEK; WITTENBERG, 2009). Com o início da Primeira Guerra Mundial, a indústria aeronáutica teve um grande salto, marcando definitivamente a importância do poderio aéreo, crescendo na Europa e Estados Unidos, criando novos nichos econômicos (FAJER, 2009). Desde então, há um trabalho contínuo de desenvolvimento em melhorias aerodinâmicas e desempenho das aeronaves, propiciando o projeto e construção de aeronaves capazes de realizarem voo transcontinentais, aviões que ultrapassam a barreira do som e até a realização de voos espaciais (RODRIGUES, 2014).

Especificamente no Brasil, o estudo em projetos de aeronave sempre se impulsionou pelo desejo em repetir os feitos de Santos Dummont (RODRIGUES, 2014). No ambiente acadêmico, com o objetivo do aprimoramento da aprendizagem relacionada diretamente à engenharia, a SAE (Sociedade de Engenheiros Automotivos) Brasil promove, desde 1999, competições a fim de gerar uma oportunidade de aprendizado na área aeronáutica através de um projeto multidisciplinar desafiador (AERODESIGN, 2020a). A competição oferece uma oportunidade ímpar aos estudantes de engenharia, organizados em equipes, de desenvolverem um projeto em todas as suas etapas, desde a sua concepção, projeto detalhado e ensaios em voo

(AERODESIGN, 2020a). O objetivo da competição é simples: projetar, construir e testar uma aeronave rádio controlada original, que seja a mais otimizada possível em todos os aspectos da missão (AERODESIGN, 2020a).

1.1.1 Delimitação do Tema

Para o ano de 2020, a SAE Brasil promoveu alterações bruscas em seu regulamento, possibilitando a utilização de motores elétricos na classe Regular da competição, classe na qual a equipe AeroDesign UTFPR Curitiba atualmente participa. A utilização de tal motor exigirá um *know-how* novo para a equipe, uma vez que em todas as competições passadas a equipe utilizou o motor a combustão do tipo *glow*, motor geralmente utilizado para tracionar veículos aéreos pequenos não tripulados, usando uma mistura de combustível composta de metanol e uma pequena quantidade de nitrometano (RAINE; THORWARTH, 2004). A alteração do motor acarreta em mudanças no desenvolvimento do projeto elétrico, como dimensionamento do chicote e bateria, além de influenciar diretamente na escolha de uma nova hélice para o grupo motopropulsor. Com isso, tornou-se importante o desenvolvimento de um sistema capaz de monitorar e controlar o comportamento desses motores em ensaios estáticos e dinâmicos, a fim de gerar dados relevantes como *output* para a área de desempenho.

Este trabalho visou o desenvolvimento de uma bancada de teste de tração, estático e dinâmico, aplicado à uma equipe de AeroDesign. A bancada é capaz de informar todas as informações necessárias para as áreas afetadas do projeto, a fim de facilitar a adequação da equipe ao motor elétrico proposto pela SAE Brasil, motor esse que será obrigatório para todas as equipes da classe regular a partir do ano de 2023 (AERODESIGN, 2020b).

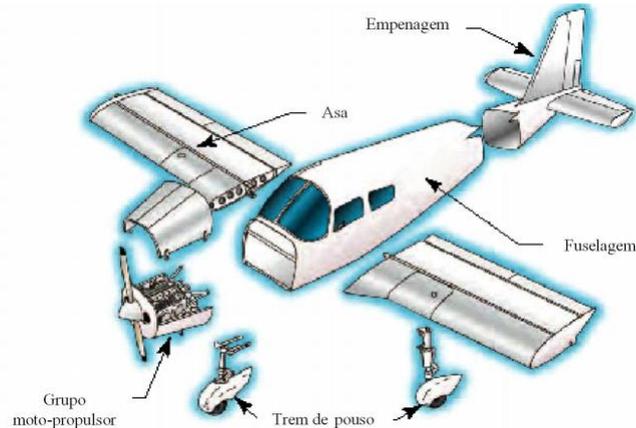
O tema deste trabalho foi o desenvolvimento de um sistema embarcado capaz de controlar e monitorar um motor elétrico do tipo *Brushless* CC sendo capaz de realizar testes dinâmicos e estáticos, a fim de informar o empuxo fornecido pelo motor em diferentes configurações de hélices. Esta bancada será de utilização da equipe de AeroDesign da Universidade Tecnológica Federal do Paraná, intitulada AeroDesign UTFPR Curitiba.

1.2 PROBLEMAS E PREMISSAS

Um avião é definido como uma aeronave de asa fixa mais pesada que o ar, movida por propulsão mecânica, mantendo as condições de voo devido à reação dinâmica do ar que

escoa através de suas asas (RODRIGUES, 2014). As aeronaves geralmente são divididas em 5 subgrupos, conforme é visto da Figura 3: asas, fuselagem, trem de pouso, empenagem e grupo motopropulsor. Esse grupo é composto pelo conjunto de motor e hélice, sendo o motor a combustão obrigatório para a classe Regular da competição SAE Brasil AeroDesign até o ano de 2019.

Figura 3 – Principais componentes de uma aeronave



Fonte: Rodrigues (2014).

Os motores de combustão interna são datados do século 19, quando Otto, em 1876, desenvolveu o motor de ignição por faísca e Diesel, em 1892, inventou o motor de ignição a compressão (HEYWOOD, 1988). A potência produzida pelo motor de combustão interna a pistão varia com o tamanho e o número de cilindros, com a rotação do motor e a densidade do ar (RODRIGUES, 2014). A hélice é colocada em movimento de rotação pelo eixo do motor e suas pás se movem através do ar como asas rotativas, criando uma força de sustentação perpendicular ao movimento de rotação denominada tração ou empuxo (RODRIGUES, 2014). Na Figura 4 é possível verificar um dos modelos de motor a combustão disponível para as equipes participantes da Competição SAE Brasil AeroDesign da classe Regular.

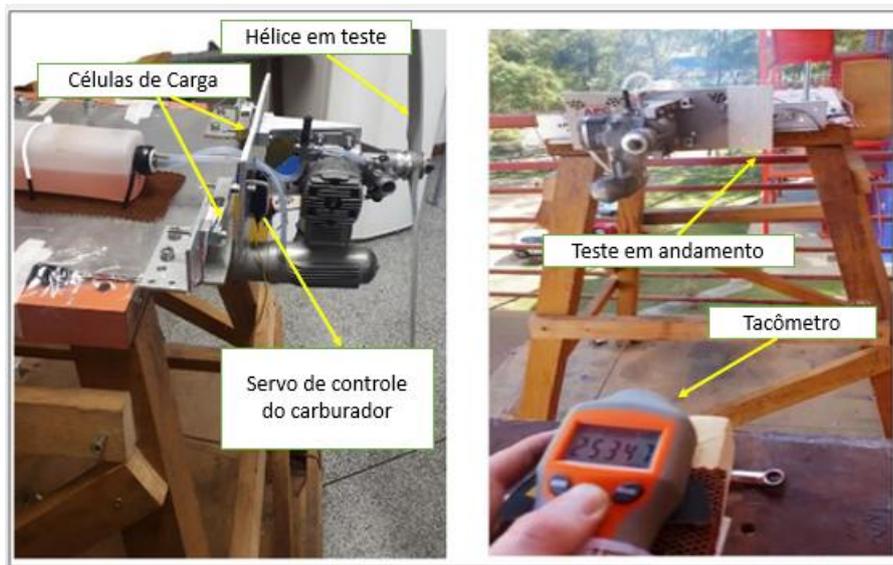
Figura 4 – Motor OS .61 tipo glow



Fonte: Rodrigues (2014).

Nas etapas preliminares de desenvolvimento do projeto da aeronave, é necessário testar o desempenho das hélices na força de tração estática do avião, assim como a tração e a potência do motor em função de uma faixa de velocidade de operação durante o voo (AMBONI; CURI, 2018). Para se obter uma aeronave com excelentes qualidades de desempenho na competição SAE Brasil AeroDesign, é de fundamental importância que a tração disponível, fornecida pela hélice, seja determinada com a maior precisão possível e uma das técnicas é pela utilização de uma bancada para testes do motor (RODRIGUES, 2014). A equipe AeroDesign UTFPR Curitiba dispõe de uma bancada de teste de tração dinâmico e estático capaz de informar o empuxo fornecido pelo motor. A bancada pode ser vista na Figura 5.

Figura 5 – Bancada de Tração desenvolvida pela equipe AeroDesign UTFPR Curitiba



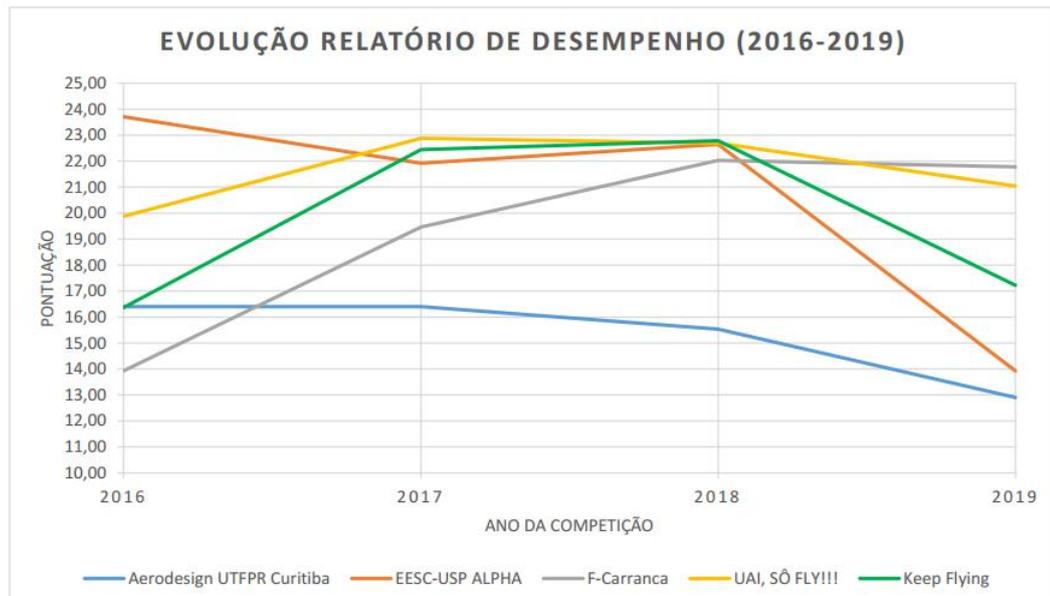
Fonte: Autoria própria (2021).

A bancada de testes utiliza células de carga para a medição da tração fornecida pelo conjunto motopropulsor, além de um servomotor utilizado para controlar a entrada de ar do motor a combustão. Conjuntamente, utiliza-se um tacômetro para a medição dos valores de rotação da hélice. Os dados obtidos são fornecidos à subárea de Desempenho, que a partir deles define a capacidade de voo da aeronave, plotando as curvas de tração estática disponível de cada hélice, possibilitando o cálculo da tração disponível em função da velocidade de voo (RODRIGUES, 2014).

Apesar da bancada de tração ter sido fundamental para o desenvolvimento dos projetos anteriores, essa forma de medição se mostrou muito problemática nos últimos anos devido à vibração excessiva, causando mau contato no controlador da bancada, além de problemas de medição na rotação da hélice devido ao erro humano. Esses problemas podem gerar incertezas

na coleta dos dados, prejudicando o desenvolvimento fiel do projeto. Na Figura 6 é mostrado o histórico de notas do relatório de Desempenho da equipe AeroDesign UTFPR Curitiba, em comparação com equipes que se mostraram vitoriosas em competições recentes da SAE Brasil AeroDesign.

Figura 6 – Evolução do Relatórios de Desempenho entre os anos de 2014 à 2019



Fonte: Autoria própria (2021).

É possível notar uma considerável diferença de pontuação entre as equipes e uma evidente queda de pontuação da equipe AeroDesign UTFPR Curitiba no ano de 2019. A partir de 2020, foi introduzido na categoria regular da competição a possibilidade da motorização elétrica devido as recentes restrições aos motores do tipo *glow* (AERODESIGN, 2020b).

Devido à essa alteração do regulamento, o remodelamento da bancada para suportar a especificação de um motor elétrico se tornou necessária. Foi observado, então, a possibilidade do desenvolvimento de uma nova bancada de testes de tração do motor, corrigindo todos os pontos problemáticos da anterior e acrescentando novas ideias de tal forma que possa responder a pergunta: qual a melhor maneira de medir o desempenho de um conjunto moto-propulsor que utiliza motor elétrico do tipo *Brushless* CC? Essas implementações serão cruciais para o aumento da competitividade da equipe na competição, tornando possível angariar melhores posições na competição SAE Brasil AeroDesign.

1.3 OBJETIVOS

1.3.1 Objetivo Geral

Este trabalho teve como objetivo desenvolver uma bancada para ensaios de tração de um motor *Brushless* CC, a fim de auxiliar o desenvolvimento de uma aeronave radiocontrolada pela equipe AeroDesign UTFPR Curitiba.

1.3.2 Objetivo Específico

O sistema embarcado é controlado por meio da placa Raspberry Pi 3b+ e, com o teste sendo conduzido na plataforma Node-RED. O *firmware* é capaz de interpretar os dados advindos dos sensores e fornecer os seguintes dados em tempo real para o usuário, incluindo: tração (N), corrente consumida pelo motor e sistema de controle (A), tensão (V) das baterias do Sistema de Controle e de Potência, potência do motor (W), Velocidade relativa do ar (m/s), rotação (RPM), temperatura (°C), altitude (m) e pressão (Pa) ambiente. O projeto conta também com uma interface gráfica (*Front-End*) que é a relação do usuário com o sistema. Também foi de escopo desse projeto o dimensionamento mecânico da bancada.

Os objetivos específicos também incluem:

- Desenvolver um sistema confiável para a coleta de dados;
- Fornecer segurança para o usuário;
- Implementar um sistema intuitivo e de alta usabilidade;
- Realizar testes de validação do sistema proposto.

1.4 JUSTIFICATIVA

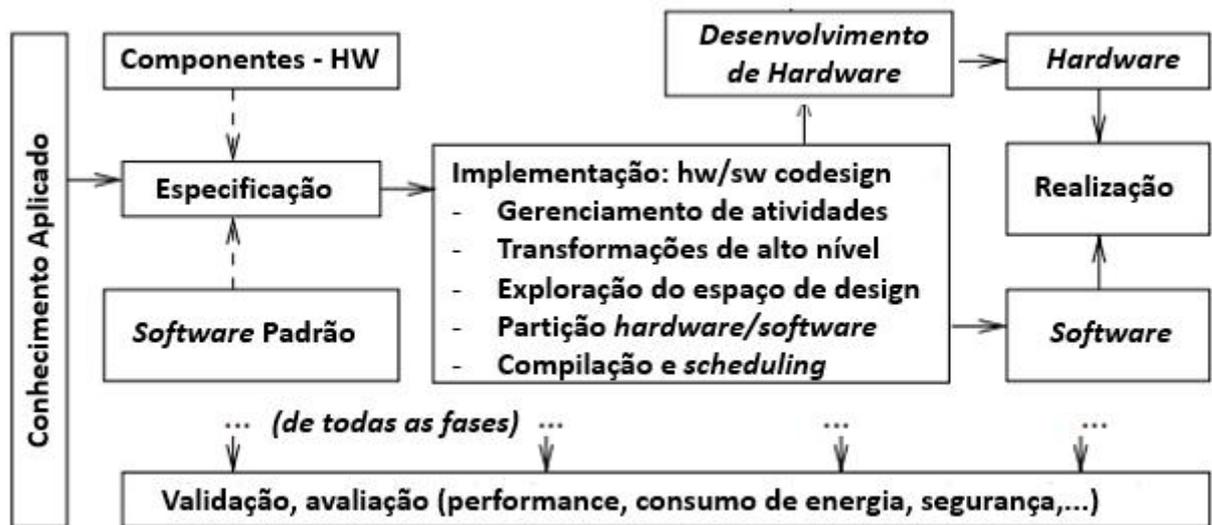
Justifica-se este trabalho como uma oportunidade para o início da aplicação dos motores elétricos na equipe AeroDesign UTFPR Curitiba, uma vez que o seu uso será obrigatório a partir do ano de 2023 na Competição SAE Brasil AeroDesign. Esta bancada facilitará a realização dos ensaios de tração estático e dinâmico dos motores *Brushless* CC, ajudando no desenvolvimento do projeto da aeronave. Alinhado às expectativas da equipe, o projeto pretende angariar um impulso na pontuação em futuras competições, aumentando o reconhecimento da universidade

Brasil fora. Além disso, esse trabalho tem como intuito cativar ainda mais alunos de Engenharia Elétrica e Eletrônica a participarem do projeto do AeroDesign, uma vez que, atualmente, a taxa de alunos desses cursos são baixas comparadas a outros cursos ofertados pela universidade.

1.5 PROCEDIMENTOS METODOLÓGICOS

O projeto desenvolvido nesse trabalho é definido como um sistema embarcado. De acordo com Marwedel (2010), sistemas embarcados são sistemas de processamento de informações que estão acoplados em um produto que normalmente não estão visíveis ao usuário. Desta maneira, foram pesquisadas metodologias de desenvolvimento específicas para esse tipo de sistema de forma a guiarem o projeto. Para Wolf (1994), a computação embarcada é única porque é um problema denominado hardware-software co-design, em que o hardware e software devem ser desenvolvidos em conjunto para se ter certeza que as implementações não só funcionem adequadamente, mas também alcancem o objetivo de desempenho, custo e confiabilidade. Levando em consideração essa característica dos sistemas embarcados, a metodologia proposta por Marwedel (2010) foi a mais precisa para o trabalho por possuir etapas bem detalhadas. Na Figura 7 é mostrado o *flowchart* da metodologia.

Figura 7 – Diagrama da metodologia proposta por Marwedel (2010)



Fonte: Tradução de Marwedel (2010).

A metodologia se inicia com as ideias do projeto capturadas e levadas às especificações. Uma das grandes características desta metodologia é a reutilização de componentes de hardware e software já disponíveis, sendo também levadas às especificações (MARWEDEL, 2010). Para

Wolf (1994), as especificações não incluem somente os requerimentos funcionais, mas também os não funcionais, incluindo velocidade, potência e custo de manufatura. No entanto, Marwedel possui uma abordagem diferente para o mesmo termo. As especificações devem ser captadas em forma de leitura de máquina seguindo uma diversidade de critérios. A escolha da linguagem usada no desenvolvimento depende do domínio de aplicação e no ambiente em que o projeto será feito (MARWEDEL, 2010). Uma das primeiras etapas do trabalho é definir a linguagem para as especificações. Com as especificações finalizadas, as atividades de implementação se iniciam. Com a implementação finalizada, o fluxograma segue para caminhos diferentes: hardware-design e software. A literatura apresentada por Marwedel não apresenta detalhadamente o que cada uma dessas etapas é responsável, portanto, há uma flexibilidade nas atividades a serem realizadas. Para a etapa de hardware-design, definiu-se como atividades:

- Diagrama elétrico do sistema;
- Dimensionamento mecânico da bancada desenvolvido no software NX Siemens®.

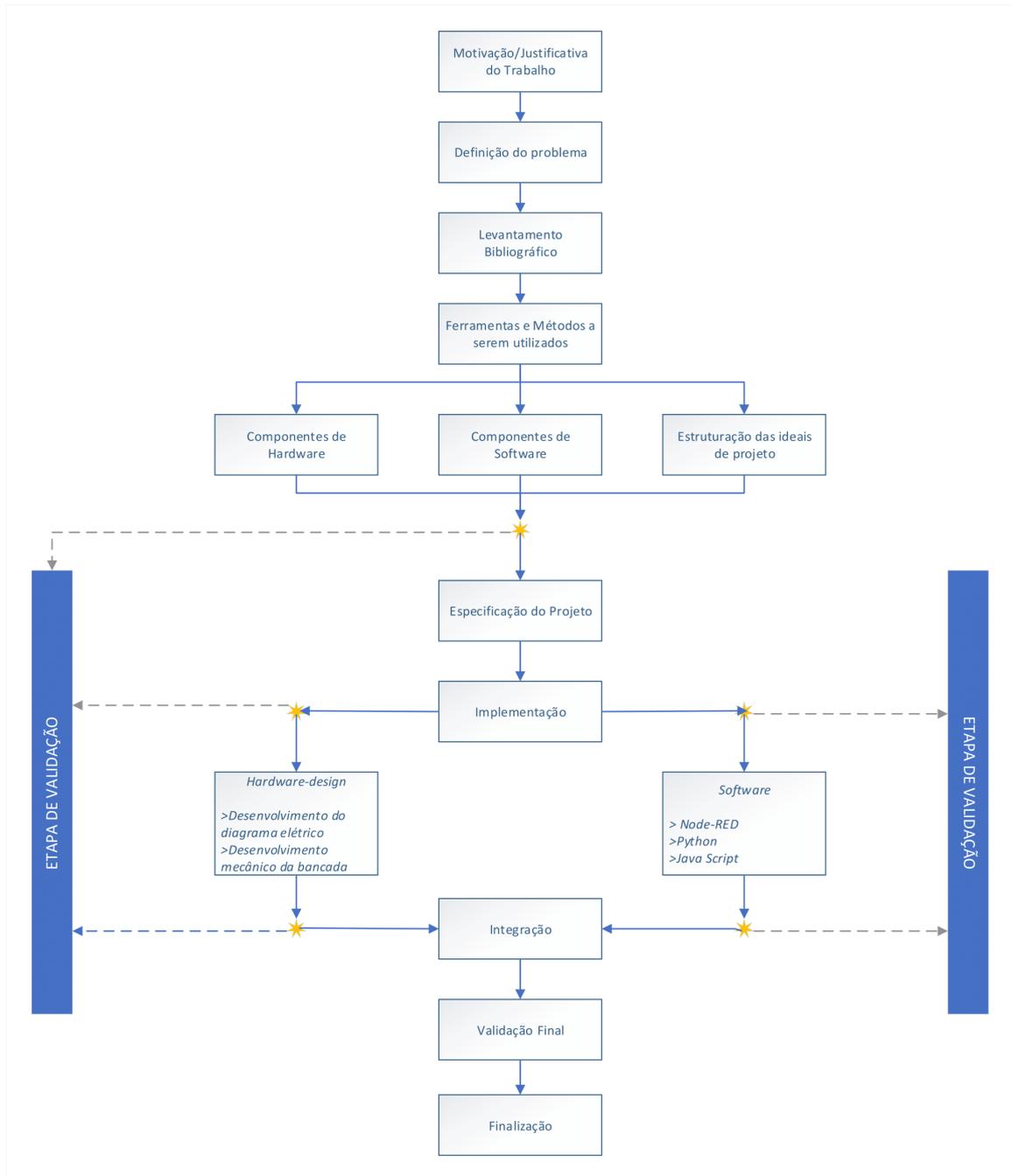
A etapa de software é de implementação do *firmware* na ferramenta Node-RED, utilizando as linguagens de programação Python e JavaScript e C. Com a finalização dessas etapas, fez-se a integração do projeto.

A etapa de validação é o processo de certificar se um certo projeto é apropriado para certo propósito, está de acordo com limites e irá performar como o esperado (MARWEDEL, 2010). De acordo com Marwedel (2010), o projeto poderia ser verificado no final de cada processo, mas essa abordagem infelizmente não funciona devido às grandes diferenças dos níveis de abstração usados nas especificações para as utilizadas na implementação. Em diversos casos, o projeto deve ser testado em ambientes reais antes de suas reais versões sejam manufaturadas. Nesta etapa, também são realizados testes de validação por meio de injeções de falhas. Também são definidos pontos estratégicos durante o projeto, nos quais são feitas as etapas de validação tanto de hardware como de software. Na Figura 8 é mostrada a metodologia de Marwedel adaptada para o trabalho.

1.6 ESTRUTURA DO TRABALHO

- Capítulo 2 - Fundamentação Teórica: é apresentado a base teórica para o entendimento completo do trabalho;

Figura 8 – Metodologia de trabalho baseado em Marwedel (2010)



Fonte: Autoria própria (2021).

- Capítulo 3 - Levantamento de Especificação do Sistema de Controle e de Potência: é apresentado as exigências do trabalho, tal como escolha de hardware e software;
- Capítulo 4 - Especificação do Sistema de Software: é desenvolvido a máquina de estados do projeto;
- Capítulo 5 - Implementação: são estudadas melhorias no código a ser implementado com base nas Especificações. Nesta etapa, dividiu-se o trabalho em dois segmentos: hardware e

software;

- Capítulo 6 - Resultados: etapa de integração entre software e hardware, com a realização de testes para a validação do sistema embarcado;
- Capítulo 7 - Conclusão: é apresentado os resultados finais do projeto, além de sugestões de melhoria para o futuro.

2 REVISÃO DA LITERATURA

Um sistema embarcado é um sistema baseado em microprocessador, construído para controlar uma função ou um *range* de funções, não projetado para ser alterado pelo usuário final da mesma maneira que um computador pessoal é (HEATH, 2002). O usuário pode realizar escolhas relativas à funcionalidade, mas não pode alterar a funcionalidade do sistema embarcado (HEATH, 2002). Uma das características de um sistema embarcado é que ambos hardware e software devem ser levados em consideração (MARWEDEL, 2010). Esse capítulo tem como objetivo realizar o levantamento teórico do trabalho, dividindo-se em quatro seções:

1. Projeto AeroDesign: nesta seção é explicado o funcionamento da competição e os requisitos da bancada;
2. Controle e Comunicação do Sistema Embarcado: nesta seção é explicado os fundamentos dos sistemas embarcados;
3. Sensores e Atuadores: nesta seção é abordado o funcionamento de todos os sensores que serão utilizados no projeto, além dos atuadores e sistema de alimentação;
4. Programação baseada em Fluxo: nesta seção é abordado a Programação baseada em Fluxo e a ferramenta Node-RED.

2.1 PROJETO AERODESIGN

Segundo Aerodesign (2020a), "O Projeto AeroDesign consiste em uma competição de engenharia, aberta a estudantes universitários de graduação e pós-graduação em Engenharias ligadas à mobilidade."

A competição SAE AeroDesign ocorre nos Estados Unidos desde 1986, tendo sido concebida e realizada pela SAE International e, a partir de 1999, esta competição passou a constar também no calendário de eventos estudantis da SAE Brasil (AERODESIGN, 2020a). A competição oferece uma oportunidade única aos estudantes, organizados em equipes, de desenvolverem um projeto aeronáutico em todas as suas etapas, desde a concepção, detalhamento do projeto, construção e testes, até colocá-lo em prova durante a competição (AERODESIGN, 2020a).

A competição pode ser dividida em Competição de Voo e Competição de Projeto. Na competição de Voo, leva-se em consideração a máxima carga que a aeronave pode levar, além de sua eficiência estrutural, peso vazio real, confiabilidade entre outros aspectos (AERODESIGN, 2020a). A competição de projeto as equipes apresentam seus estudos para determinar a pontuação máxima que a aeronave pode obter durante a competição de voo.

Para a realização do projeto, cada equipe se divide em subáreas responsáveis por diferentes estudos na constituição da aeronave. De forma geral, as equipes presentes na competição se subdividem nas seguintes áreas: Aerodinâmica, Desempenho, Estabilidade e Controle, Cargas e Aeroelasticidade, Estruturas e Ensaios Estruturais, Projeto Elétrico e Plantas. A subárea de Desempenho é responsável pela escolha do grupo motopropulsor do projeto, composto pelo motor e pela hélice. A conversão do torque fornecido pelo motor em expuxo através da hélice gera a força de tração, sendo essa força a responsável por impulsionar a aeronave em voo (RODRIGUES, 2014). A competição do AeroDesign limita a potência do motor elétrico em 700W (AERODESIGN, 2020b). Portanto, a única alternativa de se aumentar a tração disponível é escolhendo a hélice que propicie as qualidades de desempenho desejadas (RODRIGUES, 2014).

As curvas de tração disponível estática e em função da velocidade são fundamentais para a definição do grupo motopropulsor. A tração estática fornecida por cada hélice é uma das condições na qual se torna possível saber qual das hélices estudadas propicia as melhores condições para a decolagem da aeronave (RODRIGUES, 2014). Este tipo de teste pode ser determinado com a realização do ensaio estático do grupo motopropulsor ou por meio de um modelo matemático (RODRIGUES, 2014). Como informa Rodrigues (2014), o modelo analítico que fornece a tração estática de forma aproximada é definido da seguinte forma:

$$T_{v=0} = K_{T0} \frac{P_E}{nD} \quad (1)$$

A variável K_{T0} é descrita por:

$$K_{T0} = 57000(1,97 - \frac{\rho}{D}) \quad (2)$$

Na Equação (1), $T_{v=0}$ representa a tração estática em *lb*, P_E a potência disponível no eixo do motor em *hp*, n a rotação no motor em rotações por minuto, D é o diâmetro da hélice em *ft*. Na Equação (2), ρ é o passo da hélice em *ft*, e K_{T0} representa o coeficiente de tração estática. O resultado do ensaio estático deve ser comparado com o modelo matemático.

De acordo com Rodrigues (2014), "esse sistema consiste de um suporte de fixação para o motor com o acoplamento de um dinamômetro simples [...]", além de um medidor de rotação do motor.

A eficiência da hélice é resultado a razão de avanço da aeronave, função que depende da velocidade de voo, rotação do motor e do diâmetro da hélice e, como as hélices de aeromodelos são de passo fixo, sua eficiência é baixa e grande parte da potência do motor é desperdiçada (RODRIGUES, 2014). A escolha da hélice mais eficiente é fundamental na escolha na competição do AeroDesign, pois assim é possível obter a maior tração disponível, que otimizam a decolagem e possibilitam o voo com maior carga útil possível (RODRIGUES, 2014). A razão de avanço pode ser determinada pela Equação (3):

$$J = \frac{v}{nD} \quad (3)$$

em que v é a velocidade de voo em m/s.

Para se realizar a variação da tração disponível com a velocidade de voo é necessário validar o conjunto com um teste dinâmico, ou seja, colocando o motor contra o vento e coletando seus dados de tração.

2.2 CONTROLE E COMUNICAÇÃO DO SISTEMA EMBARCADO

O hardware que compõe o sistema embarcado pode ser dividido em cinco grandes categorias (GANSSLE *et al.*, 2007):

- Unidade de processamento central (CPU): O processador *master* do sistema;
- Memória: local de armazenamentos de dados;
- Caminhos de dados (*Buses*): Interconecta os componentes, fornecendo um caminho para os dados percorrerem;
- Dispositivos de entrada: Componentes eletrônicos e processadores escravos de entrada;
- Dispositivos de saída: Componentes eletrônicos e processadores escravos de saída.

Estas cinco categorias estão baseadas na arquitetura descrita pelo modelo de von Neumann, sendo fruto do trabalho publicado por John von Neumann em 1945, que definiu os

requisitos de um computador de propósito geral (*general-purpose computer*) (GANSSLE *et al.*, 2007).

2.3 SENSORES E ATUADORES

Sensores e atuadores são dois componentes críticos de todo sistema de controle em *loop* fechado; tal sistema é chamado de sistema mecatrônico (ANJANAPPA *et al.*, 2002). Sensores são necessários para mensurar sinais desconhecidos e parâmetros de um sistema de engenharia e seu ambiente (SILVA, 2015). Essencialmente, de acordo com Silva (2015), sensores são necessários para monitorar e aprender sobre um sistema. Atuadores são basicamente o “músculo” por trás de um sistema mecatrônico que aceita um comando de controle (geralmente um sinal elétrico) e produz uma mudança no estado físico gerando força, movimento, calor, fluxo, etc. Na Figura 9 é mostrado o modelo de um sistema mecatrônico simples.

Figura 9 – Modelo de um sistema mecatrônico



Fonte: Anjanappa *et al.* (2002)

2.3.1 Sensores

O sensor é um dispositivo que, quando exposto à um fenômeno físico (temperatura, deslocamento, força, etc.) produz um sinal de saída proporcional (elétrico, mecânico ou magnético) (ANJANAPPA *et al.*, 2002). Os sensores podem ser classificados entre passivos e ativos, sendo os passivos os que não necessitam de energia para funcionar e os ativos necessitam (ANJANAPPA *et al.*, 2002).

Os sensores podem ser classificados também como analógicos ou digitais, dependendo

do tipo de sinal de saída (ANJANAPPA *et al.*, 2002). Sensores analógicos produzem sinais contínuos que são proporcionais aos parâmetros captados e, tipicamente, requerem conversões analógicas-digitais antes de serem inseridas em um microcontrolador (ANJANAPPA *et al.*, 2002). Por outro lado, os sensores digitais produzem saídas digitais que podem ser diretamente interfaceadas com um microcontrolador (ANJANAPPA *et al.*, 2002).

2.3.1.1 Medição de Tração do Motor

Quando um objeto é submetido à estresse mecânico, ele contrai e expande (ANJANAPPA *et al.*, 2002). Uma célula de carga é um dispositivo que possui outros componentes internos, com o mais comum sendo o extensômetro, uma fina camada de resistência elétrica (MÜLLER *et al.*, 2010). Para medir a tensão, extensômetros são anexados à superfície que sofrem alterações, com a intensidade de variação da resistência elétrica nos extensômetros sendo proporcional à intensidade da força aplicada (MÜLLER *et al.*, 2010). A resistência R de um condutor com comprimento l e área transversal A é dada por:

$$R = \rho l / A \quad (4)$$

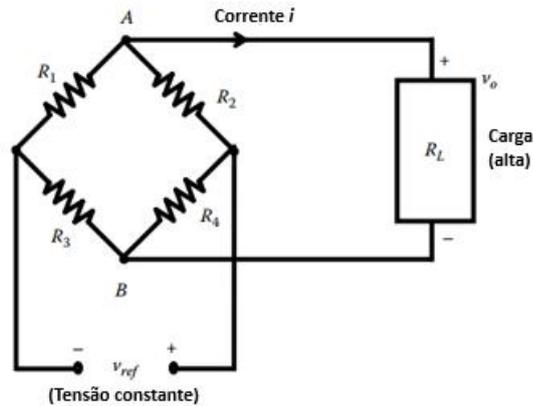
em que ρ é a resistividade do material.

As pequenas variações são um dos principais problemas de extensômetros (MÜLLER *et al.*, 2010). De acordo com Müller *et al.* (2010), para um extensômetro de 120Ω , o valor máximo de alteração seria $120,36 \Omega$. Outro problema são as indesejadas variações de resistência devido à temperatura (MÜLLER *et al.*, 2010). Com o objetivo de minimizar esses problemas, pontes de Wheatstone são usadas para medir a resistência elétrica e impedância com pequena oscilação (DALY *et al.*, 1984). O circuito é composto de 1, 2 ou 4 extensômetros, conjuntamente de um amplificador e filtros analógicos para condicionamento do sinal, como exemplificado na Figura 10. A relação de saída de uma ponte de Wheatstone é dada por:

$$v_0 = \frac{(R_1 R_4 - R_2 R_3)}{(R_1 + R_2)(R_3 + R_4)} v_{ref} \quad (5)$$

A amplificação do sinal da ponte de Wheatstone é necessária (MÜLLER *et al.*, 2010). Usualmente, uma ponte de Wheatstone de extensômetros tem 2 mV/V de sensibilidade. Com uma v_{ref} de 3 V , somente 6 mV serão lidos em escala total, requerendo, portanto, amplificação do sinal. A filtragem do sinal também é necessária em certas ocasiões, prevenindo o *aliasing* do sinal e também ruídos devido à fontes eletromagnéticas (MÜLLER *et al.*, 2010).

Figura 10 – Circuito Ponte de Wheatstone



Fonte: Silva (2015).

2.3.1.2 Medição de Temperatura

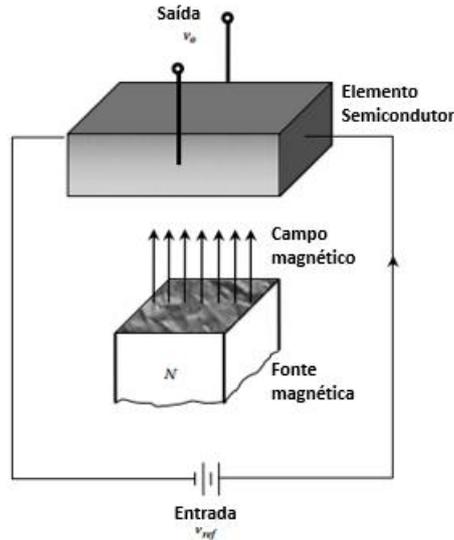
Na grande maioria dos dispositivos de medição de temperatura, a temperatura é sentida através da transferência de calor de uma fonte ao dispositivo a ser medido (SILVA, 2015). De acordo com Anjanappa *et al.* (2002), existe uma grande variedade de dispositivos capazes de medir a temperatura, sendo mais comuns o termoacoplamento, termistor, detector de temperatura via resistência (*Resistance Temperature Detecton - RTD*) e modelos infravermelhos. Existem também os sensores de temperatura de circuito integrado (*Integrated Circuit - IC*), sendo o único que possui uma função de transferência linear (KESTER *et al.*, 1999).

No passado, circuitos complexos de condicionamento analógico eram projetados para corrigir a não-linearidade, requerendo calibração manual e resistores precisos para atingir a acurácia (KESTER *et al.*, 1999). A partir das últimas décadas, sensores, sensores semicondutores integrados oferecem alta acurácia e alta linearidade com um alcance de temperatura de cerca de -55°C à $+150^{\circ}\text{C}$ possuindo amplificadores internos que podem escalar a saída para valores convenientes, como $10\text{ mV}/^{\circ}\text{C}$ (KESTER *et al.*, 1999).

2.3.1.3 Medição de Corrente do Sistema

Considere um elemento semiconductor sujeito à uma tensão contínua de referência v_{ref} . Se um campo magnético é aplicado de forma perpendicular à direção dessa tensão, uma diferença de potencial v_0 será gerada na direção ortogonal à do elemento semiconductor, apresentado na Figura 11 (SILVA, 2015). No caso, o campo magnético aplicado seria fornecido a partir de uma corrente e a diferença de potencial gerada é chamada de tensão Hall (PAUN *et al.*, 2013).

Figura 11 – Representação esquemática do sensor de efeito Hall



Fonte: Silva (2015).

Paun *et al.* (2013) destacam que os sensores de efeito Hall são amplamente usados em aplicações industriais de baixa potência. Esses sensores, integrados em tecnologia CMOS, provam ser de custo benéfico e oferecem um alto desempenho (RAMSDEN, 2006 apud PAUN *et al.*, 2013).

2.3.1.4 Medição da Pressão

De acordo com Silva (2015), os métodos mais comuns para se medir pressão são:

1. Balancear a pressão com uma força oposta e medir essa força (utilizando manômetros, líquidos e pistões);
2. Submeter a pressão à um membro flexível e medir a deflexão resultante;
3. Submeter a pressão à um membro flexível e medir a tensão mecânica resultante (por exemplo: diafragmas e capsulas).

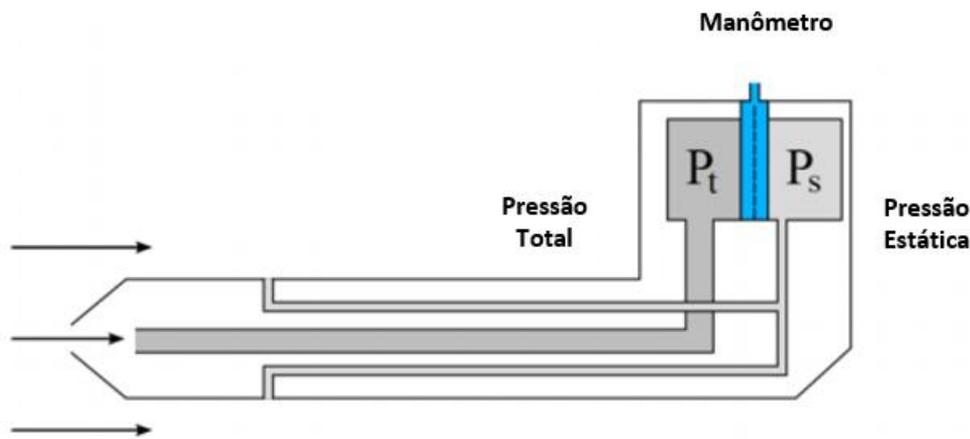
Em um sensor de pressão de diafragma, a membrana, geralmente metálica, é tensionada devido à pressão aplicada (SILVA, 2015). A pressão pode ser mensurada a partir de extensômetros montados no diafragma, sendo o princípio de diversos sensores modernos (SILVA, 2015). Mais de um extensômetro pode ser acoplado no diafragma e usando em circuito de ponte de Wheatstone (princípio explicado na seção 2.3.1.1) para providenciar a leitura de pressão, através da calibração correta (SILVA, 2015).

2.3.1.5 Medição do Escoamento de Ar

Existem diversas maneiras de se medir o fluxo de um sistema (ANJANAPPA *et al.*, 2002). Um rotâmetro e um medidor de turbina, quando colocados em um caminho de fluxo, rotacionam à uma velocidade proporcional à taxa de fluxo (ANJANAPPA *et al.*, 2002). O medidor de fluxo eletromagnético utiliza o método não-contato, em que um fluxo magnético é aplicado na direção transversal ao fluxo e o fluido age como um condutor, induzindo tensão proporcional ao fluxo do fluido (ANJANAPPA *et al.*, 2002).

O Tubo de Pitot é uma maneira de medir o fluxo, sendo um instrumento simples e barato (BROWN, 2001). Para Brown (2001), a força do sensor é a sua simplicidade, design robusto que permite uma determinação de velocidade precisa medindo a diferença de pressão entre duas entradas. O instrumento retorna a medição de pressão causada pelas partículas de ar que penetram a ponta do sensor (pressão dinâmica) e a pressão estacionária (laterais do sensor) (EZZEDDINE *et al.*, 2016). A velocidade é então retirada da diferença de pressão entre a pressão estática e dinâmica (EZZEDDINE *et al.*, 2016). Na Figura 12 é possível ver o design de um Tubo de Pitot.

Figura 12 – Representação esquemática do Tubo de Pitot



Fonte: Ezzeddine *et al.* (2016).

O tubo interno permite a leitura da pressão total do sistema e o tubo externo fornece a pressão estática (EZZEDDINE *et al.*, 2016). A velocidade é mensurada graças à fórmula de Bernoulli:

$$P_t = P_s + \frac{1}{2} \gamma \left(\frac{\Psi}{k} \right)^2 \quad (6)$$

em que:

- P_t é a pressão total (Pa);

- P_s é a pressão estática (Pa);
- γ é a densidade do fluxo (kg/m³);
- Ψ é a velocidade média do fluxo (m/s);
- k é o coeficiente do fluxo (adimensional).

Observação: Alguns símbolos foram trocados em relação à referência para evitar conflito com outras seções do trabalho.

Dada a pressão total, a pressão estática e os valores de densidade, a velocidade média é:

$$\Psi = k \sqrt{\frac{2(P_t - P_s)}{\gamma}} \quad (7)$$

2.3.1.6 Medição de Rotação

Sensores de posição linear e rotacional são dois dos mais fundamentais sistemas de medição usados em sistemas mecatrônicos (ANJANAPPA *et al.*, 2002). Os tipos de sensores são do tipo de contato como extensômetros, LVDT, RVDT, tacômetro, entre outros (ANJANAPPA *et al.*, 2002). Dos tipos de não contato, apresentam-se os encoders, sensores de efeito Hall, capacitivos, indutivos, e interferômetro (ANJANAPPA *et al.*, 2002).

2.3.2 Atuadores

Atuadores podem ser classificados como binários ou contínuos, dependendo do número de estados estáveis de saída (ANJANAPPA *et al.*, 2002). Um motor de passo é um bom exemplo de atuador contínuo, enquanto um relé eletromagnético de duas saídas é considerado binário. Motores convertem energia elétrica em energia mecânica utilizando os princípios de eletromagnetismo (ZHAO; YU, 2014). O método de conversão de energia é praticamente igual em todos os motores elétricos (ZHAO; YU, 2014).

2.3.2.1 Motores *Brushless* CC

Motores do tipo *Brushless* CC têm ganhado muita popularidade (YEDAMALE, 2019). Como o próprio nome propõe (*Brushless* - Sem escovas), esses motores não utilizam escovas para comutação e sim o princípio de comutação eletrônica (YEDAMALE, 2019). Motores do

tipo *Brushless* CC possuem diversas vantagens em relação ao motor *Brushed* e motores de indução (YEDAMALE, 2019). De acordo com Yedamale (2019), essas vantagens são descritas a seguir e mostradas na Tabela 1 e Tabela 2:

- Melhor característica velocidade versus torque;
- Alta resposta dinâmica;
- Alta eficiência;
- Longa vida útil;
- Operação silenciosa;
- Maiores alcances de velocidade.

Tabela 1 – Motor BLDC versus Motor *Brushed* CC

Característica	Motor BLDC	Motor Brushed
Comutação	Comutação eletrônica baseada na posição de sensores de efeito Hall.	Comutação via escovas.
Manutenção	Pouco requerida por não necessitar de escovas.	Manutenção periódica.
Vida útil	Longa.	Curta.
Característica Torque/Velocidade	Plana. Possibilita operação em todas as velocidades em carga nominal.	Moderadamente plana. Em altas velocidades, o atrito causado pelas escovas aumenta, reduzindo o torque útil.
Eficiência	Alta.	Moderada.
Potência de Saída/Tamanho	Alta. Possui dimensões menores devido às características térmicas superiores. O BLDC possui os enrolamentos no estator, que está conectado à carcaça, dissipando melhor o calor.	Moderada/Baixa. O calor produzido pela armadura é dissipado pela abertura de ar, conseqüentemente aumentando a temperatura nessa região e limitando as especificações de trabalho do motor.
Inércia do Rotor	Baixa por ser composto de ímãs permanentes.	Alta inércia que limita as características dinâmicas do motor.
Alcance de Velocidade	Alto. Não há limitações mecânicas, como escovas.	Baixo. Limitações mecânicas por causa das escovas.
Ruído Sonoro.	Baixo.	Arcos elétricos causados pelas escovas gera ruído causando interferência eletromagnética em dispositivos próximos.
Custo de Construção	Alto. Por conter ímãs permanentes, os custos construtivos são altos.	Baixo.
Controle	Complexo e caro.	Simple e Barato.
Requisitos de Controle	Um controlador é sempre necessário para manter o motor em funcionamento.	Não necessita de controle para velocidades fixas. Para variar a velocidade, um controlador é necessário.

Fonte: Traduzido de Yedamale (2019).

Tabela 2 – Motor BLDC versus Motor de Indução CA

Característica	Motor BLDC	Motor de Indução CA
Característica Torque/Velocidade	Plana. Possibilita operação em todas as velocidades em carga nominal.	Não linear. Baixo torque em baixas velocidades.
Potência de Saída/Tamanho	Alta. Possui dimensões menores devido às características térmicas superiores. O BLDC possui os enrolamentos no estator, que está conectado à carcaça, dissipando melhor o calor.	Moderada. Uma vez que o estator e rotor possuem enrolamentos, a potência por tamanho é menor em comparação com o BLDC.
Inércia do Rotor	Baixa por ser composto de ímãs permanentes.	Alta. Características dinâmicas ruins.
Corrente de Partida	A nominal. Não é necessário nenhum circuito limitador de corrente de partida.	Aproximadamente sete vezes superior à corrente nominal. O dispositivo de partida deve ser escolhido cuidadosamente. Normalmente utiliza a partida Estrela-Triângulo.
Escorregamento	Nenhum escorregamento é encontrado entre as frequências do rotor e estator.	O rotor trabalha em frequências inferiores ao rotor pela frequência de escorregamento, com esse fator aumentando com a carga no motor.
Requisitos de Controle	Um controlador é sempre necessário para manter o motor em funcionamento.	Não necessita de controle para velocidades fixas. Para variar a velocidade, um controlador é necessário.

Fonte: Traduzido de Yedamale (2019).

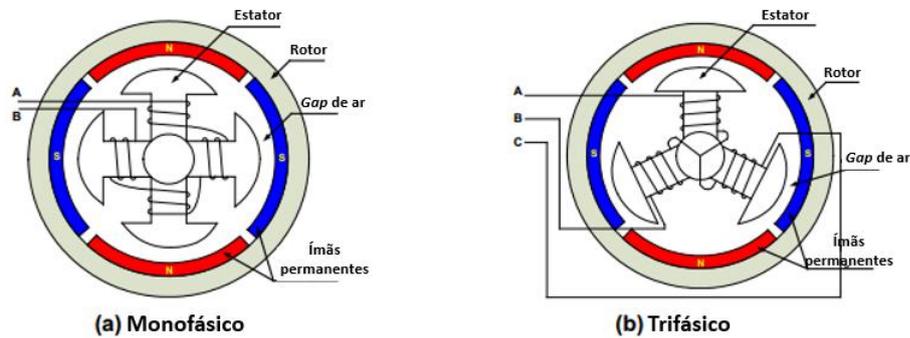
Além disso, a razão de torque por massa do motor é maior, sendo muito útil em aplicações onde peso e espaço são fatores críticos (YEDAMALE, 2019).

O motor do tipo *Brushless* CC é composto basicamente de duas partes: estator e rotor. O estator de um motor *Brushless* CC consiste em uma pilha de aços laminados com enrolamentos posicionados nos espaços que são cortados axialmente ao longo da periferia interna (YEDAMALE, 2019). Cada um dos enrolamentos é formado de várias bobinas (YEDAMALE, 2019). Existem três classificações para um motor *Brushless* DC Motor (BLDC): monofásico, bifásico e trifásico, sendo os monofásico e trifásico os mais utilizados (ZHAO; YU, 2014). A representação esquemática do motor *Brushless* CC pode ser vista na Figura 13.

Existem também dois tipos de enrolamentos no estator: trapezoidal e senoidal, que se refere ao formato do sinal da força contra eletromotriz (BEMF) (ZHAO; YU, 2014). Quando o motor BLCD rotaciona, cada enrolamento gera a tensão BEMF, que se opõem a tensão fornecida aos enrolamentos, de acordo com a Lei de Lenz. Essa tensão depende de três fatores: velocidade angular do rotor, campo magnético gerado pelos ímãs do rotor, número de voltas nos enrolamentos do estator (YEDAMALE, 2019). Na Figura 14 é mostrado o sinal trapezoidal e na Figura 15 é mostrado o sinal senoidal.

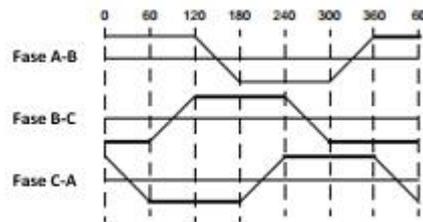
O rotor consiste de um eixo e um conjunto de ímãs permanentes que podem variar de

Figura 13 – Representação esquemática do motor *Brushless CC*



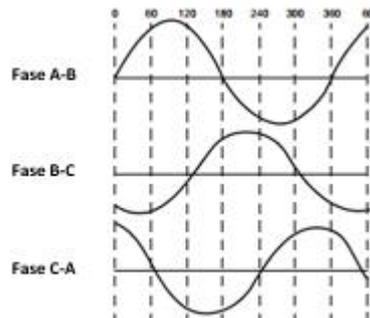
Fonte: Zhao e Yu (2014).

Figura 14 – BEMF trapezoidal



Fonte: Yedamale (2019).

Figura 15 – BEMF senoidal

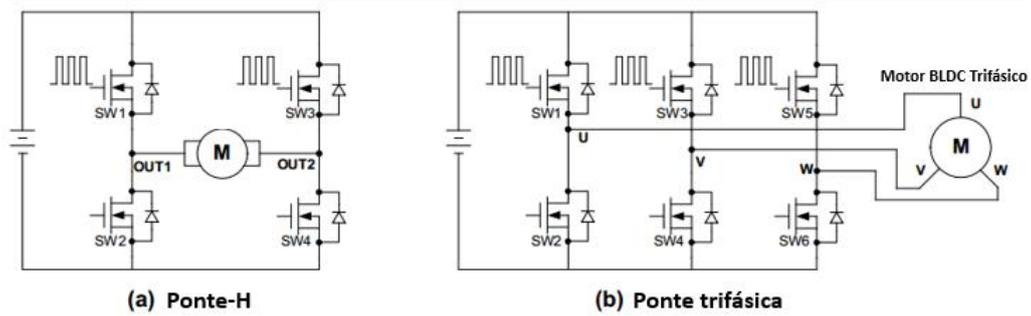


Fonte: Yedamale (2019).

2 a 8 pares de polos que alternam entre Norte e Sul (ZHAO; YU, 2014). Ímãs de ferrite são geralmente utilizados para a composição do rotor (YEDAMALE, 2019).

O sistema funciona a partir da comutação de tensão CC feita eletronicamente usando o *feedback* da posição do rotor através de sensores de efeito Hall ou um *encoder* rotativo (ZHAO; YU, 2014). Chaves são responsáveis por realizar a comutação (ZHAO; YU, 2014). Essas chaves elétricas são geralmente conectadas em uma estrutura de ponte-H em motores monofásicos e em estruturas de ponte trifásicas em motores trifásicos, como pode ser visto na Figura 16 (ZHAO; YU, 2014). Essas chaves podem ser MOSFETs, IGBTs ou transistores bipolares, dependendo dos valores de tensão e corrente do motor Bhide *et al.* (2016).

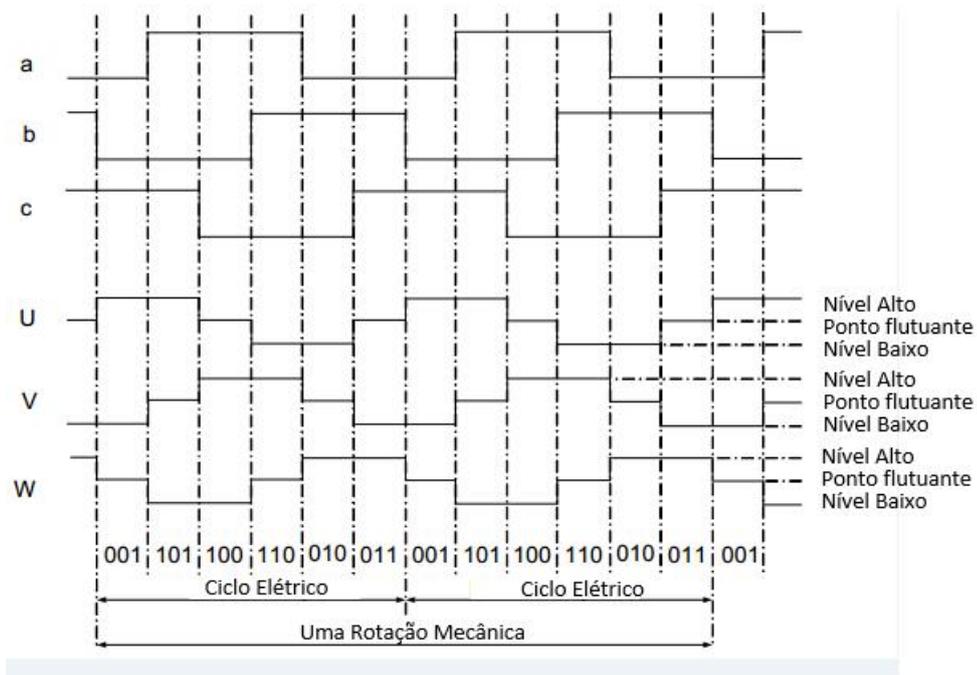
Figura 16 – Circuito controlador de motores *Brushless* CC



Fonte: Zhao e Yu (2014).

Microcontroladores são utilizados para gerar sinal de modulação por largura de pulso (PWM - *Pulse Width Modulation*) para controlar o tempo de chaveamento em uma sequência baseada no sinal de *feedback* dos sensores de efeito Hall (BHIDE *et al.*, 2016). Um motor BLDC trifásico requer três sensores de efeito Hall para detectar a posição do rotor e baseado nas posições destes sensores, existem as saídas com deslocamento de 60° e de 120° (ZHAO; YU, 2014). Os três sensores de efeito Hall — a,b,c da Figura 17 — são posicionados a 120° de diferença dentro do estator, enquanto os enrolamentos estão em formação de estrela (ZHAO; YU, 2014). Como é mostrado nas formas de onda da Figura 17, a cada 60° graus elétricos, um dos sensores muda o seu estado (BHIDE *et al.*, 2016).

Figura 17 – Motor BLDC trifásico vs tempo de chaveamento



Fonte: Zhao e Yu (2014).

Em um ciclo elétrico completo, a tensão e corrente de alimentação comutam 6 vezes, cada um após 60° graus elétricos (BHIDE *et al.*, 2016). O número de repetições de ciclos elétricos para completar um ciclo mecânico depende diretamente do número de pares de polos do rotor (BHIDE *et al.*, 2016). Para cada par de polo, um ciclo elétrico é completo, portanto, o número de ciclos elétricos por rotação mecânica é igual ao número de par de polos do rotor (BHIDE *et al.*, 2016).

Os enrolamentos - U, V, W - são energizados ou deixados em flutuação de tensão baseado nos sinais dos sensores a, b e c (BHIDE *et al.*, 2016). Para variar a velocidade de rotação, o sinal PWM é utilizado.

Como descrito anteriormente, um microcontrolador pode ser o responsável pela geração desse sinal, que é então aplicado aos terminais de *gate* das chaves comutadoras. De acordo com Bhide *et al.* (2016), o método de PWM é muito vantajoso em aplicações que envolvem o controle de motores. Quando a razão de trabalho, também conhecida como *duty cycle*, é alterada, a tensão média aplicada no estator se altera, mudando a velocidade de rotação do motor.

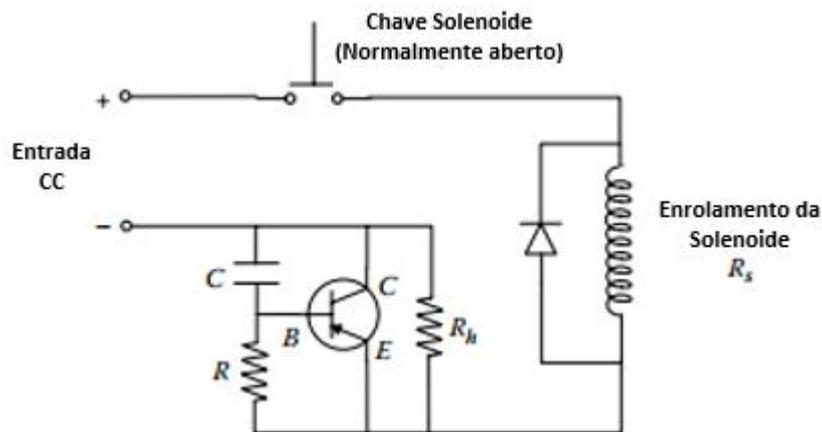
2.3.2.2 Relé Eletromagnético

Quando uma aplicação requer uma chave com baixa resistência de contato, o relé eletromagnético é a melhor opção (GANSSLE *et al.*, 2007). A implicação prática de uma resistência de contato é o calor gerado, logo, a potência dissipada (GANSSLE *et al.*, 2007). Quando passa corrente pelo contato, a resistência causa aquecimento *Joule* proporcional a $I \times R$ (GANSSLE *et al.*, 2007). Um ímã auxiliar efetivamente polariza o relé e quando a bobina é energizada com a polaridade adequada, esse elemento é atraído para perto da bobina (GANSSLE *et al.*, 2007). O sistema deve ser capaz de aguentar a tensão de *fly-back*, resultado do descarregamento da bobina (GANSSLE *et al.*, 2007). A melhor maneira de evitar esse problema é colocando um diodo entre a alimentação do relé, fixando a tensão à níveis tolerados por transistores (GANSSLE *et al.*, 2007), como é apresentado na Figura 18.

2.3.2.3 Bateria

Baterias armazenam energia a partir de reações químicas, sendo chamadas de baterias eletroquímicas (MIT, 2020). A reação química envolve a remoção de elétrons de um material e adicionando elétrons em outro, sendo esse processo chamado de oxirredução. A energia é

Figura 18 – Diodo *Fly-Back*



Fonte: Silva (2015).

extraída à medida que os elétrons passam por uma carga (um motor, por exemplo), de um lado para o outro. A reação química somente acontece se os íons podem fluir livremente entre um cátodo e ânodo. Para que isso aconteça, ambos os eletrodos devem estar mergulhados em um fluido chamado de eletrólito. Além disso, há uma separação entre o cátodo e ânodo, chamada de membrana, que possibilita a passagem somente de íons (MIT, 2020).

De acordo com Aerodesign (2020b), para o sistema de controle de voo, como o sistema embarcado, a equipe deve utilizar apenas baterias dentro os tipos permitidos:

- Níquel Cádmio (NiCd);
- Níquel Metal Hidreto (NiMh);
- Lítio Ferro Polímero (LiFePo4).

As baterias do tipo LiPo não devem ser utilizadas no sistema de controle da aeronave. Os sistemas propulsivos podem utilizar baterias dentro os tipos permitidos conforme a seguir:

- Níquel Cádmio (NiCd);
- Níquel Metal Hidreto (NiMh);
- Lítio Ferro Polímero (LiFePo4);
- Lítio Polímero (LiPo).

A Tabela 3 mostra a comparação entre todas as baterias que são permitidas pela Competição da SAE AeroDesign Brasil, segundo o regulamento atual.

Tabela 3 – Comparação entre baterias permitidas pela competição SAE AeroDesign Brasil

Características	LiFePo4	NiCd	LiPo	Ni-MH
Tensão (V/Célula)	2,5-3,65V	1,20	3,6	1,20
Ciclos de Vida	2000	1000	300-500	300-500
Energia Específica (kW/kg)	90-120	45-80	100-130	60-120
Corrente de Descarga	1C, 25C em algumas células	Pode estar acima de 1C	2C	1C
Comentários	Muito segura. Curva de descarga linear.	Robusta. Requer manutenção constante.	Pouco segura. Requer sistema de proteção.	Mais delicada que a NiCd. Manutenção baixa e maior capacidade energética.

Fonte: Autoria própria (2021).

A Tabela 3 utiliza as seguintes referências: (GOMES, 2018), (UNIVERSITY, 2020a), (UNIVERSITY, 2020c) e (UNIVERSITY, 2020b).

Ao se utilizar baterias, é sempre importante saber manuseá-las. Problemas em baterias são geralmente causados por erro humano (MIT, 2020). A seguir estão alguns erros comuns, que podem causar graves problemas de segurança:

- Plugar a bateria no carregador com a polaridade invertida;
- Utilizar o carregador errado, ocasionando problemas de sobrecarga;
- Não utilizar um sistema de gerenciamento de bateria, então ocasionando a sobtensão;
- Descarregar a bateria muito rapidamente;
- Curto circuito externo (por exemplo, derrubar uma ferramenta entre os dois polos da bateria).

Com uma potência alta concentrada em um volume pequeno, as baterias com base em Lítio apresentam risco de explosão caso sejam manuseadas de maneira incorreta (GOMES, 2018). De acordo com Gomes (2018), um exemplo é a perfuração da bateria. A perfuração, se atingir a membrana de separação entre o ânodo e cátodo, oferece um caminho alternativo para o fluxo de elétrons e essa perda de energia se converte em calor em uma reação térmica incontrolável, ocasionando em uma explosão (MIT, 2020).

2.4 PROGRAMAÇÃO BASEADA EM FLUXO

A programação baseada em fluxo (PBF) é um paradigma de programação criado no final dos anos 60 por J. Paul Rodker Morrison, que se utiliza do princípio de redes de processos de com módulos em "caixas pretas", enviando informações em pacotes sendo transmitidos em conexões pré-definidas, onde essas conexões estão externas ao processo (MORRISON, 2020).

Os processos de caixa preta podem ser reconectados diversas vezes para diferentes aplicações sem ter seu conteúdo interno modificado Morrison (2020), como pode ser visto na Figura 19 (AL-FEDAGHI, 2015 apud MORRISON, 2020).

Figura 19 – Componente na Programação baseada em Fluxo



Fonte: Al-Fedaghi (2015).

Os dados no PBF são chamados de “pacotes de informação” (ou PI) e uma aplicação é construída de um conjunto de programas passando PI entre eles. A entrada e saída no componente são chamadas de portas, pois recebem e enviam pacotes de informações (AL-FEDAGHI, 2015 apud MORRISON, 2020).

2.4.1 Programação baseada em fluxo com Node-RED

O Node-RED é uma ferramenta de programação com editor baseada em *browser* que torna fácil a criação de módulos de programação, realizando a interação com hardware, APIs e serviços *online* (NODE-RED, 2020a). O Node-RED é construído em *Node.js*, utilizando das vantagens de ser orientado a eventos para ser utilizado em conjunto com hardware de baixo custo via rede, como o Raspberry Pi (NODE-RED, 2020a).

2.4.1.1 Nós

Os nós são a representação gráfica de um componente na PBF. Esses elementos são acionados recebendo uma entrada de um nó anterior no fluxo, ou esperando um evento externo, como uma solicitação HTTP, por exemplo (NODE-RED, 2020b). As mensagens são então

processadas e podem enviar essa mensagem adiante no fluxo. A Figura 20 mostra a representação gráfica de um nó.

Figura 20 – Representação gráfica de um nó



Fonte: Autoria própria (2021).

2.4.1.2 Fluxo

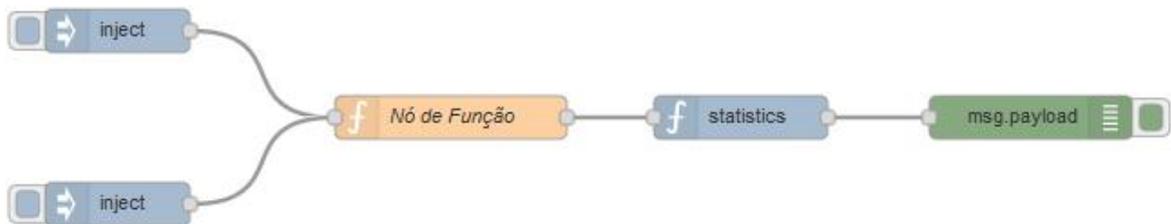
Dentro da ferramenta, as abas são chamadas de fluxo. O termo fluxo é também informalmente utilizado para representar um conjunto de nós conectados. Dessa maneira, é possível ter vários fluxos inseridos dentro de somente uma aba de fluxo, como pode ser visto nas Figuras 21 e 22.

Figura 21 – Representação do fluxo no Editor



Fonte: Autoria própria (2021).

Figura 22 – Representação gráfica de um fluxo composto de nós



Fonte: Autoria própria (2021).

2.4.1.3 Contexto

Contexto é uma maneira de armazenar informações que pode ser compartilhado entre nós mesmo não existindo uma conexão entre eles. Existem três tipos de contexto:

- Nó: somente visível no nó que define o valor;
- Fluxo: visível dentro de todos os nós do mesmo fluxo;
- Global: visível dentro de todos os nós.

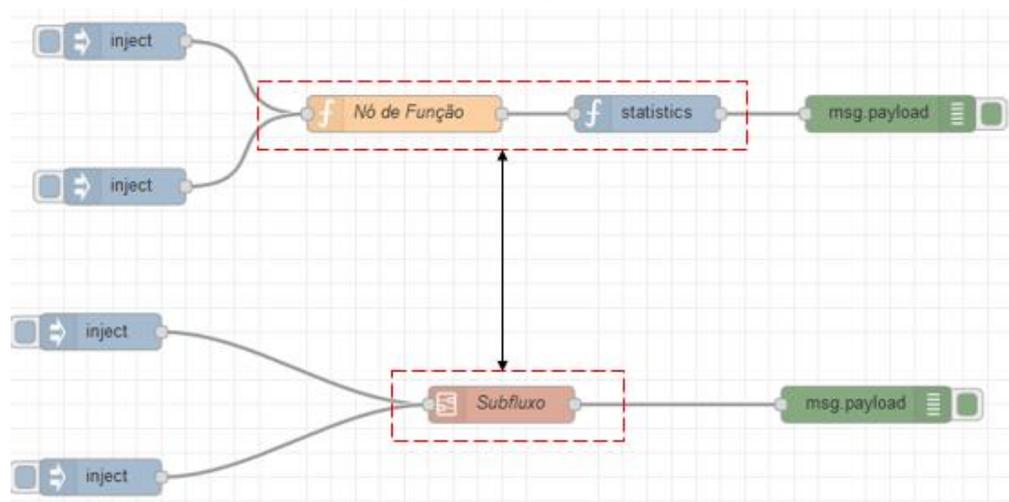
2.4.1.4 Mensagem

Mensagens são as informações que passam de um nó para o outro em um fluxo. São objetos *JavaScript* que podem ter um conjunto de propriedades, sendo referenciadas como *msg* dentro do editor. Por definição, elas possuem uma propriedade chamada de *payload* que carregam a informação mais importante.

2.4.1.5 Subfluxo

Os subfluxos são um conjunto de nós que está englobado em um único nó. Eles são usados para facilitar a visualização gráfica de um fluxo. A Figura 23 mostra a representação gráfica de um subfluxo.

Figura 23 – Representação gráfica de um Subfluxo



Fonte: Autoria própria (2021).

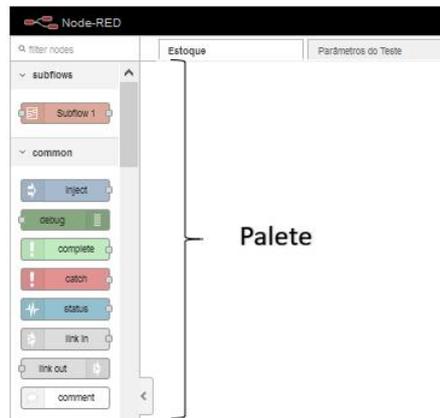
2.4.1.6 Fiação ou Conexões

Os fios são responsáveis por conectar os nós entre si, representando o caminho para as mensagens prosseguirem.

2.4.1.7 Palete

O palete está à esquerda do editor e tem como objetivo mostrar todos os nós disponíveis para a construção dos fluxos. Nós adicionais podem ser inseridos utilizando a Gerenciados de Paletes ou por meio da linha de comando, como mostra a Figura 24.

Figura 24 – Palette



Fonte: Autoria própria (2021).

2.4.1.8 Área de Trabalho

A área de trabalho é a região onde os fluxos são construídos “arrastando” componentes do paleta e conectando-os por meio dos fios. A área de trabalho possui várias abas na parte superior, um para cada fluxo específico, como pode ser visto na Figura 25.

Figura 25 – Visão geral do editor



Fonte: Autoria própria (2021).

3 LEVANTAMENTO DE ESPECIFICAÇÃO DE COMPONENTES DO SISTEMA DE CONTROLE E POTÊNCIA

No Apêndice A é mostrado o resultado da pesquisa de campo. Essa pesquisa serviu como ponto de partida para o levantamento da especificação do sistema embarcado proposto.

3.1 LEVANTAMENTO DE HARDWARE

Uma das características de sistemas embarcados é que tanto hardware e software devem ser levados em consideração e a reutilização de ambos é a base para a metodologia baseada em plataforma, proposto por (MARWEDEL, 2010). O objetivo desta seção é apresentar o levantamento dos Sistemas de Controle e Potência. O Sistema de Controle diz respeito aos dispositivos que estão conectados com a unidade de controle do sistema embarcado. O Sistema de Potência se refere aos dispositivos envolvidos diretamente no circuito do motor.

3.1.1 Sistema de Controle

3.1.1.1 Unidade Controladora Principal

Para a unidade controladora da bancada foi visada uma placa controladora acessível, com a capacidade operacional para multitarefas e facilidade de utilização. Além disso, é de suma importância o dispositivo ter conectividade IoT. O Raspberry Pi 3B+ se mostrou o mais adequado para o projeto, possuindo um processador 64-bit de quatro núcleos, banda de Wi-Fi dupla de 2,4 GHz e 5 GHz, com conexão Ethernet (FOUNDATION, 2020). Conforme Figura 26, o dispositivo conta com 40 pinos GPIO, um suporte para cartão SD e a possibilidade de rodar o sistema operacional Linux (FOUNDATION, 2020).

A unidade controladora é responsável pelo tratamento dos valores dos sensores e interface *Front-End*, além do envio do sinal PWM para o Controlador de Velocidade Eletrônica (ESC) do motor.

Figura 26 – Raspberry Pi 3B+

Fonte: Foundation (2020).

3.1.1.2 Unidade Controladora Auxiliar

A unidade controladora auxiliar tem o objetivo de coletar os dados vindos dos sensores periféricos e enviar à placa controladora principal. Para esta função foi selecionado o Arduino Uno (Figura 27), placa controladora baseada no processador ATmega328P, com 14 saídas/entradas digitais e 6 saídas/entradas analógicas (ARDUINO, 2021). A possibilidade de leitura de sensores analógicos sem a necessidade de um conversor Analógico/Digital foi uma das razões da escolha, além da facilidade de uso e grande quantidade de bibliotecas disponíveis para leitura dos sensores.

Figura 27 – Arduino Uno

Fonte: Arduino (2021).

3.1.1.3 Sensores e Periféricos

A bancada conta com uma célula de carga de 5 kg (Figura 28), responsável pela medição da tração do motor *Brushless* CC. O sinal de tensão por deslocamento da célula de carga é de baixa amplitude, necessitando de um amplificador operacional para possibilitar o sinal de ser lido pela placa controladora. O HX711 é um conversor analógico-digital de precisão de 24-bits, desenvolvido para células de carga para fazer interface direta com as pontes de Wheatstone do sensor (WINGS, 2021). A tensão de alimentação do sensor é entre 2,7-5,5 V, possuindo uma integração ideal com o Arduino Uno. O HX711 possui dois canais de entrada para sinais, com o Canal A podendo ter um ganho de 64 ou 128, e o Canal B tem um ganho fixo de 32. Os maiores ganhos são necessários para garantir que os pequenos valores de saída do sensor sejam lidos (WINGS, 2021). A comunicação entre o periférico e placa controladora é via protocolo I²C.

Figura 28 – Célula de Carga e HX711



Fonte: Amazon (2020).

Para a medição da velocidade do ar é utilizado um Tubo de Pitot. O tubo possui duas saídas de ar que são direcionadas para o sensor MPXV7002DP, que é um sensor de diferença de pressão. Uma das saídas do Tubo de Pitot estabelece a pressão dinâmica e a outra a pressão estática, explicado em mais detalhes na seção 2.3.1.5. O sensor MPXV7002DP é piezoresistivo capaz de gerar uma saída de tensão proporcional a diferença de pressão estática e dinâmica. A alimentação do sensor é de 4,75 VCC até 5,25 VCC, possuindo uma corrente consumida de 1 mA CC (NPX SEMICONDUCTOR, 2017). A saída é analógica e precisa ser convertida para digital antes da comunicação com a placa controladora. Na Figura 29 é possível ver o conjunto escolhido.

Para a medição de rotação da hélice do motor foi escolhido o Módulo Sensor de Efeito

Figura 29 – Tubo de Pitot e módulo MPX7002DP



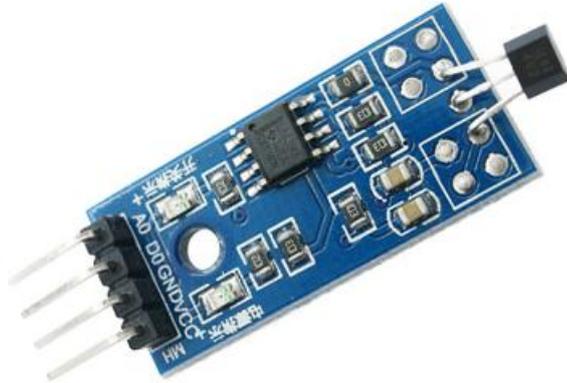
Fonte: HobbyKing (2020).

Hall 3144E. Esse sensor é posicionado um ímã permanente próximo à hélice, proporcionando uma variação do campo magnético quando o motor estiver em movimento, variação essa que é sentida pelo módulo. Este módulo conta com o CI monolítico 3144E, chave sensitiva de efeito Hall, especificado para operação no meio automotiva e industrial com alcance de temperatura de -40°C à $+85^{\circ}\text{C}$ (ALLEGRO MICROSYSTEMS, 2002).

A saída desse dispositivo chaveia para nível baixo quando o campo magnético no sensor Hall excede o limite do ponto de operação B_{PO} de 350 Gauss, com a saída do dispositivo nesse ponto sendo tipicamente 175 mV (ALLEGRO MICROSYSTEMS, 2002). Quando o campo magnético é reduzido para baixo do limite do ponto de liberação B_{RP} , o dispositivo chaveia para nível alto (ALLEGRO MICROSYSTEMS, 2002). A diferença entre esses dois pontos é chamada de histerese, que permite uma saída limpa mesmo com a presença de vibração mecânica e ruído elétrico (ALLEGRO MICROSYSTEMS, 2002). O Módulo Sensor de Efeito Hall 3144E conta com o CI LM293 com comparador, que fornece uma saída digital e também analógica. O módulo é apresentado na Figura 30.

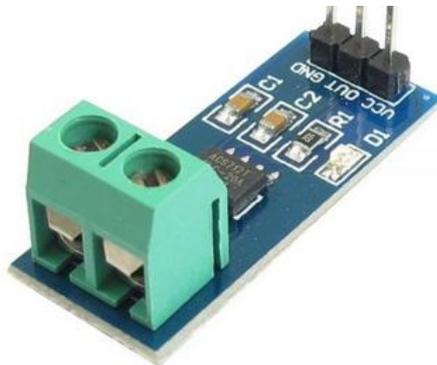
Para a medição de corrente na entrada do Raspberry Pi 3B+ foi utilizado o Sensor de Corrente ACS712 5A, apresentado na Figura 31. Este módulo conta com o CI ACS712, consistindo de um sensor de Efeito Hall preciso, baixo-*offset* e linear com um caminho de condução de cobre que, quando passado por uma corrente elétrica, gera um campo magnético que é sentido pelo sensor Hall integrado, convertendo em uma tensão proporcional (ALLEGRO MICROSYSTEMS, 2007). A tensão de alimentação é de 4,5 VCC à 5,5 VCC, com consumo de no máximo 13 mA (ALLEGRO MICROSYSTEMS, 2007). A saída fornecida pelo Módulo é analógica.

Figura 30 – Módulo Sensor de Efeito Hall 3144E



Fonte: TiggerCOMP (2020).

Figura 31 – Sensor de Corrente ACS712 5A



Fonte: Robotica (2020).

São utilizados dois Módulos de Sensor de tensão 0-25V CC. Um deles é utilizado para medir a tensão na bateria do Sistema de Potência e o outro a bateria do Sistema de Controle. O Módulo de tensão pode ser visto na Figura 32. O sensor utiliza o princípio de divisor de tensão para gerar uma saída de tensão analógica 5 vezes menor que a tensão de entrada. O sensor é considerado passivo, pois não necessita de uma tensão de alimentação.

Figura 32 – Sensor de Tensão 0-25V CC



Fonte: Arduoeleetro (2020).

Para medição de temperatura na bateria do sistema de potência é utilizado o CI LM35,

Figura 35 – Motor Turnigy D3548 / 4 1100KV *Brushless* Outrunner



Fonte: HobbyKing (2021b).

3.1.2.2 Controlador de Velocidade (ESC)

O ESC escolhido foi o Hobbyking 80A ESC 4A SBEC, apresentado na Figura 36. Este controlador suporta uma corrente de 80 A e uma corrente máxima de 100 A por até 10 segundos (HOBBYKING, 2021a). Pode ser alimentado com baterias de 2 à 6 células e possui um peso estimado de 90 gramas (HOBBYKING, 2021a). A escolha deste controlador se deve à sua maleabilidade de sua corrente de operação, podendo ser utilizado com uma grande variedade de motores.

Figura 36 – Hobbyking 80A ESC 4A SBEC



Fonte: HobbyKing (2021a).

3.1.2.3 Bateria

A bateria escolhida para alimentação do motor foi a Powerbolt 14,8 V 4000 mAh 60C 4S Lipo. A escolha dessa bateria se deve à sua compactação e peso, sendo uma ótima opção a ser

usada futuramente na equipe e também pela alta taxa de descarga de 60C, podendo descarregar seguramente até 240 A por um tempo determinado.

3.1.2.4 Relé Eletromagnético

Para controle do motor, é utilizado um relé eletromagnético DNI8122 70A 12V. Este relé suporta correntes de até 70 A, sendo compatível com o ESC escolhido.

3.1.2.5 Sensor de Corrente do Motor

O sensor de corrente escolhido para o motor é o Allegro ACS758 de 100 A, apresentado na Figura 37.

Figura 37 – Sensor Allegro ACS758



Fonte: Allegro Microsystems (2020).

Como a bancada poderá ser aplicado a diferentes motores e baterias, optou-se por um sensor com um range de leitura alto de corrente para abranger todos os tipos de configuração de grupo motopropulsor. O sensor Allegro ACS758, possui alimentação de entrada de 3,0 a 5,5 V (ALLEGRO MICROSYSTEMS, 2020).

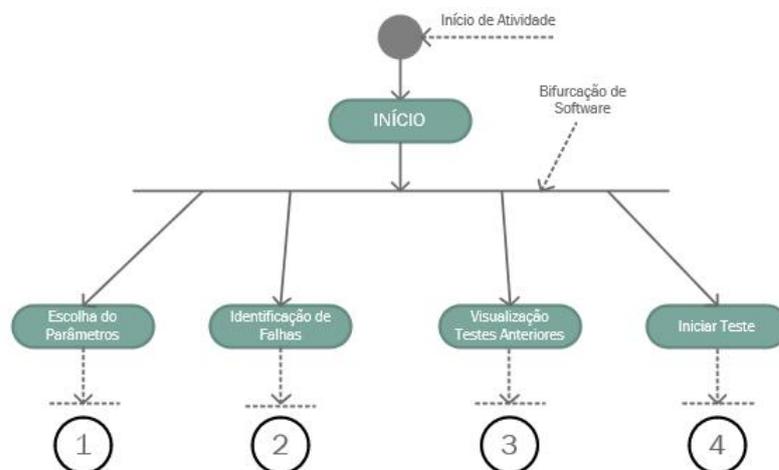
4 ESPECIFICAÇÃO DO SISTEMA DE SOFTWARE

De acordo com Marwedel (2010), as especificações de um sistema embarcado devem ter completude, ausência de contradição e devem derivar implementações da especificação de uma maneira sistemática. Com uma abordagem mais sistemática para as primeiras etapas de modelamento do sistema, a *Unified Modeling Language* - UML se apresentou uma boa alternativa de linguagem (MARWEDEL, 2010). Portanto, esta seção demonstrará a funcionalidade do software por meio da linguagem UML. A UML fornece um suporte no desenvolvimento do processo do software, possuindo um grande número de tipos de diagramas e é uma linguagem gráfica complexa (MARWEDEL, 2010). Neste trabalho, é utilizado o UML de Diagrama de Atividades, que são, de acordo com Marwedel (2010), Redes de Petri estendidas. As barras horizontais representam junções ou bifurcações e o losango representa decisões (MARWEDEL, 2010). É utilizada a ferramenta Microsoft Visio[®] para o desenvolvimento dos diagramas UML.

4.1 VISÃO GERAL

O software foi dividido em quatro partes. Cada parte é representada como uma aba dentro da ferramenta Node-RED, podendo cada um ser acessada a qualquer momento pelo usuário. Por razões intuitivas, foi tratado nesse texto as etapas apresentadas na Figura 38, indo de cima para baixo e da esquerda para a direita.

Figura 38 – Divisão do software do sistema

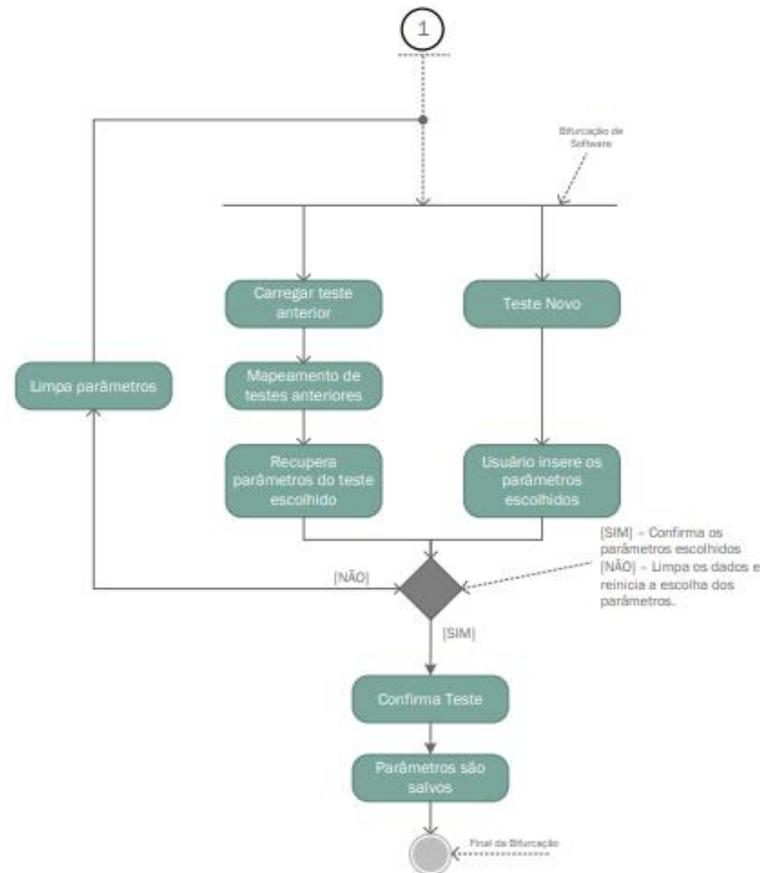


Fonte: Autoria própria (2021).

4.2 ESCOLHA DOS PARÂMETROS

A etapa escolha dos parâmetros, apresentado na Figura 39, é responsável pela coleta dos parâmetros de teste do usuário.

Figura 39 – Fluxograma das escolhas de teste



Fonte: Autoria própria (2021).

A etapa inicia com uma bifurcação que depende da escolha do usuário. O usuário tem a opção de iniciar um teste novo, preenchendo todos os parâmetros especificados na Figura 40. Essa opção cabe em casos nos quais uma configuração será testada pela primeira vez. Para otimização de tempo, é oferecida uma opção de carregamento de testes anteriores, em que o usuário tem a opção de buscar no base de dados um teste anterior e carregar os parâmetros contidos nesse teste para reduzir o tempo de preenchimento.

Em caso de preenchimento correto dos parâmetros, o usuário deve confirmar o teste para que o programa salve os valores em variáveis do sistema que serão usadas em etapas posteriores. Caso o usuário preencha erroneamente ou escolhido um teste anterior de forma incorreta, ele pode limpar os dados e reiniciar o preenchimento.

sensor considerado crítico falhar, o teste não poderá ser iniciado. Caso um sensor não crítico falhe, o usuário é informado de qual sensor falhou, mas o teste pode prosseguir, mesmo cenário para todos os sensores estarem em pleno funcionamento.

4.4 ETAPA 3 - VISUALIZAR TESTES ANTERIORES

A terceira etapa diz respeito ao software apresentar ao usuário a opção de visualizar os dados de testes realizados anteriormente. Ao entrar nessa aba, o usuário encontra a opção de carregamento de teste, com uma lista de testes realizados anteriormente. Com o teste escolhido, o programa busca no base de dados onde estão contidos os testes e é feito o mapeamento dos valores do arquivo CSV. Os valores são extraídos e tratados dentro do programa, apresentando uma visualização gráfica para o usuário. Na Figura 42 é apresentado o fluxograma dessa etapa. O objetivo dessa etapa é tornar o ambiente de testes o mais completo possível para o usuário, não necessitando utilizar outra ferramenta para plotar os dados para análise.

Figura 42 – Visualizar Testes Anteriores



Fonte: Autoria própria (2021).

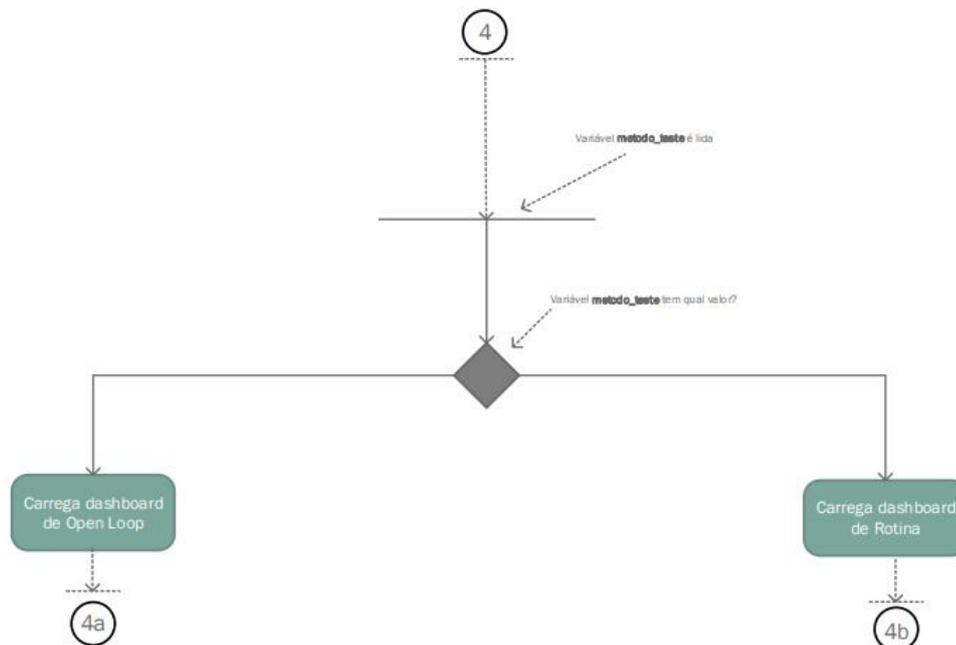
4.5 ETAPA 4 - INICIAR TESTE

A aba Iniciar Teste é o ambiente onde o usuário executa o teste desejado. Inicia-se com a leitura da variável "Formato do Teste", apresentado na seção 4.2 Escolha dos Parâmetros, para

se obter a informação se o teste escolhido foi o de *Open Loop* ou de Rotina. Com esta informação extraída, o programa carrega a *dashboard* específica de acordo com o método escolhido. Decidiu-se fazer dessa maneira, uma vez que a operação de *Open Loop* apresenta a opção do usuário controlar o motor da forma desejada. O método de Rotina não apresenta essa opção, uma vez que o teste será automatizado de acordo com a rotina escolhida.

Caso o usuário opte na Etapa 1 pelo formato *Open Loop*, o *dashboard* é carregado com as instruções de realização do teste. O usuário deve iniciar o teste clicando em um botão gráfico que estará presente na tela. No momento da confirmação, o software, inicialmente, lê os valores instantâneos de tensão na bateria de alimentação do sistema do sistema de potência. Caso o valor da tensão da bateria esteja baixo do limite permitido, o teste não pode iniciar e o usuário é informado do problema, sendo requerido o carregamento da bateria. Caso as baterias estejam devidamente carregadas, o teste continua com a calibração do ESC. Esse fluxograma pode ser visto na Figura 43.

Figura 43 – Fluxograma da etapa 4

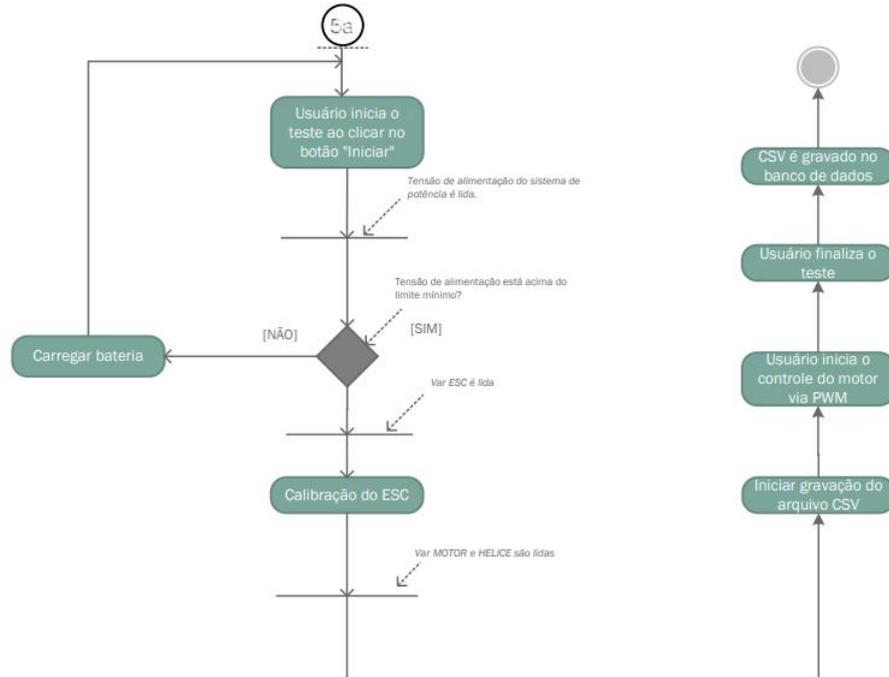


Fonte: Autoria própria (2021).

Previamente, o software buscará a variável ESC e os parâmetros de calibração do controlador. Cada ESC possui uma configuração diferente, portanto se faz necessário a calibração individual desses dispositivos. Posteriormente, o programa coleta os valores dos parâmetros *Hélice*, *Motor* e *Bateria* para montar o nome do arquivo CSV e iniciar a gravação dos dados do teste. O usuário controla a velocidade do motor por meio de uma ferramenta gráfica que simula

um acelerador. Quando achar conveniente, o usuário deve pressionar um botão “Finalizar Teste” para que os dados sejam gravados e o teste finalizado. Essa etapa pode ser vista na Figura 44.

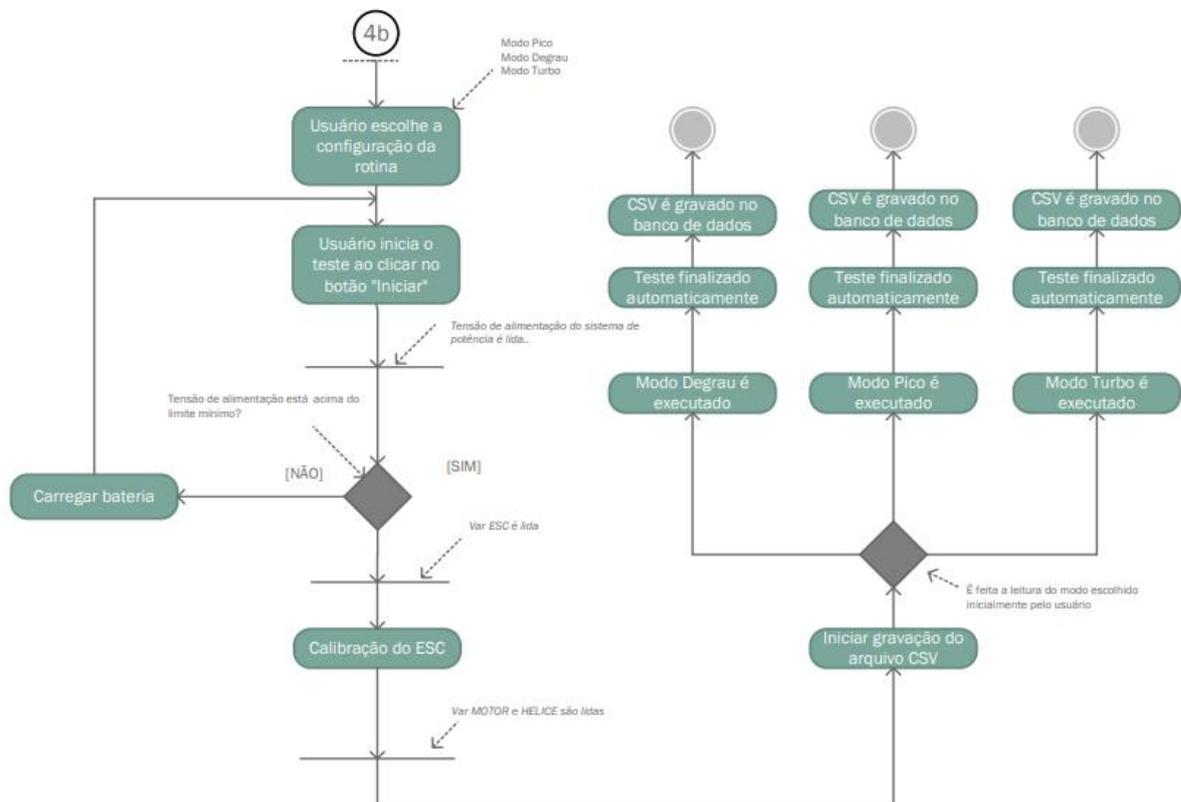
Figura 44 – Fluxograma da Etapa 4a



Fonte: Autoria própria (2021).

A Etapa 4b (Figura 45) inicia com uma escolha do usuário de um dos modos de rotina oferecidos. Com o modo definido, o usuário inicia o teste clicando no botão gráfico "Iniciar". As sequências posteriores são idênticas ao método *Open Loop*. A partir do início da gravação do arquivo CSV é que o código se altera. O algoritmo analisa qual o modo escolhido pelo usuário e realiza o carregamento do teste automatizado escolhido. Detalhes dos testes automatizados podem ser vistos na etapa de Implementação. Com o teste automatizado executado, o processo é finalizado e o arquivo CSV gravado no base de dados.

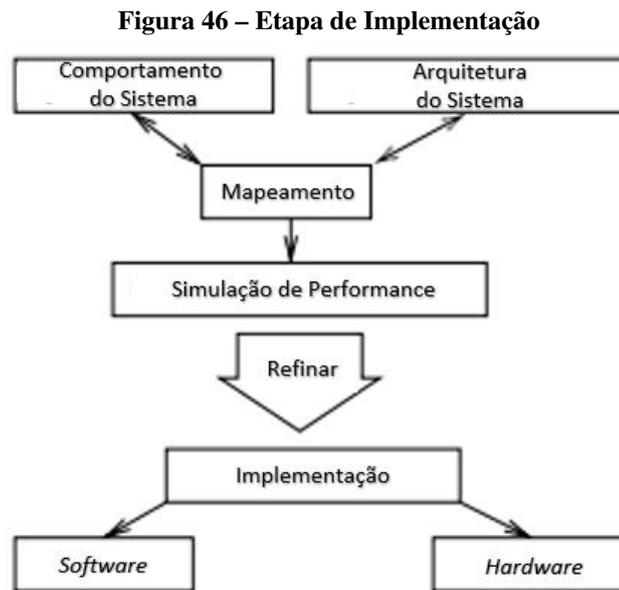
Figura 45 – Fluxograma da etapa 4b



Fonte: Autoria própria (2021).

5 IMPLEMENTAÇÃO

A parte de implementação tem como objetivo apresentar a combinação ideal de hardware e software resultando em um produto eficiente que cumpra com os objetivos apresentados na seção de Especificação do Sistema de software, subdividindo-se em *hardware design* e de *software design*, como mostra a Figura 46.



Fonte: Traduzido de Marwedel (2010).

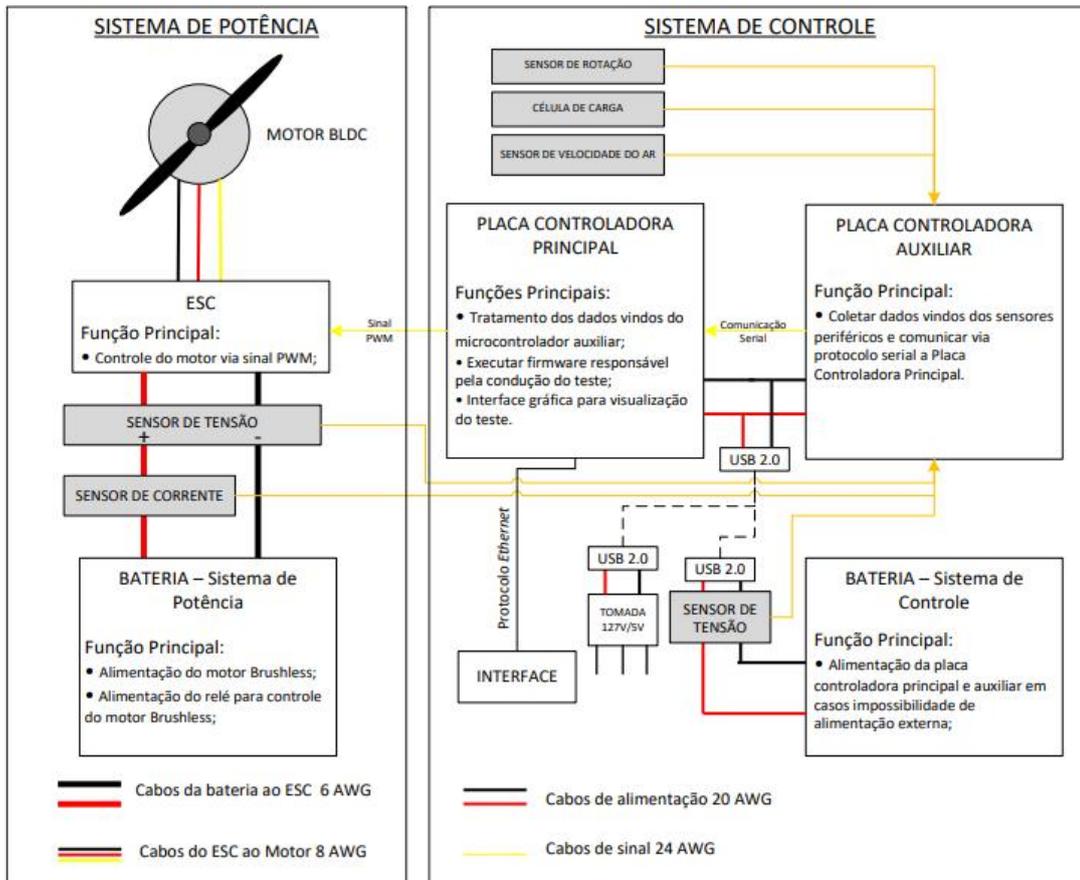
A bancada implementada é de carga estática, com opção de ser controlada com funções de controle para a reprodução de condições de uso. Na etapa de *hardware design*, apresentando o diagrama elétrico para o sistema de potência e de controle, além do projeto mecânico da bancada. Na etapa de *design* é detalhado o *firmware* desenvolvido para a realização dos testes. A visão de blocos do projeto pode ser vista na Figura 47.

5.1 HARDWARE DESIGN

5.1.1 Diagrama Elétrico - Sistema de Controle

O sistema de controle é composto de um sistema principal e de uma placa controladora auxiliar. O sistema principal é o Raspberry Pi 3B+, responsável pela execução do *firmware* de realização dos testes e também de tratamento dos dados vindos da placa controladora auxiliar. Este sistema também tem como função enviar o sinal de controle PWM para o ESC via porta

Figura 47 – Visão em blocos do projeto



Fonte: Autoria própria (2021).

GPIO 16 e de controle de liga/desliga do motor por meio de um relé eletromagnético por meio da porta GPIO 20, como pode ser visto na Figura 48.

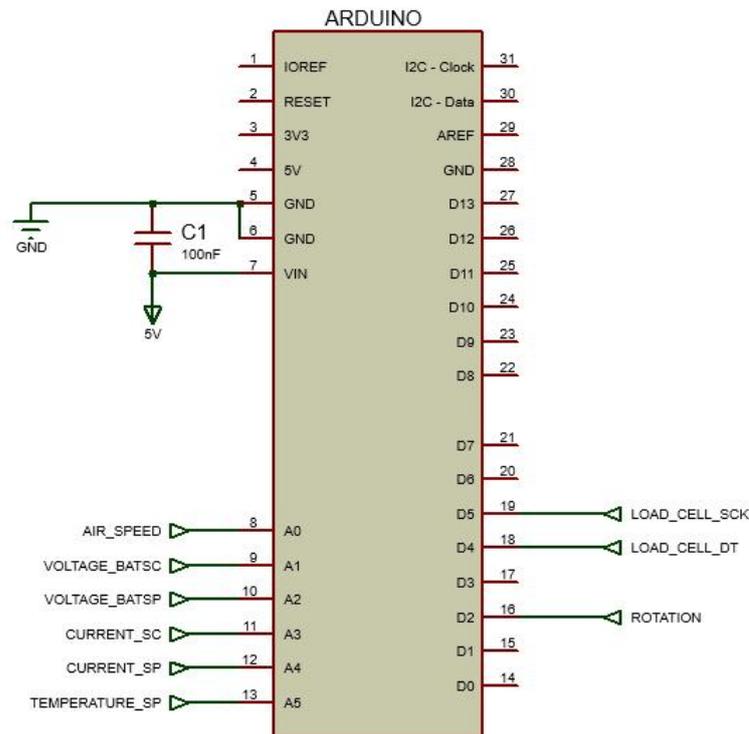
O único sensor que está ligado ao sistema principal é o BMP180, responsável por fornecer dados de temperatura, pressão e altitude ambiente. Decidiu-se conectar este sensor ao Raspberry Pi 3B+, pois não há necessidade de uma grande taxa amostral desses dados.

A placa controladora auxiliar é o Arduino, responsável pela coleta dos dados dos sensores que requerem alta taxa amostral e envio dos dados para o sistema principal. Inicialmente, o objetivo seria utilizar o Raspberry Pi 3B+ como o único sistema microprocessado, porém notou-se que com a complexidade do *firmware* e também pela necessidade de velocidade dos dados advindos dos sensores, o processador ficaria sobrecarregado.

O sensor BMP180 se comunica com o Raspberry Pi 3B+ via protocolo I²C utilizando o GPIO 2 para a linha SDA e GPIO 3 para a linha SCL do barramento, como pode ser observado analisando conjuntamente a Figura 48 e Figura 49. Na Figura 50 é possível ver o diagrama elétrico da Placa controladora auxiliar.

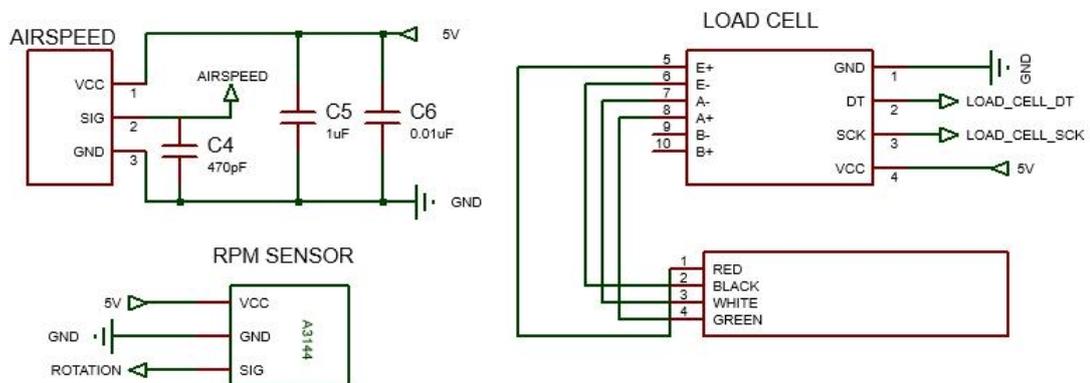
Na Figura 51 é apresentado o esquema elétrico dos sensores conectados próximos ao

Figura 50 – Diagrama dos pinos da placa controladora auxiliar



Fonte: Autoria própria (2021).

Figura 51 – Sensores próximos ao motor

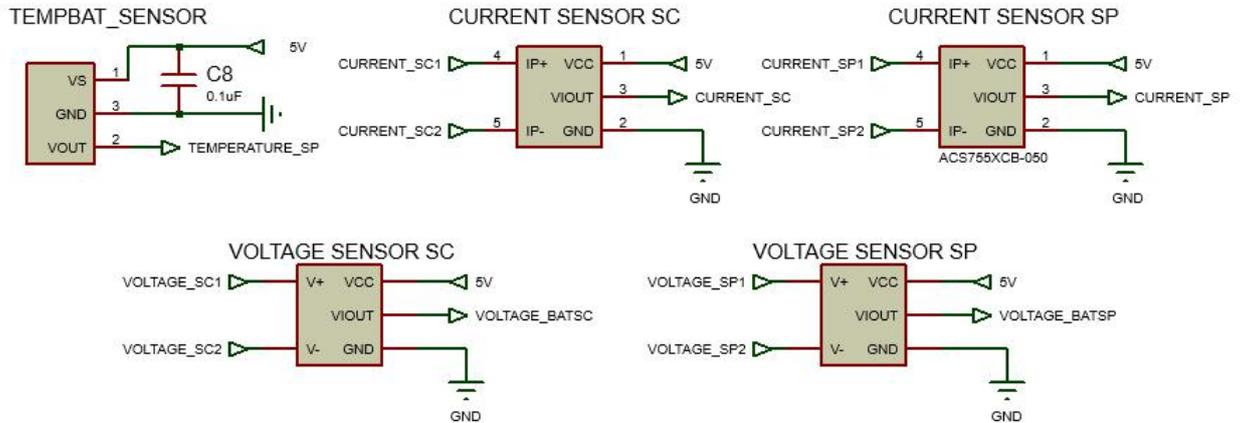


Fonte: Autoria própria (2021).

5.1.2 Diagrama Elétrico - Sistema de Potência

O sistema de potência consiste de todos os dispositivos que atuam diretamente no controle do motor *Brushless* CC. O sistema é alimentado com uma bateria Powerbolt 14,8V 4000mAh 60C. O sistema possui um conjunto de capacitores de desacoplamento para filtragem de ruídos, devido ao chaveamento em alta frequência do ESC. Utilizou-se um relé de 12 V 70 A normalmente aberto para controle do motor, acionando com o início do teste e desligando ao final do teste. Foi escolhido um conversor CC-CC Buck para a redução da tensão da bateria para

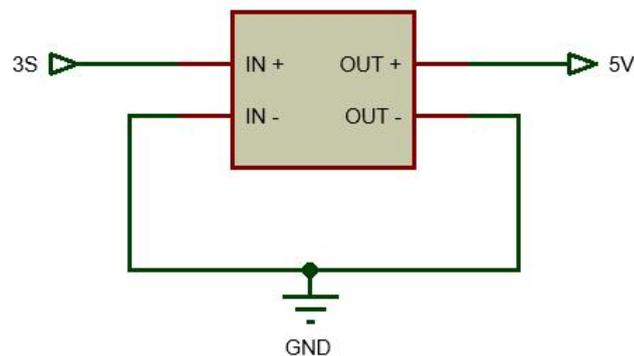
Figura 52 – Sensores de monitoramento



Fonte: Autoria própria (2021).

Figura 53 – Conversor Buck

CONVERSOR BUCK

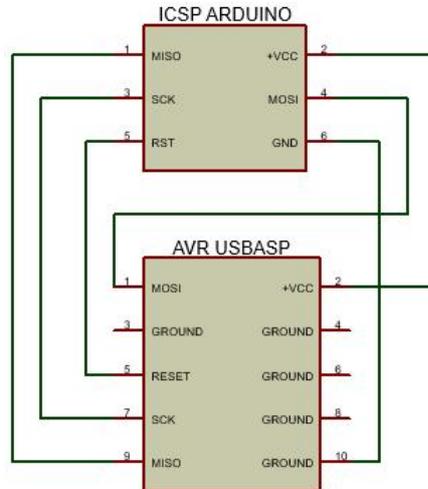


Fonte: Autoria própria (2021).

o valor aceitável para a alimentação da bobina deste relé. Em paralelo com a bobina do motor, foi inserido um diodo 1N4007 como *Fly-back*, a fim de evitar a corrente de descarga da bobina quando a mesma é desligada. A Figura 55 mostra a disposição do circuito de alimentação do sistema de potência.

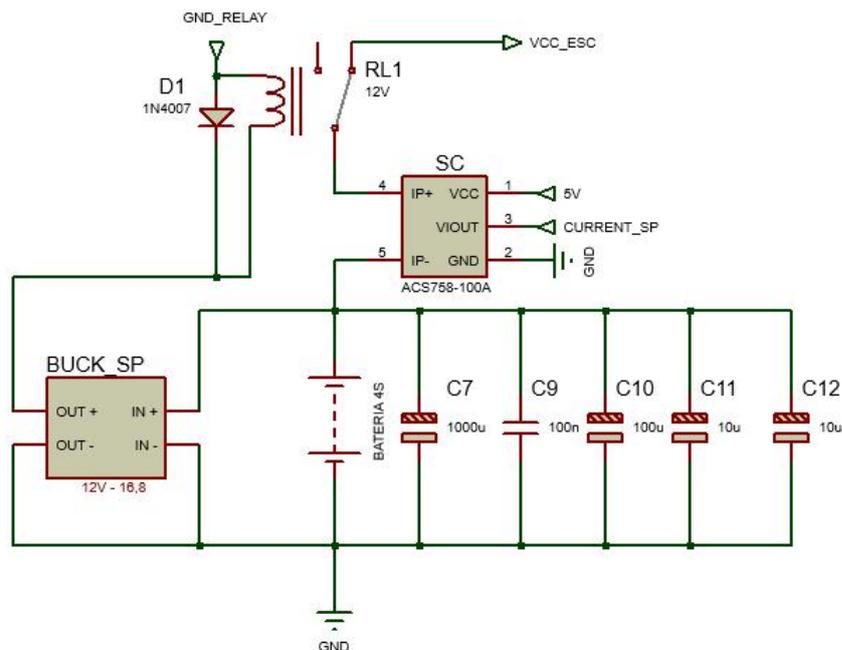
No momento em que o sistema é ligado, a tensão da bateria não alimenta o ESC, pois o relé está desenergizado. Para que o relé seja acionado, o usuário envia um comando para que o relé seja chaveado e assim o motor alimentado. A Figura 56 apresenta o circuito responsável por realizar esse comando. O circuito é constituído do transistor 2N2222 e um resistor de 1200 Ω. No estado inicial do sistema, o sinal SIG RELAY envia um nível lógico baixo para a base do transistor, mantendo o circuito aberto. No momento em que o usuário iniciar o teste, o sinal SIG RELAY enviará o nível lógico alto, com tensão de 3,3 V, chaveando o transistor e conectando

Figura 54 – Circuito para gravação do *bootloader* do Arduino



Fonte: Autoria própria (2021).

Figura 55 – Circuito de Alimentação do Sistema de Potência



Fonte: Autoria própria (2021).

o GND com a bobina do relé, possibilitando a passagem de corrente, logo acionando o relé. A bobina do relé possui $73,6 \Omega$, requerendo uma corrente de $0,165 \text{ A}$ com uma alimentação de 12 V . Por segurança, o valor da corrente foi arredondado para $0,2 \text{ A}$. De acordo com o *datasheet* do transistor, o ganho h_{FE} é considerado 100 .

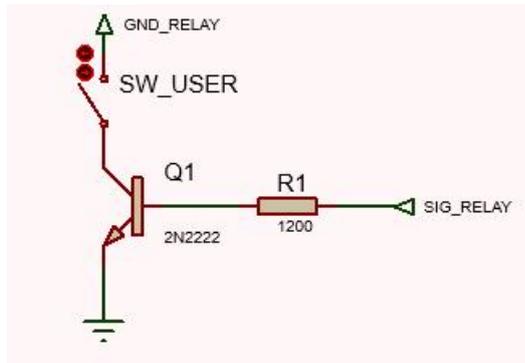
$$I_B = I_C / h_{FE} = 0,2 / 100 = 2 \times 10^{-3} \text{ A} \quad (8)$$

O nível lógico alto do Raspberry Pi 3B+ é de $3,3 \text{ V}$. Considerando a tensão entre base e emissor sendo de $0,7 \text{ V}$:

$$R_B = \frac{V_{BB} - V_{BE}}{I_B} = \frac{3,3 - 0,7}{2 \times 10^{-3}} \approx 1300\Omega \quad (9)$$

Por ser difícil encontrar um resistor especificamente de 1300 Ω , foi escolhido um de 1200 Ω , como é mostrado na Figura 56.

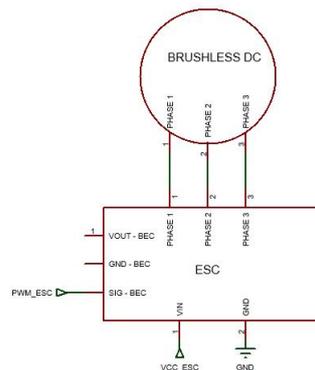
Figura 56 – Circuito de acionamento do relé



Fonte: Autoria própria (2021).

Foi posicionado o sensor de corrente Allegro ACS758 em série com a entrada do relé, a fim de medir a corrente de entrada no motor. Na Figura 57 é possível visualizar o circuito de ligação do ESC ao motor. Logo após o sensor de corrente, é feita a ligação no polo positivo do ESC, energizando assim o motor. No ESC, a entrada de PWM é ligada na porta GPIO 16 da placa principal.

Figura 57 – ESC e motor BLDC



Fonte: Autoria própria (2021).

5.1.3 Bancada Mecânica

A bancada foi projetada para ser fixada a qualquer superfície plana rígida de forma rápida, fácil e segura. O projeto mecânico da bancada consistiu de um suporte para o motor, uma base de fixação do suporte e uma estrutura para a fixação do conjunto na superfície plana,

projetado na ferramenta NX Siemens®. O suporte do motor é uma estrutura de fixação triangular, com uma célula de carga posicionada na parte superior acoplada ao motor, como pode ser visto na Figura 58. A estrutura deve ser capaz de fornecer rigidez necessária para suportar os movimentos de tração do motor sem sofrer flexão, pois o mínimo deslocamento influencia nos valores registrados pela célula de carga.

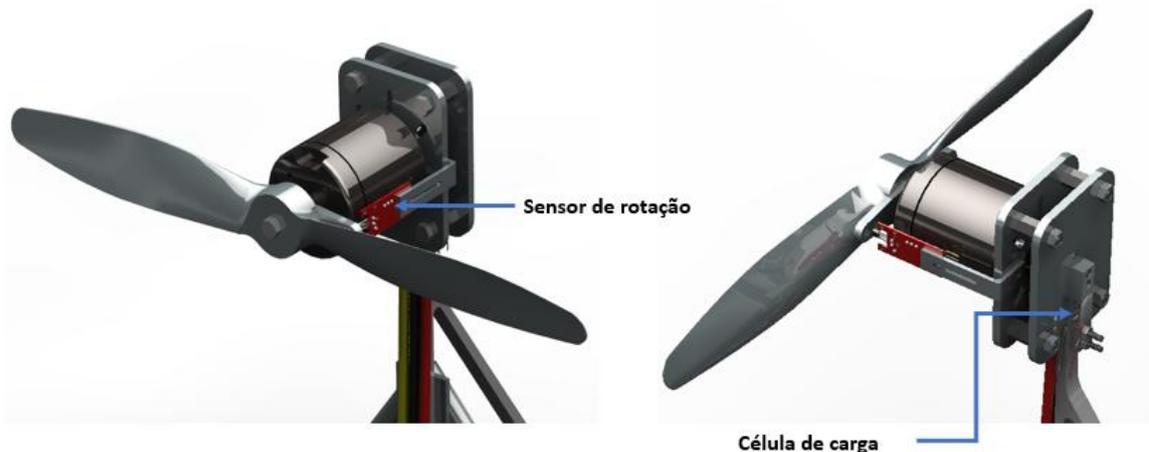
Figura 58 – Estrutura de fixação do motor e célula de carga



Fonte: Autoria própria (2021).

Para o acoplamento da célula de carga com o motor, foi desenvolvida uma chapa de alumínio, mostrada na Figura 59. Nessa mesma, posicionou-se o suporte do sensor de rotação, fabricado em impressora 3D. O sensor pode ser posicionado em qualquer local do motor, pois a chapa de fixação foi prevista com diferentes pontos de fixação do mesmo, por razões de diferentes motores e configurações. Além disso, outra peça desenvolvida em 3D foi acoplada no eixo do motor, logo atrás da hélice, contendo os ímãs para medição da rotação.

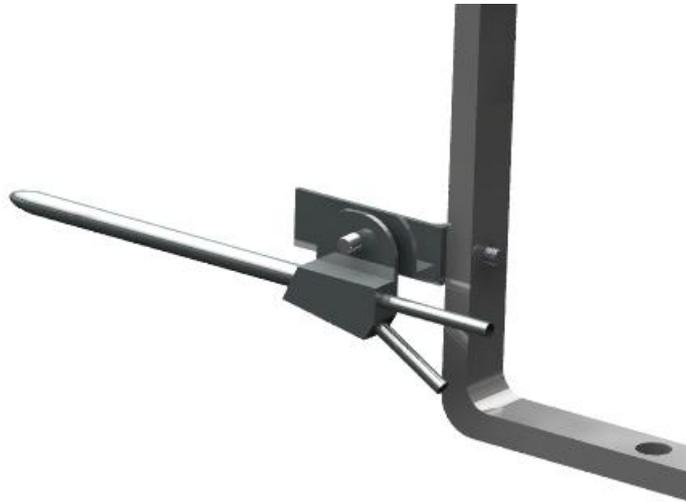
Figura 59 – Acoplamento do motor



Fonte: Autoria própria (2021).

A fim de evitar o fluxo de ar turbulento advindo da hélice, o sensor de velocidade do ar foi posicionado na parte inferior da estrutura triangular, paralelo ao sentido do ar, como é mostrado na Figura 60.

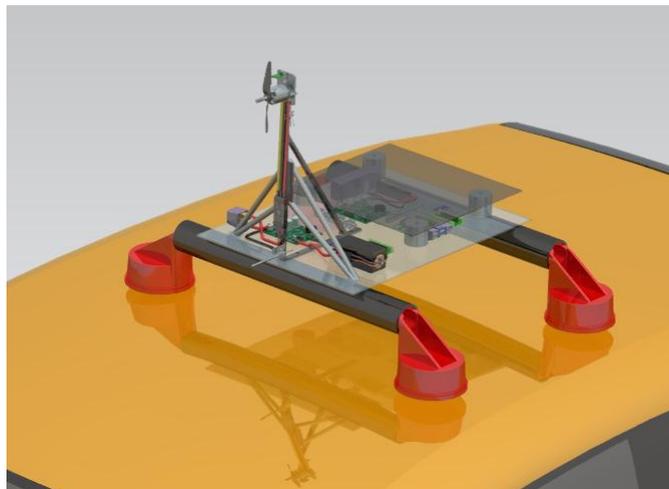
Figura 60 – Fixação do Tubo de Pitot



Fonte: Autoria própria (2021).

Para a base onde é fixado o suporte do motor, utilizou-se um acrílico transparente com 4 mm de espessura. Essa mesma base serve de suporte para o sistema eletrônico embarcado. Todo o conjunto foi acoplado à duas barras parafusadas longitudinalmente com ventosas presas nas pontas. O sistema de ventosas faz a fixação da bancada na superfície, como é demonstrado na Figura 61 e Figura 62.

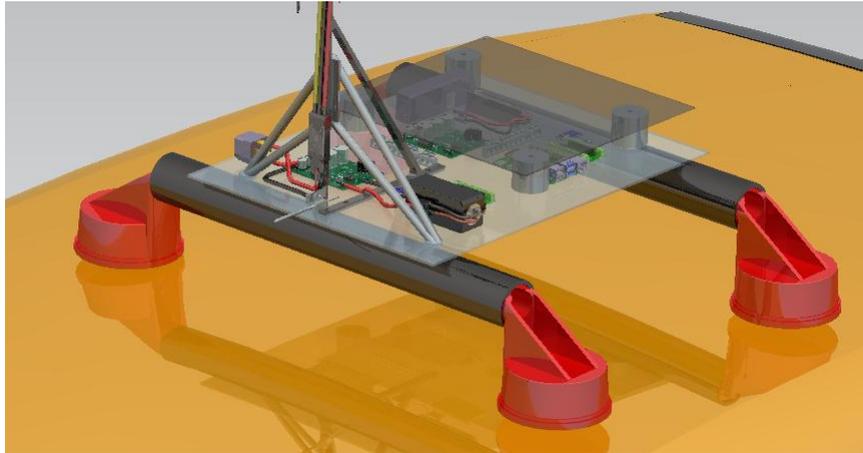
Figura 61 – Bancada montada na parte superior de um veículo



Fonte: Autoria própria (2021).

Uma chapa de acrílico de 2 mm insulfilmada foi colocada na parte superior a fim de diminuir o contato do sistema eletrônico com o ar.

Figura 62 – Estrutura de fixação da bancada por meio de ventosas



Fonte: Autoria própria (2021).

5.2 SOFTWARE DESIGN

O software foi desenvolvido, em sua grande maioria, na ferramenta Node-RED, através do qual foram criados os procedimentos de teste, interface para o usuário e o tratamento dos dados dos sensores. A realização do teste é toda feita utilizando a ferramenta Node-RED sendo executada na placa controladora principal. Como os testes a serem executados pela equipe AeroDesign UTFPR Curitiba serão executados tanto em laboratórios como em veículos em movimento, o acesso a placa controladora se torna limitado, deixando a opção de utilizar cabo HDMI para a transmissão da tela do microcontrolador limitada, pela dificuldade de energização de um monitor durante o teste. Para isso, será utilizada a rede LAN do Raspberry Pi 3B+ por meio do protocolo Ethernet. Um cabo Ethernet de 2 metros foi disponibilizado para equipe. Este cabo deve ser conectado à um notebook e a visualização da tela é possível utilizando o software VNC Viewer por meio de um IP configurável.

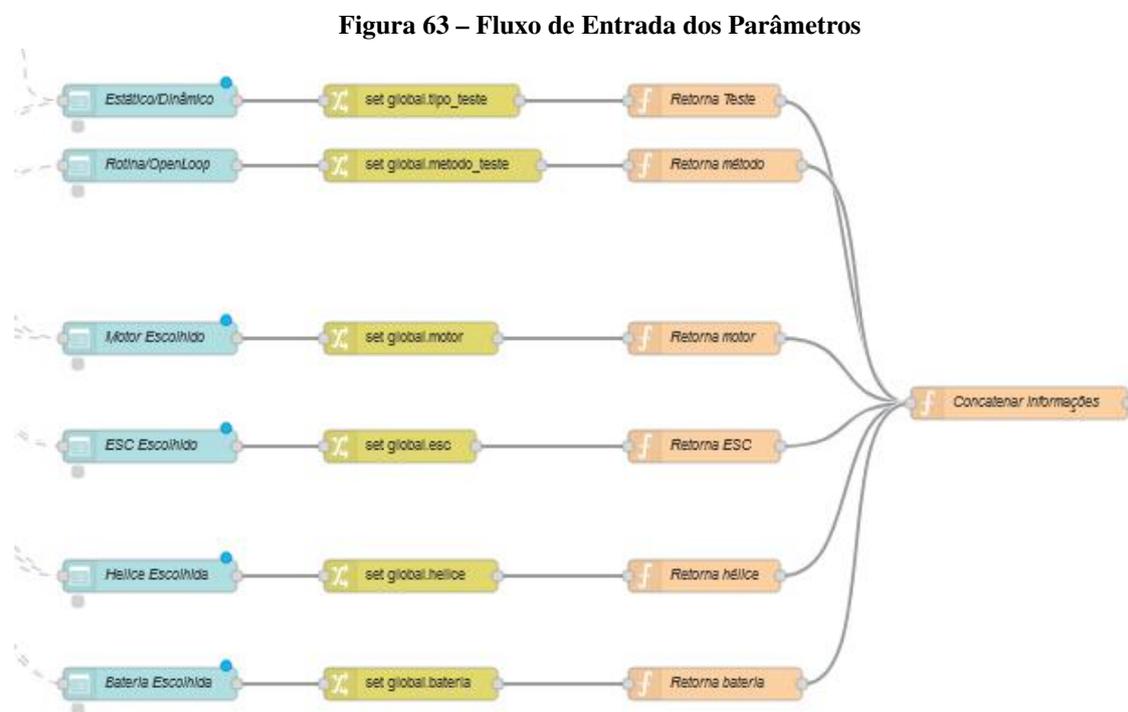
5.2.1 Entrada de Parâmetros

Esta aba é responsável por extrair os parâmetros básicos de testes. Como a bancada será utilizada para a realização de todos os motores a serem adquiridos pela equipe, foi disponibilizado um sistema de escolha do grupo motopropulsor utilizado. Os parâmetros considerados básicos são:

- Tipo de Teste;
- Formato do Teste;

- Motor escolhido;
- ESC escolhido;
- Hélice escolhida;
- Bateria Escolhida;
- Configuração do Teste.

O usuário tem a opção de escolher cada um dos parâmetros baseado no base de dados. Na Figura 63 é mostrado o fluxo do software. Para mais detalhamento do código, verificar Apêndice B.



Fonte: Autoria própria (2021).

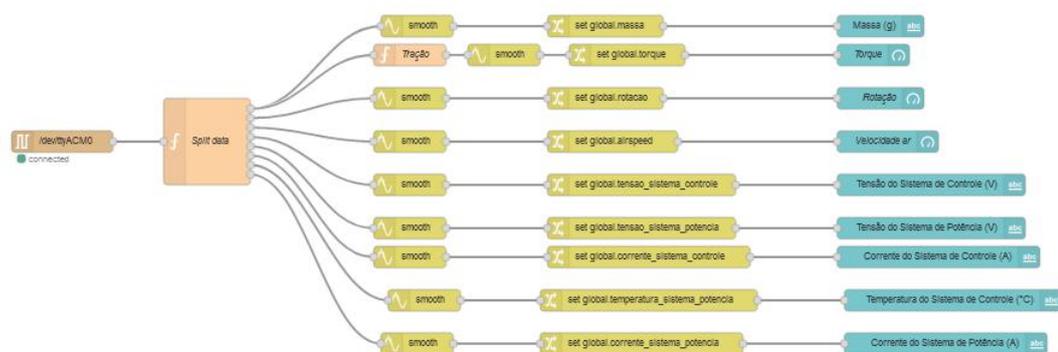
5.2.2 Coleta de dados dos sensores

Como descrito anteriormente, a placa controladora auxiliar é responsável pela coleta dos dados vindos dos sensores periféricos. O software para a coleta dos dados foi desenvolvido em linguagem de programação C, junto à IDE do Arduino. Foi criado uma função em *loop* para cada um dos sensores. A saída de cada sensor é então formatada e enviada com uma taxa de 100 ms para a porta serial do Arduino. O Node-RED fornece a possibilidade de leitura da porta serial

utilizando o nó *serial in*. A única informação necessária para informar ao nó é a porta serial onde esses dados estão sendo transmitidos.

Cada mensagem serial é passada para um nó do tipo *function*, denominado de “*Split data*” dentro do código, que coleta os dados vindos como mensagem do nó “*serial in*” e designar cada dado valor à variável correspondente à cada sensor. O fluxo completo de tratamento dos dados dos sensores pode ser visto na Figura 64. Para maiores detalhes do software, verificar Apêndice C.

Figura 64 – Fluxo de tratamento dos valores advindos da placa controladora auxiliar



Fonte: Autoria própria (2021).

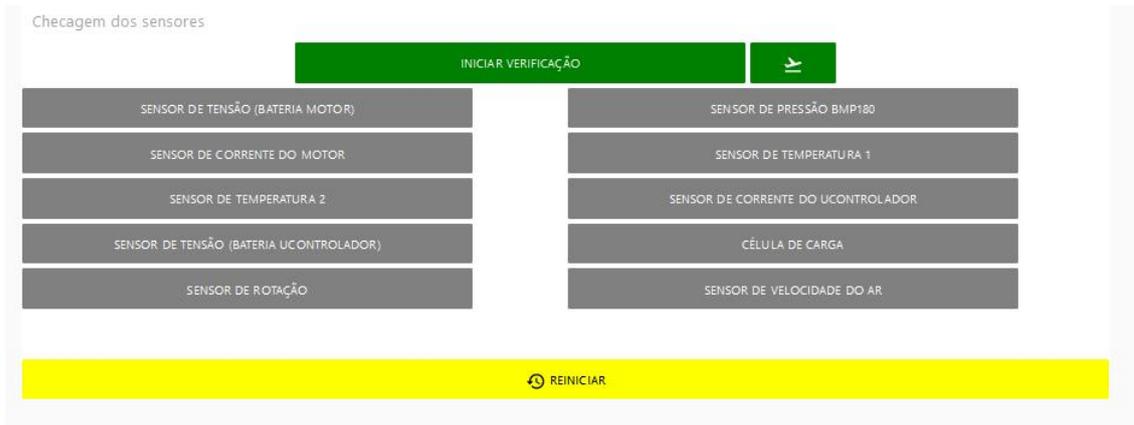
5.2.3 Etapa de Verificação

A etapa de verificação foi criada com o intuito de verificar a funcionalidade de todos os sensores presentes na bancada, a fim de informar o usuário antes do início do teste. O programa de verificação inicia após o usuário pressionar o botão “Iniciar Verificação”, presente na interface da Figura 65.

A calibração inicia com o nó do tipo *function* que analisa o valor vindo do sensor, informando via mensagem do tipo *payload* se o valor está dentro do limite estabelecido de funcionamento. Para cada sensor, existe um nó de função específico com a definição do que é aceitável ou não. Na Figura 66 é demonstrada a verificação do sensor de tensão da bateria do motor. Este sensor é considerado crítico por ser responsável direto por fornecer os dados de tensão do motor, utilizado para calcular a potência elétrica do mesmo.

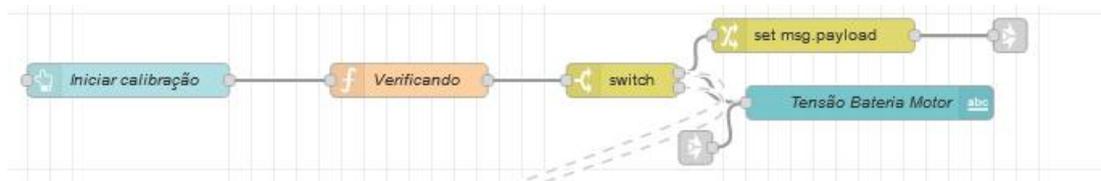
Após a passagem pelo nó “Verificando”, o próximo nó do fluxo é *switch*. Este nó avalia a mensagem do tipo *payload* na entrada e, dependendo do valor, direciona para um caminho específico. O valor de saída do nó de verificação é um texto “OK” e “NOT OK”. Caso o valor

Figura 65 – Interface da etapa de Verificação



Fonte: Autoria própria (2021).

Figura 66 – Fluxo de verificação do sensor de bateria do motor



Fonte: Autoria própria (2021).

seja “NOT OK”, o fluxo segue para um caminho de alimentação da variável de contexto global “falha”, informando que o teste não pode ser iniciado sem a correção desse sensor, por ser crucial para a segurança e execução do teste. A verificação prossegue para todos os outros sensores presentes no projeto, e ao final é apresentado um aviso ao usuário sobre o *status* dos sensores.

5.2.4 Modos de teste

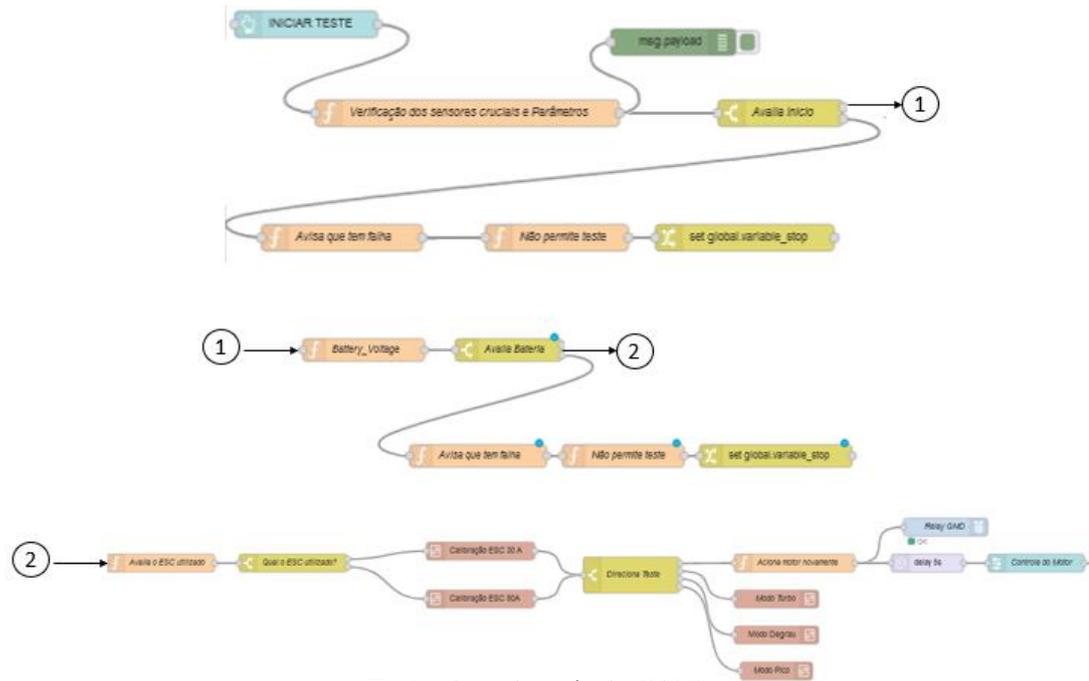
São oferecidas duas opções para o usuário executar o teste: Modo *Open Loop* e Modo Rotina, que são escolhidos na etapa de “Entrada de Parâmetros”. Após a verificação dos sensores, o usuário deverá ir na aba “Display” para que o teste possa ser iniciado.

Ao usuário pressionar o botão Iniciar, é dado início ao fluxo de realização do teste. A primeira etapa é a verificação de todos os sensores considerados cruciais para a realização do teste, por meio do nó “Verificação dos sensores cruciais e Parâmetros”, como pode ser visto na Figura 67. Para mais detalhes do código, verificar Apêndice D.

5.2.5 Etapa de Visualização de Dados

A etapa de visualização de dados é utilizada para os usuários observarem os resultados plotados de forma gráfica, sejam eles do teste atual como de qualquer teste já feito pela equipe

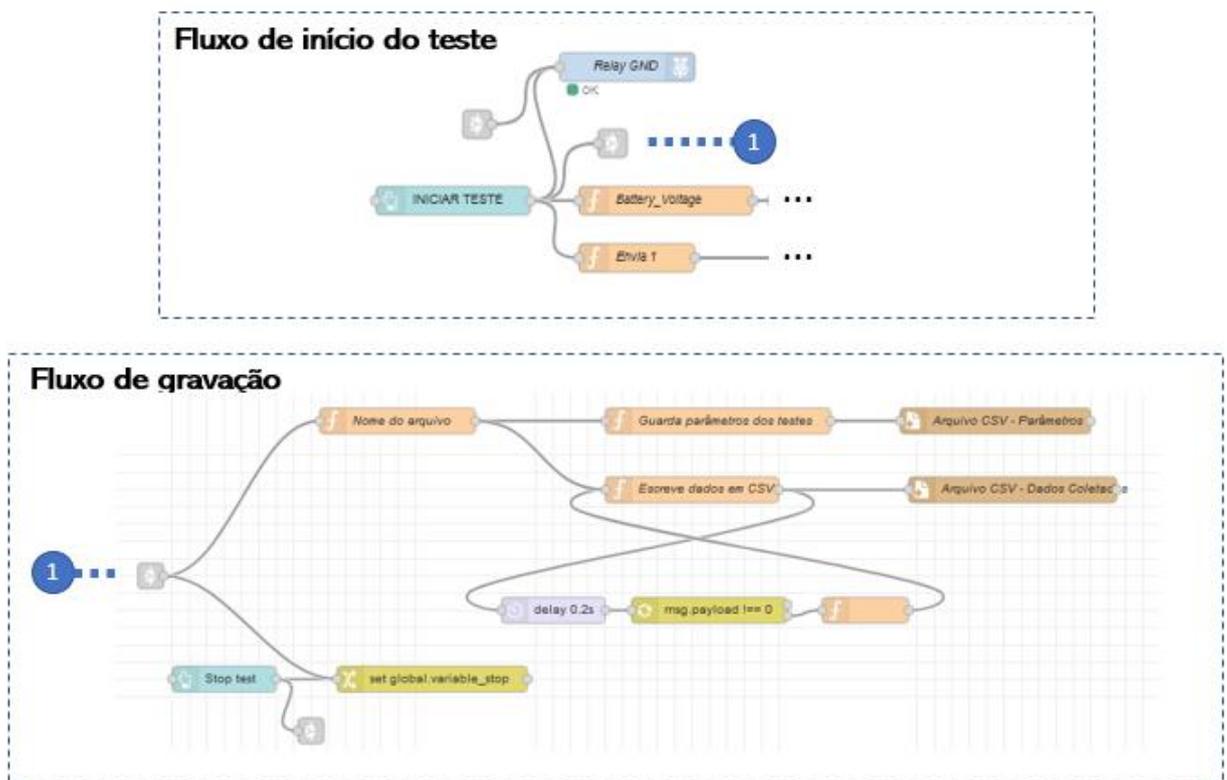
Figura 67 – Fluxo de realização do teste



Fonte: Autoria própria (2021).

que esteja no base de dados. Inicialmente, foi abordada a gravação dos resultados dos testes. Na etapa Modos de teste, seção 5.2.4, é informado que o botão Iniciar começa o teste. Nesse mesmo momento, este botão inicia o fluxo de gravação dos parâmetros do teste e também dos dados dos sensores, como pode ser visto na Figura 68. Para mais detalhes do código, verificar Apêndice F.

Figura 68 – Fluxo de gravação do teste



Fonte: Autoria própria (2021)

6 RESULTADOS

6.1 MONTAGEM FÍSICA DA BANCADA

6.1.1 Montagem mecânica

Inicialmente, a estrutura de fixação do motor contava somente com uma estrutura triangular, conforme a Figura 69.

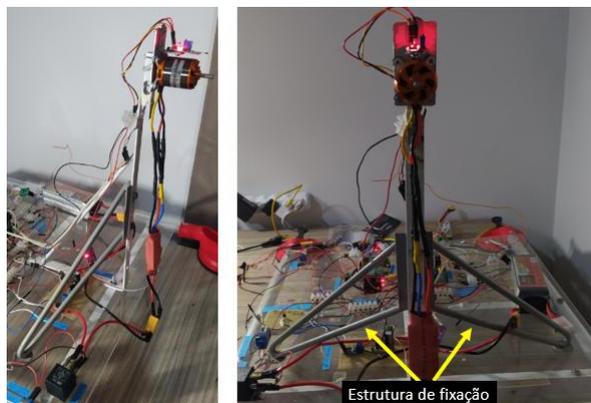
Figura 69 – Suporte de fixação previsto anteriormente no projeto



Fonte: Autoria própria (2021).

A estrutura foi montada fisicamente, no entanto se mostrou muito instável quando posta em movimento. Para solucionar o problema, foram postas vigas que restringem o movimento de rotação do suporte do motor. A Figura 70 mostra a bancada com o suporte reforçado.

Figura 70 – Estrutura de fixação



Fonte: Autoria própria (2021).

A nova fixação se mostrou muito estável quando o motor foi posto em situação de tração

e também ao ser montado no topo do carro. Essa nova montagem reduz erros da leitura de tração da célula de carga, uma vez que a base onde a célula está posicionada deve ser a mais rígida possível para evitar distorções de leitura. Para a fixação da bancada em uma mesa ou na parte superior de um veículo, foi adaptado barras com ventosas em suas pontas, como mostra a Figura 71. O sistema se mostrou confiável ao ser preso no topo de veículo. A único problema encontrado foi a aderência das ventosas, se comportando bem em superfícies planas, lisas, não porosas sem desenhos ou texturas, pinturas lisas e brilhantes somente. Entretanto, não forneceram resultado apropriado em mesas de madeira.

Figura 71 – Sistema de ventosas



Fonte: Autoria própria (2021).

A chapa de acrílico utilizada para a montagem de todo o sistema elétrico e integração do suporte do motor apresentou características satisfatórias a respeito de design, no entanto sofreu flambagem quando todos os componentes estavam montados sobre ela. Para corrigir esse problema, foram dispostas duas barras estabilizadoras transversais na parte de fixação do motor, área que mais sofreu com o fenômeno. Na Figura 72 é possível verificar as duas barras transversais. Com a montagem, a bancada se mostrou mais estável quando posta em ambientes de maior vibração, como em testes dinâmicos em cima do veículo.

A fixação do motor é apresentada na Figura 73. Foi utilizado o montante incluído no *kit* do motor para fazer a montagem na chapa desenvolvida no projeto. Foi desenvolvida uma chapa onde a célula de carga fica posicionada. Em uma chapa logo à frente, espaçada utilizando peças espaçadoras, foi preso o sensor de rotação e o motor.

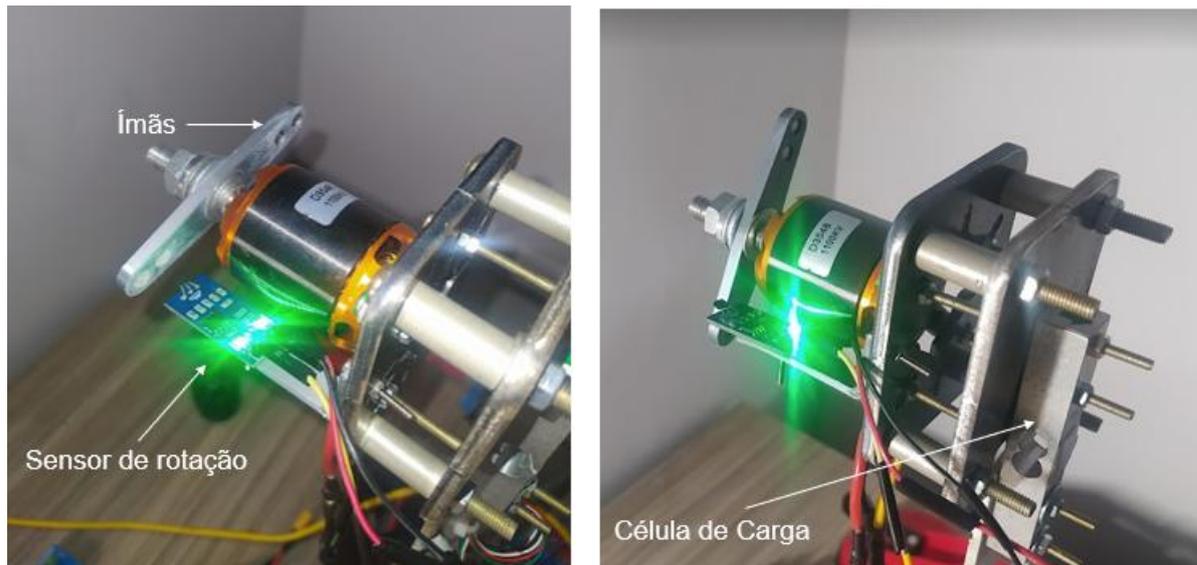
O sensor de rotação foi fixado ao lado esquerdo do motor, com a parte sensível de efeito Hall preso logo após o fim da carcaça do motor. Com isso, a ponta sensível fica mais próxima à peça que contém os ímãs responsáveis por causar o distúrbio de fluxo magnético proporcionando a leitura das rotações do motor. Na Figura 74 é mostrado o sistema de coleta de

Figura 72 – Barras estabilizadoras para diminuir movimento de flambagem da bancada



Fonte: Autoria própria (2021).

Figura 73 – Estrutura de fixação do motor, célula de carga e sensor de rotação



Fonte: Autoria própria (2021).

rotações do motor. Na Figura 75 é possível ver a bancada montada na parte superior do carro para testes dinâmicos.

Para evitar a exposição do circuito e proteger a bancada de eventuais objetos caindo sobre o sistema elétrico, selecionou-se uma chapa de acrílico de 2 mm que cobre toda a região, apresentado na Figura 76.

Figura 74 – Sistema de coleta de rotações do motor



Fonte: Autorial própria (2021).

Figura 75 – Bancada montada para testes dinâmicos



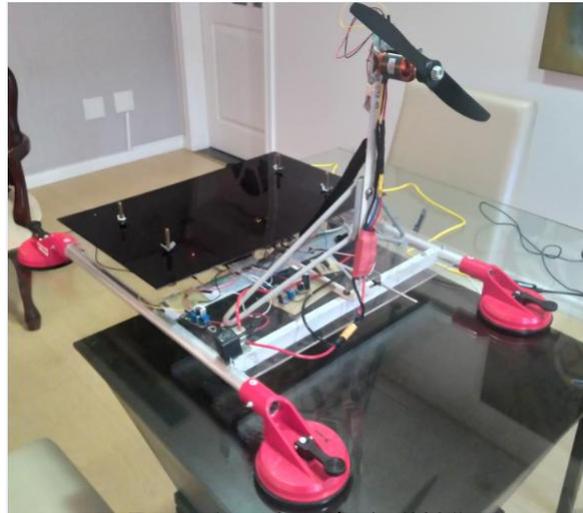
Fonte: Autorial própria (2021).

6.1.2 Montagem elétrica

O sistema elétrico foi disposto sobre a chapa de acrílico de 4 mm de espessura, não sendo projetado placa de circuito impresso para o projeto. A disposição dos controladores e dos periféricos foi feita visando facilidade de montagem e desmontagem do chicote elétrico. Um dos objetivos de design foi alterar minimamente o circuito elétrico em caso de alteração do grupo motopropulsor a ser testado, sendo utilizados conectores do tipo XT60 para a rápida desacoplagem da bateria, ESC e motor.

Para facilitar o diagnóstico de problemas, foram colocados barramentos para os sinais vindos dos sensores, não necessitando ter acesso direto aos controladores para identificar falha no chicote elétrico. Barramentos para a alimentação e *ground* também foram fornecidos. Para fins

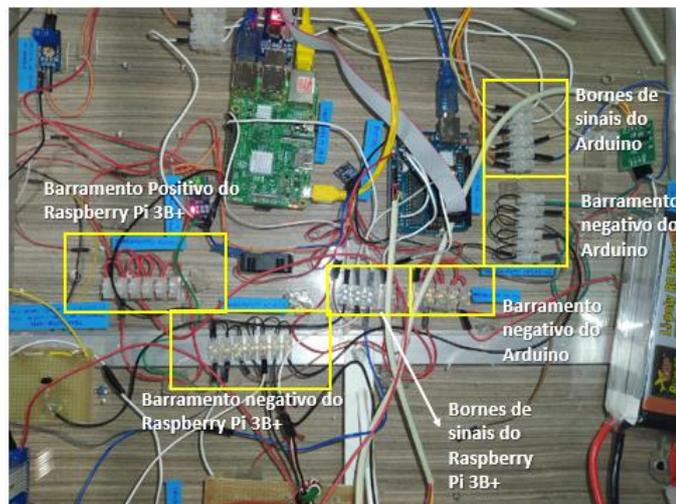
Figura 76 – Chapa de acrílico protegendo o circuito eletrônico



Fonte: Autoria própria (2021).

didáticos, foram colocadas etiquetas de identificação em todas as partes do circuito, explicando a funcionalidade de cada área para facilitar a adaptação ao projeto pela equipe do AeroDesign UTFPR Curitiba. A Figura 77 apresenta a vista superior do circuito elétrico montado.

Figura 77 – Vista superior do circuito elétrico

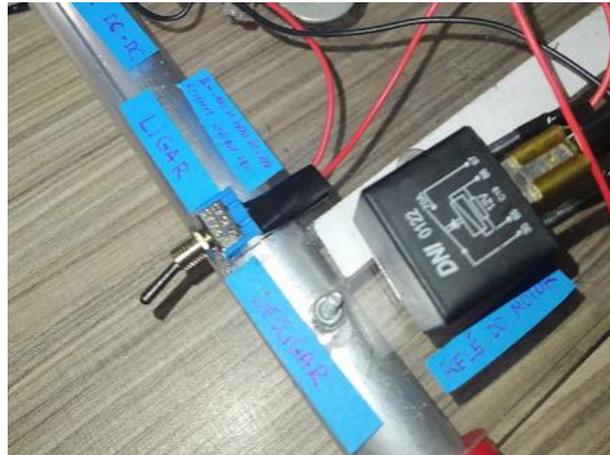


Fonte: Autoria própria (2021).

Uma vez que toda a bancada passará por vibrações constantes nos testes, um ponto de preocupação são as terminações dos cabos nos equipamentos da bancada. Para resolver isso, foram utilizados os terminais adequados para cada dispositivo que garanta a máxima qualidade de fixação dos cabos. Um ponto levantado pelos membros da equipe AeroDesign UTFPR Curitiba foi em caso de falha da comunicação Ethernet entre a bancada e o notebook durante a realização do teste. Com a perda da comunicação, o motor ficaria ligado sendo impossibilitado o seu desligamento de forma fácil, a única alternativa seria manualmente retirando a alimentação do

motor. No entanto, com a chapa superior colocada, o acesso à essa área é limitado, necessitando então de outra alternativa. Para isso, colocou-se uma chave em série com o sinal que aciona o relé eletromagnético. Em caso de falha de comunicação, o simples acionamento da chave irá desligar a alimentação do motor, mas mantendo o restante do sistema embarcado alimentado. Para retomar o teste, basta corrigir o problema de comunicação e colocar a chave na posição inicial. A chave é apresentada na Figura 78.

Figura 78 – Chave utilizada para desligamento instantâneo do motor



Fonte: A autoria própria (2021).

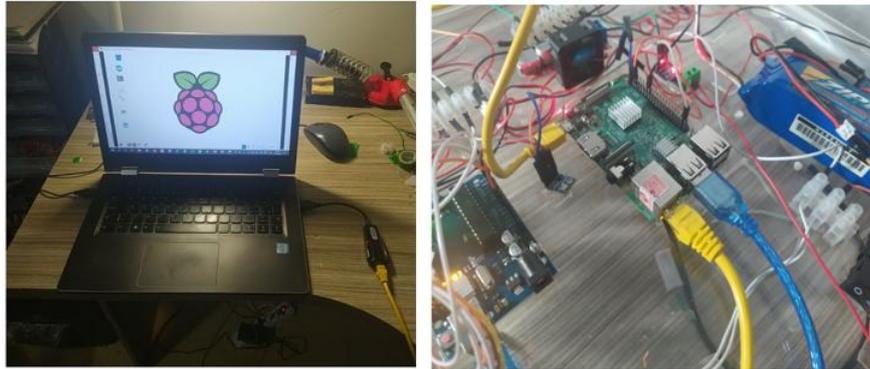
6.2 COMUNICAÇÃO E INTERFACE GRÁFICA

A conexão com o Raspberry Pi 3B+ é feita via cabo Ethernet, portanto conecta-se um lado do cabo ao notebook e o outro ao Raspberry Pi 3B+. Para a conexão ser bem sucedida, é necessário alterar as opções de adaptador, modificando as propriedades da conexão Ethernet e associar a conexão à um IP. Desta maneira, é possível utilizar o software VNC Viewer, permitindo o acesso remoto ao IP designado, tendo acesso à tela do Raspberry Pi 3B+, como é possível ver na Figura 79. À esquerda da Figura 79 está o notebook utilizado conectado ao Raspberry Pi 3B+ via VNC Viewer e à direita a conexão feita ao Raspberry Pi 3B+.

Caso o usuário não deseje utilizar o cabo Ethernet para a conexão, é possível realizar isso pelo Wi-Fi. Basta conectar o Raspberry Pi 3B+ à internet e habilitar o VNC Server dentro das configurações. Uma vez conectado à internet, o próprio Raspberry Pi 3B+ gera um IP vinculado à rede, sendo possível acessar também via aplicativo VNC Viewer. Os passos podem ser vistos na Figura 80.

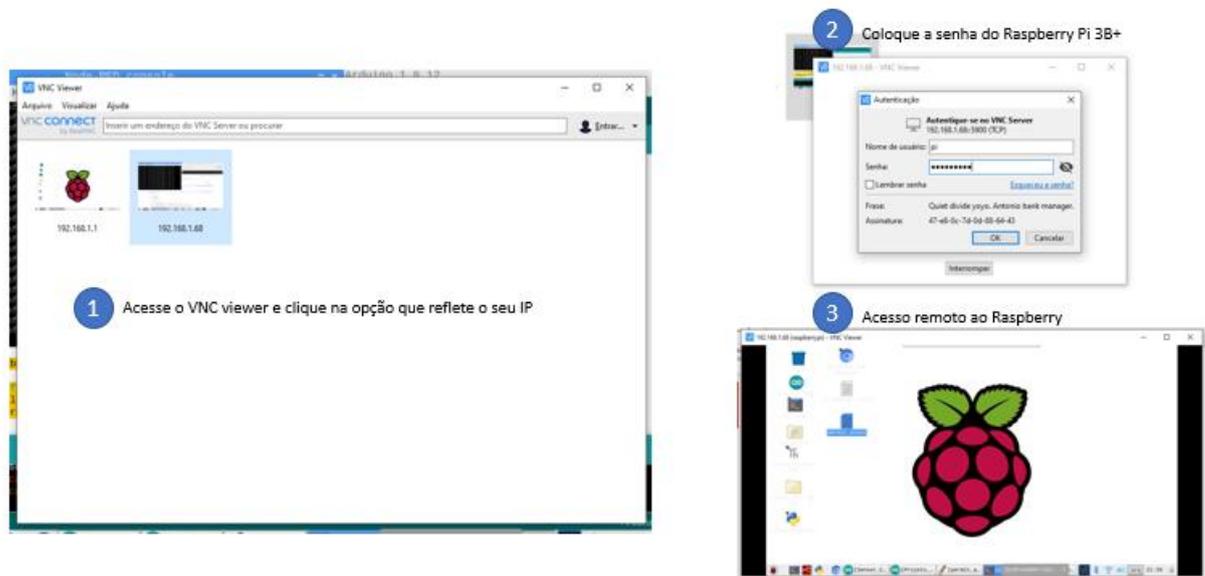
Uma vez ligado o sistema, o USBasp carrega o programa inicializador para o Arduino

Figura 79 – Conexão via cabo Ethernet com o Raspberry Pi 3B+



Fonte: Autoria própria (2021).

Figura 80 – Passos para ter o acesso remoto do Raspberry Pi 3B+

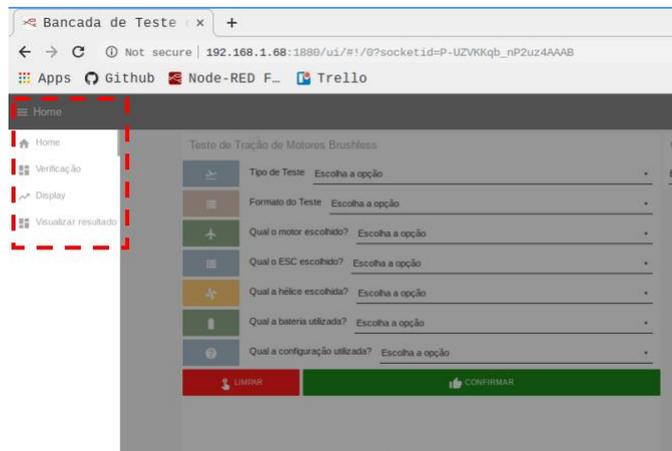


Fonte: Autoria própria (2021).

e os dados dos sensores já começam a ser coletados. Para ter acesso à interface do programa de testes é necessário inicializar o aplicativo do Node Red. Com a conexão feita, o usuário deverá iniciar o Node-RED para que seja criado o editor em ambiente *browser*. A ferramenta está configurada para gerar um ponto de IP no link <http://192.168.1.1:1880>, podendo ser acessado pelo Chromium Web Browser do Linux. O caminho <http://192.168.1.1:1880> é utilizado para o editor do Node-RED. Para ter acesso ao *Front-End* o caminho <http://192.168.1.1:1880/ui> é utilizado. A Figura 82 apresenta a tela ao acessar esse caminho. Caso a conexão com o Raspberry Pi 3B+ seja via Wi-Fi, deverá ser acessado o mesmo IP utilizado para entrar no VNC Viewer. A aba Home é a entrada do software, na qual o usuário informa os parâmetros de realização do teste. À esquerda da Figura 81 são apresentadas todas as abas do projeto: Home, Verificação,

Display e Visualizar Resultado. O usuário tem a disponibilidade de acessar qualquer uma das abas em qualquer momento do teste, no entanto a permissão à todas as funcionalidades depende de se certos critérios foram satisfeitos. Esses critérios são detalhados na sequência.

Figura 81 – Abas do Projeto



Fonte: Autoria própria (2021).

Na parte central da Figura 82 são mostrados os parâmetros de testes a serem escolhidas pelo usuário. Mais à direita está a opção “Carregar Teste”, na qual o usuário pode selecionar os parâmetros do teste anterior. O software avalia se a configuração escolhida possui problemas de consistência na escolha do grupo motopropulsor, avisando o usuário, mas não impedindo a realização do teste.

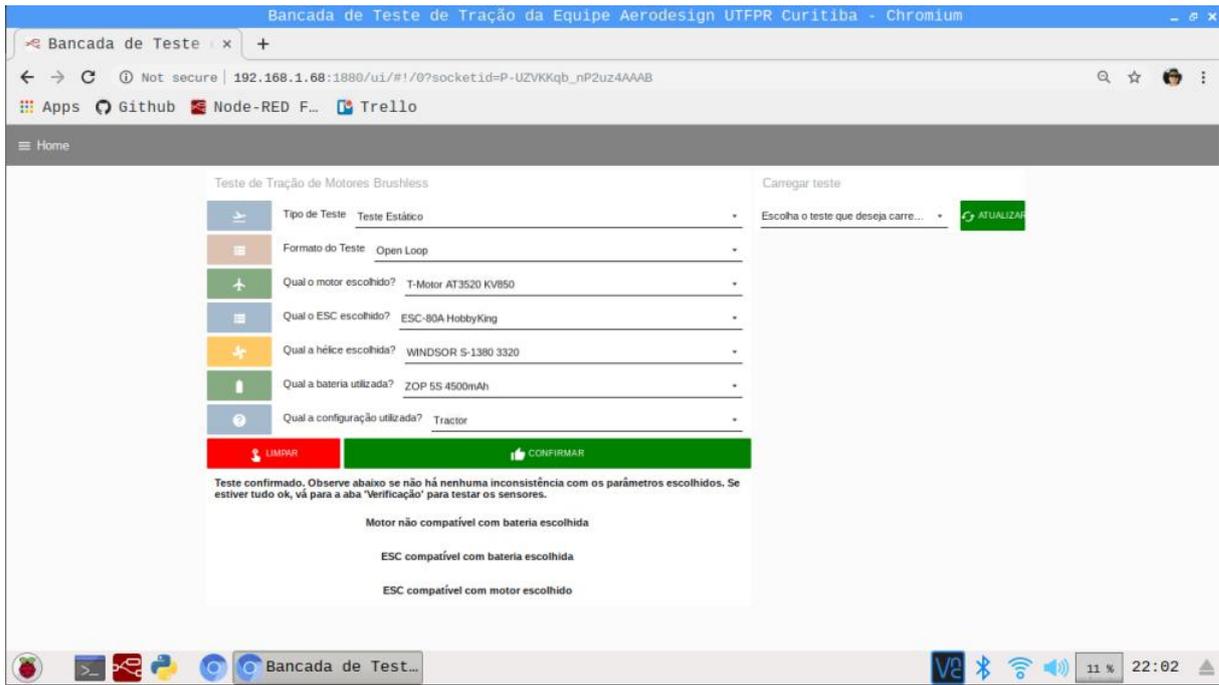
Caso falte a entrada de um ou mais parâmetros o usuário é avisado e o teste não é habilitado até o dado ser incluído. A falta de qualquer parâmetro influencia diretamente na realização do teste. A Figura 84 mostra a mensagem de erro apresentada ao usuário. Com essa etapa finalizada, o usuário segue para a verificação dos sensores.

Ao clicar no botão “Iniciar Verificação”, é iniciado um fluxo que obtém a leitura de todos os sensores e analisa se as leituras estão condizentes. Mais detalhes do funcionamento desse código podem ser encontrados na seção 5.2.3. Para essa verificação, tem-se três situações possíveis:

1. Todos os sensores estarem funcionando corretamente;
2. Um ou mais sensores não críticos não estarem funcionando corretamente;
3. Um ou mais sensores críticos não estarem funcionando corretamente.

Caso todos os sensores estejam funcionando ou como no segundo caso de sensores não críticos com problema, o usuário recebe uma mensagem de que nenhum sensor crítico está

Figura 82 – Acesso ao *Front-End* para realização dos Testes



Fonte: Autoria própria (2021).

Figura 83 – Aviso ao usuário pela falta de informação na entrada



Fonte: Autoria própria (2021).

falhado. Fica a critério do usuário em dar prosseguimento sem a correção dos sensores falhados. O terceiro caso leva em conta sensores críticos, sendo eles: sensor de tensão de bateria do motor, sensor de corrente do motor, sensor de temperatura da bateria e célula de carga. Para essa situação, é informado ao usuário que a correção do estado de falha desses sensores é necessária, senão o teste não pode começar.

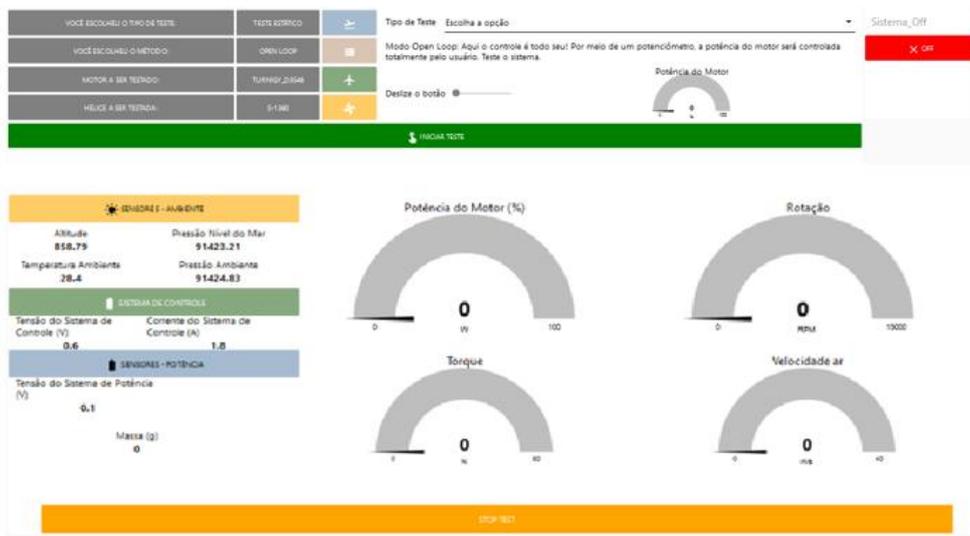
A tela de Display fornece os parâmetros do teste na parte superior esquerda e requer a escolha do Tipo de Teste na lateral direita. Na Figura 85 é demonstrada a interface gráfica da aba de “Display”, em caso do Modo Rotina ser o escolhido pelo usuário. Na Figura 86 é possível ver a tela de para o caso do teste ser *Open Loop*.

Figura 84 – Tipos de avisos para a verificação dos sensores



Fonte: Autoria própria (2021).

Figura 85 – Interface de realização dos testes - Modo *Open Loop*



Fonte: Autoria própria (2021).

Para iniciar o teste o usuário deve pressionar o botão “Iniciar Teste”. Neste momento, é feita a gravação dos parâmetros do teste em um arquivo CSV e salvo em uma pasta dentro do Raspberry Pi 3B+, como é possível ver na Figura 87.

Quando o teste finaliza, seja por meio do usuário pressionado o botão “Stop Test” ou pelo Modo Rotina, os valores dos sensores são armazenados em outro arquivo CSV gravado em outra pasta no Raspberry Pi 3B+, como é possível ver na Figura 88.

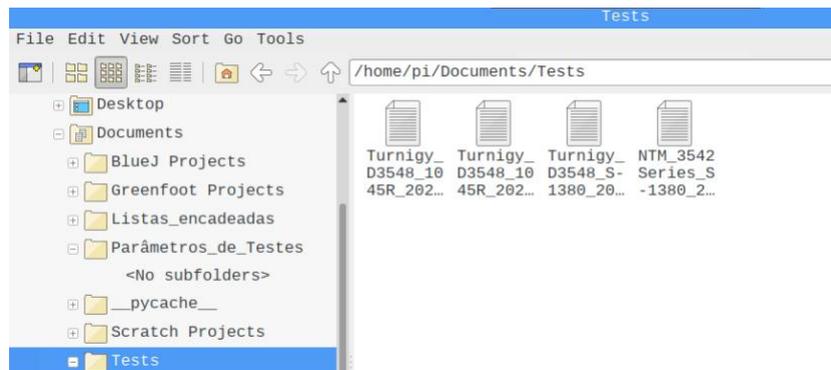
A Visualização dos dados é demonstrada na Figura 89. Uma lista suspensa com todos os testes já gravados no base de dados é fornecida ao usuário, como mostra a Figura 89. Escolhendo

Figura 86 – Interface de realização dos testes - Modo Rotina



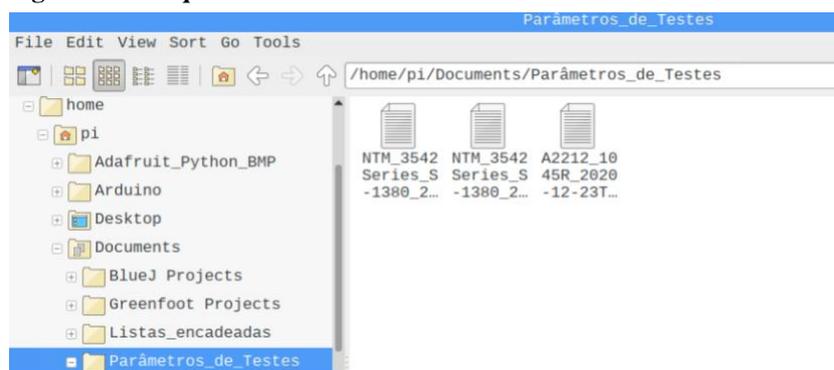
Fonte: Autoria própria (2021).

Figura 87 – Arquivos CSV contendo os parâmetros dos testes



Fonte: Autoria própria (2021).

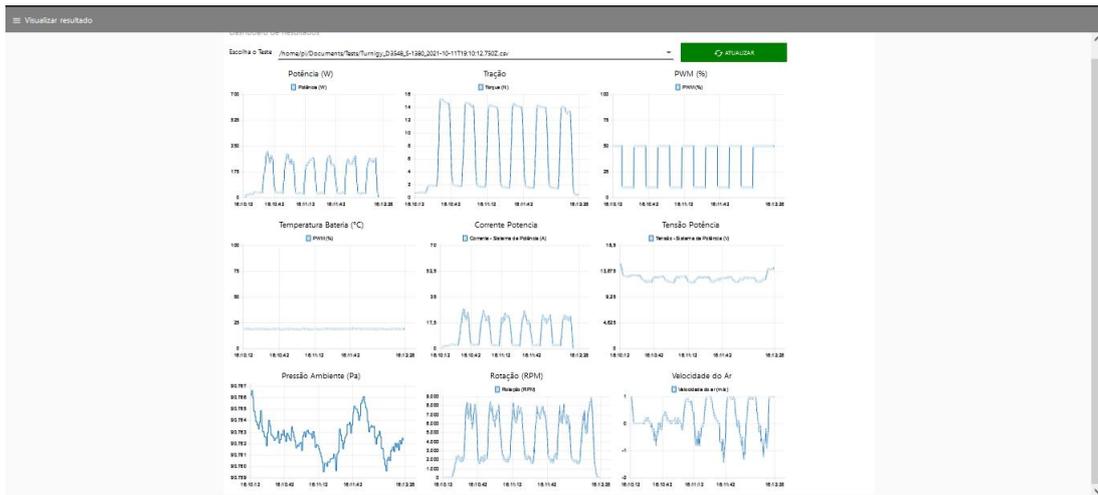
Figura 88 – Arquivos CSV contendo os valores dos sensores durante o teste



Fonte: Autoria própria (2021).

o teste desejado na lista, os valores são carregados e plotados.

Figura 89 – Dashboard de Visualização dos Dados



Fonte: Autoria própria (2021).

6.3 TESTES

Com a parte mecânica e elétrica montada, iniciou-se as coletas de dados dos testes propostos. Por limitações técnicas, não foi possível realizar a calibração dos sensores de velocidade do ar e de rotação. Os testes em ambiente externo — testes dinâmicos — não foram realizados por razão da pandemia e necessidade de mais integrantes da equipe no suporte à realização do teste. Além disso, as condições extremas de teste foram desconsideradas por não possuir o ambiente adequado de segurança, uma vez que grande parte dos testes foram feitos em ambiente domiciliar por razão da pandemia.

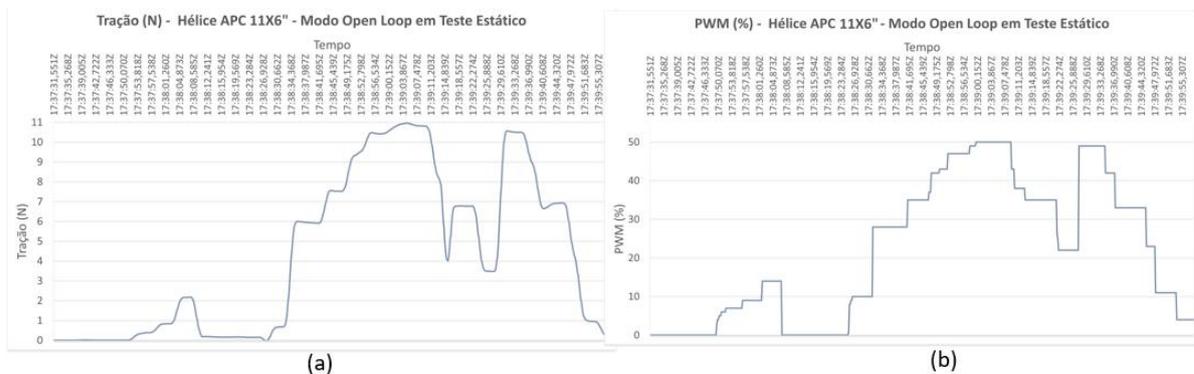
Foram realizados testes com duas hélices diferentes, a fim de verificar as diferenças de resposta de ambas. A primeira hélice foi da fabricante APC, de tamanho 11x6". A segunda hélice foi a Windsor S-1380, de tamanho 13x8". Cada conjunto motopropulsor passou pelos testes de *Open Loop* e pelos 3 modos de Rotina: degrau, pico e turbo. Os testes estáticos foram realizados conectando o sistema de controle na rede externa, portanto os valores de corrente da bateria de controle e tensão da bateria de controle são apresentados somente na seção de teste dinâmico. Além disso, no teste estático o intuito é realizar os testes em ambiente fechado sem influência do ar, não sendo necessária a coleta da velocidade do ar. Da mesma forma, esses valores podem ser encontrados na seção de Teste dinâmico.

6.3.1 Teste Estático com a Hélice APC 11x6

6.3.1.1 *Open Loop*

O teste de *Open Loop* foi feito variando o PWM aplicado na faixa entre 0 a 50%. A Figura 90 apresenta os resultados da tração e do PWM lado a lado. Importante notar como a tração segue diretamente as mudanças do PWM, por razão da maior tensão aplicada ao motor, consequentemente aumentando a sua rotação e a força de empuxo da hélice. O conjunto atingiu uma tração máxima de 10,96 N nesse teste.

Figura 90 – Resultados de (a) Tração e (b) PWM durante o teste de *Open Loop*



Fonte: Autoria própria (2021).

Diferentemente dos outros sensores, o sensor de rotação apresentou oscilações além do normal, apesar de girar em torno de rotações plausíveis. A Figura 91 (a), apresenta como a rotação tende a seguir também a curva de PWM. No entanto seus valores oscilaram demasiadamente. A Figura 91 (b) apresenta a potência elétrica do conjunto, que atingiu seu pico máximo de 239,82 W.

Figura 91 – Resultados de (a) Rotação e (b) Potência durante o teste de *Open Loop*

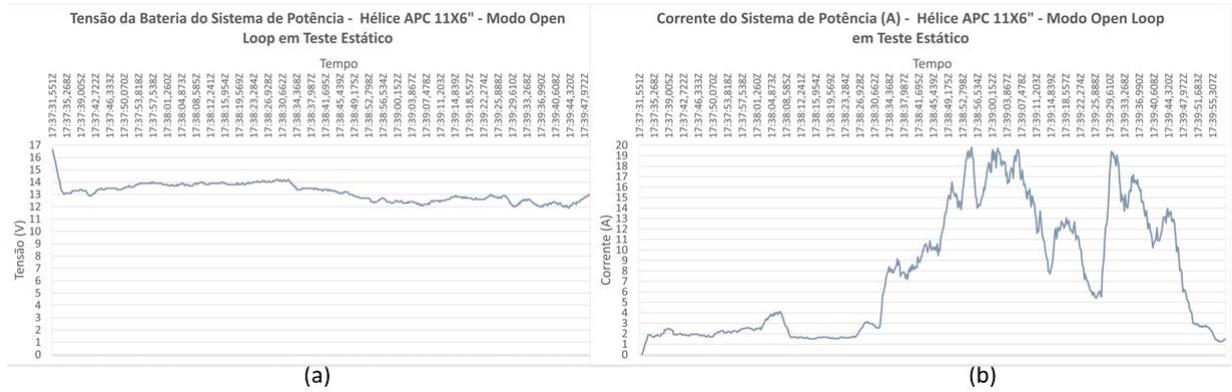


Fonte: Autoria própria (2021).

Os dados de potência são calculados multiplicando-se a tensão e corrente apresentados

na Figura 92. O gráfico está mais linear devido à taxa de impressão desses dados vindos do Node-RED, que multiplica instantaneamente os valores vindos da porta serial e plota em taxa menor que o restante dos dados.

Figura 92 – Resultados de (a) Tensão e (b) Corrente durante o teste de *Open Loop*

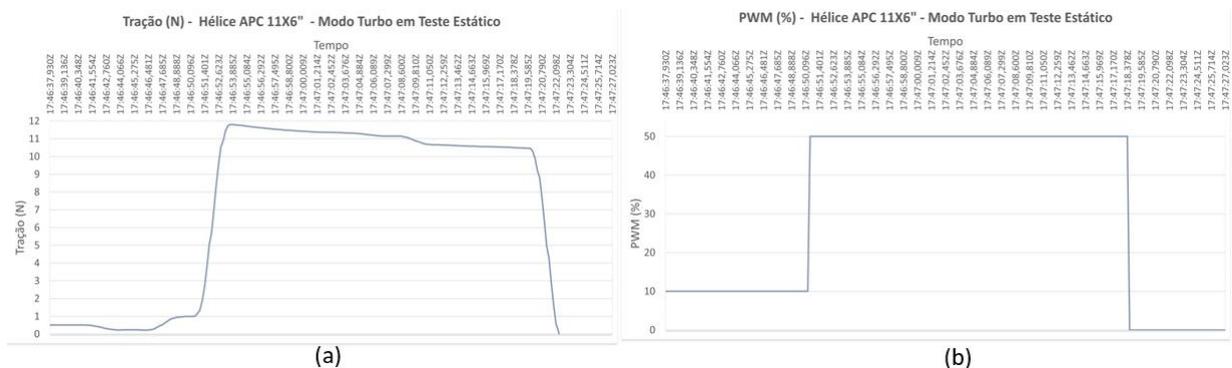


Fonte: Autoria própria (2021).

6.3.1.2 Modo Turbo

O modo Turbo foi projetado para alimentar o motor em potência máxima por 30 segundos, no entanto por limitações técnicas e de segurança, preferiu-se reduzir a potência em 50% de PWM. Essa alteração não modifica o intuito do teste, que visa verificar o comportamento do motor em situações contínuas de tração. Na Figura 93 é possível verificar a curva de tração e PWM lado a lado.

Figura 93 – Resultados de (a) Tração e (b) PWM durante o teste de Rotina - Modo Turbo



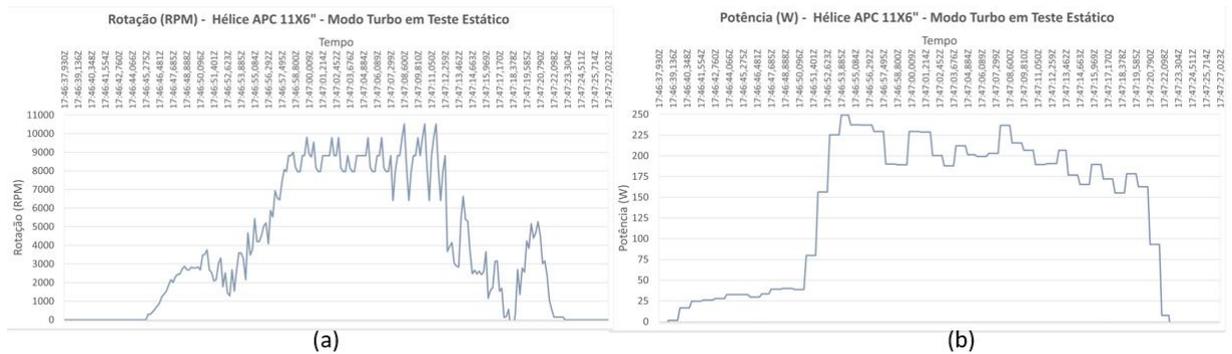
Fonte: Autoria própria (2021).

É verificado que a tração ao iniciar o teste atinge seu ponto máximo e reduz ao longo da realização do teste. Algumas hipóteses foram levantadas, tais como diminuição da carga

da bateria e também pelo aumento de fluxo turbulento ao longo da realização do teste, por consequência da movimentação da hélice em alta rotação e pelo ambiente ser fechado.

A corrente sofreu oscilações mesmo durante o teste mantendo o mesmo nível de PWM, valores esses que afetaram, consequentemente, a potência, dados que podem ser observados na Figuras 94 e 95.

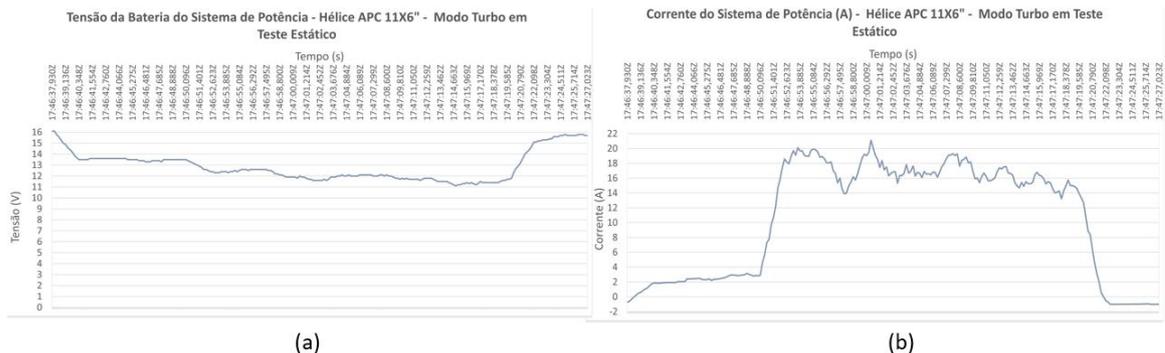
Figura 94 – Resultados de (a) Rotação e (b) Potência durante o teste de Rotina - Modo Turbo



Fonte: Autoria própria (2021).

A rotação apresentou valores não condizentes no período final do teste, reduzindo drasticamente e não acompanhando o resto dos dados. Durante o teste, foi possível observar oscilações no torque muito possivelmente por razão do ar turbulento do ambiente.

Figura 95 – Resultados de (a) Tensão e (b) Corrente durante o teste de Rotina - Modo Turbo

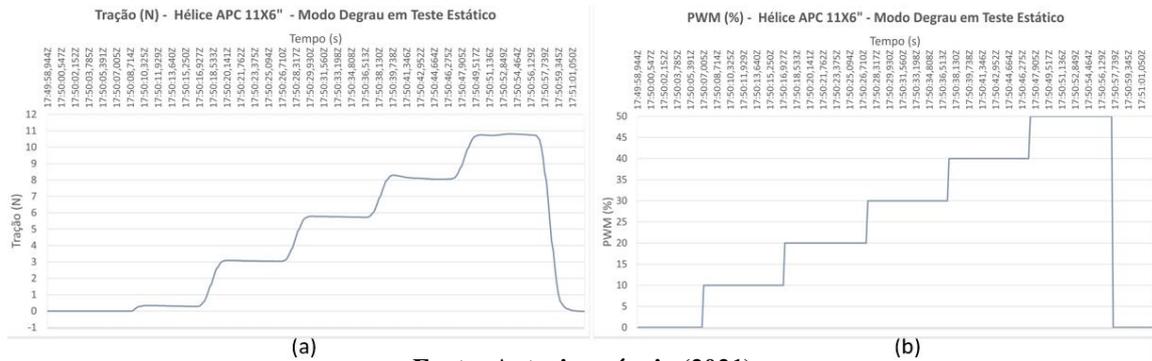


Fonte: Autoria própria (2021).

6.3.1.3 Modo Degrau

O modo Degrau foi projetado para aumentar o PWM de 10 em 10%, a cada 10 segundos, a potência distribuída ao motor. Pelas mesmas razões explicadas anteriormente, o valor máximo limitado no teste foi de 50%. Os gráficos são apresentados na Figuras 96, 97 e 98.

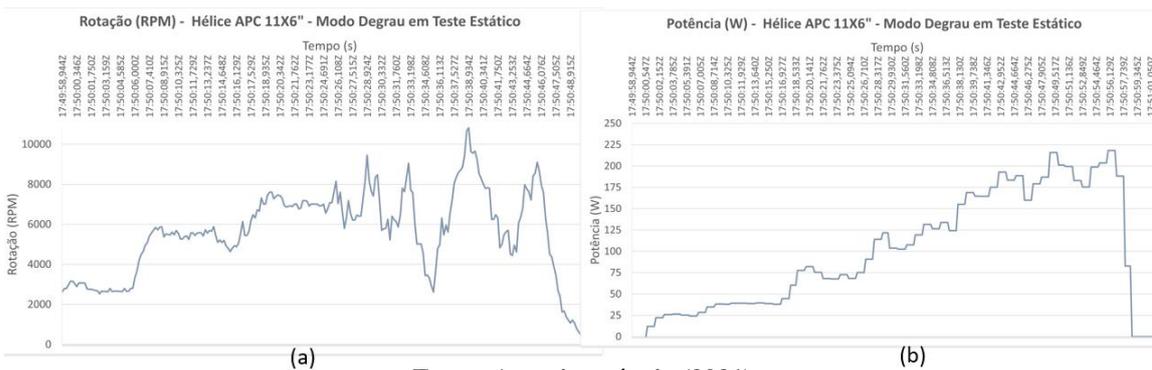
Figura 96 – Resultados de (a) Tração e (b) PWM durante o teste de Rotina - Modo Degrau



Fonte: Autoria própria (2021).

A tração segue precisamente os degraus de PWM aplicados no motor, mesmo efeito seguido nos outros gráficos, rotação não incluído.

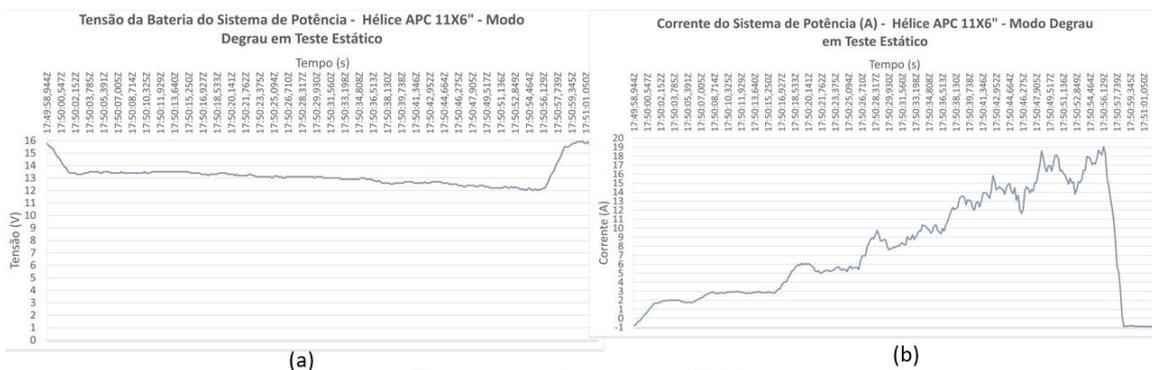
Figura 97 – Resultados de (a) Rotação e (b) Potência durante o teste de Rotina - Modo Degrau



Fonte: Autoria própria (2021).

Os dados tenderam a sequência de subida de rotação, no entanto não foi possível de fazer o levantamento preciso de seus dados para cada valor do PWM aplicado. Interessante observar a queda de tensão de aproximadamente 2 V ao iniciar o teste, devido à alimentação do motor e posteriormente, a elevação da tensão pelo desligamento do motor.

Figura 98 – Resultados de (a) Tensão e (b) Corrente durante o teste de Rotina - Modo Degrau

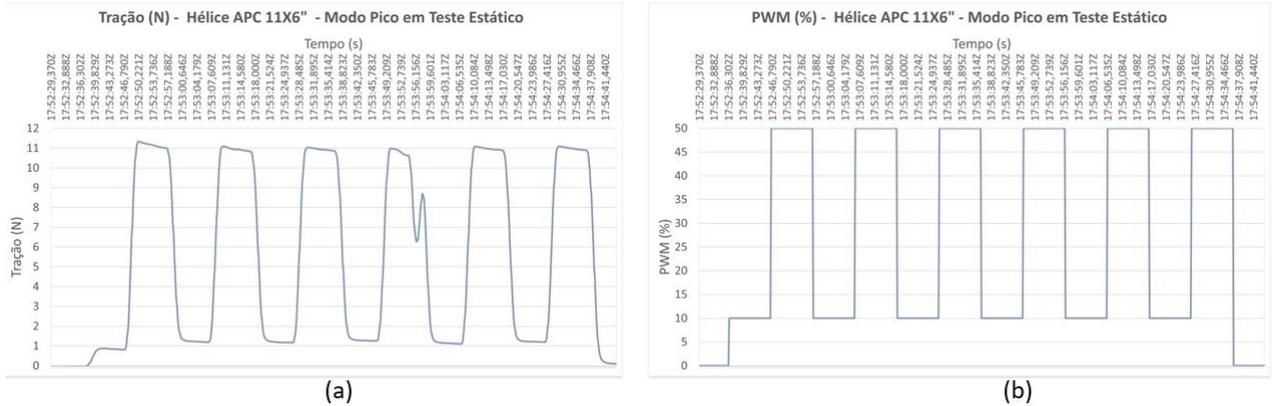


Fonte: Autoria própria (2021).

6.3.1.4 Modo Pico

O modo Pico, projetado para dar impulsos de potência no motor, fornecendo tal pico em 100% de PWM. No entanto, pelas mesmas razões já explicadas nos testes anteriores, escolheu-se reduzir o pico para 50%. Os gráficos da tração e PWM podem ser vistos na Figura 99.

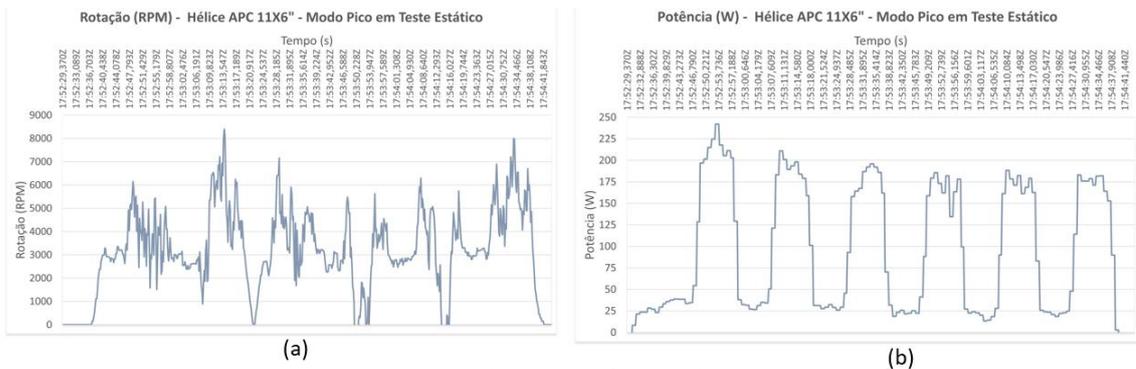
Figura 99 – Resultados de (a) Tração e (b) PWM durante o teste de Rotina - Modo Pico



Fonte: Autoria própria (2021).

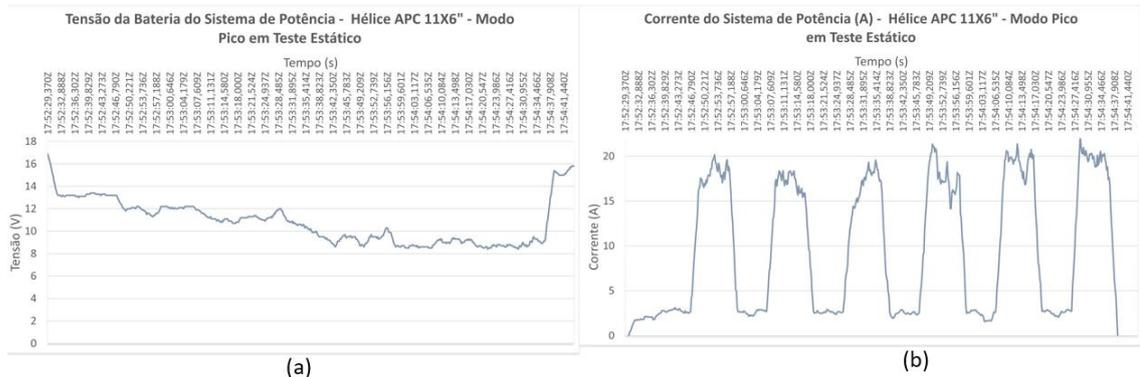
Da mesma forma que em testes anteriores, a tração seguiu o formato da curva de PWM. Durante a quarta curva, é possível observar um afundamento da tração mesmo com o PWM se mantendo o mesmo. Apesar disso, não foi observada uma queda evidente nos valores de corrente e de tensão, apresentados na Figura 101. Portanto a oscilação pode ter sido causada por uma falha pontual na célula de carga. A medição de rotação apresentou oscilação durante o teste, mas durante os períodos nos quais o motor ficou em PWM de 10%, é possível observar no gráfico que a rotação seguiu a mesma tendência, conforme a Figura 100.

Figura 100 – Resultados de (a) Rotação e (b) Potência durante o teste de Rotina - Modo Pico



Fonte: Autoria própria (2021).

Figura 101 – Resultados de (a) Tensão e (b) Corrente durante o teste de Rotina - Modo Pico



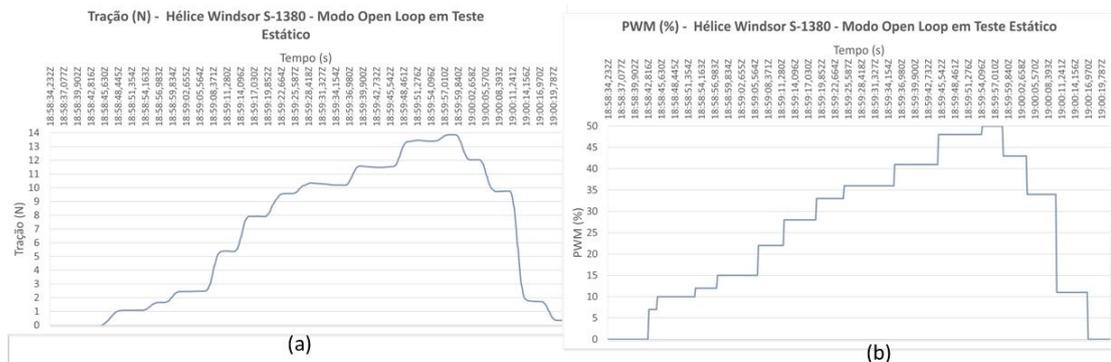
Fonte: Autoria própria (2021).

6.3.2 Teste Estático com a Hélice WINDSOR S-1380

6.3.2.1 *Open Loop*

Para o teste de *Open Loop* desse conjunto motopropulsor, também se elevou o PWM até um nível máximo de 50%. Como é possível verificar na Figura 102, a curva de tração seguiu diretamente os valores de PWM.

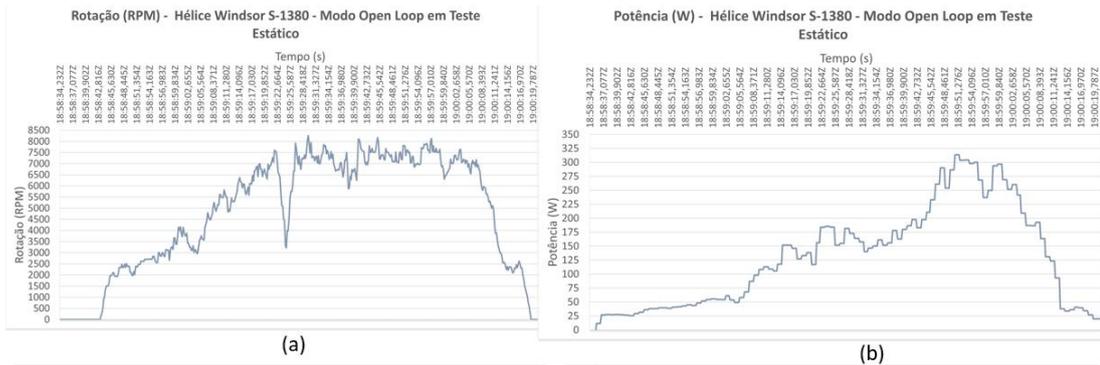
Figura 102 – Resultados de (a) Tração e (b) PWM durante o teste de *Open Loop*



Fonte: Autoria própria (2021).

A rotação teve uma resposta mais adequada em comparação com a hélice APC 11x6", possuindo oscilações menores. Durante o teste é possível verificar pela Figura 103, que uma queda brusca no valor de rotação aconteceu no meio do teste, mas que foi rapidamente recuperada após alguns segundos.

Figura 103 – Resultados de (a) Rotação e (b) Potência durante o teste de *Open Loop*



Fonte: Autoria própria (2021).

A corrente e tensão do sistema de potência podem ser visualizados na Figura 104.

Figura 104 – Resultados de (a) Tensão e (b) Corrente durante o teste de *Open Loop*

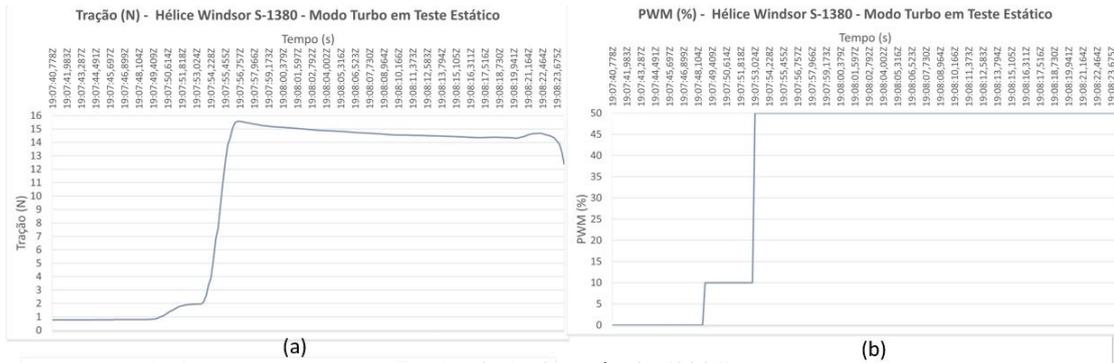


Fonte: Autoria própria (2021).

6.3.2.2 Modo Turbo

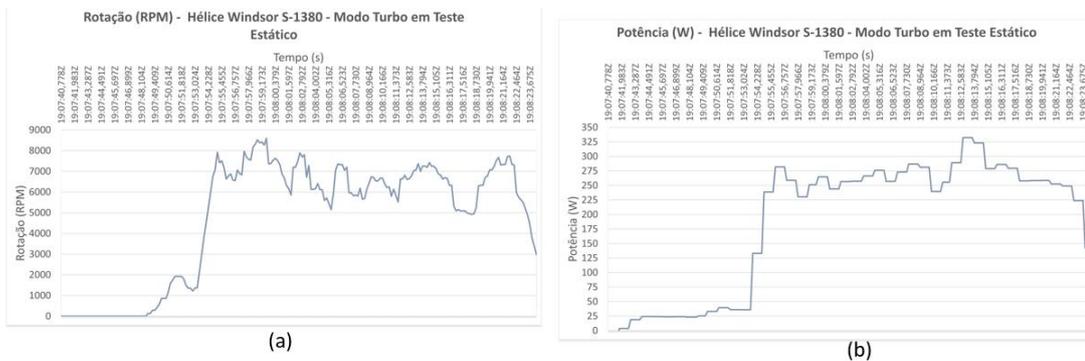
Da mesma forma que o teste para a hélice 11x6", o teste em modo Turbo foi alterado para ter seu valor máximo fixo em 50%. É possível verificar que a corrente se manteve na faixa de 20 a 24 A durante todo o teste, possuindo uma variação mais brusca no terço final. No entanto, a variação na corrente não afetou a tração disposta na Figura 105. A potência, juntamente da rotação, pode ser vista na Figura 106. A tensão e corrente são apresentadas na Figura 107.

Figura 105 – Resultados de (a) Tração e (b) PWM durante o teste de Rotina - Modo Turbo



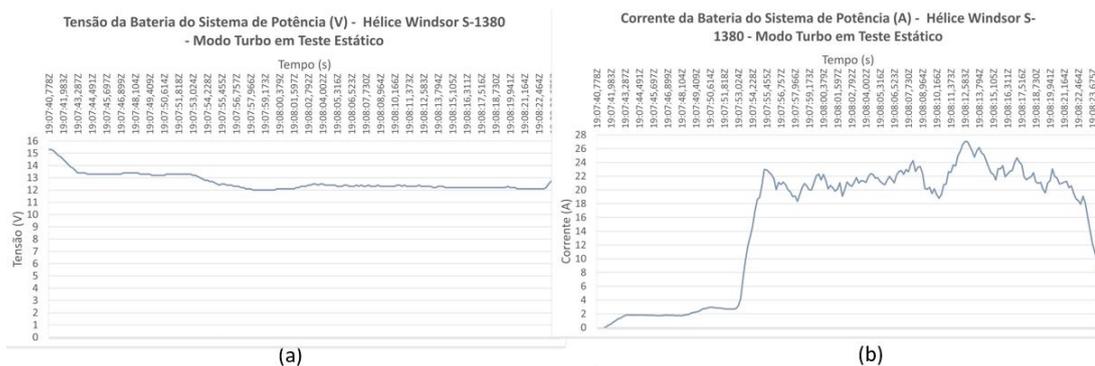
Fonte: Autoria própria (2021).

Figura 106 – Resultados de (a) Rotação e (b) Potência durante o teste de Rotina - Modo Turbo



Fonte: Autoria própria (2021).

Figura 107 – Resultados de (a) Tensão e (b) Corrente durante o teste de Rotina - Modo Turbo



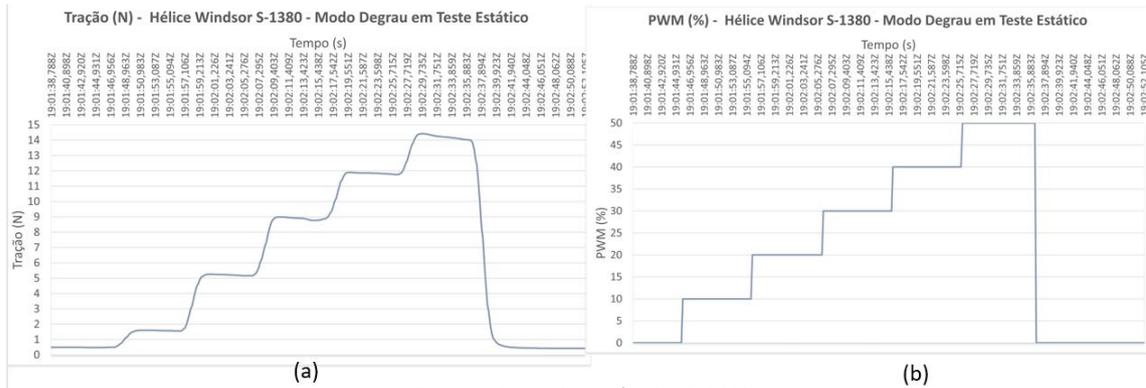
Fonte: Autoria própria (2021).

6.3.2.3 Modo Degrau

Da mesma forma que o teste feito para a hélice APC 11x6", o degrau foi limitado em 50%. Especificamente para esse caso, o usuário pressionou o botão "Parar Teste" alguns segundos após a finalização do ciclo, então é possível visualizar o comportamento dos sensores durante esse

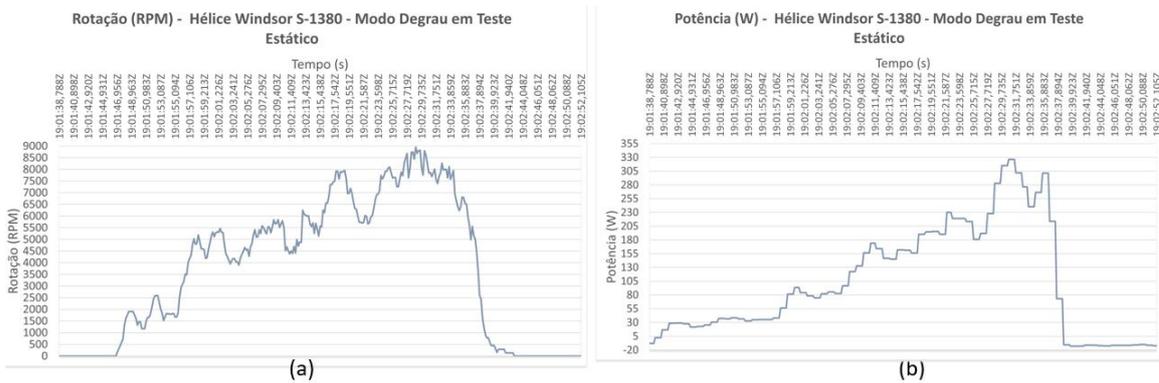
período. A tensão, apresentada na Figura 110, mostra sua evidente subida após o desligamento do motor. A rotação apresentou resultados satisfatórios nesse teste, sendo encontrados na Figura 109.

Figura 108 – Resultados de (a) Tração e (b) PWM durante o teste de Rotina - Modo Degrau



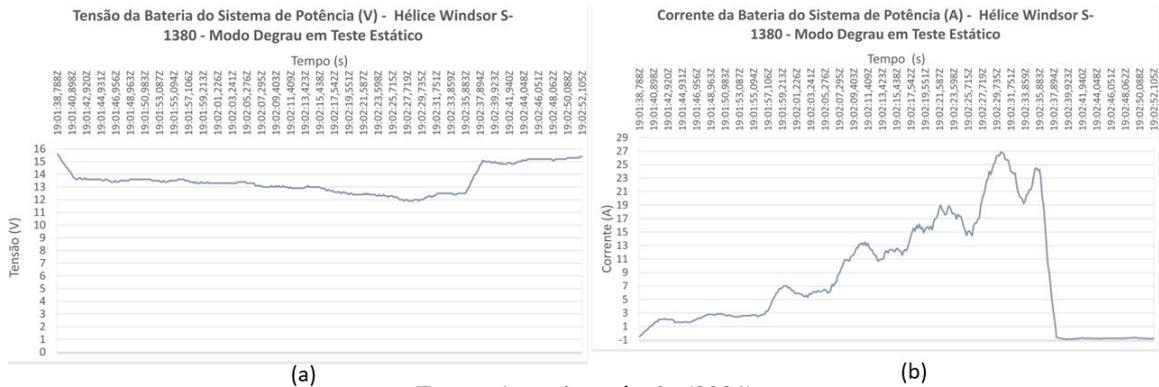
Fonte: Autoria própria (2021).

Figura 109 – Resultados de (a) Rotação e (b) Potência durante o teste de Rotina - Modo Degrau



Fonte: Autoria própria (2021).

Figura 110 – Resultados de (a) Tensão e (b) Corrente durante o teste de Rotina - Modo Degrau

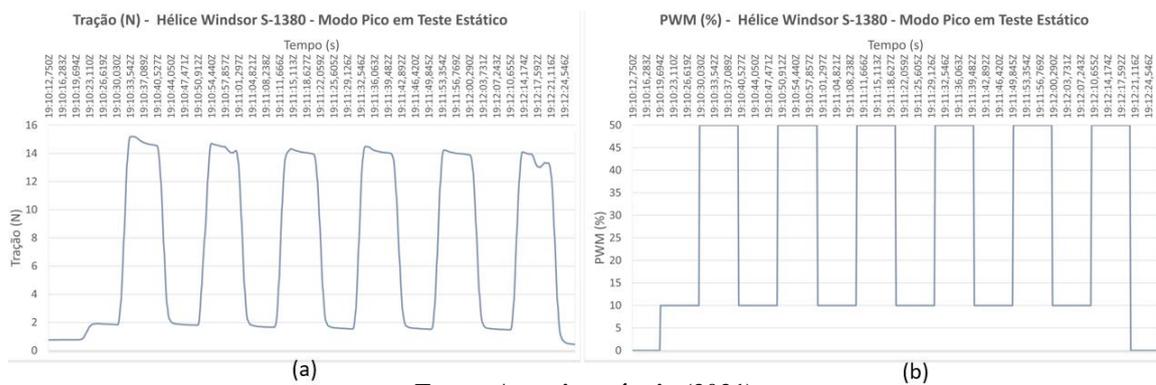


Fonte: Autoria própria (2021).

6.3.2.4 Modo Pico

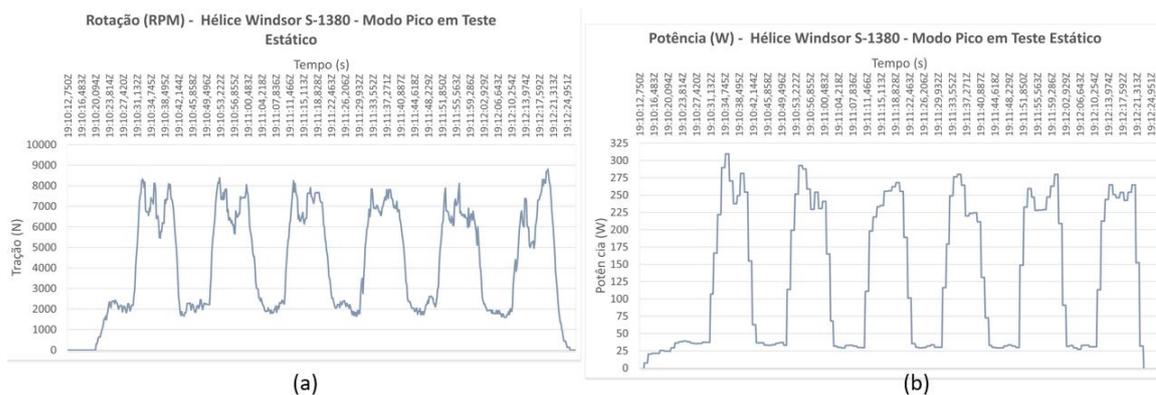
Da mesma forma que o teste realizado para a hélice 11x6", o modo Pico foi reduzido para um PWM máximo de 50 %. Os resultados de rotação, apresentados na Figura 112, apresentam resultados muito satisfatórios, assim como o restante dos sensores. Uma possível razão da melhora nos resultados se deve à hélice Windsor S-1380 ser dedicada à motores elétricos, enquanto a 11x6" foi construída para motores a combustão.

Figura 111 – Resultados de (a) Tração e (b) PWM durante o teste de Rotina - Modo Pico



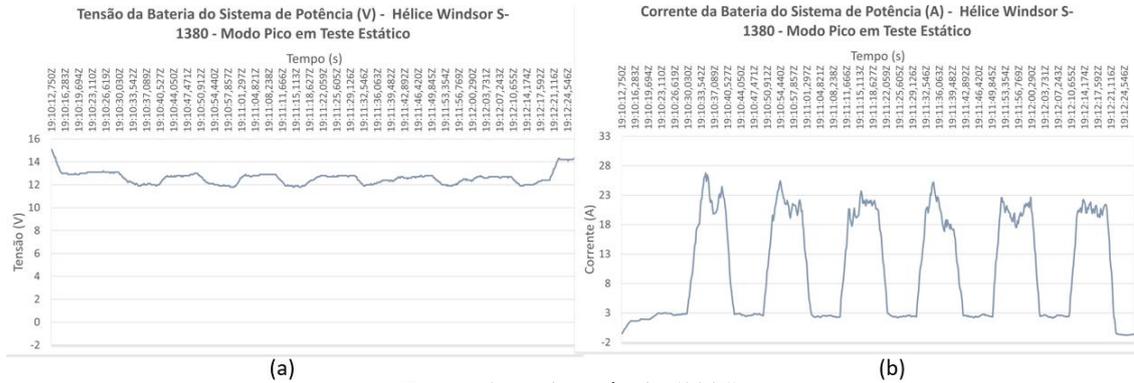
Fonte: Autoria própria (2021).

Figura 112 – Resultados de (a) Rotação e (b) Potência durante o teste de Rotina - Modo Pico



Fonte: Autoria própria (2021).

Figura 113 – Resultados de (a) Tensão e (b) Corrente durante o teste de Rotina - Modo Pico



Fonte: Autoria própria (2021).

6.4 TESTE DINÂMICO COM A HÉLICE WINDSOR S-1380

Para validar a bancada para testes dinâmicos, realizou-se um teste em potência reduzida com a bancada no topo do veículo, somente no modo *Open Loop*. O teste foi feito aumentando-se gradativamente a velocidade do veículo para verificar a influência do vento na tração do conjunto motopropulsor. Além disso, foi possível verificar o comportamento do sistema funcionando totalmente embarcado, uma vez que os testes estáticos foram feitos com o sistema sendo alimentado pela rede elétrica via tomada. As Figuras 114 e 115 mostram a montagem da bancada no topo do veículo.

Figura 114 – Montagem da bancada no topo do veículo para testes dinâmicos



Fonte: Autoria própria (2021).

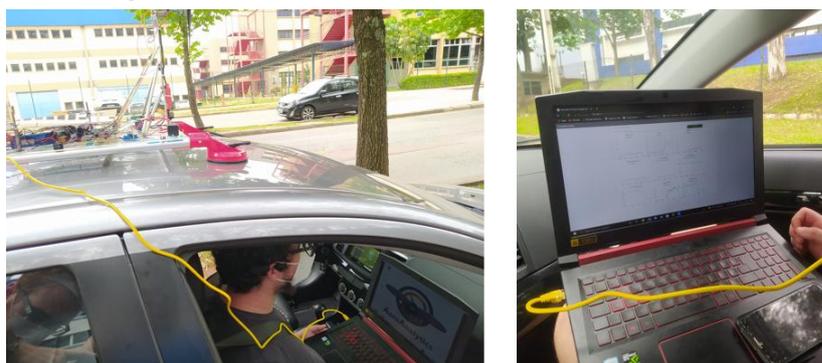
Figura 115 – Vista lateral da montagem da bancada no topo do veículo para testes dinâmicos



Fonte: Autoria própria (2021).

A comunicação à distância com a bancada via cabo Ethernet se mostrou satisfatória, enviando e recebendo dados instantaneamente da placa controladora principal. Após a realização do teste foi possível verificar e validar os dados utilizando a aba “Visualizar Resultado”. Na Figura 116 é possível ver a conexão via cabo Ethernet com a bancada e a tela de Visualização de Dados.

Figura 116 – Conexão com a bancada e leitura dos dados

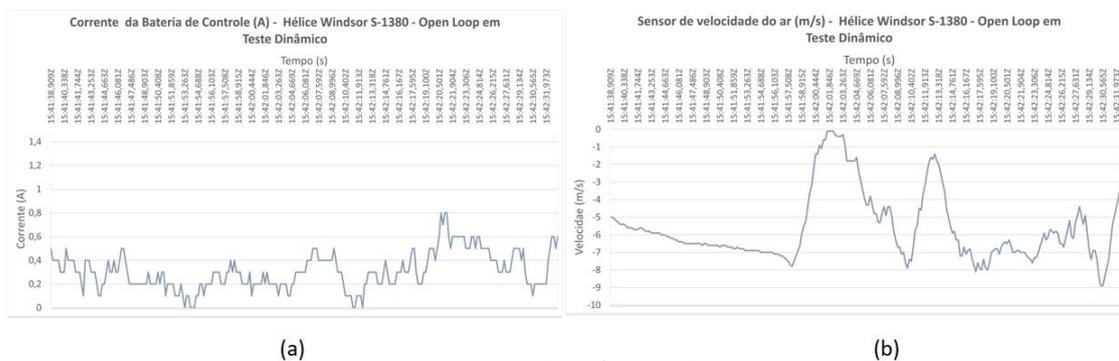


Fonte: Autoria própria (2021).

Foi possível observar que o sensor de velocidade do ar não possui, no momento, a calibração mais adequada. Apesar do movimento do veículo, o sensor não forneceu dados adequados. Isso se deve, também, à variação excessiva do vento que estava se movimentando a lés-sudeste, valor esse coletado pelo sensor, contabilizando velocidade negativa. A corrente e

tensão do sistema de controle se mostraram constantes durante toda a realização do teste. Os resultados são apresentados na Figura 117.

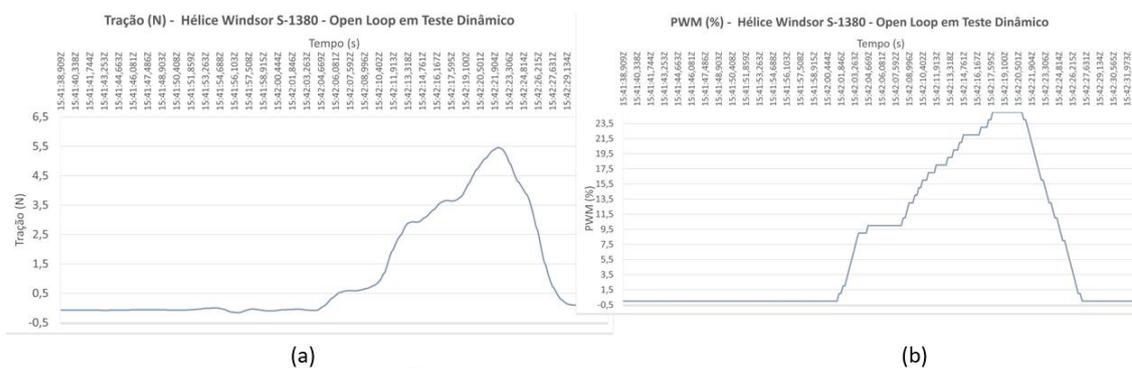
Figura 117 – Resultados da (a) Corrente do Sistema de controle e (b) Velocidade do ar durante o teste de *Open Loop* - Teste Dinâmico



(a) (b) Fonte: Autoria própria (2021).

Uma das preocupações da realização dos testes dinâmicos seria a vibração dos componentes da bancada, principalmente do suporte do sensor de rotação preso próximo à hélice do motor. No entanto, os dados de rotação se mostraram constantes e condizentes com as características do motor. Apesar da ação do vento, para um PWM de 25 % a tração possuiu valores muito parecidos com os do teste estático, com um valor de 5,45 N.

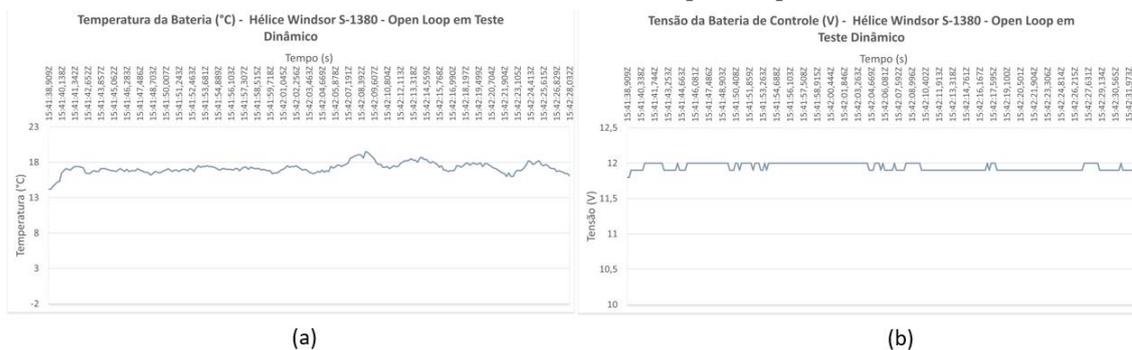
Figura 118 – Resultados de (a) Tração e (b) PWM durante o teste de *Open Loop* - Teste Dinâmico



(a) (b) Fonte: Autoria própria (2021).

Sugere-se em avaliações futuras a realização de teste em dias menos turbulentos de vento, em que somente o movimento do veículo iria influenciar a leitura do sensor de velocidade do ar. Observou-se um aumento da temperatura durante o início do teste, de aproximadamente 3°C, como é possível ver na Figura 119.

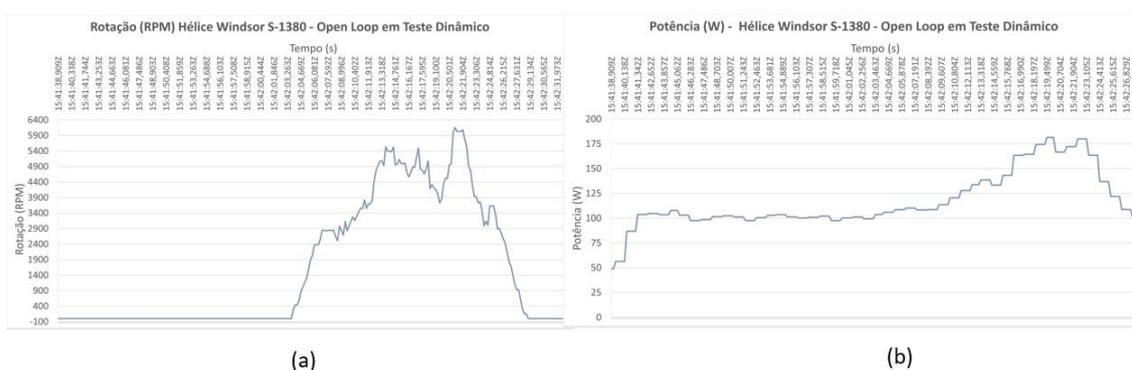
Figura 119 – Resultados de (a) Temperatura da Bateria e (b) Tensão do Sistema de Controle durante o teste de *Open Loop* - Teste Dinâmico



Fonte: Autoria própria (2021).

Na Figura 120 é apresentado os dados de Rotação e Potência durante o teste. Interessante notar que mesmo com a movimentação da bancada a leitura de rotação não foi afetada, demonstrando que o sistema de fixação do sensor de rotação foi satisfatório.

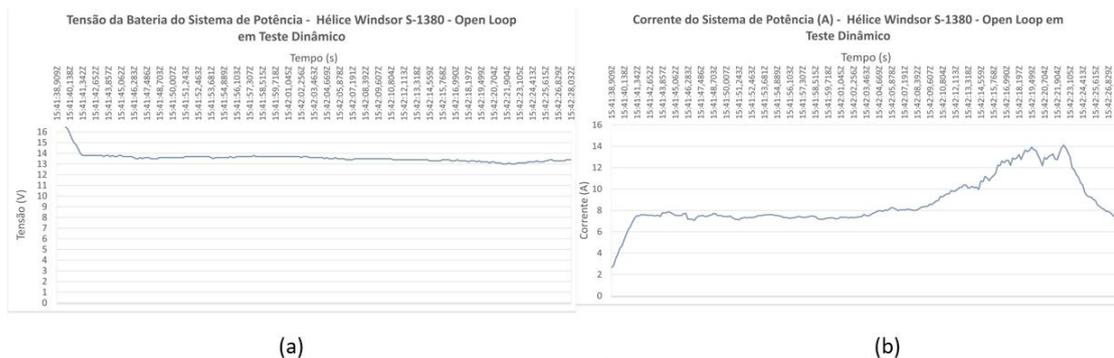
Figura 120 – Resultados de (a) Rotação e (b) Potência durante o teste de *Open Loop* - Teste Dinâmico



Fonte: Autoria própria (2021).

Os dados de tensão do sistema de potência e corrente do sistema de potência podem ser vistos na Figura 121.

Figura 121 – Resultados de (a) Tensão e (b) Corrente durante o teste de *Open Loop* - Teste Dinâmico



Fonte: Autoria própria (2021).

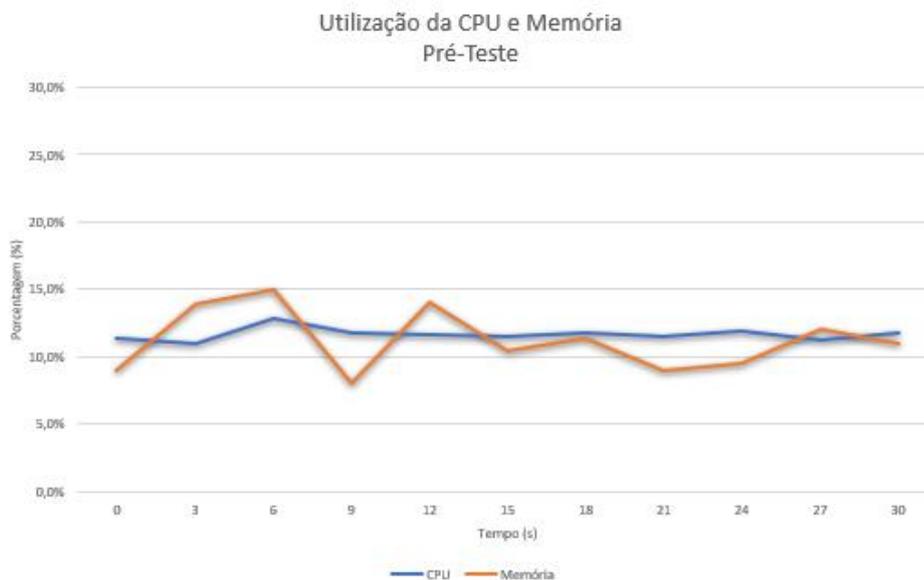
6.5 DESEMPENHO DO SOFTWARE

Foram coletadas as porcentagens de utilização de CPU e Memória durante 3 períodos de utilização da bancada:

- Pré-Teste: O sistema é ligado e nenhum programa ou arquivo é executado;
- Iniciando Node-RED: O programa Node-RED é executado;
- Durante Teste: A interface gráfica é executada e o teste realizado.

Para a coleta desses valores foi utilizado o comando *top* no *prompt* de comando, que fornece todas as rotinas que consomem memória e CPU da placa controladora principal. Após a inicialização do Raspberry Pi 3B+ a CPU oscila pouco, permanecendo entre 12-13% durante esse período, enquanto a memória oscila um pouco mais, mas permanecendo majoritariamente entre 10-15%. A Figura 122 mostra os valores coletados.

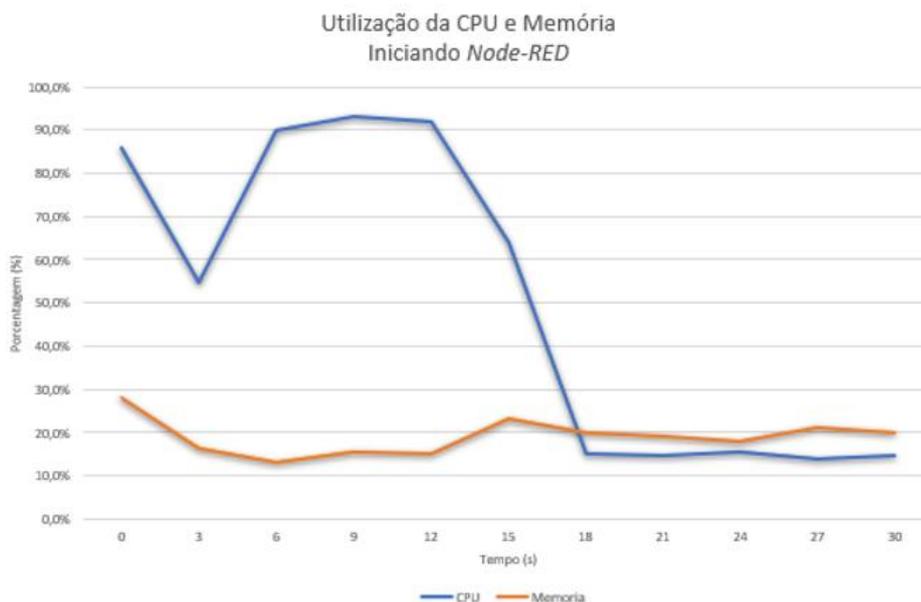
Figura 122 – Utilização da CPU e Memória - Pré-Teste



Fonte: Autoria própria (2021).

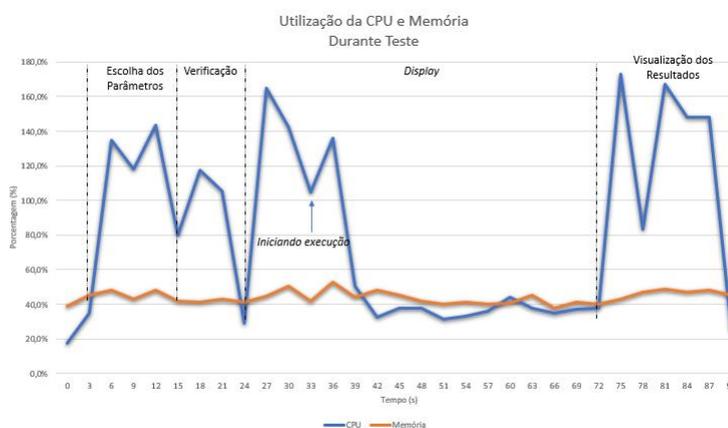
Iniciando o Node-RED, a CPU tem sua porcentagem de utilização atingindo o 90%, ficando nessa faixa por 6 segundos, como é possível ver na Figura 123. O tempo de inicialização do Node-RED dura 18 segundos, até o momento que o sistema retorna para o regime permanente. É possível analisar que com a simples execução do software, a memória sobe para 20%.

Ao iniciar o Chromium Web Browser para acessar a interface do usuário, é notado um grande aumento da utilização da CPU. Isso se deve ao carregamento inicial da página. Portanto,

Figura 123 – Utilização da CPU e Memória - Iniciando Node-RED

Fonte: Autoria própria (2021).

toda vez que é alternado entre as abas, a utilização da CPU sobe bruscamente, mas logo depois decai com a página carregada. O maior exemplo disso é durante a etapa "Display", apresentado na Figura 124. A página é carregada e o processamento sobe para 162 %.

Figura 124 – Utilização da CPU e Memória - Durante o teste

Fonte: Autoria própria (2021).

Com a inicialização do teste — usuário pressionando o botão 'Iniciar Teste' — há novamente um pico de 140%, mas logo depois o processamento cai para a faixa de 40% e permanece nesse valor até o momento que o teste é finalizado e a aba de 'Visualização dos Resultados' é carregada.

O consumo de corrente do sistema com o teste em andamento foi medido. O valor coletado foi de 250 mA na saída da bateria do sistema de controle. Como a bateria do sistema de controle tem ampacidade de 2500 mAh, o sistema tem autonomia de 10 horas.

7 CONCLUSÕES E CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo o desenvolvimento de uma bancada de tração para motores elétricos do tipo *Brushless* CC a serem aplicados no desenvolvimento de projetos pela equipe AeroDesign UTFPR Curitiba. O sistema foi desenvolvido para testes estáticos e dinâmicos, a fim de fornecer todos os dados necessários para a escolha do grupo motopropulsor. Utilizou-se a placa controladora Raspberry Pi 3B+ para a criação do *Front-End* por meio do software Node-RED e o Arduino para coleta de dados.

O projeto se mostrou condizente com as propostas dos objetivos gerais e específicos. Uma das prioridades era tornar o projeto seguro para o usuário. Diversas etapas do software foram implementadas com o objetivo de reduzir o risco com o uso da bancada. A etapa de análise dos parâmetros do conjunto propulsor, inseridos pelo usuário, que avaliam se os dispositivos escolhidos são compatíveis, foi aprovada pelos integrantes. A incompatibilidade dos equipamentos impossibilita o teste de acontecer, e essa limitação foi aprovada pela equipe. Outro objetivo específico do projeto foi de facilidade de utilização. Para validar tal objetivo, ao final do projeto, todo o sistema foi deixado nas mãos de integrantes com pouco conhecimento no funcionamento da bancada e reportaram como fácil o manuseio de todo o sistema. As diversas etapas de validação do software facilitaram no desenvolvimento de um sistema confiável para coleta de dados. Cada sensor foi avaliado individualmente diversas vezes durante o projeto, sendo ainda fornecido para o usuário uma etapa de verificação dos sensores a fim de informar se um sensor está com leituras inconsistentes.

Dificuldades foram constantes durante a realização do projeto. O projeto foi todo desenvolvido durante o período da pandemia de COVID-19, o que dificultou o acesso à faculdade, impossibilitando melhores calibrações dos sensores contidos no projeto. Como consequência da pandemia de COVID-19, houve um aumento excessivo de preço de componentes o que acabou encarecendo o projeto. O custo para o desenvolvimento do projeto é apresentado em mais detalhes no Apêndice F. Para superar o problema de falta de contato com os integrantes da equipe durante o projeto, foram realizadas reuniões à distância a fim de que todos os integrantes da equipe do AeroDesign estivessem alinhados com o objetivo do projeto e se concordavam com os caminhos tomados. Outro grande obstáculo foi o local de realização dos testes, que por muitas vezes foi o próprio quarto do proponente por restrição de outros locais. Felizmente, no período final do projeto, houve a possibilidade do retorno à faculdade onde contou-se com a ajuda de

dois integrantes da equipe para a execução dos testes dinâmicos.

A calibração do sensor de corrente foi um grande obstáculo devido à sensibilidade de leitura. A utilização do software Node-RED foi muito desafiadora por existir pouco conhecimento disponível na internet, mas ao mesmo tempo foi útil, pois apresentou o grande potencial que a ferramenta possui. A montagem da bancada passou por diversas iterações desde a sua versão inicial até a final. Durante o desenvolvimento, a bancada foi colocada em diferentes modos de falha para verificação da confiabilidade, passando assim por diversas mudanças, sempre buscando a maior segurança para o usuário.

Como sugestão de trabalhos futuros, pode-se citar:

- Implementação do banco de dados em SQL - Atualmente os dados são salvos em uma pasta interna no Raspberry Pi 3B+, que se mostrou pouco eficiente em ambientes de teste. Uma sugestão é a implementação de um banco SQL para facilitar o acesso a esses dados;
- Placa de Circuito Impresso - A bancada atual tem seu sistema elétrico montado sobre uma chapa de acrílico de 4 mm, sendo utilizados muitos cabos para as conexões. Uma possibilidade é a substituição de todo esse circuito por uma placa de circuito impresso;
- Sistema de Login - Ao inicializar o Node-RED, é gerado um link de IP para acesso remoto à interface do aplicativo. No entanto, todos os dispositivos que estiverem na mesma rede podem ter acesso a esse IP e alterar os dados do teste, o que pode torná-lo inseguro. Portanto, uma sugestão seria bloquear esse acesso ao link de IP possibilitando o acesso somente a quem tiver o login e senha;
- Sistema de Gerenciamento de Bateria - O sistema atual conta com duas baterias, uma responsável pelo sistema de controle e outra pelo sistema de potência. O usuário tem informação da carga da bateria somente quando inicializa o sistema e abre a tela de User Interface do Node-RED. Uma sugestão para trabalhos futuros seria montar um sistema de gerenciamento de bateria, informando o usuário do status atual da bateria, para deixar o teste ainda mais seguro;
- Calibração do Sensor de Rotação e de Velocidade do Ar - Por limitações técnicas, não foi possível realizar a calibração precisa de ambos os sensores. Para o sensor de rotação, é necessário um tacômetro que meça com alta precisão a rotação do motor. Para o Tubo de

Pitot, é necessário um ambiente controlado onde seja possível saber a velocidade de ar para comparar com as leituras do sensor.

REFERÊNCIAS

AERODESIGN, SAE Brasil. **22° Competição SAE Brasil Aerodesign 2020**. 2020. "Acesso em: 09 jul. 2020.". Disponível em: http://saebrasil.org.br/2020/Aero2020/Regulamento_SAE_BRASIL_AeroDesign_2020_Rev02.pdf.

AERODESIGN, SAE Brasil. 22° competição sae brasil aerodesign 2020 - motorização elétrica. 2020.

AL-FEDAGHI, S. On a flow-based paradigm in modeling and programming. **International Journal of Advanced Computer Science and Applications**, v. 6, n. 06, 2015.

ALEGRO MICROSYSTEMS. **Fully integrated, Hall effect-based linear current sensor with 2.1 kVRMS voltage isolation and a low-resistance current conductor**. [S.l.], 2007.

ALLEGRO MICROSYSTEMS. **Sensitive Hall-Effect switches for high-temperature operation**. [S.l.], 2002.

ALLEGRO MICROSYSTEMS. **Thermally enhanced, fully integrated, Hall-effect-based linear current sensor IC with 100 current conductor**. [S.l.], 2020.

AMAZON. **Degraw 5kg load cell and HX711 combo pack kit - load cell amplifier ADC weight sensor for Arduino scale - everything needed for accurate force measurement**. 2020. Acesso em: 09 out 2020. Disponível em: <https://www.amazon.com/Degraw-Load-Cell-HX711-Combo/dp/B075317R45>.

AMBONI, O. R.; CURI, E.I.M. Desenvolvimento de uma bancada para ensaios de um motor a combustão utilizado em aeromodelo rádio controlado. **Revista Vinc**, 2018.

ANJANAPPA, M.; DATTA, K.; SONG, T. Introduction to sensors and actuators. *In: ____*. [S.l.: s.n.], 2002. p. 16–1.

ARDUINO. **Arduino Uno Rev3**. 2021. Disponível em: <https://store.arduino.cc/usa/arduino-uno-rev3,note=>.

ARDUOELETRO. **Sensor De Tensão (voltagem) 0 - 25 Vdc Para Arduino Pic**. 2020. (Acesso em: 09 out. 2020.). Disponível em: <https://www.arduoletro.com/sensor-de-tensao-voltagem-0-25-vdc-arduino-pic>.

BD, Gear mart. **LM35 precision centigrade temperature sensors**. 2020. "Acesso em: 09 out. 2020.". Disponível em: <https://gearmartbd.com/product/lm35-precision-centigrade-temperature-sensors/>.

BHIDE, G.; SUTAR, A. R.; MANE, J.J. Implementation and study of bldc motor drive system. 05 2016.

BROWN, G. Henry darcy and the pitot tube. *In: . [S.l.: s.n.]*, 2001. p. 360–366. ISBN 978-0-7844-0594-9.

DALY, J. W.; RILEY, W. F.; MCCONNELL, K. G.; SAUNDERS, H. Instrumentation for engineering measurements. *In: . [S.l.: s.n.]*, 1984.

EZZEDDINE, W.; SCHUTZ, J.; REZG, N. Pitot sensor airflow measurement accuracy. *In: ____*. *[S.l.: s.n.]*, 2016. p. 1401–1405. ISBN 978-1-138-02997-2.

FAJER, M. Sistemas de investigação dos acidentes aeronáuticos da aviação geral: uma análise comparativa. **Faculdade de Saúde Pública, Universidade de São Paulo, São Paulo**, 2009.

FOUNDATION, Raspberry Pi. **Raspberry Pi 3 Model B+**. 2020. Acesso em: 09 out. 2020. Disponível em: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/>.

GANSSLE, J.; NOERGAARD, T.; EADY, F.; KATZ, D. J.; GENTILE, R.; ARNOLD, K.; HYDER, K.; PERRIN, B. **Embedded Hardware: Know It All**. USA: Newnes, 2007. ISBN 0750685840.

GOMES, M. E. **Sistema Embarcado para Controle de Motores Brushless DC sensored e sensorless para Veículos Portáteis de Transporte Individual**. 2018. 1-155 p., Brasília, 2018.

HEATH, Steve. **Embedded Systems Design**. 2nd. ed. USA: Butterworth-Heinemann, 2002. ISBN 0750655461.

HEYWOOD, John B. **Internal Combustion Engine Fundamental**. USA: McGraw-Hill, Inc, 1988. 917 p.

HOBBY, Limeira. **Bateria Zippy Li-Po 3S 11,1V X 2200mAh 25C**. 2021. Acesso em: 05 dez. 2021. Disponível em: <https://www.limeirahobby.com.br/bateria-zippy-li-po-3s-11-1v-x-2200ma-25c-p20132>.

HOBBYKING. **HK Pilot analog air speed sensor and Pitot set tubo**. 2020. Acesso em: 25 jan. 2021. Disponível em: https://hobbyking.com/pt_pt/hk-pilot-analog-air-speed-sensor-and-pitot-tube-set.html?__store=pt_pt.

HOBBYKING. **Hobby King 80A ESC 4A SBEC**. 2021. Acesso em: 23 jan. 2021. Disponível em: https://hobbyking.com/pt_pt/hobbyking-80a-2-6s-esc-4a-sbec.html.

HOBBYKING. **Turnigy D3548 / 4 1100KV Brushless Outrunner Motor**. 2021. "Acesso em: 23 jan. 2021". Disponível em: https://hobbyking.com/pt_pt/turnigy-d3548-4-1100kv-brushless-outrunner-motor.html?__store=pt_pt".

K., Klaus; HARALD, H.; ACHATZ, R.; BROY, M. **Model-Based engineering of embedded systems: The SPES 2020 methodology**. 1nd. ed. [S.l.]: Springer Heidelberg New York Dordrecht London, 2012.

KESTER, Walt; BRYANT, James; JUNG, Walt. Section 7 temperature sensors. *In: . [S.l.: s.n.]*, 1999.

MARWEDEL, Peter. **Embedded system design: embedded systems foundations of Cyber-Physical systems**. 2nd. ed. [S.l.]: Springer Publishing Company, Incorporated, 2010. ISBN 9789400702561.

MIT. **Como crimpar um cabo de rede ethernet**. 2020. Acesso em: 14 nov. 2020. Disponível em: <https://tecnoblog.net/276245/como-crimpar-um-cabo-de-rede-ethernet/>.

MORRISON, J Paul Rodker. **Flow-based Programming**. 2020. Acesso em: 14 out. 2020. Disponível em: <https://jpaulm.github.io/fbp/index.html>.

MÜLLER, I.; BRITO, R. M. de; PEREIRA, C.; BRUSAMARELLO, V. Load cells in force sensing analysis - theory and a novel application. **Instrumentation Measurement Magazine, IEEE**, v. 13, p. 15 – 19, n.03, 2010.

NODE-RED. **Low-code programming for event-driven applications**. 2020. Acesso em: 09 out. 2020. Disponível em: <https://nodered.org>.

NODE-RED. **Node-RED Concepts**. 2020. Acesso em: 09 out. 2020. Disponível em: <https://nodered.org/docs/user-guide/concepts>.

NPX SEMICONDUCTOR. **MPXV7002 integrated Silicon pressure sensor on-chip signal conditioned, temperature compensated and calibrated**. [S.l.], 2017. Rev. 4.

PAUN, Maria-Alexandra; SALLESE, Jean-Michel; KAYAL, Maher. Hall effect sensors design, integration and behavior analysis. **Journal of Sensor and Actuator Networks**, v. 2, p. 85–97, 03 2013.

PETRESCU, R.; AVERSA, R.; AKASH, B.; BUCINELL, R.; RODRÍGUEZ, J. C.; BERTO, F.; MIRSAYAR, M.; APICELLA, A.; PETRESCU, F. I. History of aviation-a short review. **Journal of Aircraft and Spacecraft Technology**, v. 2017, n.06, 2017.

RAINE, R.; THORWARTH, H. Performance and combustion characteristics of a glow-ignition two-stroke engine. **SAE Technical Papers**, n.03, 2004.

RAMSDEN, Ed. Hall-effect sensors: Theory and applications. *In: . [s.n.]*, 2006. Disponível em: <http://www.books24x7.com/marc.asp?bookid=32304>.

ROBOTICA, Ardu. **Sensor De Corrente Acs712 – 5a Arduino Pic**. 2020. (Acesso em: 09 out. 2020.). Disponível em: <https://ardurobotica.com.br/produto/sensor-de-corrente-ac712-5a-arduino-pic/>.

RODRIGUES, LEMJ. Fundamentos da engenharia aeronáutica com aplicações ao projeto sae-aerodesign: Aerodinâmica e desempenho. **Salto/SP**, v. 1, 2014.

SILVA, C.W. de. **Sensors and Actuators: Engineering System Instrumentation, Second Edition**. CRC Press, 2015. ISBN 9781466506824. Disponível em: <https://books.google.com.br/books?id=ukZOCgAAQBAJ>.

TEXAS INSTRUMENTS. **LM35 precision centigrade temperature sensors**. *[S.l.]*, 2017.

TIGGERCOMP. **Módulo Sensor Hall 3144 E**. 2020. Acesso em: 09 out. 2020. Disponível em: http://www.tiggercomp.com.br/novaloja/product_info.php?products_id=9125.

TORENBEEK, E.; WITTENBERG, H. **Flight Physics: Essentials of Aeronautical Disciplines and Technology, with Historical Notes**. Springer Netherlands, 2009. ISBN 9781402086649. Disponível em: <https://books.google.com.br/books?id=VIjEZPyGRIAC>.

UNIVERSITY, Battery. **BU-206: Lithium-polymer: Substance or Hype?** 2020. Acesso em: 18 jul. 2020. Disponível em: https://batteryuniversity.com/learn/article/the_li_polymer_battery_substance_or_hype.

UNIVERSITY, Battery. **BU 215: Summary table of nickel based batteries**. 2020. Acesso em: 18 jul. 2021. Disponível em: https://batteryuniversity.com/learn/article/bu_215_summary_table_of_nickel_based_batteries.

UNIVERSITY, Battery. **BU 216: Summary table of lithium based batteries**. 2020. Acesso em: 18 jul. 2020. Disponível em: https://batteryuniversity.com/learn/article/bu_216_summary_table_of_lithium_based_batteries.

WINGS, Electronic. **HX711 24-bit ADC for Weigh Scales**. 2021. Acesso em: 23 jan. 2021. Disponível em: <https://www.electronicwings.com/components/hx711-24-bit-adc-for-weigh-scales>.

Wolf, W. H. Hardware-software co-design of embedded systems. **Proceedings of the IEEE**, v. 82, n. 7, p. 967–989, 1994.

YEDAMALE, Padmaraja. Brushless dc (bldc) motor fundamentals. *In: . [S.l.: s.n.]*, 2019.

ZHAO, Jian; YU, Yangwei. Brushless dc motor fundamentals. **Monolithic Power Systems**, p. 19, 2014.

APÊNDICES

APÊNDICE A – PESQUISA DE CAMPO

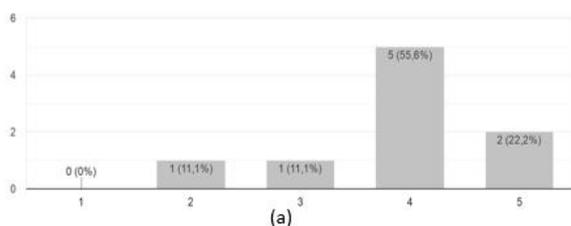
Como ponto de partida para o levantamento de especificação do sistema embarcado, foi realizada uma pesquisa de campo com os integrantes da equipe AeroDesign UTFPR Curitiba, iniciando com três perguntas objetivas para observar a preferência visual e de montagem do projeto:

1. Classifique, de 1 a 5, o quão importante é a bancada ser portátil;
2. Classifique, de 1 a 5, a importância da bancada possuir um baixo custo;
3. Classifique, de 1 a 5, a importância da bancada contar com uma interface gráfica que mostre todos os gráficos em tempo real.

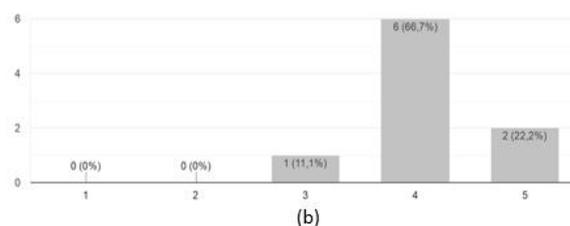
A Figura 125 apresenta os resultados, que foram respondidos por nove integrantes da equipe:

Figura 125 – Pesquisa objetiva

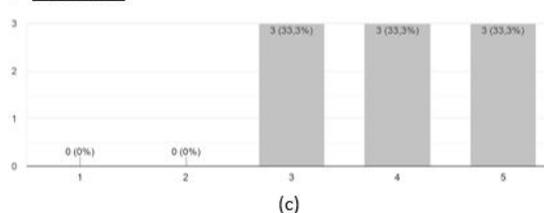
Classifique, de 1 a 5, o quão importante é a bancada ser portátil



Classifique, de 1 a 5, a importância da bancada possuir um baixo custo



Classifique, de 1 a 5, a importância da bancada contar com uma interface gráfica que mostre todos os gráficos em tempo real



Fonte: Autoria Própria (2021).

Como é possível ver, a equipe se apresenta positiva à bancada ser portátil. Essa preferência se deve à facilidade na realização dos ensaios em possuir um design compacto e que se apresenta flexível em qualquer local que o teste se realize. A equipe também optou por um projeto de custo reduzido, tornando-se prioridade do projeto em obter componentes baratos e fáceis de serem encontrados. A última pergunta foi relacionada à visualização gráfica dos

dados coletados no teste, aprovada pelos integrantes. Portanto, foi um dos focos do trabalho o desenvolvimento de uma interface visual.

A segunda série de perguntas foi relacionada aos procedimentos de teste. Foram duas perguntas objetivas feitas, respondida pelos mesmos nove integrantes da pesquisa anterior. Foram fornecidas opções de como salvar os dados e formas de realização dos testes, que podem ser observados na Figura 126:

Figura 126 – Pesquisa objetiva



Fonte: Autoria Própria (2021).

A equipe mostrou a preferência em um salvamento dos dados de forma automatizada, ou seja, com o final do teste esses dados devem ser enviados à uma plataforma de preferência dos integrantes. O outro levantamento demonstrou a importância para a equipe da flexibilidade dos testes. Portanto, os seguintes pontos foram definidos como requisitos do projeto:

- Portabilidade;
- Baixo custo;
- Interface gráfica;
- Salvamento automático;
- Testes em Rotina e *Open Loop*.

APÊNDICE B – ENTRADA DE PARÂMETROS - DETALHAMENTO DE SOFTWARE

Esta aba é responsável por extrair os parâmetros básicos de testes. Como a bancada será utilizada para a realização de todos os motores a serem adquiridos pela equipe, deverá ser disponibilizado um sistema de escolha do grupo motopropulsor utilizado. Os parâmetros considerados básicos são:

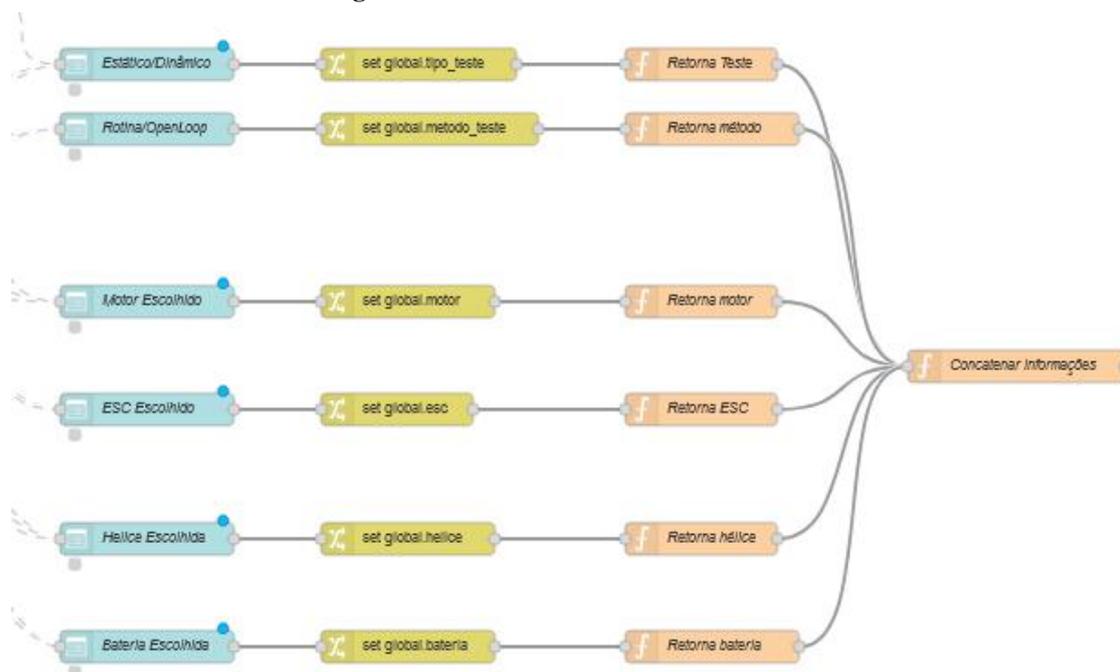
- Tipo de Teste;
- Formato do Teste;
- Motor escolhido;
- ESC escolhido;
- Hélice escolhida;
- Bateria Escolhida;
- Configuração Escolhida.

O usuário terá a opção de escolher cada um dos parâmetros baseado no base de dados. Na Figura 127 é mostrado o fluxo do software. O nó mais à esquerda é responsável por fornecer opção de escolha ao usuário em forma de lista suspensa. Após a opção do usuário, uma mensagem do tipo *payload* é enviada ao próximo nó do tipo *change*, que alterará a mensagem para uma variável de contexto global para que seja utilizada em qualquer parte do código. O próximo nó é do tipo *function* que registra a escolha do usuário e prepara uma *string* para que seja usada como texto posteriormente no *firmware*. A Figura 128 mostra a parte interna do fluxo de escolha do tipo de teste.

Outra opção de definição dos parâmetros oferecida ao usuário é a de carregar valores colocados em testes anteriores, mostrado na Figura 129. Ao pressionar o botão Atualizar, o fluxo segue para no nó de listagem, que realiza o mapeamento dos testes já feitos pelo usuário, que estão presentes no base de dados. O mapeamento extrai o caminho de cada teste anterior, dispondo o resultado em forma de lista suspensa.

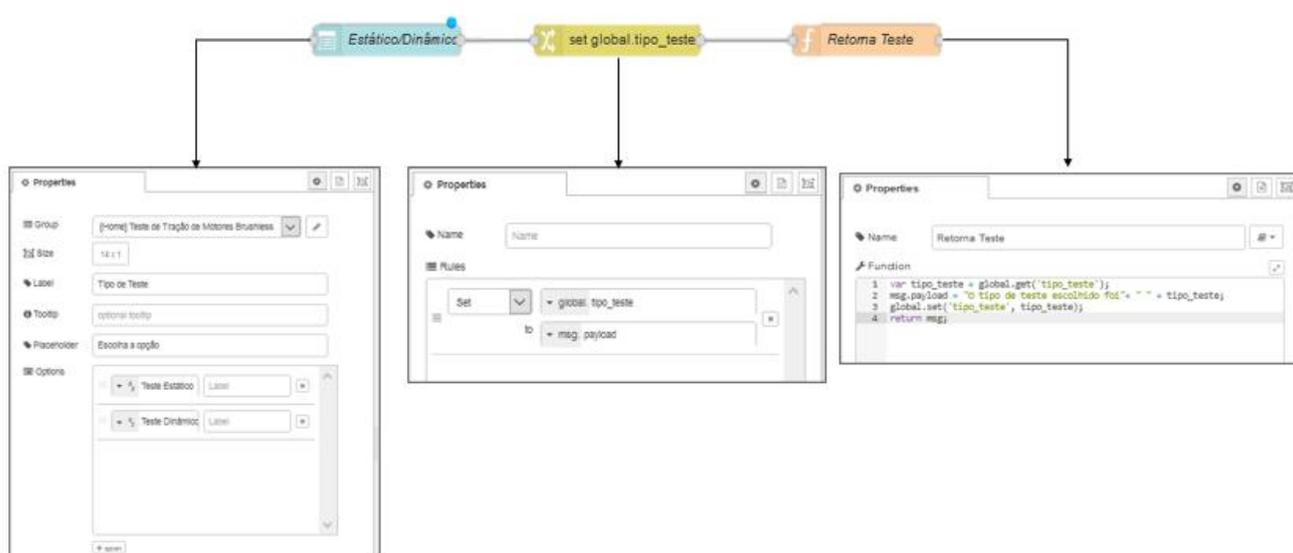
Para a confirmação dos parâmetros, o usuário deverá pressionar o botão confirmar que irá gravar cada parâmetro em uma variável de contexto global específica. Na Figura 130 é possível ver a interface gráfica criada para a escolha dos parâmetros do teste.

Figura 127 – Fluxo de Entrada dos Parâmetros



Fonte: Autoria Própria (2021).

Figura 128 – Fluxo para a escolha do Tipo de Teste

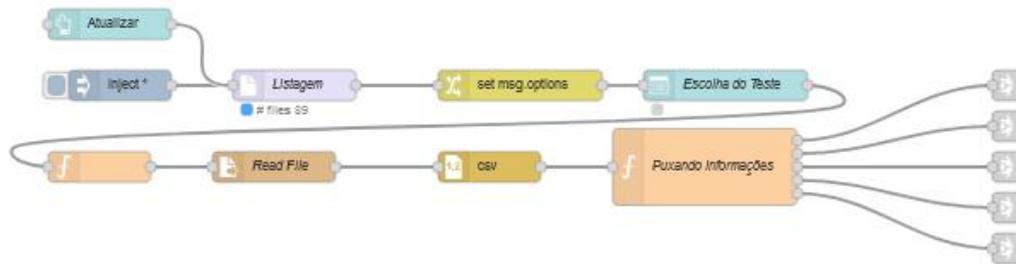


Fonte: Autoria Própria (2021).

Após a confirmação do teste, o software avalia se todos os parâmetros inseridos pelo usuário são condizentes entre si, avaliando:

- Bateria versus ESC;
- Bateria versus Motor;

Figura 129 – Fluxo de leitura dos parâmetros de testes anteriores



Fonte: Autoria Própria (2021).

Figura 130 – Interface para a escolha dos parâmetros do teste

Fonte: Autoria Própria (2021).

- Motor versus ESC;

Em cada um dos nós da função apresentada na Figura 131 é feita a extração dos parâmetros elétricos das escolhas do usuário, como:

- Tensão mínima, tensão máxima, corrente máxima e corrente *idle* do motor;
- Tensão mínima, tensão máxima e corrente máxima do ESC;
- Tensão mínima e nominal da bateria.

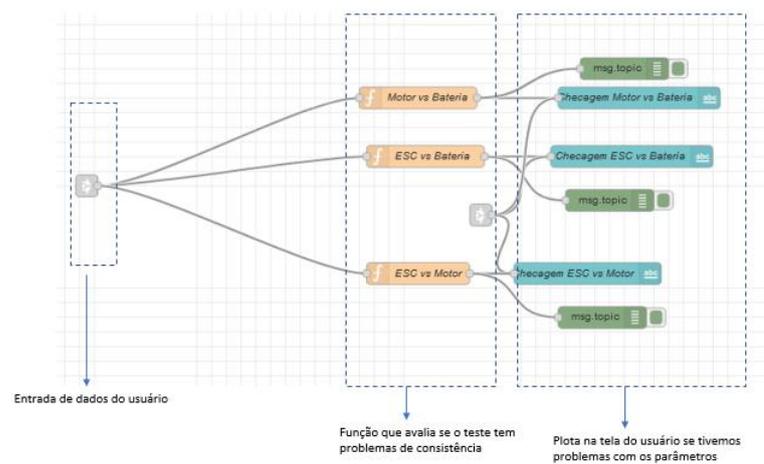
Para cada combinação de motor, ESC e bateria, o software avalia se os componentes escolhidos podem causar danos, como é apresentado na Figura 132. O usuário é informado e fica a critério do mesmo dar continuidade ao teste ou rearranjar os componentes.

Figura 131 – Extração dos parâmetros elétricos do motor, ESC e bateria



Fonte: Autoria Própria (2021).

Figura 132 – Avaliação dos parâmetros



Fonte: Autoria Própria (2021).

APÊNDICE C – COLETA DE DADOS DOS SENSORES - DETALHAMENTO DE SOFTWARE

O microcontrolador auxiliar é responsável pela coleta dos dados vindos dos sensores periféricos. O software para a coleta dos dados foi desenvolvido em linguagem de programação C, junto à IDE do Arduino. Foi criado um uma função em *loop* para cada um dos sensores. A saída de cada sensor é então formatada e enviada com uma taxa de 100 ms para a porta serial do Arduino, apresentado na Figura 133. O Node-RED fornece a possibilidade de leitura da porta serial utilizando o nó *serial in*. A única informação necessária para informar ao nó é a porta serial onde esses dados estão sendo transmitidos.

Figura 133 – Código para envio dos dados ao Microcontrolador Principal

```
void loop() {
  massa = Load_Cell();
  rotacao = RPM_3144E();
  velocidade = pitot();
  tensao_bateria_control = tensao_bateria_controle();
  tensao_bateria_pot = tensao_bateria_potencia();
  corrente_bateria_control = sensor_corrente_controle();
  Serial.print(massa);
  Serial.print(",");
  Serial.print(rotacao);
  Serial.print(",");
  Serial.print(velocidade);
  Serial.print(",");
  Serial.print(tensao_bateria_control);
  Serial.print(",");
  Serial.print(tensao_bateria_pot);
  Serial.print(",");
  Serial.print(corrente_bateria_control);
  Serial.print(",");
  Serial.print(corrente_bateria_potencia);
  Serial.print(",");
  Serial.print(temperatura_bateria);
  Serial.print("\n");
  delay(100);
}
```

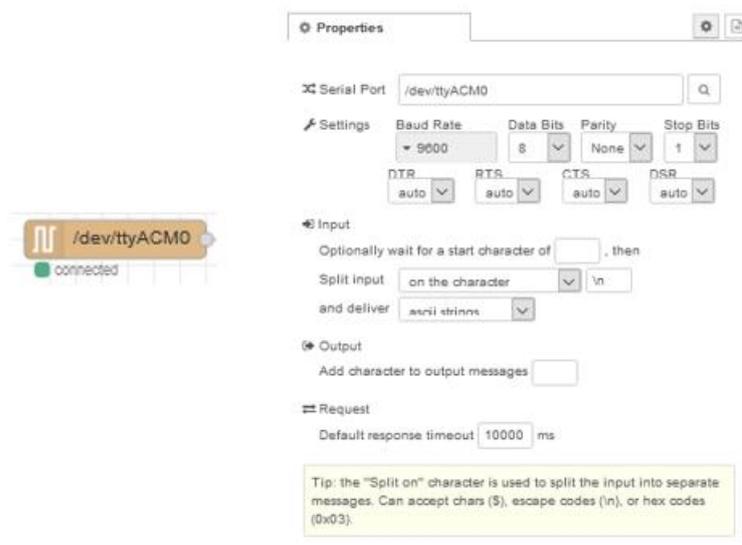
Fonte: Autoria Própria (2021).

Na Figura 134, é possível ver o nó *Serial In* utilizado e seus atributos necessários para a funcionalidade. Esse nó envia uma mensagem do tipo *payload* contendo uma *string* do tipo ASCII, dividindo cada mensagem de *input* pelo caracter $\backslash n$. Observe que este caracter é inserido no final do código de envio dos dados ao microcontrolador principal da Figura 133.

Cada mensagem serial é passada para um nó do tipo *function*, denominado de *Split data* dentro do código, que irá coleta os dados vindos como mensagem do nó *Serial in* e designar cada dado valor a variável correspondente a cada sensor. O fluxo completo de tratamento dos dados dos sensores pode ser visto na Figura 135.

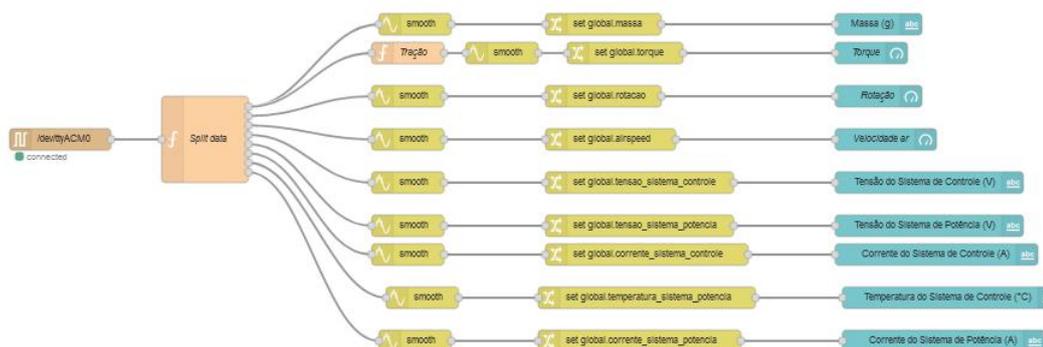
Dentro do nó *Split Data*, apresentado na Figura 136, é criada uma variável *output* que irá armazenar o valor de cada valor do *string* em um vetor. Através da função *parseInt* e

Figura 134 – Nó "serial in"



Fonte: Autoria Própria (2021).

Figura 135 – Fluxo de tratamento dos valores advindos do microcontrolador auxiliar



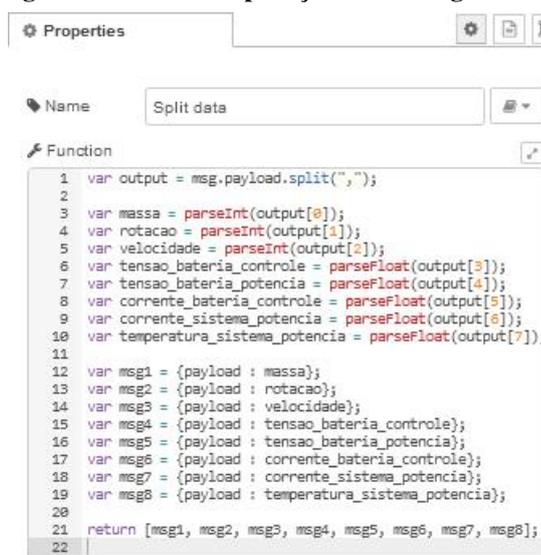
Fonte: Autoria Própria (2021).

parseFloat, o valor de cada unidade do vetor é extraído e colocado em uma variável local. O nó foi definido com oito saídas, transmitindo uma mensagem to tipo *payload* de cada sensor.

Esses valores são direcionados então para o nó "*smooth*". Esse nó fornece várias funções, como média de valores, valor máximo e mínimo e filtros passa baixa e passa alta. Para esse caso, este nó será utilizado para suavizar os dados vindos dos sensores, retornando a média dos dez valores mais recentes. Na Figura 137 é possível ver as propriedades internas desse nó.

O valor é então enviado ao próximo nó que irá tornar cada variável como contexto global, podendo ser utilizada em qualquer aba do projeto. Além dos sensores conectados ao Microcontrolador Auxiliar, o projeto contou com o sensor BMP180 conectado diretamente ao Raspberry Pi 3B+ via protocolo I²C. A leitura desse sensor é feita conforme o fluxo apresentado na Figura 138.

Figura 136 – Nó de separação da mensagem serial



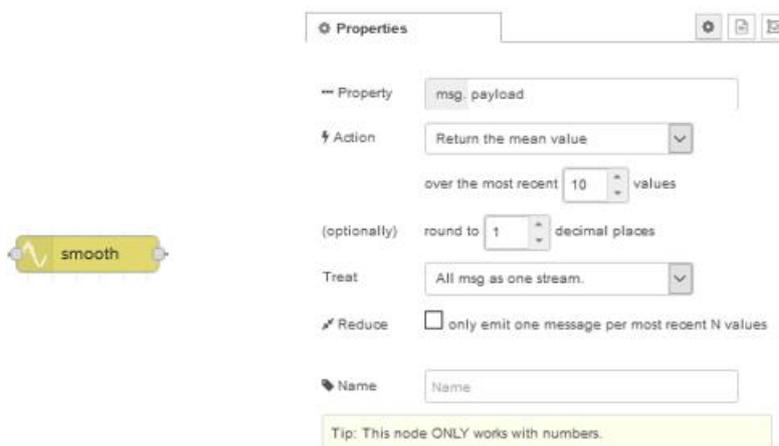
```

1 var output = msg.payload.split(",");
2
3 var massa = parseInt(output[0]);
4 var rotacao = parseInt(output[1]);
5 var velocidade = parseInt(output[2]);
6 var tensao_bateria_controle = parseFloat(output[3]);
7 var tensao_bateria_potencia = parseFloat(output[4]);
8 var corrente_bateria_controle = parseFloat(output[5]);
9 var corrente_sistema_potencia = parseFloat(output[6]);
10 var temperatura_sistema_potencia = parseFloat(output[7]);
11
12 var msg1 = {payload : massa};
13 var msg2 = {payload : rotacao};
14 var msg3 = {payload : velocidade};
15 var msg4 = {payload : tensao_bateria_controle};
16 var msg5 = {payload : tensao_bateria_potencia};
17 var msg6 = {payload : corrente_bateria_controle};
18 var msg7 = {payload : corrente_sistema_potencia};
19 var msg8 = {payload : temperatura_sistema_potencia};
20
21 return [msg1, msg2, msg3, msg4, msg5, msg6, msg7, msg8];
22

```

Fonte: Autoria Própria (2021).

Figura 137 – Nó de suavização dos valores dos sensores

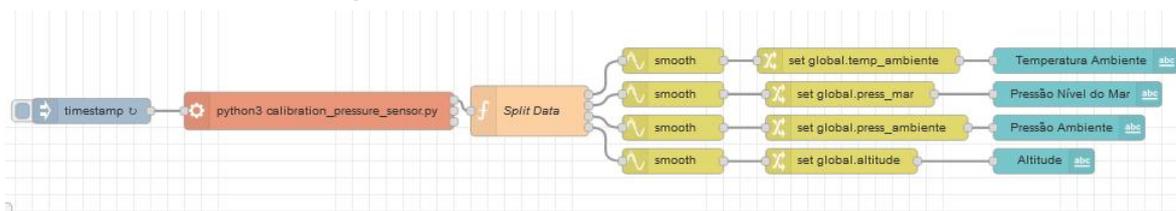


Fonte: Autoria Própria (2021).

Este fluxo executa um código no sistema de comando do Raspberry Pi 3B+, que então executa o código em Python do sensor BMP180. O sensor possui 4 valores como saída: temperatura ambiente, pressão ambiente, pressão à nível do mar e altitude. A biblioteca Adafruit BMP085 é a utilizada para coletados os dados dos sensores. A saída é formada a fim de facilitar a leitura dos dados no Node-RED. O nó de tipo *inject* é responsável pelo início do *loop* de execução desse código, realizando a mesma a cada 1 segundo. O nó *exec*, apresentado na Figura 139, é utilizado para a execução do código em Python e a mensagem então é enviada como uma *string*. Na Figura 140 é possível ver um trecho do código executado no Raspberry Pi 3B+.

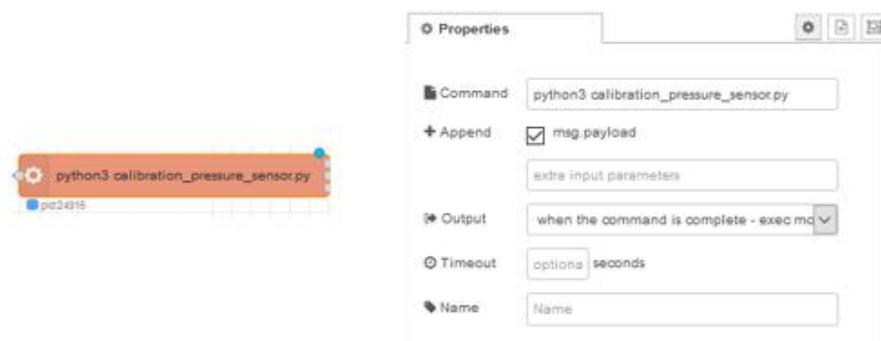
Após a etapa de execução no código, a saída do nó *exec* é então processada no nó de

Figura 138 – Fluxo de leitura do sensor BMP180



Fonte: Autoria Própria (2021).

Figura 139 – Nó do tipo exec



Fonte: Autoria Própria (2021).

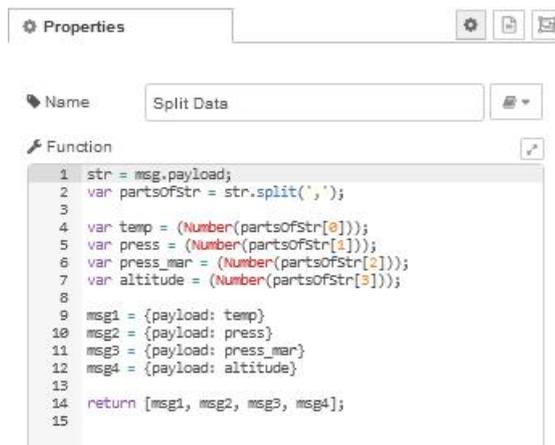
Figura 140 – Código do sensor BMP180 executado em Python

```
import sys
import Adafruit_BMP.BMP085 as BMP085
try:
    sensor = BMP085.BMP085()
    calibração_temperatura = sensor.read_temperature()
    calibração_pressão = sensor.read_pressure()
    calibração_pressãonivelmar = sensor.read_sealevel_pressure()
    calibração_altitude = sensor.read_altitude()
except:
    calibração_temperatura = 0
    calibração_pressão = 0
    calibração_pressãonivelmar = 0
    calibração_altitude = 0
finally:
    sys.stdout.write("%f,%f,%f,%f"%(calibração_temperatura, calibração_pressão, calibração_pressãonivelmar, calibração_altitude))
```

Fonte: Autoria Própria (2021).

function, também chamado de "Split data", realizando a separação dos valores vindos como mensagem do tipo *payload*. O objetivo é o mesmo da extração de dados dos sensores do microcontrolador auxiliar. A *string* é reconstruída em um vetor e o valor de cada parâmetro vindo do sensor é designado para uma variável específica, como é possível ver na Figura 141.

Posteriormente, os dados são suavizados utilizando o nó "smooth", e as variáveis são transformadas em globais.

Figura 141 – Nó de separação dos valores do sensor BMP180

The image shows a screenshot of a Node-RED function node. The node is titled "Split Data". The function code is as follows:

```
1 str = msg.payload;
2 var partsOfStr = str.split(',');
3
4 var temp = (Number(partsOfStr[0]));
5 var press = (Number(partsOfStr[1]));
6 var press_mar = (Number(partsOfStr[2]));
7 var altitude = (Number(partsOfStr[3]));
8
9 msg1 = {payload: temp}
10 msg2 = {payload: press}
11 msg3 = {payload: press_mar}
12 msg4 = {payload: altitude}
13
14 return [msg1, msg2, msg3, msg4];
15
```

Fonte: Autoria Própria (2021).

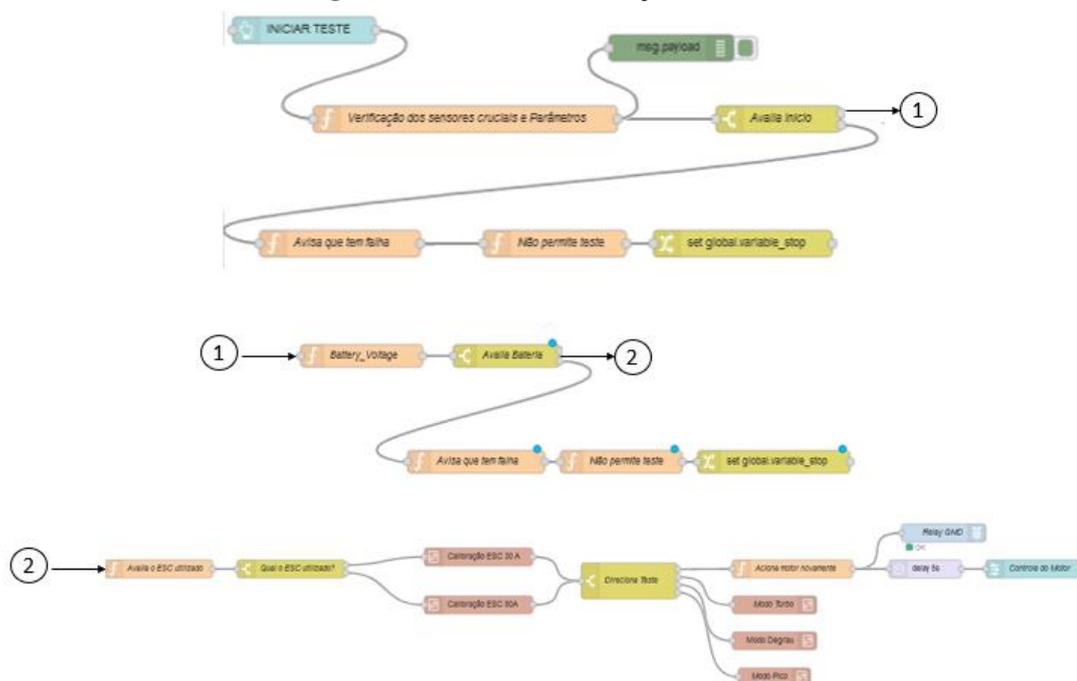
APÊNDICE D – MODOS DE TESTE - DETALHAMENTO DE SOFTWARE

São oferecidas duas opções para o usuário executar o teste: Modo *Open Loop* e Modo Rotina, que são escolhidos na etapa de "Entrada de Parâmetros". Após a verificação dos sensores, o usuário deverá ir na aba "Display" para que o teste possa ser iniciado.

O usuário irá pressionar o botão Iniciar, dando início ao fluxo de realização do teste. A primeira etapa é a verificação de todos os sensores considerados cruciais para a realização do teste, por meio do nó "Verificação dos sensores cruciais e Parâmetros", como pode ser visto na Figura 142. Os sensores considerados cruciais são:

- Sensor de corrente do Sistema de Potência;
- Sensor de tensão do Sistema de Potência;
- Sensor de temperatura da bateria do Sistema de Potência.

Figura 142 – Fluxo de realização do teste



Fonte: Autoria Própria (2021).

Além disso, nesse mesmo nó, é verificado se todos os parâmetros iniciais, especificados na seção 5.2.1 foram preenchidos. Caso algum sensor esteja com problema ou algum parâmetro não foi preenchido, o teste não é iniciado e uma mensagem de erro aparecerá na tela do usuário. Caso os sensores e parâmetros estejam corretos, o fluxo segue para a verificação da tensão da bateria do Sistema de Potência, por meio do nó "Battery Voltage". Caso a bateria esteja abaixo

do nível aceitável do fabricante, o teste não é iniciado e uma mensagem de erro aparecerá na tela do usuário. Em caso de funcionamento, o fluxo finaliza a parte de verificação e inicia a calibração do ESC. A função "Avalia o ESC utilizado" extrai qual o ESC escolhido pelo usuário na etapa de Entrada de Parâmetros e por meio do nó "Qual o ESC utilizado", e a calibração do ESC é feita. Cada ESC possui uma maneira de calibração específica. Dessa maneira, fez-se um subfluxo para cada ESC contido no base de dados da equipe. A Figura 143 apresenta o subfluxo de calibração do *ESC Hobbyking 80A*.



Com o ESC devidamente calibrado, o fluxo segue para o nó "Direciona Teste", no qual é feita a extração do modo de teste escolhido pelo usuário e o teste é realizado. Em caso de *Open Loop*, o usuário tem o controle do motor por meio de um botão deslizante, sem qualquer intervenção do software. Esse sinal é enviado diretamente ao ESC. Para casos de Modo Rotina, o usuário escolhe qual o modelo de rotina ele deseja. A bancada possui os seguintes modelos:

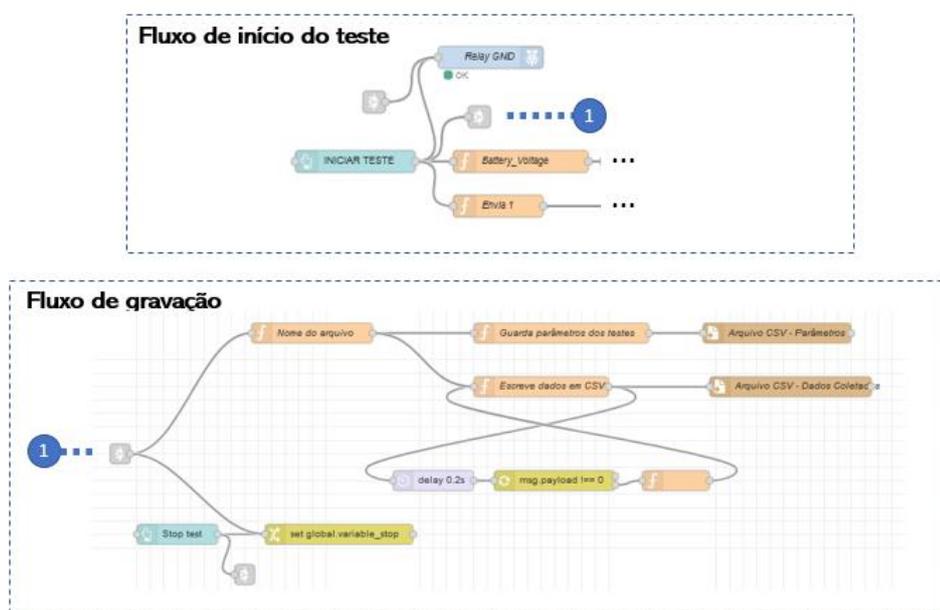
- Modo Turbo;
- Modo Degrau;
- Modo Pico.

O modo turbo coloca o motor em potência máxima durante trinta segundos. O intuito desse teste é ver a capacidade de descarga do motor nessas condições. O Modo Escada testa o motor quando aplicada uma potência escalonada, aumentando de 10 em 10 % até atingir a potência máxima e o teste ser finalizado. O Modo Pico coloca o motor em situações de mudanças bruscas de torque, simulando retomadas de aceleração em voo.

APÊNDICE E – ETAPA DE VISUALIZAÇÃO DE DADOS - DETALHAMENTO DE SOFTWARE

A etapa de visualização de dados será utilizada para os usuários observarem os resultados plotados de forma gráfica, sejam eles do teste atual como de qualquer teste já feito pela equipe que esteja no base de dados. Inicialmente, é abordada a gravação dos resultados dos testes. Na etapa Modos de teste, seção 5.2.4, é informado que o botão Iniciar começa o teste. Nesse mesmo momento, este botão inicia o fluxo de gravação dos parâmetros do teste e também dos dados dos sensores, como pode ser visto na Figura 144.

Figura 144 – Fluxo de gravação do teste



Fonte: Autoria Própria (2021).

O fluxo superior é utilizado para gravar os parâmetros escolhidos pelo usuário na etapa de "Entrada de Parâmetros", apresentado na seção 5.2.1. Como é mostrado nessa etapa, todas as variáveis são gravadas e salvas globalmente no software. O nó "Nome de arquivo", apresentado na Figura 145, constrói uma *string* com o nome do teste, com os atributos de horário do início do teste, hélice e motor utilizado, evitando nomes duplicado, passando uma mensagem do tipo *filename*, que será interpretado no nó "File" como o caminho a gravar o arquivo CSV. Com isso, o software sofre uma bifurcação. Com os dados extraídos, o nó "Guarda parâmetros do Teste" constrói uma *string* com todos os parâmetros, enviando uma mensagem do tipo *payload* para o nó 'File'.

O outro caminho da bifurcação vai para o nó "Escreve dados em CSV". Este nó extrai

Figura 145 – Código construtor do nome do Arquivo CSV



Fonte: Autoria Própria (2021).

os valores instantâneos dos sensores e constrói uma *string*, enviando como mensagem do tipo *payload*, como mostra a Figura 146.

Figura 146 – Código para gravar os parâmetros do teste



Fonte: Autoria Própria (2021).

Após este nó, é escrito o valor dos sensores no nó "Arquivo CSV - Dados Coletados" e ao mesmo tempo o fluxo segue para uma estrutura em *loop*, utilizando o nó *while-loop*. Caso a condição expressa no nó seja verdadeira, o *loop* continua até o momento que seja falso. Esse fluxo foi criado com o intuito de gravar os valores coletados em um período de execução do teste. No momento em que o usuário pressiona o botão iniciar na tela de Display, uma mensagem do tipo *payload* é enviado para a variável de contexto global *variable*. Esta variável é responsável por informar se o teste está em andamento ou não. No momento que o teste finaliza ou que o usuário determina o seu fim, o botão "Stop Test" clicado armazenará o valor 0 nesta variável. Desta maneira, o *loop* é quebrado, finalizando a escritura dos dados no arquivo CSV. O tempo de cada leitura escolhido foi de 200 ms.

Caso o usuário tenha o intuito de visualizar de forma gráfica os dados do teste que acabou de executar, ou de testes anteriores, ele terá a possibilidade através da aba "Visualização de dados". Nesta aba, são listados todos os testes executados e gravados no base de dados. A listagem de cada um dos testes é feita por meio do nó "File Lister". Ao fornecer um caminho de pasta presente no Raspberry Pi 3B+, o nó extrai todos os arquivos com extensão específica,

Figura 147 – Código para gravar os dados em CSV



```

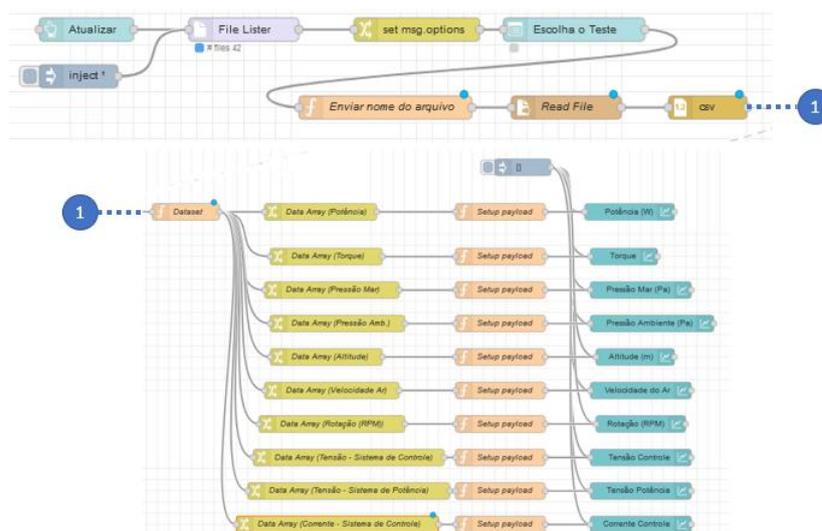
1 var continuidade = msg.payload;
2 var pressao_mar = global.get('press_mar');
3 var pressao_amb = global.get('press_ambiente');
4 var altitude = global.get('altitude');
5 var timestamp = global.get('timestamp');
6 var potencia = global.get('potencia');
7 var massa = global.get('massa');
8 var torque = global.get('torque');
9 var rotacao = global.get('rotacao');
10 var airdspeed = global.get('airspeed');
11 var tensao_controle = global.get('tensao_sistema_controle');
12 var tensao_potencia = global.get('tensao_sistema_potencia');
13 var corrente_controle = global.get('corrente_sistema_controle');
14
15- if(continuidade===0){
16-   msg.payload=0;
17- }
18
19- else{
20-   var data = timestamp+","+pressao_mar+","+pressao_amb+","+altitude+","+potencia+","+torque+","+rotacao+","+airspeed+","+tensao_controle+","+tensao_potencia+","+corrente_controle;
21-   msg.payload = data;
22- }
23 return msg;

```

Fonte: Autoria Própria (2021).

para nosso caso somente arquivos CSV. Para que todos os caminhos sejam mostrados ao usuário em forma de lista suspensa, é utilizado o nó *dropdown*. Quando esse nó recebe uma entrada em forma de *array* como mensagem do tipo *options*, todos os valores contidos no *array* são dispostos na lista. Para que a mensagem chegue como extensão *options*, é utilizado o nó *Change*. Com o usuário selecionando o arquivo na lista suspensa, o caminho é enviado ao nó *Read File*. Os dados são passados ao nó CSV que transforma os valores em CSV para *string* e a leitura é então passada aos *dashboards*.

Figura 148 – Fluxo de Visualização dos Dados



Fonte: Autoria Própria (2021).

APÊNDICE F – ORÇAMENTO DO PROJETO

Na Tabela 4 é possível ver o custo do Sistema de Controle.

Tabela 4 – Custo do Sistema de Controle

Dispositivo	Preço (Unidade)	Quantidade	Valor Total
Raspberry Pi 3B+	R\$ 450,00	1	R\$ 450,00
Arduino Uno	R\$ 50,00	1	R\$ 50,00
Célula de Carga + HX711	R\$ 25,00	1	R\$ 25,00
Tubo de Pitot + MPXV7002DP	R\$ 150,00	1	R\$ 150,00
Sensor de Efeito Hall 3144E	R\$ 12,00	1	R\$ 12,00
Sensor de Corrente ACS712	R\$ 18,00	1	R\$ 18,00
Sensor de Tensão 0-25V	R\$ 8,00	2	R\$ 16,00
Sensor de Temperatura LM35	R\$ 12,00	1	R\$ 12,00
Bateria Zippy 11,1V 2500mAh	R\$ 112,00	1	R\$ 112,00
Sensor BMP180	R\$ 12,00	1	R\$ 12,00
Conversor Buck 5V	R\$ 20,00	1	R\$ 20,00
Placa AVR USBASP	R\$ 35,00	1	R\$ 35,00
Regulador de Tensão LM7812	R\$ 6,00	1	R\$ 6,00
Total			R\$ 918,00

Fonte: Autoria Própria (2021).

Na Tabela 5 é mostrado o custo do Sistema de Potência.

Tabela 5 – Custo do Sistema de Potência

Dispositivo	Preço (Unidade)	Quantidade	Valor Total
Conjunto Motor + ESC + Hélice	R\$ 596,00	1	R\$ 596,00
Bateria ZOP Power 18,5V	R\$ 386,00	1	R\$ 386,00
Relé Eletromagnético 80/60A 12V	R\$ 25,00	1	R\$ 25,00
Sensor de Corrente ACS758	R\$ 55,00	1	R\$ 55,00
Capacitor 10uF	R\$ 0,10	3	R\$ 0,30
Capacitor 1000uF	R\$ 0,50	1	R\$ 0,50
Capacitor 100nF	R\$ 0,10	1	R\$ 0,10
Capacitor 100uF	R\$ 1,00	1	R\$ 1,00
Diodo N4007	R\$ 0,10	1	R\$ 0,10
Resistor 1,2kOhm	R\$ 0,10	1	R\$ 0,10
Transistor 2N2222	R\$ 1,00	1	R\$ 1,00
Total			R\$ 1.064,10

Fonte: Autoria Própria (2021).

Na Tabela 6 é apresentado o custo da parte mecânica da bancada, tal como os gastos com chicote elétrico.

Tabela 6 – Custos da parte mecânica e chicote elétrico

Dispositivo	Preço (Unidade)	Quantidade	Valor Total
Suporte de Fixação	R\$ 155,00	1	R\$ 155,00
Placa de Acrílico	R\$ 50,00	1	R\$ 50,00
Chicote Elétrico (cabos, terminais e plugs)	R\$ 100,00	-	R\$ 100,00
Total			R\$ 305,00

Fonte: Autoria Própria (2021).

Com isso, o custo total do projeto foi de R\$2287,10.