

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

**JHONATAS WILLIAN GONÇALVES DE MORAES**

**PROJETO E DESENVOLVIMENTO DE UM VEÍCULO AUTÔNOMO  
DE SUPERFÍCIE PARA COLETA DE AMOSTRAS DE ÁGUA**

**APUCARANA**

**2023**

**JHONATAS WILLIAN GONÇALVES DE MORAES**

**PROJETO E DESENVOLVIMENTO DE UM VEÍCULO AUTÔNOMO  
DE SUPERFÍCIE PARA COLETA DE AMOSTRAS DE ÁGUA**

**Design and Development of an Autonomous Surface Vehicle to Collect  
Water Samples**

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

Orientador: Prof. Dr. Thales Eugenio Portes de  
Almeida

**APUCARANA  
2023**



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

Esta licença permite remixe, adaptação e criação a partir do trabalho, para fins não comerciais, desde que sejam atribuídos créditos ao(s) autor(es) e que licenciem as novas criações sob termos idênticos. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

**JHONATAS WILLIAN GONÇALVES DE MORAES**

**PROJETO E DESENVOLVIMENTO DE UM VEÍCULO AUTÔNOMO  
DE SUPERFÍCIE PARA COLETA DE AMOSTRAS DE ÁGUA**

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

Data de aprovação: 15/junho/2023

---

Thales Eugenio Portes de Almeida  
Doutor em Engenharia Elétrica  
Universidade Tecnológica Federal do Paraná - Campus Apucarana

---

Fernando Barreto  
Doutor em Engenharia Elétrica e Informática Industrial  
Universidade Tecnológica Federal do Paraná - Campus Apucarana

---

Vinícius Dário Bacon  
Doutor em Engenharia Elétrica  
Universidade Tecnológica Federal do Paraná - Campus Apucarana

**APUCARANA  
2023**

Dedico este trabalho aos meus pais, Janethe e Jorge que me apoiaram com os estudos antes e durante toda a graduação. Dedico também a minha tia Leontina e ao meu tio Cláudio por estarem comigo em um dos momentos mais difíceis de minha vida, e também por me levantarem para continuar a minha carreira.



## **AGRADECIMENTOS**

Agradeço primeiramente a Deus por me dar forças e motivação para concluir este trabalho dentro de minhas expectativas podendo demonstrar todo o esforço realizado em sua construção. Agradeço principalmente a minha família por enfrentarem junto comigo as dificuldades de distância e os desafios que foram passados durante todo o período de graduação.

Agradeço ao meu orientador Prof. Dr. Thales Eugênio Portes de Almeida pela orientação com a ideia inicial deste trabalho, atendendo as expectativas de desenvolvimento de produto. Agradeço aos meus amigos mais próximos Arthur Higino, Beatriz Martins, Mateus Milanez, Matheus Marques, Michael Nunes e Samuel Oliveira pelos bons momentos de estudos e diversão em que passamos juntos durante a graduação.

Agradeço também a coordenação de engenharia elétrica por sempre proporcionar desafios que trouxessem boas experiências na formação de um engenheiro com as soluções aplicadas.

## RESUMO

Com o cenário de evolução das tecnologias de geolocalização e automatização de processos torna-se mais importante a aplicação destas tecnologias em projetos que são capazes de uni-las de maneira concisa. O projeto apresentado neste trabalho consiste na construção de um veículo autônomo de superfície capaz de coletar amostras de água e informações sobre as condições climáticas em determinadas coordenadas. O veículo é equipado com sensores de temperatura, pressão atmosférica e umidade relativa do ar, além de um GPS e um magnetômetro, onde juntos são capazes de registrar em um cartão micro SD um pequeno relatório dos dados obtidos. A metodologia utilizada para a construção do veículo incluiu a seleção dos componentes necessários, a fabricação das peças em impressão 3D, a montagem dos componentes e a programação do sistema de controle. Foram realizados testes em piscina e lago para avaliar o desempenho do veículo, e os resultados foram satisfatórios. O projeto apresentado neste trabalho é uma demonstração do potencial das tecnologias de geolocalização e automatização de processos aplicadas a projetos de coleta de dados ambientais. A utilização de sensores e tecnologias já conhecidas em um modelo de estudo pode ser replicada em outras aplicações, contribuindo para o desenvolvimento de soluções sustentáveis e eficientes para o monitoramento ambiental.

**Palavras-chave:** veículos autônomos; água - qualidade - medição; motores; sistema de posicionamento global; engenharia de protótipos .

## ABSTRACT

With the scenario of evolution of geolocation technologies and process automation, the application of these technologies in projects that are able to unite them in a concise way becomes more important. The project presented in this work consists of the construction of an autonomous surface vehicle capable of collecting water samples and information on weather conditions at certain coordinates. The vehicle is equipped with temperature, atmospheric pressure and relative humidity sensors, in addition to a GPS and a magnetometer, which together are capable of recording a small report of the data obtained on a microSD card. The methodology used to build the vehicle included selecting the necessary components, manufacturing the parts in 3D printing, assembling the components and programming the control system. Tests were carried out in swimming pool and lake to evaluate the vehicle's performance, and the results were satisfactory. The project presented in this work is a demonstration of the potential of geolocation technologies and process automation applied to environmental data collection projects. The use of already known sensors and technologies in a study model can be replicated in other applications, contributing to the development of sustainable and efficient solutions for environmental monitoring.

**Keywords:** automated vehicles; water quality - measurement; motors; global positioning system; prototypes, engineering.

## LISTA DE FIGURAS

|                                                                                            |    |
|--------------------------------------------------------------------------------------------|----|
| Figura 1 – Construção da barragem da usina de Itaipu. . . . .                              | 15 |
| Figura 2 – Posição dos satélites para triangularização. . . . .                            | 20 |
| Figura 3 – Diferença do polo norte geográfico e o polo norte magnético. . . .              | 21 |
| Figura 4 – Mapa indicando a variação de declinação magnética ao longo do<br>globo. . . . . | 22 |
| Figura 5 – Influência da interferência ferro duro. . . . .                                 | 23 |
| Figura 6 – Influência da interferência ferro macio. . . . .                                | 23 |
| Figura 7 – Display OLED 128x64. . . . .                                                    | 28 |
| Figura 8 – Módulo GPS. . . . .                                                             | 28 |
| Figura 9 – Módulo Magnetômetro QMC5883L. . . . .                                           | 29 |
| Figura 10 – Flutuadores de espuma de polietileno cortados. . . . .                         | 31 |
| Figura 11 – Vista frontal do motor sem escova com a hélice. . . . .                        | 32 |
| Figura 12 – Vista traseira do motor sem escova. . . . .                                    | 33 |
| Figura 13 – Controlador eletrônico de velocidade usado no projeto. . . . .                 | 33 |
| Figura 14 – Impressora 3D utilizada na confecção de peças para o projeto. . .              | 34 |
| Figura 15 – Circuito com optoacopladores e mosfets. . . . .                                | 35 |
| Figura 16 – Circuito para sensor DS18S20. . . . .                                          | 36 |
| Figura 17 – Tela de abertura desenhada. . . . .                                            | 37 |
| Figura 18 – Display OLED mostrando a tela de abertura. . . . .                             | 37 |
| Figura 19 – Ícone de Bandeira. . . . .                                                     | 38 |
| Figura 20 – Ícone de bússola. . . . .                                                      | 38 |
| Figura 21 – Ícone de cartão de memória. . . . .                                            | 38 |
| Figura 22 – Ícone para ajuste do parâmetro de controle P. . . . .                          | 38 |
| Figura 23 – Ícone para ajuste do parâmetro de controle I. . . . .                          | 39 |
| Figura 24 – Ícone para ajuste do parâmetro de controle D. . . . .                          | 39 |
| Figura 25 – Ícone para ajuste do parâmetro de velocidade máxima. . . . .                   | 39 |
| Figura 26 – Ícone da tela de informações. . . . .                                          | 39 |
| Figura 27 – Ícone da tela para testes. . . . .                                             | 39 |
| Figura 28 – Ícone para ação dos motores individualmente. . . . .                           | 40 |
| Figura 29 – Estrutura de suporte dos motores e espumas. . . . .                            | 40 |

|                                                                                                                    |    |
|--------------------------------------------------------------------------------------------------------------------|----|
| Figura 30 – Anel de sustentação. . . . .                                                                           | 41 |
| Figura 31 – Chassi de união central. . . . .                                                                       | 41 |
| Figura 32 – Chassi superior. . . . .                                                                               | 42 |
| Figura 33 – Caixa das placas. . . . .                                                                              | 43 |
| Figura 34 – Caixa das baterias. . . . .                                                                            | 43 |
| Figura 35 – Caixa das bombas de água. . . . .                                                                      | 44 |
| Figura 36 – Suporte de garrafas. . . . .                                                                           | 45 |
| Figura 37 – Tampa para garrafa com respiro. . . . .                                                                | 45 |
| Figura 38 – Alojamento para magnetômetro (QMC5883) e sensor de umidade,<br>temperatura e pressão (BME280). . . . . | 46 |
| Figura 39 – Suporte para barra de rosca em forma de X. . . . .                                                     | 47 |
| Figura 40 – Alojamento mangueiras e sensor de temperatura. . . . .                                                 | 47 |
| Figura 41 – Montagem de todos os elementos por meio de software. . . . .                                           | 48 |
| Figura 42 – Curvas amostradas e corrigidas obtidas com magnetômetro. . . . .                                       | 50 |
| Figura 43 – Modelo de sistema semelhante ao encontrado no projeto. . . . .                                         | 52 |
| Figura 44 – Planilha com interface gráfica por VBA construída. . . . .                                             | 55 |
| Figura 45 – Vista superior da placa montada com os componentes. . . . .                                            | 57 |
| Figura 46 – Verso da placa montada com os componentes. . . . .                                                     | 58 |
| Figura 47 – Montagem das duas primeiras bombas na caixa. . . . .                                                   | 58 |
| Figura 48 – Bombas montadas com a trava central e a trava superior. . . . .                                        | 59 |
| Figura 49 – Bombas montadas na caixa e seladas com silicone. . . . .                                               | 60 |
| Figura 50 – Flutuadores montados nos anéis de sustentação. . . . .                                                 | 61 |
| Figura 51 – Vista lateral destacando os flutuadores e um dos motores. . . . .                                      | 61 |
| Figura 52 – Suporte de garrafas montado. . . . .                                                                   | 62 |
| Figura 53 – Montagem do suporte das garrafas com as garrafas posicionadas. . . . .                                 | 62 |
| Figura 54 – Vista interna do sensor BME280. . . . .                                                                | 63 |
| Figura 55 – Vista superior com o sensor posicionado e tampa montada. . . . .                                       | 64 |
| Figura 56 – Vista do suporte de mangueiras e sensor montado. . . . .                                               | 65 |
| Figura 57 – Vista isométrica do projeto montado. . . . .                                                           | 65 |
| Figura 58 – Dados obtidos de posição e distancia do alvo. . . . .                                                  | 67 |
| Figura 59 – Coordenadas amostradas e apontadas. . . . .                                                            | 68 |

|                                                                                                             |    |
|-------------------------------------------------------------------------------------------------------------|----|
| Figura 60 – Esquema de posicionamento do veículo para simulação de coleta de água. . . . .                  | 69 |
| Figura 61 – Piscina em que foi realizado o primeiro teste em água. . . . .                                  | 70 |
| Figura 62 – Veículo posicionado dentro da piscina. . . . .                                                  | 70 |
| Figura 63 – Altura de submersão do veículo. . . . .                                                         | 71 |
| Figura 64 – Vista do lago do Recanto Fritzke por satélite. . . . .                                          | 72 |
| Figura 65 – Rota programada para o primeiro teste no lago. . . . .                                          | 73 |
| Figura 66 – Veículo durante o processo de navegação. . . . .                                                | 74 |
| Figura 67 – Rota do veículo durante a navegação do primeiro teste no lago. . .                              | 74 |
| Figura 68 – Dados obtidos com a amostragem do primeiro teste no lago. . . .                                 | 75 |
| Figura 69 – Diferença dos ângulos apontados para os ângulos alvos durante o primeiro teste no lago. . . . . | 76 |
| Figura 70 – Amostragem dos valores de erro durante o primeiro teste no lago. .                              | 77 |
| Figura 71 – Amostra de água coletada do lago. . . . .                                                       | 78 |
| Figura 72 – Rota programada para o segundo teste no lago. . . . .                                           | 79 |
| Figura 73 – Rota do veículo durante a navegação do segundo teste no lago. . .                               | 80 |
| Figura 74 – Dados obtidos com a amostragem do segundo teste no lago. . . .                                  | 81 |
| Figura 75 – Diferença dos ângulos apontados para os ângulos alvos durante o segundo teste no lago. . . . .  | 82 |
| Figura 76 – Amostragem dos valores de erro durante o segundo teste no lago. .                               | 83 |
| Figura 77 – Veículo armazenado dentro do baú da moto para transporte. . . .                                 | 84 |
| Figura 78 – Local em que houve o dano durante transporte. . . . .                                           | 84 |

## **LISTA DE TABELAS**

|                                                                                      |           |
|--------------------------------------------------------------------------------------|-----------|
| <b>Tabela 1 – Lista de materiais . . . . .</b>                                       | <b>56</b> |
| <b>Tabela 2 – Pontos de alvo para o primeiro teste no estacionamento . . . . .</b>   | <b>66</b> |
| <b>Tabela 3 – Relatório de amostras no primeiro teste (estacionamento) . . . . .</b> | <b>67</b> |
| <b>Tabela 4 – Pontos de alvo para o primeiro teste no lago . . . . .</b>             | <b>73</b> |
| <b>Tabela 5 – Relatório de amostras do primeiro teste no lago. . . . .</b>           | <b>77</b> |
| <b>Tabela 6 – Pontos de alvo para o segundo teste no lago . . . . .</b>              | <b>78</b> |
| <b>Tabela 7 – Relatório de amostras do segundo teste no lago. . . . .</b>            | <b>80</b> |
| <b>Tabela 8 – Dados das células de baterias. . . . .</b>                             | <b>83</b> |

## LISTA DE ABREVIATURAS E SIGLAS

### Siglas

|        |                                                            |
|--------|------------------------------------------------------------|
| BLDC   | <i>Brushless Direct Current</i>                            |
| CAD    | <i>Computer Aided Design</i>                               |
| CETES  | Companhia Ambiental do Estado de São Paulo                 |
| EEPROM | <i>Electrically-Erasable Programmable Read-Only Memory</i> |
| ESC    | <i>Electronic Speed Controller</i>                         |
| GPS    | <i>Global Positioning System</i>                           |
| I2C    | <i>Inter-Integrated Circuit</i>                            |
| IOC    | <i>Initial Operational Capability</i>                      |
| OLED   | <i>Organic Light-Emitting Diode</i>                        |
| PID    | Proporcional Integral Derivativo                           |
| PVC    | Policloreto de Vinila                                      |
| SD     | <i>Secure Digital</i>                                      |
| SPI    | <i>Serial Peripheral Interface</i>                         |
| SWATH  | <i>Small Waterplane Area Twin Hull</i>                     |
| WMM    | <i>World Magnetic Model</i>                                |



## SUMÁRIO

|            |                                                    |           |
|------------|----------------------------------------------------|-----------|
| <b>1</b>   | <b>INTRODUÇÃO</b>                                  | <b>14</b> |
| <b>1.1</b> | <b>Objetivos</b>                                   | <b>16</b> |
| 1.1.1      | Objetivo geral                                     | 16        |
| 1.1.2      | Objetivo específico                                | 16        |
| <b>1.2</b> | <b>Justificativa</b>                               | <b>17</b> |
| <b>1.3</b> | <b>Estrutura do trabalho</b>                       | <b>18</b> |
| <b>2</b>   | <b>REFERENCIAL TEÓRICO</b>                         | <b>19</b> |
| <b>2.1</b> | <b>GPS e Geolocalização</b>                        | <b>19</b> |
| <b>2.2</b> | <b>Orientação magnética</b>                        | <b>20</b> |
| <b>2.3</b> | <b>Princípio de Arquimedes</b>                     | <b>23</b> |
| <b>2.4</b> | <b>Controle PID</b>                                | <b>24</b> |
| <b>3</b>   | <b>MATERIAIS E MÉTODOS</b>                         | <b>26</b> |
| <b>3.1</b> | <b>Materiais</b>                                   | <b>26</b> |
| 3.1.1      | Microcontrolador escolhido                         | 26        |
| 3.1.2      | Linguagem de programação                           | 26        |
| 3.1.3      | Bibliotecas                                        | 27        |
| 3.1.3.1    | U8g2                                               | 27        |
| 3.1.3.2    | TinyGPSPlus                                        | 27        |
| 3.1.3.3    | QMC5883L Compass                                   | 28        |
| 3.1.3.4    | ESP32 Servo                                        | 29        |
| 3.1.3.5    | Adafruit BME280                                    | 29        |
| 3.1.3.6    | DallasTemperature                                  | 30        |
| 3.1.3.7    | Bibliotecas nativas                                | 30        |
| 3.1.4      | Solução para flutuação e modelo de casco utilizado | 30        |
| 3.1.5      | Sistema de propulsão                               | 31        |
| 3.1.6      | Software de desenho 3D                             | 33        |
| 3.1.7      | Impressão 3D                                       | 34        |
| <b>3.2</b> | <b>Métodos</b>                                     | <b>35</b> |
| <b>3.3</b> | <b>Elaboração dos circuitos eletrônicos</b>        | <b>35</b> |
| 3.3.1      | Subcircuito de mosfets                             | 35        |

|            |                                                                                                       |           |
|------------|-------------------------------------------------------------------------------------------------------|-----------|
| 3.3.2      | Sub circuito para uso do sensor de temperatura da água DS18S20 . . . . .                              | 36        |
| <b>3.4</b> | <b>Construção do menu de interface . . . . .</b>                                                      | <b>36</b> |
| 3.4.1      | Tela de abertura . . . . .                                                                            | 36        |
| 3.4.2      | Ícones de menu . . . . .                                                                              | 37        |
| <b>3.5</b> | <b>Modelagem e construção mecânica . . . . .</b>                                                      | <b>40</b> |
| 3.5.1      | Estrutura dos Motores e Espuma . . . . .                                                              | 40        |
| 3.5.2      | Chassi de união central . . . . .                                                                     | 41        |
| 3.5.3      | Chassi superior . . . . .                                                                             | 42        |
| 3.5.4      | Caixa para as placas . . . . .                                                                        | 42        |
| 3.5.5      | Caixa para as baterias . . . . .                                                                      | 42        |
| 3.5.6      | Caixa para as bombas . . . . .                                                                        | 44        |
| 3.5.7      | Suporte e tampas para garrafas . . . . .                                                              | 45        |
| 3.5.8      | Alojamento para magnetômetro e sensor de temperatura, umidade e pressão                               | 46        |
| 3.5.9      | Suporte para mangueiras e sensor de temperatura da água . . . . .                                     | 46        |
| 3.5.10     | Montagem dos modelos projetados . . . . .                                                             | 47        |
| <b>3.6</b> | <b>Algoritmos implementados no projeto . . . . .</b>                                                  | <b>48</b> |
| 3.6.1      | Rotina de calibração do magnetômetro . . . . .                                                        | 48        |
| 3.6.2      | Calculo de angulo com relação ao norte geográfico . . . . .                                           | 49        |
| 3.6.3      | Leitura de coordenadas do cartão micro SD . . . . .                                                   | 50        |
| 3.6.4      | Cálculo da direção entre a coordenada atual e coordenada alvo e a respec-<br>tiva distância . . . . . | 51        |
| 3.6.5      | Controle de direção . . . . .                                                                         | 51        |
| 3.6.6      | Algoritmo para realização de atividades nos pontos escolhidos . . . . .                               | 53        |
| 3.6.7      | Algoritmo de navegação . . . . .                                                                      | 54        |
| <b>3.7</b> | <b>Programa criado em VBA e excel para a entrada de coordenadas</b>                                   | <b>54</b> |
| <b>4</b>   | <b>RESULTADOS . . . . .</b>                                                                           | <b>56</b> |
| <b>4.1</b> | <b>Montagem da placa . . . . .</b>                                                                    | <b>57</b> |
| <b>4.2</b> | <b>Montagem das bombas . . . . .</b>                                                                  | <b>58</b> |
| <b>4.3</b> | <b>Montagem dos flutuadores e motores . . . . .</b>                                                   | <b>60</b> |
| <b>4.4</b> | <b>Montagem dos suportes de garrafas . . . . .</b>                                                    | <b>61</b> |
| <b>4.5</b> | <b>Montagem do suporte do magnetômetro e sensor de temperatura,<br/>umidade e pressão . . . . .</b>   | <b>62</b> |

|             |                                                                            |            |
|-------------|----------------------------------------------------------------------------|------------|
| <b>4.6</b>  | <b>Montagem do suporte de mangueiras e sensor de temperatura . .</b>       | <b>64</b>  |
| <b>4.7</b>  | <b>Montagem final . . . . .</b>                                            | <b>65</b>  |
| <b>4.8</b>  | <b>Testes em ambiente seguro (sem água) . . . . .</b>                      | <b>66</b>  |
| 4.8.1       | Primeiras dificuldades encontradas . . . . .                               | 67         |
| <b>4.9</b>  | <b>Testes fora da água para ajuste do tempo das bombas . . . . .</b>       | <b>68</b>  |
| <b>4.10</b> | <b>Testes em piscina pequena com água . . . . .</b>                        | <b>69</b>  |
| 4.10.1      | Dificuldades encontradas o teste realizado em água dentro de piscina . . . | 71         |
| <b>4.11</b> | <b>Testes do veículo em condições reais . . . . .</b>                      | <b>72</b>  |
| 4.11.1      | Primeiro teste no lago . . . . .                                           | 72         |
| 4.11.2      | Segundo teste no lago . . . . .                                            | 76         |
| 4.11.3      | Desempenho das baterias . . . . .                                          | 82         |
| 4.11.4      | Dificuldades encontradas para os testes no lago . . . . .                  | 83         |
| <b>5</b>    | <b>CONCLUSÃO . . . . .</b>                                                 | <b>86</b>  |
|             | <b>REFERÊNCIAS . . . . .</b>                                               | <b>87</b>  |
|             | <b>APÊNDICE A DIAGRAMA E SILK PCB DA PLACA PRINCIPAL</b>                   | <b>91</b>  |
|             | <b>APÊNDICE B DIAGRAMA E SILK PCB DA PLACA DE IN-</b>                      |            |
|             | <b>TERFACE . . . . .</b>                                                   | <b>95</b>  |
|             | <b>APÊNDICE C DIAGRAMA E SILK PCB DA PLACA DAS BA-</b>                     |            |
|             | <b>TERIAS . . . . .</b>                                                    | <b>99</b>  |
|             | <b>APÊNDICE D CÓDIGO ESCRITO PARA O PROGRAMA . .</b>                       | <b>103</b> |

## 1 INTRODUÇÃO

O escopo do trabalho apresentado é estudar o desenvolvimento do projeto de um veículo autônomo de superfície do qual tem como objetivo melhorar a qualidade de amostragem de água em reservatórios sem perturbação. Com um embasamento de desenvolvimento de mínimo produto viável, é importante estabelecer alguns objetivos e metas no estudo deste trabalho, onde o projeto precisa atender as necessidades básicas estipuladas e trazer ao respectivo usuário a possibilidade de realizar com sucesso o que é esperado sem grandes dificuldades. Ao levar isso em conta, espera-se que o projeto final seja capaz de com algumas instruções navegar do ponto A ao ponto B sucessivamente até a quantidade de pontos desejada e depois retornar a sua origem, podendo o número de destinos ser configurado e ajustado conforme a necessidade. Por fim, também deve ser possível em cada um dos pontos de parada coletar medições de temperatura, pressão, umidade relativa, e marcar a posição em que foi feita cada uma das medidas bem como ser capaz de realizar amostragem de água nos pontos escolhidos.

Atualmente os veículos autônomos são usados para diversas aplicações, como pode ser visto em carros autônomos que fazem entregas (RINALDI, 2021) ou então em atividades até mais complexas que consistem em transportar passageiros por meio de serviço de táxi ou similar como tem sido iniciado pela Ford nos Estados Unidos (Sá, 2021).

Fazendo um paralelo com o que acontece com os carros, os veículos autônomos de superfície (VAS) também estão em desenvolvimento a um tempo, sendo que já é possível encontrar projetos como o apresentado pela Holos Brasil (SILVEIRA, 2017), onde consiste de um modelo que possui módulos fotovoltaicos que dão a possibilidade do veículo passar longos períodos navegando sem precisar parar para recarregar as suas baterias. Veículos Autônomos de superfície vem sendo colocados em pauta, já que a sua existência proporciona operações que antes seriam perigosas e custosas de tempo para atividades mais simples que precisam ser direcionadas e precisas (RASHID *et al.*, 2022).

A coleta de amostras de água se torna interessante quando se coloca como temática a necessidade de monitorar a qualidade da água de diversos pontos com uma certa variedade de dados e com uma grande diversidade de ambientes para o mesmo efluente, sendo que cada ponto do efluente pode mostrar um comportamento diferente quando comparados entre si (JUNIOR; LIMA; OLIVEIRA, 2012).

Quando existe a construção de barragens em reservatórios ocorre uma mudança na geografia no local, sendo que começa a ser demonstrada uma diferença de como as substâncias são sedimentadas e distribuídas ao longo do reservatório de água. Um dos eventos em que isso se tornou um fator que pode ser considerado foi durante a construção da barragem da usina hidroelétrica de Itaipu conforme visto na Figura 1.

Uma das formas de se tornar possível a operação de um veículo autônomo é demonstrada por Almeida (2014), apresentando uma abordagem de uso de sensores de baixo custo, usando

**Figura 1 – Construção da barragem da usina de Itaipu.**



**Fonte: (GUTIERREZ, 1980).**

tecnologia MEMS, com altas taxas de ruído e assim apresentando os métodos de filtragem. Neste caso o veículo possui a forma de trimarã, garantindo a sua estabilidade na navegação.

Também é apresentado por Oliveira (2021) uma outra maneira de se realizar a construção do veículo de superfície, porém o trabalho realizado mostra a construção do veículo controlado a distância, onde neste caso depende completamente de ter um operador mesmo que a distância dedicado a controlar o veículo durante a sua operação.

Outro trabalho interessante é o mostrado por Martins (2020) onde o autor também faz a construção do veículo autônomo fazendo do uso de um modelo de casco diferenciado, e também usando um motor de barco de maior porte. Um ponto importante a se observar neste trabalho é a funcionalidade de se coletar amostras de água por meio de bombas, e assim trazer de volta ao usuário as amostras coletadas não as testando diretamente no local como os demais trabalhos vistos.

O projeto mostrado por Amran *et al.* (2021) aborda um modelo de veículo subaquático que é capaz de realizar amostragem diretamente no local em que o veículo está por meio de sensores de pH, turbidez da água e temperatura. Este tem por vantagem poder verificar diretamente as amostras sem a interferência de ter carregado elas até o usuário, e assim não corrompendo as amostras por variações de características como por exemplo, a temperatura.

De forma a mostrar algumas aplicações deste trabalho, pode-se tomar como base a portaria de consolidação nº 5 (SAÚDE, 2017) da qual se discute a forma em que os parâmetros da água potável devem ser avaliados. Dentro dos parâmetros avaliados estão as características físicas, químicas e biológicas, onde os parâmetros físicos são a cor e a turbidez, os parâmetros químicos são o pH, o cloro livre e o fluoreto, e os biológicos representando a presença de coliformes totais no reservatório de água, sendo que todos estes na água potável devem estar dentro dos limites especificados pela portaria fazendo os testes e amostragens nas respectivas periodicidades citadas no documento.

Periodicamente, órgãos como a CETESB (ANA; CETESB, 2011) realizam o controle de amostragem de reservatórios com o intuito de avaliar a qualidade da água que está presente no local. Tal amostragem costuma ser feita sobre pontes, nas margens do reservatório, ou então por meio de embarcações tripuladas, onde a coleta é realizada de forma manual pelos operadores. Esta operação pode se tornar um pouco arriscada devido a necessidade do acesso humano no reservatório, bem como custosa devido a necessidade de locomoção de equipamentos e embarcações para o local de amostragem.

## **1.1 Objetivos**

### **1.1.1 Objetivo geral**

Mostrar o desenvolvimento de um projeto de um veículo autônomo de superfície, desde a sua concepção, desenhos, estrutura, programação e testes finais de modo que seja possível entender as tecnologias aplicadas reforçando e praticando as habilidades de engenharia em planejar e corrigir possíveis problemas durante a sua execução. Todos os procedimentos de rotinas elaboradas para o projeto serão realizado por meio de algoritmos de controle e atuação.

### **1.1.2 Objetivo específico**

Traçando os objetivos específicos deste trabalho, é necessário fazer um estudo e projeto de construção de um Veículo Autônomo de Superfície tendo em vista a introdução de elementos para geolocalização e orientação espacial sobre corpos d'água rasos e sem turbulências significativas. Dessa forma, planeja-se usar do recurso de automação do veículo para coletar amostras de água em reservatórios para posterior análise em laboratório, bem como dados estáticos de temperatura, umidade relativa e pressão atmosférica no instante da coleta. Dessa forma devem ser satisfeitos os seguintes objetivos específicos:

- Apresentar e justificar as tecnologias de comunicação utilizadas;

- Condicionar os dados fornecidos pelos sensores para que seja possível obter coordenadas de posição e orientação;
- Tornar possível a programação de missões em que o veículo fará a coleta de amostras na região escolhida;
- Validar a capacidade de autonomia do veículo, capacidade de controlar direções mesmo com pequenas perturbações, e de retorno ao ponto de origem.

Com estes objetivos alcançados, há a possibilidade de em trabalhos futuros atender maiores especificações e demandas de possíveis novas aplicações para o projeto.

## 1.2 Justificativa

Apresentar o projeto como solução para atender a atual demanda existente de se monitorar a qualidade de água em reservatórios diminuindo os custos necessários e aumentando a segurança para os operadores, bem como aumentar a quantidade de informações obtidas de um determinado local com poucas operações. Também aplicar os conhecimentos adquiridos durante a formação em bacharel em engenharia elétrica focando na aplicação de conceitos de eletrônica, desenvolvimento de produtos e projetos, programação e física.

O processo de amostragem de água em reservatórios para controle de qualidade pode ser uma tarefa complicada por diversos fatores, sendo alguns deles:

- Contaminação;
- Variação temporal;
- Dificuldades de acesso;
- Dificuldades na preservação das amostras;
- Custos;
- Habilidade técnica.

Destas dificuldades, a maioria delas pode ser constante no processo de amostragem de água, sendo que além de tudo, é uma atividade que pode tomar demasiado tempo em sua execução. Visando solucionar alguns destes problemas torna-se interessante a aplicação de veículos autônomos de superfície neste tipo de atividade, onde o veículo é capaz de ter uma boa acuracidade na coleta dos dados, ter uma velocidade de coleta razoável, necessitar de menos habilidade técnica dos operadores, além de dar maior segurança aos que são interessados na amostragem das respectivas fontes. As operações do veículo podem ser todas construídas por meio de algoritmos computacionais de controle e operação, dando assim a automatização de todos os processos usados no projeto.

### **1.3 Estrutura do trabalho**

O presente trabalho se divide em capítulos com a revisão de todo o conteúdo mais fundamental do trabalho, sendo visto no capítulo 2, abordando alguns pontos sobre a orientação do veículo autônomo de superfície e um pouco sobre a técnica de controle, seguindo a discussão com os materiais e métodos no capítulo 3, onde são abordados todos os elementos escolhidos para a construção do trabalho, bem como a apresentação das etapas desenvolvidas para a montagem do projeto. Ainda neste capítulo é apresentado como que os recursos já apresentados são encaixados de forma que sejam capazes de atender as necessidades do projeto. Por fim, no capítulo 4 são apresentados os resultados do processo de construção e testes do projeto em algumas condições de operação, bem como as dificuldades encontradas em cada uma das etapas, e finalizando o trabalho com o texto apresentado no capítulo 5 dando os apontamentos finais de todo o projeto realizado e discutido ao longo do texto.



## 2 REFERENCIAL TEÓRICO

Para realizar os primeiros estudos de viabilidade de aplicação, foi necessário o entendimento de algumas tecnologias como GPS, magnetômetro, e alguns conceitos físicos de corpos flutuantes em meios aquáticos.

### 2.1 GPS e Geolocalização

O GPS (*Global Positioning System*) é um sistema de navegação por satélite que permite determinar a localização geográfica exata de um objeto, pessoa ou veículo em qualquer lugar do planeta. O sistema é composto por uma rede de satélites orbitando a Terra, estações de controle em solo e receptores GPS que estão localizados nos dispositivos utilizados pelos usuários. O funcionamento do GPS é baseado no conceito de triangulação e os satélites GPS transmitem sinais de rádio que contêm informações sobre sua posição e hora (ZOGG, 2009). O receptor GPS no dispositivo do usuário recebe os sinais de pelo menos quatro satélites e, a partir das informações contidas nesses sinais, é capaz de determinar a distância entre o receptor e cada um dos satélites. Usando a técnica de triangulação, o receptor pode então calcular sua própria posição geográfica com uma precisão menor que 20 metros, levando em consideração as informações de distância dos satélites. Para isso, o receptor precisa de informações precisas sobre a posição dos satélites, bem como do tempo exato em que os sinais foram emitidos. Os satélites GPS estão equipados com relógios atômicos altamente precisos que permitem que eles sincronizem seus sinais com um alto grau de precisão. O receptor GPS, por sua vez, possui um relógio interno que é sincronizado com o sinal do satélite. A partir das informações de distância e do tempo sincronizado, o receptor pode calcular sua posição geográfica com grande precisão. O GPS é utilizado em diversas aplicações, desde navegação veicular até aeronáutica, mapeamento, geodésia, monitoramento ambiental, entre outras. Atualmente, existem também dispositivos portáteis que integram GPS, como *smartphones* e relógios inteligentes, permitindo que qualquer pessoa possa determinar sua localização em qualquer lugar do mundo.

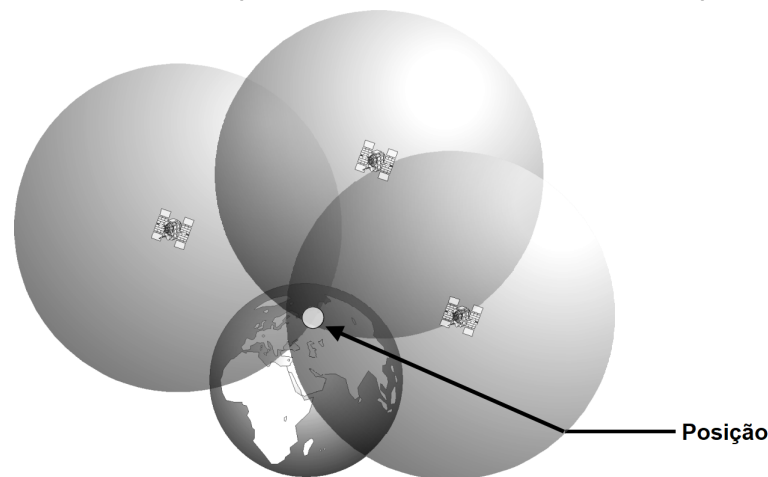
Basicamente o GPS opera comparando as diferenças entre o tempo em que o sinal foi emitido e o tempo em que o sinal foi recebido. Sabendo a velocidade de transmissão do sinal, e com mais de uma fonte de sinal, é possível saber por meio de triangularização em qual ponto um dispositivo está. O conceito é semelhante ao que acontece durante uma tempestade com raios, em que com o instante em que é visto o relâmpago, e com o tempo que leva até ouvi-lo é possível estimar a distância do ponto atual até o ponto que foi atingido pelo raio em questão.

De acordo com o que é introduzido por (EL-RABBANY, 2002), um sistema de GPS válido é composto por um conjunto de pelo menos 24 satélites dos quais compõem uma constelação, sendo esta a capacidade operacional inicial, conhecida por IOC (*Initial Operational*

*Capabiliy*). A IOC foi finalizada em 1993, onde é distribuída por 4 satélites em cada um dos seis planos orbitais do planeta Terra.

Com isso, é importante levar em conta que para se ter informações de latitude e longitude é necessária a conexão com pelo menos 3 satélites, e caso a altitude também precise ser conhecida, é necessário a conexão com 4 satélites. Uma ilustração disso pode ser visto na Figura 2. Com as informações de Latitude e Longitude é possível calcular distâncias e até mesmo realizar medidas geográficas com uma precisão razoável.

**Figura 2 – Posição dos satélites para triangularização.**



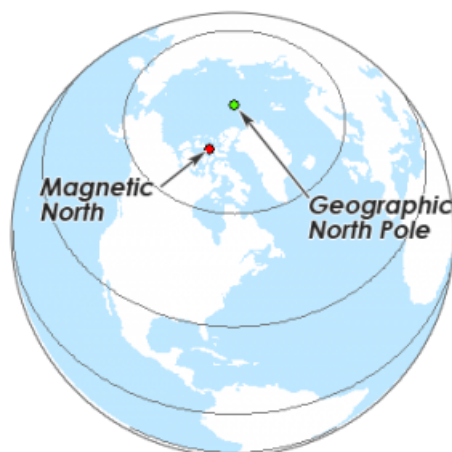
**Fonte: Adaptado de (ZOGG, 2009).**

## **2.2 Orientação magnética**

Uma das possíveis formas de se ter um referencial de direção em qualquer lugar do planeta é por meio do campo magnético terrestre. Este por sua vez existe devido a grande massa de ferro presente no núcleo do planeta. O polo norte geográfico fica aproximadamente no meio do oceano ártico (GEOGRAPHY, 2022), onde é o ponto em que as linhas de longitude convergem. Já o polo norte magnético fica sobre a ilha Ellesmere, situada no norte do Canadá, na região de Qikiqtaaluk do território de Nunavut. Este é o ponto em que as linhas magnéticas "entram" no plano terrestre e também este é o ponto para qual as bússolas tradicionais apontam. Assim como as bússolas, um magnetômetro bem calibrado pode cumprir a mesma função, dando informações sobre as direções em que existe um fluxo magnético próximo. O magnetômetro é um instrumento utilizado para medir o campo magnético em um determinado local por meio de sua intensidade. Ele é utilizado em diversas áreas, como geologia, exploração mineral, estudos de oceanografia, arqueologia, entre outras áreas de estudo.

Entre os dois polos já discutidos existe uma inclinação angular, chamada de "inclinação magnética" que se trata da diferença entre a direção do polo geográfico e o polo magnético, porém dependendo do local em que é realizada a medida, o valor pode variar, inclusive no

**Figura 3 – Diferença do polo norte geográfico e o polo norte magnético.**



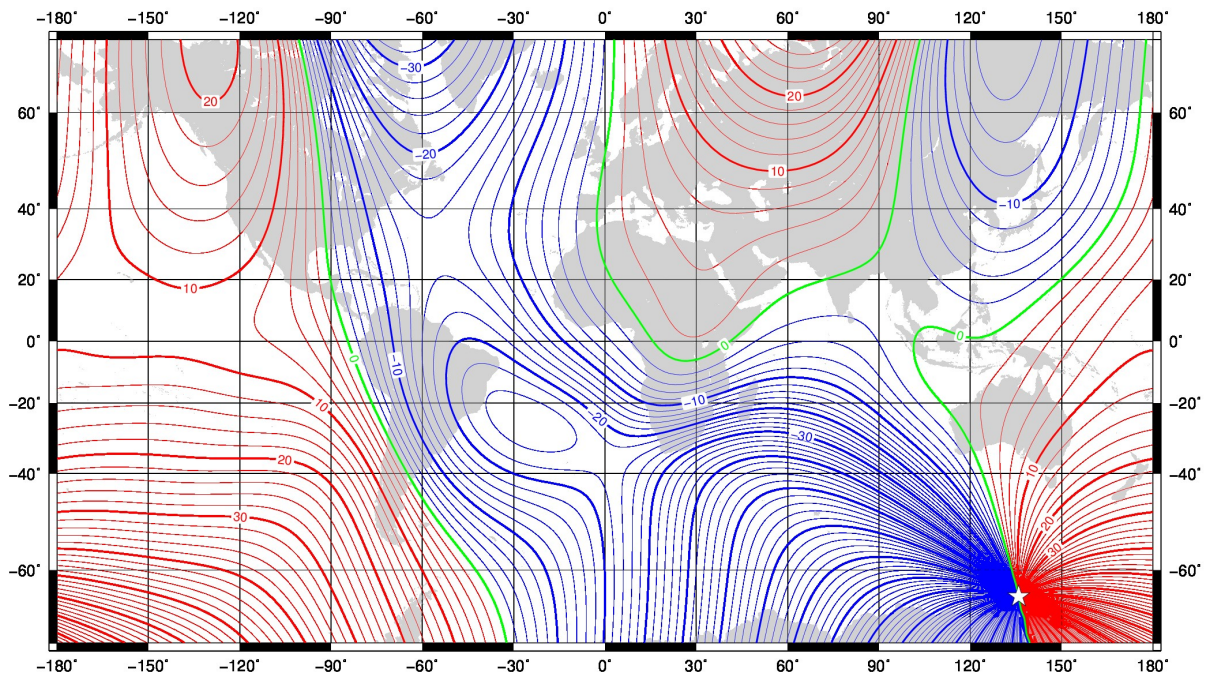
**Fonte: Adaptado de (GEOGRAPHY, 2022).**

decorrer do tempo. A diferença entre o polo geográfico e a direção em que uma bússola aponta é chamada de "declinação magnética". Alguns métodos de cálculo já foram criados para se obter o valor angular de declinação magnética em determinado ponto do globo, porém estes métodos de cálculo possuem validade e não podem ser usados indefinidamente. Um dos métodos atualmente usados é o WMM 2020 (World Magnetic Model) do qual oferece uma precisão suficiente para a aplicação em dispositivos modernos. Um mapa mostrando a variação da declinação magnética pode ser visto na Figura 4, onde neste mapa é indicado as curvas de declinação magnética observadas em 2020, sendo que este método de cálculo é válido até 2025. As curvas em vermelho indicam uma declinação magnética positiva (leste), enquanto as curvas em azul indicam uma declinação magnética negativa (oeste). Cada uma das curvas possui um passo de 2 graus de declinação entre si.

Existem vários tipos de magnetômetros, cada um com sua própria técnica de medição. Alguns utilizam um magnetômetro de *fluxgate*, que mede a variação do campo magnético em uma bobina quando é submetida a um campo magnético externo. Outros utilizam um magnetômetro de prótons, que mede a frequência de precessão de prótons em um campo magnético uniforme (KHOSRAVI; ALIZADEH; POURMAHDIAN, 2020). O magnetômetro em questão a ser aplicado neste trabalho se trata de um modelo composto por sensores magneto resistivos de três eixos, onde basicamente o campo magnético aplicado é convertido em uma diferença de potencial (CORPORATION, 2016) que por sua vez pode ser convertido para um sinal digital e assim é possível medir a intensidade do campo, sabendo a sua direção.

Uma das aplicações mais importantes do magnetômetro é na navegação, pois ele pode ajudar a determinar a direção do campo magnético da Terra. No entanto, é importante considerar a declinação magnética ao utilizar um magnetômetro para navegação para determinar corretamente a direção do norte verdadeiro. Caso contrário, a direção indicada pelo magnetômetro pode estar deslocada em relação à direção correta.

**Figura 4 – Mapa indicando a variação de declinação magnética ao longo do globo.**

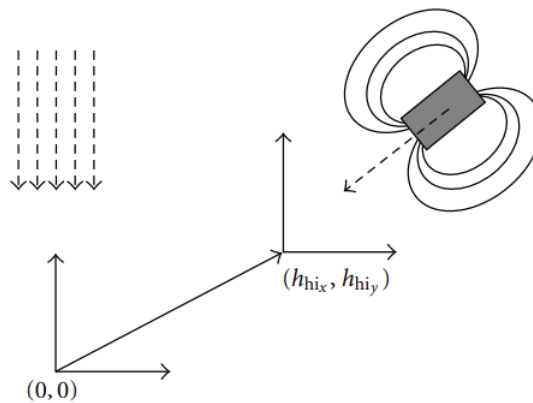


**Fonte: Adaptado de (SURVEY, 2020).**

Para calibrar o magnetômetro são conhecidas diversas abordagens matemáticas e complexas como a apresentada por Liu e Zhu (2014), onde são aplicadas técnicas de calibração e de compensação de erro dos valores medidos considerando todos os três eixos, manipulando as coordenadas deslocadas antes da calibração de forma que seja possível se ter uma transformação referenciada nos valores corretos. Para o escopo do trabalho, por se tratar de um deslocamento apenas no plano de superfície da água, pode ser tomada uma abordagem mais simples. Como mostrado por ShawnHymel (2022) é possível tomar um caminho mais resumido para se conseguir sucesso na calibração. Basicamente o método consiste na consideração de dois fatores que causam as medidas errôneas, sendo elas chamadas por ferro duro (*hard iron*) ou ferro macio (*soft iron*).

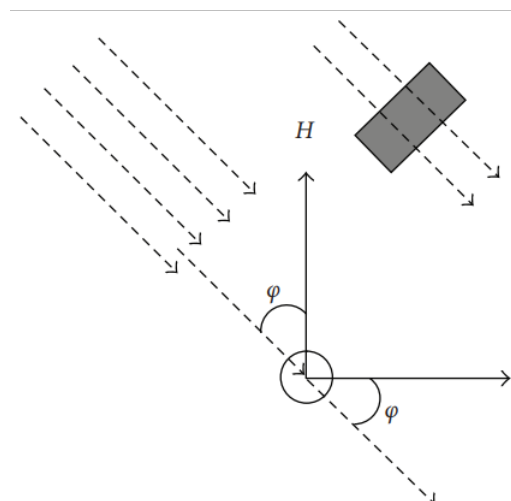
A interferência por ferro duro provém de fontes de campo magnético fixos, como ímãs, motores, alto falantes ou equipamentos que possuem tal efeito por natureza. Este por sua vez faz com que o valor médio medido por um magnetômetro fique fora da origem, ou seja, o valor entre o valor mínimo e o valor máximo não está localizado em zero quando o campo magnético terrestre é medido como indicado na Figura 5. Já a interferência por ferro macio ocorre devido a presença de materiais metálicos próximos ao magnetômetro, onde esta por sua vez tem um comportamento de condutor do campo magnético terrestre fazendo com que o campo seja no mesmo sentido do material ferromagnético conforme visto na Figura 6. Assim, como resultado da interferência, os valores máximos e mínimos não possuem simetria quando comparado entre os eixos medidos (x e y por exemplo) em que quando plotados em um plano, mostram um comportamento elíptico.

**Figura 5 – Influência da interferência ferro duro.**



Fonte: Adaptado de (RENAUDIN; AFZAL; LACHAPELLE, 2010).

**Figura 6 – Influência da interferência ferro macio.**



Fonte: Adaptado de (RENAUDIN; AFZAL; LACHAPELLE, 2010).

### 2.3 Princípio de Arquimedes

O princípio de Arquimedes é uma lei física formulada pelo famoso matemático e cientista grego Arquimedes. Esse princípio descreve o comportamento de corpos imersos em fluidos, como líquidos ou gases, e é amplamente utilizado para entender o conceito de flutuação.

Segundo o princípio de Arquimedes, um corpo total ou parcialmente imerso em um fluido sofre uma força de empuxo vertical para cima igual ao peso do fluido deslocado pelo corpo. Em outras palavras, o empuxo é igual ao peso do fluido que é deslocado pelo corpo submerso. Isso significa que, se um objeto flutuante, como um navio, deslocar uma quantidade de água equivalente ao seu próprio peso, ele estará em equilíbrio e não afundará. Se o objeto deslocar uma quantidade de água menor do que seu peso, ele afundará parcialmente, e se

deslocar uma quantidade maior de água do que seu peso, ele flutuará totalmente acima da superfície do líquido. (HALLIDAY; RESNICK; WALKER, 2013)

Assim como no caso do navio, o peso total do veículo autônomo de superfície deve ser equilibrado pelo empuxo gerado pelo fluido (água) que ele desloca. Para calcular a quantidade de peso que um veículo autônomo de superfície pode suportar deve-se primeiro determinar o volume das partes que ficarão submersas e a sua respectiva densidade, a densidade da água que pode ser aproximada para  $1000\text{kg}/\text{m}^3$  e o peso do fluido deslocado, que no caso é a água. Com essas informações é possível determinar o peso máximo que o veículo autônomo de superfície pode suportar sem afundar.

## 2.4 Controle PID

O controle PID, que significa Proporcional-Integral-Derivativo, é um dos métodos mais comuns e amplamente utilizados na área de controle de sistemas. Ele é amplamente aplicado em uma variedade de indústrias, como automação industrial, robótica, controle de processos e engenharia de sistemas. O controle PID é projetado para regular e estabilizar um sistema dinâmico, mantendo a variável controlada (também conhecida como variável de processo) o mais próximo possível de um valor de referência desejado. Isso é feito ajustando-se continuamente uma saída de controle com base no erro entre a variável controlada e o valor de referência. (DORF; BISHOP, 2022)

Dividindo o controle PID nas três componentes, pode-se descrevê-las da forma listada a seguir.

- Proporcional (P): A ação proporcional como o próprio nome diz é proporcional ao erro atual entre a variável controlada e o valor de referência. Quanto maior o erro, maior é a ação proporcional aplicada. Essa ação fornece uma resposta rápida, mas pode resultar em um erro de regime permanente se usada sozinha.
- Integral (I): A ação integral leva em consideração a soma do erro ao longo do tempo. Isso permite que o controlador responda não apenas ao erro atual, mas também a erros passados acumulados conforme a amostragem. A ação integral é eficaz para corrigir erros estáveis, como desvios constantes.
- Derivativo (D): A ação derivativa leva em consideração a taxa de mudança do erro ao longo do tempo. Isso ajuda a prever tendências futuras do erro e ajustar a ação de controle de acordo. A ação derivativa é eficaz para fornecer uma resposta rápida a mudanças rápidas no erro.

A combinação dessas três ações - proporcional, integral e derivativa - permite que o controle PID se adapte a diferentes condições do sistema e minimize o erro de controle. A contribuição de cada componente é ajustada por meio de constantes de ajuste conhecidas

como ganhos. A sintonia adequada dos parâmetros do controlador PID é crucial para obter um desempenho ideal, e existem várias técnicas de sintonia disponíveis, incluindo métodos manuais, métodos baseados em regras e algoritmos de otimização (FADALI; VISIOLI, 2013).

### 3 MATERIAIS E MÉTODOS

Para solucionar as dificuldades e alcançar os objetivos já discutidos no início deste trabalho, é necessário estipular as ferramentas e materiais que são necessários para a construção de um protótipo que seja capaz de realizar pequenas missões de coletas de dados. Os principais itens que precisam ser colocados como tópicos de estudo estão relacionados a escolha de estrutura de um veículo, a sua forma de locomoção, a sua forma de controle e sensoriamento, também como cada uma dessas coisas interage com o objetivo de tornar o sistema algo operacional e responsivo.

#### 3.1 Materiais

De modo a tornar a jornada de desenvolvimento do projeto algo possível, foram selecionadas algumas ferramentas e materiais que são úteis durante o processo de desenho mecânico, fabricação de peças e programação computacional dos elementos utilizados no protótipo.

##### 3.1.1 Microcontrolador escolhido

Na intenção de se ter um microcontrolador que não traga dificuldades com relação a desempenho e disponibilidade de interface, foi escolhido ESP32 como microcontrolador principal deste projeto. Nele deve ser gravado o programa que é responsável pela iteração entre todos os elementos presentes no trabalho. Este microcontrolador foi escolhido também por ser uma opção moderna, com bastante espaço para armazenamento para o programa escrito, e também por ser compatível com diversas bibliotecas que originalmente foram criadas para a plataforma *Arduino*. Este fator torna o desenvolvimento do trabalho mais focado na aplicação dos primeiros conceitos motivadores deste trabalho sem ter a necessidade de se dedicar um demasiado tempo explorando formas de comunicação e interpretação dos dados fornecidos pelos periféricos conectados ao microcontrolador. O ESP32 é uma opção com um bom desempenho computacional, do qual é importante pois existe a necessidade de se realizar operações quase que simultâneas, onde o tempo de cada operação é um fator que pode ser um problema caso haja alguma demora em suas execuções.

##### 3.1.2 Linguagem de programação

Com o objetivo de se usar a *Arduino IDE* para programar o microcontrolador escolhido, há a necessidade de fazer o uso da linguagem de programação C++. Esta linguagem de programação é a principal usada neste trabalho, sendo que esta é uma variação da linguagem C, porém contemplando a orientação a objetos, já que isto é essencial para o uso dos



recursos que estão presentes no projeto. A linguagem C++ associada as técnicas de programação direcionada a microcontroladores baseada na linguagem C faz uma interface do nível de linguagem de baixo nível (linguagem interna em microcontroladores) com uma linguagem de nível mais alto, que é próximo do que é falado pelos humanos. Para fazer a programação do microcontrolador, é importante ter em mãos um compilador que faz a conversão da linguagem C ou C++ em linguagem de máquina. Conforme (SOUZA, 2022) cita em sua publicação, o *Arduino IDE* pode ser usado como compilador que faz a conversão da linguagem C++ para o ESP32 usado como microcontrolador deste trabalho.

### 3.1.3 Bibliotecas

De modo a tornar ágil a construção do software usado no veículo autônomo de superfície foram usadas algumas bibliotecas compatíveis com ESP32 para a administração de todos os recursos necessários.

#### 3.1.3.1 U8g2

A biblioteca U8g2 foi desenvolvida por Oliver Kraus (KRAUS, 2021), sendo esta uma evolução da U8Glib, da qual é amplamente usada em pequenos projetos realizados pela comunidade. É notável a sua fama ao ver a quantidade de colaborações que a biblioteca em questão possui. Esta biblioteca é capaz de fazer a interface com vários tipos de displays gráficos, dando a capacidade de desenhar bitmaps nos displays, textos, números e caracteres especiais nas mais diversas fontes e tamanhos de letra. Como principal recurso da biblioteca usado neste trabalho está a simples escrita de texto, o recurso de desenho de caixas e também a possibilidade de se desenhar bitmaps em qualquer parte do display. Na Figura 7 é mostrado alguns elementos gráficos com o uso da biblioteca discutida.

#### 3.1.3.2 TinyGPSPlus

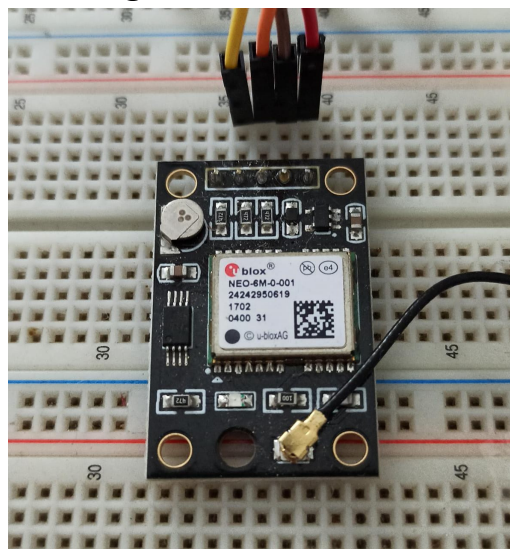
A biblioteca usada para interpretação dos dados fornecidos pelo módulo de GPS é a *TinyGPSPlus*. Esta biblioteca foi escrita por (HART, 2013), onde com ela é possível converter o dado fornecido pelo módulo Neo6M para dados específicos, como latitude, longitude, quantidade de satélites conectados e até mesmo informações de tempo. Com o auxílio desta biblioteca é possível obter principalmente as informações de posição, que são essenciais para a aplicação do veículo autônomo de superfície. Um módulo de GPS NEO6M é mostrado na Figura 8.

**Figura 7 – Display OLED 128x64.**



**Fonte: Do Autor (2023).**

**Figura 8 – Módulo GPS.**



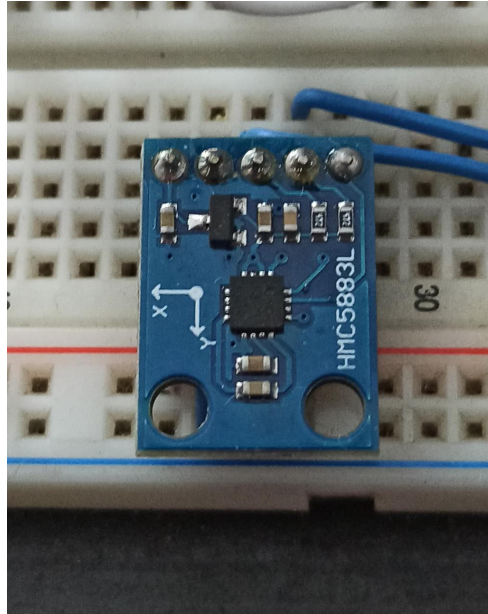
**Fonte: Do Autor (2023).**

### 3.1.3.3 QMC5883L Compass

Para fazer a comunicação e interpretação de dados fornecidos pelo magnetômetro, está sendo usada a biblioteca *QMC5883 Compass* (MPROGRAMS; NãVEKE, 2022) onde esta biblioteca basicamente converte os dados fornecidos pelo módulo magnetômetro de hexadecimal para valores que são correspondentes a intensidade magnética lida em cada um dos três eixos do magnetômetro. Com esta biblioteca também é possível implementar os valores de calibração, que nada mais é do que os valores mínimos e máximos lidos pelo magnetômetro em cada um dos eixos, sendo que a calibração é feita automaticamente e então é descontado os

erros encontrados nos valores brutos lidos pelo magnetômetro. A Figura 9 mostra o módulo de sensor com o magnetômetro discutido, do qual é compatível com esta biblioteca.

**Figura 9 – Módulo Magnetômetro QMC5883L.**



**Fonte: Do Autor (2023).**

#### 3.1.3.4 ESP32 Servo

Para facilitar a aplicação dos controladores eletrônicos de velocidade (*ESC*), é usada uma biblioteca que faz a interface de controle para motores servo. Como o controle dos *ESCs* são basicamente por tempo de pulso que consequentemente correspondem ao ângulo de inclinação de um servo entre 0 a 180 graus, torna-se simples e prática a aplicação fazendo o uso de uma biblioteca assim. A biblioteca usada no desenvolvimento do projeto do veículo é a Servo ESP32 (BENNETT, 2017) sendo que esta biblioteca é uma adaptação da biblioteca nativa usada em arduino, porém para o ESP32.

#### 3.1.3.5 Adafruit BME280

Para fazer a aplicação do leitor de temperatura, pressão atmosférica e umidade relativa implementados pelo módulo BME280 é usada uma biblioteca da Adafruit para BME280 que faz esta interface (ADAFRUIT, 2021). Esta biblioteca contém internamente a biblioteca Adafruit Sensor (ADAFRUIT, 2023) que faz o auxílio de conversão de unidades para os valores lidos pelos sensores. A biblioteca BME280 basicamente faz a interface de leitura e configuração do BME280 disponibilizando os seus respectivos valores lidos de forma prática e rápida.

### 3.1.3.6 DallasTemperature

Com o intuito de se fazer a interface para usar o sensor de temperatura DS18S20 é necessário o uso da biblioteca *Dallas Temperature* (BURTON, 2020) onde esta biblioteca faz a interface do sensor por meio do protocolo de comunicação *OneWire* do qual o sensor é conectado. A biblioteca é bastante simples e permite a obtenção dos valores de temperatura em graus Celsius e em graus Fahrenheit de forma direta.

### 3.1.3.7 Bibliotecas nativas

Algumas bibliotecas como *EEPROM*, *SD*, *I2C*, *SPI*, *OneWire* e *Math* são bibliotecas nativas da ArduinoIDE. Estas bibliotecas são disponibilizadas com o intuito de oferecer recursos de acesso a memória EEPROM do micro controlador, acesso, leitura e gravação de arquivos em cartão SD por meio do protocolo de comunicação SPI, e alguns outros protocolos sendo a dois fios (I2C) e a um fio (OneWire). Também a biblioteca *Math* que disponibiliza ao usuário funções matemáticas mais complexas como trigonometria por exemplo.

## 3.1.4 Solução para flutuação e modelo de casco utilizado

Buscando uma solução economicamente viável para se construir uma estrutura de veículo que deve flutuar sobre a água, foram escolhidos flutuadores de piscina para atender as necessidades do projeto, se comportando como boias posicionadas nas laterais do veículo. Este formato característico é conhecido como *SWATH (Small Waterplane Area Twin Hull)* e é apresentado no estudo mostrado por Rashid *et al.* (2022). Uma estrutura neste formato também é vista em catamarãs da qual fornece maior estabilidade em veículos de superfície, já que a área do corpo em contato com a superfície da água é reduzida e distribuída em dois pontos afastados do centro, diminuindo os efeitos e as influências de ondas e outras perturbações. Devido ao formato de flutuadores de piscina ser cilíndrico, é uma excelente opção ao levar em conta que deve ser fixada em uma estrutura planejada para ser feita em um protótipo, dispensando a necessidade de se ter um casco selado que seja capaz de sustentar o veículo sobre a água. Os flutuadores não absorvem água e possuem um volume considerável, já que são compostos por espuma de polietileno fazendo com que seu peso seja quase desprezível. Como o formato dos flutuadores é cilíndrico, é possível calcular facilmente o seu volume, e consequentemente o peso que ele é capaz de sustentar sem afundar na água. Para o desenvolvimento do projeto são usados dois flutuadores de 1,60m, cortados em seções de 35cm, totalizando 8 peças com 35cm. Assim os flutuadores podem ser divididos simetricamente, com 4 peças de 35cm em cada um dos dois lados do veículo. Os flutuadores cortados podem ser vistos na Figura 10.

**Figura 10 – Flutuadores de espuma de polietileno cortados.**



**Fonte: Do Autor (2023).**

Para determinar o peso máximo que o veículo suporta por flutuador, pode ser usado o princípio de Arquimedes, onde sabe-se que os flutuadores tem seção transversal cilíndrica, com cilindro externo de  $60mm$  de diâmetro, com furo central de  $25mm$  de diâmetro. Portanto o valor da área da seção transversal do flutuador é  $0,0023m^2$ , como o comprimento de cada cilindro é  $35cm$ , resulta em um volume de  $0,8178 * 10^{-3}m^3$  por cilindro. Para simplificação dos cálculos, considera-se a densidade dos cilindros como nula já que são extremamente leves quando comparado com o volume, praticamente não interferindo no cálculo o peso dos cilindros. Dessa forma a força que será resultante para cima para cada um dos cilindros será relacionada ao peso do volume de água deslocada por cilindro, que é de  $0,8178 * 10^{-3}m^3$ . Considerando a densidade da água como  $1000kg/m^3$ , o peso suportado resultante em cada um dos cilindros é de  $817,8g$  e como são 8 cilindros no total, o peso máximo suportado pelos cilindros sem que o veículo afunde será de aproximadamente  $6,542kg$ . Isso é mais do que suficiente para suportar todo o peso do veículo e das amostras coletadas sem grandes problemas.

### 3.1.5 Sistema de propulsão

No projeto em questão é necessária a implementação de um sistema de propulsão viável, que seja eficiente e que possua a potência necessária para fazer o deslocamento do veículo, sendo que os motores devem preferencialmente ficar submersos com o objetivo de simplificar a construção do projeto. Com o objetivo de atender essas necessidades, a escolha de motores de corrente contínua sem escovas é uma das melhores opções, devido a não precisarem ter energia passando por escovas que acabariam por ficarem submersas, diminuindo a sua vida útil e aumentando o risco de curto-circuitos. Como os motores de corrente contínua sem escova,

conhecidos também como *BLDC* (*Brushless DC*), usam apenas os enrolamentos do estator fixos na estrutura e os ímãs permanentes fixados no rotor, é garantido maior segurança e confiabilidade em seu uso.

Para este projeto, os motores devem ser capazes de deslocar um determinado volume de água para o sentido oposto ao de movimento do veículo, fazendo assim o deslocamento do conjunto completo. Tanto o deslocamento como o controle de direção é aplicado por meio de dois motores, que são acoplados no eixo radial do projeto com o objetivo de tornar mais estável e responsivo a mudanças de direções. Os motores podem ser instalados no meio da estrutura dos flutuadores, assim, com a diferença de velocidade de rotação entre os motores, o veículo é capaz de apontar para diferentes direções. Os motores são montados com pequenas hélices que são responsáveis por fazer o deslocamento da água como pode ser visto na Figura 11. Ambos os motores precisam ser selados e capazes de suportar a submersão na água e como os motores não possuem escova, são as melhores opções para aplicações deste tipo já que não existem contatos expostos na água que poderiam ocasionar a falha dos mesmos. A vista traseira do motor é mostrada na Figura 12 onde a saída dos cabos é selada também com silicone.

**Figura 11 – Vista frontal do motor sem escova com a hélice.**

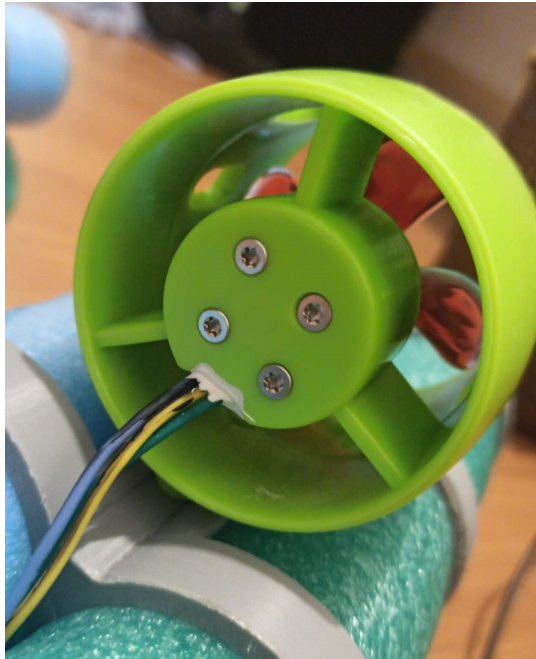


**Fonte: Do Autor (2023).**

Os motores *BLDC* precisam ser acionados por controladores eletrônicos de velocidade que são responsáveis por converter um sinal de pulso modulado para um sinal de três fases. O controlador eletrônico de velocidade pode ser abreviado para *ESC* (*Electronic Speed Controller*) é geralmente um dispositivo pequeno e energeticamente eficiente. O modelo usado no presente trabalho é mostrado na Figura 13, sendo que este controlador é da marca ZTW do modelo Shark 40A, do qual é um controlador específico para aplicações com veículos aquáti-



**Figura 12 – Vista traseira do motor sem escova.**



**Fonte: Do Autor (2023).**

cos. No manual do controlador (SCIENCE; TECHNOLOGY, 2020) é possível encontrar mais detalhes de aplicação e configuração do dispositivo.

**Figura 13 – Controlador eletrônico de velocidade usado no projeto.**



**Fonte: Do Autor (2023).**

### 3.1.6 Software de desenho 3D

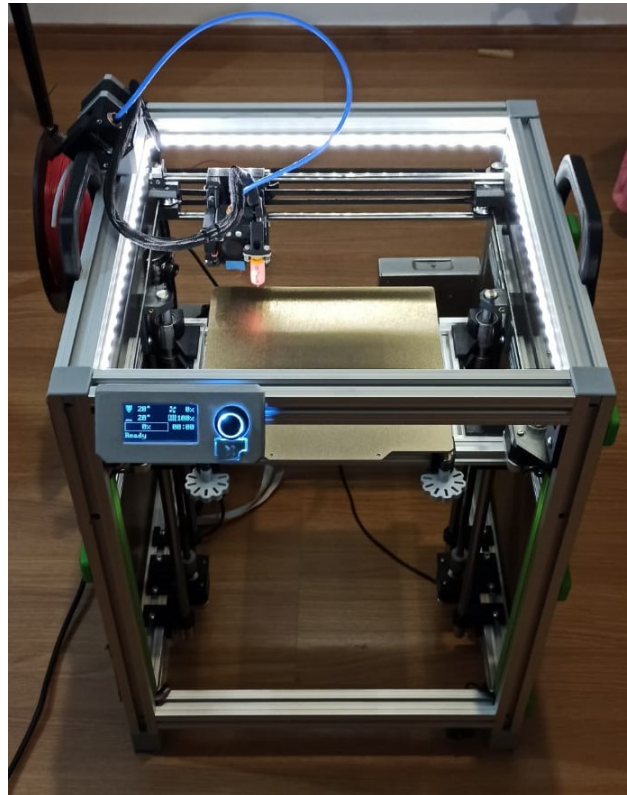
Para realizar testes de montagem e também dar a possibilidade de se projetar as peças antes de sua confecção, o aplicativo *Autodesk Fusion360* foi escolhido como software

de desenho para todas as peças mecânicas. A universidade em parceria com a *Autodesk*, empresa responsável pela criação deste software, disponibilizam de maneira gratuita a licença do aplicativo para fins estudantis, sendo que este software possui grande compatibilidade com formatos utilizados para desenho 3D. Além de atender muito bem as necessidades do projeto, este aplicativo não necessita de muitas habilidades para o seu uso, sendo uma das formas mais práticas de realizar a etapa de projeto mecânico do trabalho.

### 3.1.7 Impressão 3D

Para que seja possível confeccionar todas as peças mecânicas projetadas e utilizadas na construção do veículo autônomo de superfície, foi utilizada uma impressora 3D desenvolvida também pelo mesmo autor deste trabalho, onde é mostrada na Figura 14. Esta impressora possui volume de impressão de até 220x220x220mm, podendo este volume ser fechado para assim otimizar a possibilidade de uso de materiais de impressão. O material usado em todas as partes impressas é o ABS, onde este material possui uma resistência mecânica considerável, além também de ser bastante eficiente com relação ao seu custo.

**Figura 14 – Impressora 3D utilizada na confecção de peças para o projeto.**



**Fonte: Do Autor (2023).**

A impressora em questão utiliza um bico com diâmetro de 0,6mm, onde este é um diâmetro maior do que o mais utilizado em outras impressoras do mercado (0,4mm), possibilitando a confecção das peças de maneira rápida e suficientemente precisa.



## 3.2 Métodos

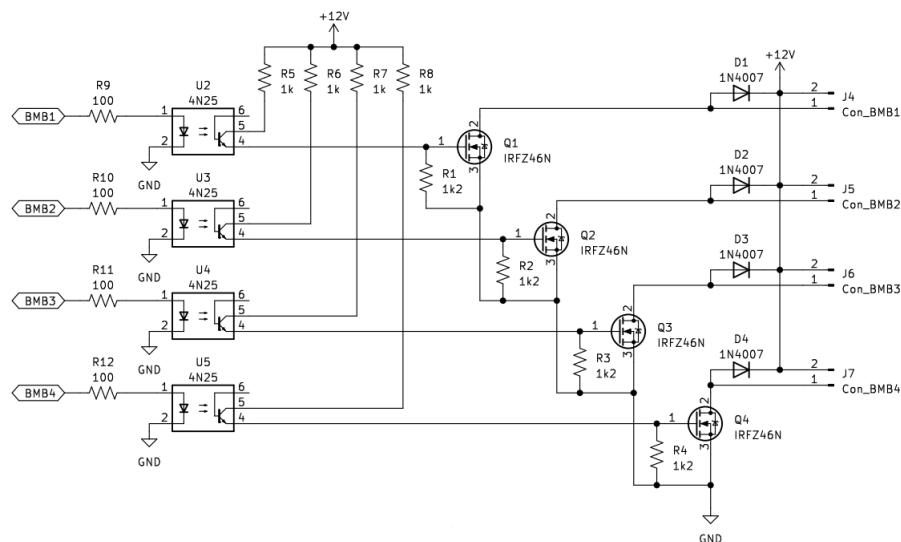
### 3.3 Elaboração dos circuitos eletrônicos

Com o objetivo de criar um ambiente de interface entre o microcontrolador ESP32 e os diversos componentes do projeto, foi elaborada uma pequena placa, com a mesma ideia de um shield para arduino, onde nesta placa deveriam estar dispostos os alojamentos para conexão do ESP32, fixação do leitor de cartões, integração dos circuitos para controle das bombas de água, e também o suporte para conexão dos demais elementos por meio de cabos e plugs da maneira mais profissional possível com as habilidades obtidas durante a formação. O esquemático completo da placa é mostrado no Apêndice A.

#### 3.3.1 Subcircuito de mosfets

Para que seja possível acionar as bombas de água por meio do microcontrolador, foi elaborado o circuito mostrado na Figura 15, onde os mosfets IRFZ46N são acionados por meio de optoacopladores 4N25, assim não é necessário que os pinos do microcontrolador estejam em contato diretamente com os mosfets.

**Figura 15 – Circuito com optoacopladores e mosfets.**



**Fonte: Do Autor (2023).**

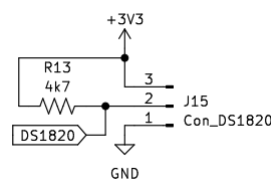
Este recurso de conexão dos mosfets por meio de optoacopladores torna o ESP32 mais protegido de possíveis falhas de curto que podem ocorrer no lado do mosfet, evitando inclusive possíveis interferências eletromagnéticas que podem ocorrer no acionamento dos motores das bombas. Porém apesar de que a corrente de saída do microcontrolador precisa ser apenas o suficiente para acionar um led interno no optoacoplador, este led possui corrente de operação de 60mA. No circuito é instalado também um diodo em antiparalelo com

os mosfets chaveadores com o objetivo de absorver correntes reversas providas dos motores acionados, assim, evitando que os mosfets tenham picos de tensão pelos terminais dos motores.

### 3.3.2 Sub circuito para uso do sensor de temperatura da água DS18S20

Para a conexão do sensor de temperatura DS18S20, foi usado um circuito muito simples, que compõe apenas um resistor de  $4k7$  que é conectado entre o pino de dados e a tensão de alimentação. Esta especificação é dada pelo datasheet do sensor de modo que seja possível a operação pelo método de comunicação a um fio (*One-Wire*).

**Figura 16 – Circuito para sensor DS18S20.**



**Fonte: Do Autor (2023).**

## 3.4 Construção do menu de interface

Para ser possível o ajuste de parâmetros do dispositivo, bem como dar comandos ao veículo e ter uma interface gráfica que seja intuitiva e funcional para realizar pequenos ajustes em campo, foi usado o display OLED de 128x64 pixels para tornar tal tarefa algo possível.

Com o uso da biblioteca `u8g2` é possível fazer desenhos no display com o uso de imagens convertidos para bitmaps. As imagens foram desenhadas manualmente com tamanhos de 128x64 pixels para imagens que ocupam o display inteiro, ou então em 16x16 pixels para os ícones. Assim, juntamente com as funcionalidades de texto disponibilizadas pela biblioteca citada, é possível fazer as operações conforme discutido a seguir. Todos os elementos foram desenhados com o uso de um site online gratuito chamado *Photopea*, porém poderia também ser feito da mesma forma em softwares mais completos como *Photoshop* ou então *GIMP*, porém, de forma a manter a praticidade sem necessidade de instalação, foi usada a plataforma mencionada

### 3.4.1 Tela de abertura

Ao iniciar o sistema, a primeira tela a ser mostrada é a tela de abertura. Ela consiste de uma pequena ilustração do que seria o veículo autônomo de superfície já montado, bem como também o texto descritivo do título do projeto. A tela desenhada mostrada na Figura 17 foi confeccionada e escrita completamente na plataforma citada no início desta seção.

**Figura 17 – Tela de abertura desenhada.**



**Fonte: Do Autor (2023).**

Esta tela foi implementada no veículo para aparecer durante 2 segundos, onde acaba tendo como funcionalidade principal mostrar ao usuário o escopo do programa desenvolvido e o projeto aplicado. A tela mostrada no display OLED pode ser vista na figura Figura 18

**Figura 18 – Display OLED mostrando a tela de abertura.**



**Fonte: Do Autor (2023).**

### 3.4.2 Ícones de menu

Os ícones de menu foram desenhados buscando trazer de forma ilustrativa os elementos gráficos ao display de forma que ficasse intuitivo e pudesse dar a oportunidade de se implementar elementos gráficos, já que o menu desenvolvido apenas com elementos textuais o poderia tornar confuso. Na Figura 19 é mostrado o ícone de uma bandeira, representando o comando de início do roteiro de coleta de amostras. Assim como este, cada um dos itens representa um sub menu com atividades que podem ser ajustadas. No caso do ícone de bandeira em questão, o subcomando realizado é o de iniciar o roteiro de navegação.

**Figura 19 – Ícone de Bandeira.****Fonte: Do Autor (2023).**

Com o objetivo de ilustrar a calibração do magnetômetro, foi inserido o ícone mostrado na figura Figura 20, onde este submenu possui como atividade a calibração do magnetômetro do qual mais instruções são dadas nas telas seguintes deste menu.

**Figura 20 – Ícone de bússola.****Fonte: Do Autor (2023).**

Para ilustrar a leitura dos dados de coordenadas e roteiro de navegação a partir do cartão micro SD, foi desenhado o ícone da Figura 21, onde é indicado um cartão de memória, representando a leitura dos dados a partir do micro SD. Caso o cartão não esteja inserido, é emitida uma mensagem de erro notificando para verificar o cartão onde se caso for inserido ou reinserido o cartão, será necessário reiniciar o dispositivo para que a leitura seja feita de maneira correta. Caso o arquivo "coordenadas.txt" não seja encontrado no cartão de memória ou caso o arquivo esteja vazio, será indicado que 0 pontos de coleta foram lidos. Caso esteja tudo correto para a leitura dos pontos desejados, após a leitura do cartão será indicado no display quantos pontos de coleta foram lidos, e caso seja um número maior que zero, será habilitada a opção de iniciar o roteiro de navegação.

**Figura 21 – Ícone de cartão de memória.****Fonte: Do Autor (2023).**

Os ícones indicados na Figura 22, Figura 23 e Figura 24 são os itens que representam a configuração dos parâmetros proporcional, integrador e derivativo respectivamente. Para que os valores sejam limitados ao que é estabelecido no programa, os ajustes aqui podem ser feitos entre 0% a 100% do valor máximo de cada parâmetro, assim, é possível inclusive desabilitar o efeito do controlador para ocasiões de testes.

**Figura 22 – Ícone para ajuste do parâmetro de controle P.****Fonte: Do Autor (2023).**

Da mesma forma que os ajustes de PID, também há um ajuste de velocidade base indicado pelo ícone mostrado na Figura 25 onde aqui é inserido o percentual da velocidade

**Figura 23 – Ícone para ajuste do parâmetro de controle I.**



**Fonte: Do Autor (2023).**

**Figura 24 – Ícone para ajuste do parâmetro de controle D.**



**Fonte: Do Autor (2023).**

máxima que pode ser atingido. Neste parâmetro, o ajuste pode ser feito de 0% a 80%, sendo que no máximo 80% da velocidade máxima pode ser ajustada. Isso se dá para poder existir uma margem entre a velocidade de deslocamento máxima e o ajuste do controle de direção do veículo.

**Figura 25 – Ícone para ajuste do parâmetro de velocidade máxima.**



**Fonte: Do Autor (2023).**

No item associado ao mostrado na Figura 26 é mostrada uma tela com o nome do projeto (Veículo Autônomo de Superfície), o seu objetivo (Trabalho de Conclusão de Curso) e o seu respectivo autor (Jhonatas Willian Gonçalves de Moraes), de forma abreviada. O objetivo desta tela é apenas ser um informativo de autoria do projeto.

**Figura 26 – Ícone da tela de informações.**



**Fonte: Do Autor (2023).**

Sobre o item associado a Figura 27 trata-se de um menu de testes que mostra na tela a direção apontada atual, uma direção de alvo fixa, os valores de PID ajustados e a velocidade ajustada.

**Figura 27 – Ícone da tela para testes.**



**Fonte: Do Autor (2023).**

Por fim, o menu possui dois itens com a Figura 28, sendo cada um deles correspondente com um dos motores laterais. Por meio do menu deste item é possível ter um ajuste fixo da velocidade atual dos motores para efeito de testes.

**Figura 28 – Ícone para ação dos motores individualmente.**



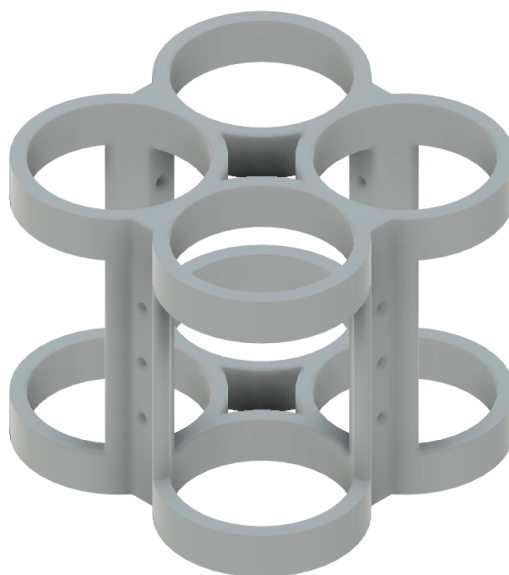
**Fonte: Do Autor (2023).**

### **3.5 Modelagem e construção mecânica**

#### **3.5.1 Estrutura dos Motores e Espuma**

Com os motores escolhidos, o projeto estrutural se iniciou pela elaboração da parte que suporta as espumas que fazem a sustentação sobre a água, e também deixando com suporte para a fixação dos motores e do chassi em duas etapas. Para suportar as espumas com raio de  $60mm$ , a estrutura foi pensada de forma que pudesse ser leve, e adaptável para a instalação de periféricos. Os locais onde seriam colocados os insertos de  $5mm$  foram feitos com furos de  $4,7mm$ , e os furos dos quais haveriam parafusos instalados, ficaram com furos de  $3,5mm$  (para parafusos de  $3mm$ ). A estrutura citada é vista na Figura 29. Foram feitas duas peças idênticas, sendo uma para cada lado.

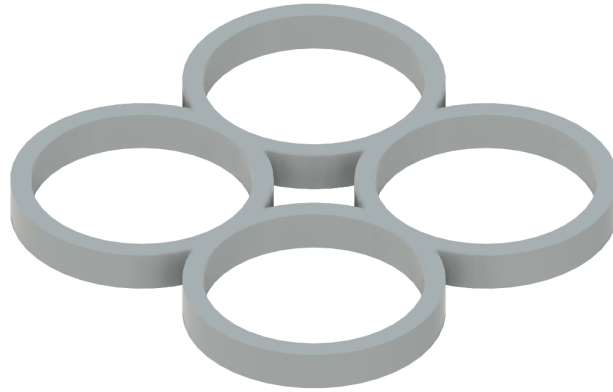
**Figura 29 – Estrutura de suporte dos motores e espumas.**



**Fonte: Do Autor (2023).**

Além da estrutura de suporte aos motores, também foram construídos os anéis de sustentação e estabilidade das espumas, sendo que esta é apenas um espelho da face lateral da estrutura de espumas. Esta tem as mesmas medidas da estrutura dos motores, porém foram feitas 4 peças idênticas para serem distribuídas próximo das pontas das espumas. O anel de sustentação pode ser visto na Figura 30.

**Figura 30 – Anel de sustentação.**

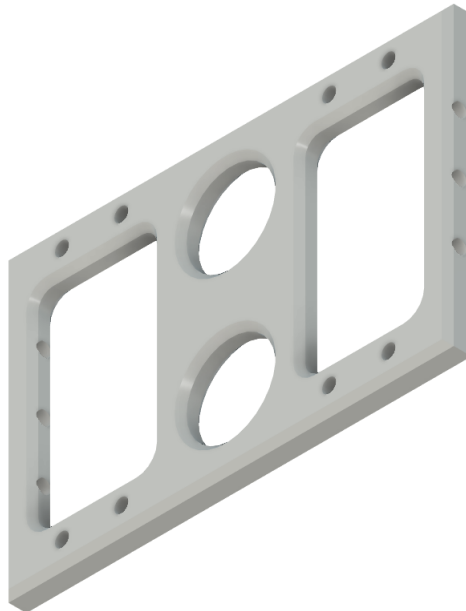


**Fonte: Do Autor (2023).**

### 3.5.2 Chassi de união central

Para a sustentação das partes laterais, suportar o alojamento das bombas de água e alguns outros itens, foi elaborado um chassi de união central do qual faz a junção das duas estruturas laterais, mantendo uma distancia entre os motores (distancia central entre as estruturas) de  $250mm$ . Esta parte do chassi possui apenas furos para os insertos de  $5mm$ , sendo que aqui é considerado que as demais partes são fixadas a ela. O chassi desenhado pode ser visto na Figura 31.

**Figura 31 – Chassi de união central.**

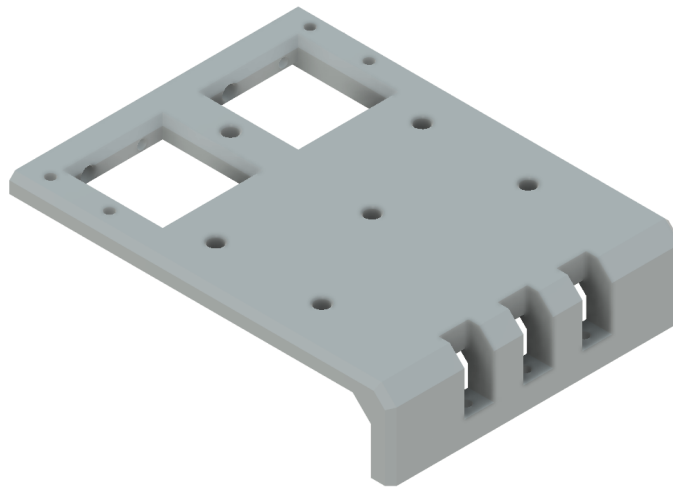


**Fonte: Do Autor (2023).**

### 3.5.3 Chassi superior

A construção da parte superior do chassi precisou ser dividida em duas partes, devido a limitações de tamanho em impressão 3D, sendo assim, essa parte do chassi consiste em duas partes exatamente iguais que se encaixam em seu centro, e entregam pontos de apoio com a estrutura de suporte dos motores. Uma das peças do chassi superior pode ser visto na Figura 32.

**Figura 32 – Chassi superior.**



**Fonte: Do Autor (2023).**

### 3.5.4 Caixa para as placas

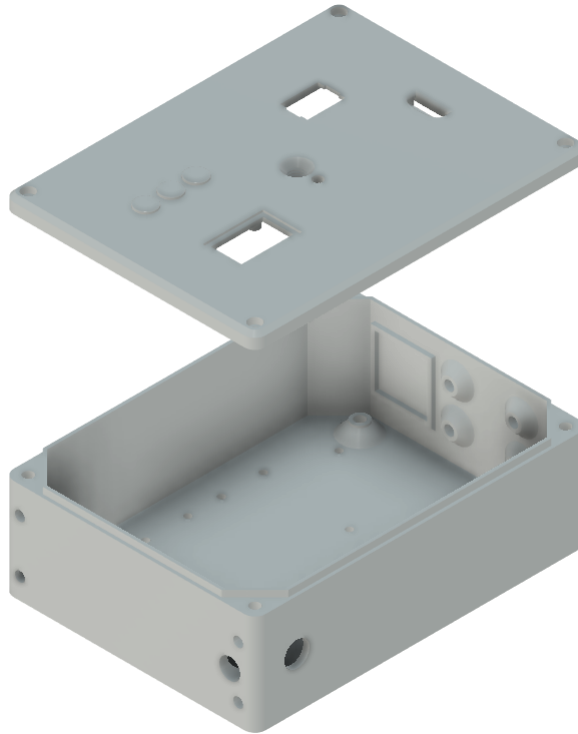
Tendo em mente o local para a instalação e alojamento das placas principais, ESCs e alguns outros componentes foi elaborada uma caixa para o alojamento destes componentes. A caixa possui furos para fixação de insertos para a tampa, das placas internas, sendo elas GPS e placa principal, e também externamente nas laterais para o acoplamento do suporte das baterias. A caixa para as placas foi projetado para que pudesse ser parafusada a placa já pré-projetada, mantendo as distancias corretas para os furos previstos na placa principal. A caixa pode ser vista na Figura 33, onde pode se ter uma vista explodida, apresentando também a tampa. Nota-se que foi considerado um encaixe para que dificulte a entrada de água por meio da borda da tampa, assim, aumentando a confiabilidade de se colocar o veículo na água.

### 3.5.5 Caixa para as baterias

A Caixa para o alojamento das baterias foi planejada para ser instalada nas duas laterais da caixa das placas, sendo assim, ainda é mantida a simetria do projeto e evitando que um lado possa ficar mais pesado que o outro. As caixas das baterias foram projetados de uma



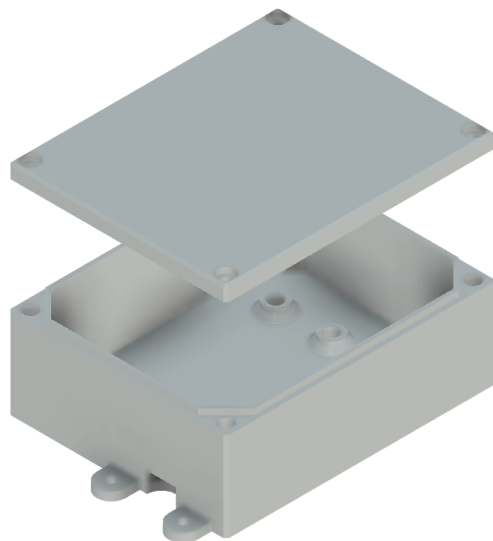
**Figura 33 – Caixa das placas.**



**Fonte: Do Autor (2023).**

forma que pudesse ser usado o mesmo desenho espelhado do outro lado, assim, foi necessário realizar apenas um dos desenhos mecânicos. Este alojamento também foi projetado para que houvesse um encaixe na tampa que dificultasse a entrada indesejada de água para dentro do espaço em que as baterias ficam instaladas. O suporte utilizado neste caso é para três baterias de íons de lítio de tamanho 18650. Na Figura 34 é possível ver o modelo desenhado.

**Figura 34 – Caixa das baterias.**

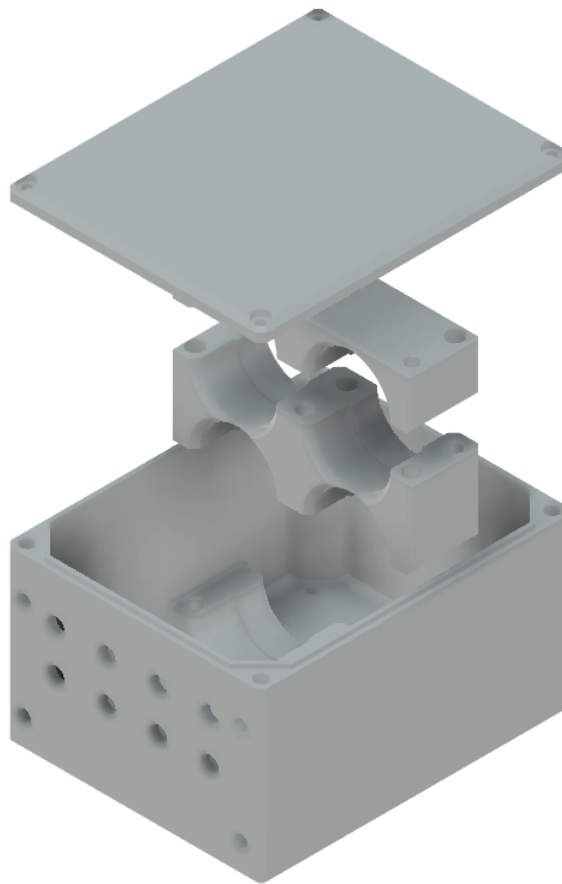


**Fonte: Do Autor (2023).**

### 3.5.6 Caixa para as bombas

O local onde seriam instalados as bombas para sucção de água fica em baixo do chassi superior, e em cima do chassi central, assim, abaixando o centro de massa do veículo, que por sua vez implica na estabilidade do veículo. Como foram usadas bombas pequenas, é necessário que todas as quatro bombas caibam dentro de um invólucro que fique apenas para o transporte de água, assim, a água deve entrar por mangueiras, indo até o local da bomba, e saindo por outra mangueira, evitando quaisquer vazamentos de água que possam existir no meio do percurso. Sendo assim, as quatro bombas são distribuídas entre duas em baixo, alojadas no suporte na base da caixa, sendo travadas por meio de uma trava central entre as bombas, na qual esta mesma trava já opera como um alojamento para mais duas bombas que ficam na parte superior, onde finalmente são travadas por mais uma trava superior, onde todas estas são parafusadas de maneira simétrica. Após a sua respectiva instalação é colocada a tampa superior, onde esta possui o encaixe para evitar a entrada indesejada de água no invólucro das bombas de água. A caixa para as bombas é mostrado em vista explodida na Figura 35.

**Figura 35 – Caixa das bombas de água.**

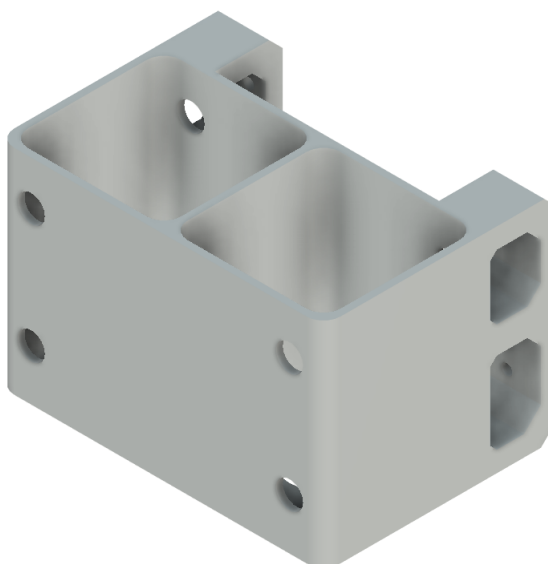


**Fonte: Do Autor (2023).**

### 3.5.7 Suporte e tampas para garrafas

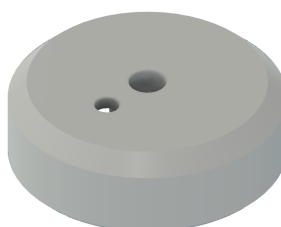
Para disponibilizar o local onde são armazenadas as amostras de água, foi elaborado um pequeno "cesto" onde podem ser colocadas duas garrafas de água com aproximadamente 200mL. O suporte foi projetado para poder ser encaixado na parte frontal e traseira do veículo, sendo parafusado na caixa para as bombas, de forma que haja um fácil acesso das mangueiras até as garrafas. O suporte para as garrafas foi projetado de forma que seja compatível com qualquer um dos dois lados do veículo (frontal ou traseiro) onde este pode ser visto na Figura 36. Para que seja possível fixar as mangueiras na tampa das garrafas, também foi projetado um modelo de tampa com respiro, onde a mangueira fique firme sem a necessidade de se usar cola, evitando assim a contaminação das amostras. O modelo das tampas projetado pode ser visto na Figura 37.

**Figura 36 – Suporte de garrafas.**



Fonte: Do Autor (2023).

**Figura 37 – Tampa para garrafa com respiro.**

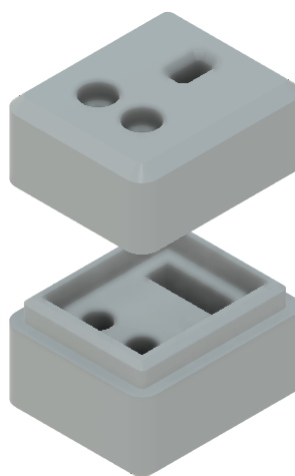


Fonte: Do Autor (2023).

### 3.5.8 Alojamento para magnetômetro e sensor de temperatura, umidade e pressão

Para que seja possível manter o sensor magnetômetro com uma distância segura dos demais circuitos, de maneira que não hajam influências eletromagnéticas nas medidas, foi projetada uma torre para que pudesse ser alojado este elemento com segurança na parte superior do veículo. Este alojamento é capaz de guardar tanto o magnetômetro como o sensor de temperatura, umidade e pressão também com uma certa segurança em relação a entrada de água. Para a sua sustentação, é usada uma barra de rosca de 8 milímetros com aproximadamente 10 centímetros de comprimento, colocada sobre a tampa da caixa das placas. A barra de rosca é encaixada tanto do lado do alojamento aqui citado como na tampa da caixa das placas, mantendo assim uma certa possibilidade de ajuste de direção de leitura do magnetômetro, bem como uma rigidez e estabilidade suficientes para a aplicação. É mostrado na Figura 38 o modelo do alojamento discutido. Também é importante mencionar que este elemento está posicionado ao centro do veículo, assim, todo deslocamento angular é refletido diretamente ao sensor.

**Figura 38 – Alojamento para magnetômetro (QMC5883) e sensor de umidade, temperatura e pressão (BME280).**



**Fonte: Do Autor (2023).**

### 3.5.9 Suporte para mangueiras e sensor de temperatura da água

Para coletar as amostras de água, e também a temperatura da superfície da água no ponto de coleta, foi elaborado um suporte para barra de rosca em X, e também um alojamento para a instalação das mangueiras das quatro bombas de água e do sensor de temperatura da água. Neste caso, os dois são acoplados por meio de uma barra de rosca de 8 milímetros de espessura e de aproximadamente 10 centímetros de comprimento. O suporte em forma de X é instalado na parte inferior do chassi central, colocando também o ponto de coleta na parte

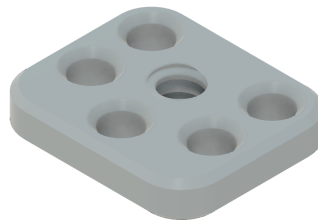
central do veículo. Os itens suporte em X e alojamento para mangueiras e sensor citados podem ser observados nos modelos apresentados nas figuras Figura 39 e Figura 40 respectivamente.

**Figura 39 – Suporte para barra de rosca em forma de X.**



Fonte: Do Autor (2023).

**Figura 40 – Alojamento mangueiras e sensor de temperatura.**



Fonte: Do Autor (2023).

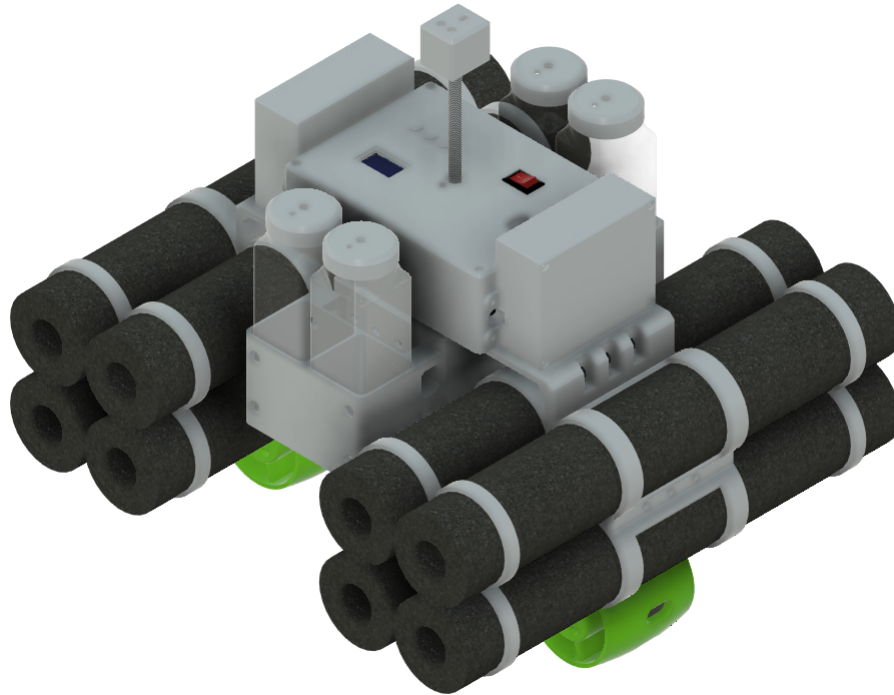
#### 3.5.10 Montagem dos modelos projetados

Com o auxílio do software *Fusion 360* foi feita a montagem detalhada de todos os elementos já discutidos, bem como também foi inserido os modelos de CAD dos motores e de outros elementos para ilustrar as suas posições no projeto. A montagem é mostrada na Figura 41 onde é mostrada a vista isométrica do veículo autônomo de superfície.

A concepção de todas as peças desenhadas no software foram feitas por meio de impressão 3D, onde as peças puderam ser dinamicamente fabricadas sem a necessidade de terceirização do processo, bem como a diminuição de custos com a prototipação das partes.

Ao todo foi gasto na impressão 3D de todas as peças a quantidade aproximada 2kg de filamento, sendo que todas as peças foram fabricadas em ABS, já que é um material interessante para a aplicação em protótipos e peças com destinação mecânica.

**Figura 41 – Montagem de todos os elementos por meio de software.**



**Fonte: Do Autor (2023).**

### **3.6 Algoritmos implementados no projeto**

O algoritmo do projeto é dividido em algumas partes essenciais para o controle de direção e locomoção do veículo autônomo de superfície por meio do acionamento dos motores com base nas orientações de GPS e de direção com o magnetômetro.

#### **3.6.1 Rotina de calibração do magnetômetro**

Com o intuito de remover as taxas de erro discutidas no Capítulo 2 deste trabalho, há uma parte no código que faz a calibração dos fatores de erro do magnetômetro com base nos valores medidos. Esta calibração é essencial, já que o campo magnético terrestre é diferente dependendo da localidade em que o veículo é posicionado. Isso se dá pela presença de materiais metálicos, magnéticos ou então por interferências eletromagnéticas próximas do sensor magnetômetro.

A parte do código que faz essa calibração pode ser acessada pela navegação do menu principal. Neste algoritmo de calibração, inicialmente são ajustados os parâmetros da biblioteca em seus valores máximos para que não influenciem na amostragem. A seguir, no decorrer de um loop de repetição são coletados os valores máximos e mínimos lidos pelo magnetômetro nos eixos x e y que são paralelos a superfície da terra, assim, a cada repetição é verificado se o valor medido é maior que o valor máximo já registrado, ou então se é menor que o valor

mínimo já registrado, sendo armazenado o ultimo valor toda vez que este extrapolar o atingido em uma repetição anterior. Com isso, vão sendo feitas várias medidas, sempre armazenando os maiores e menores valores para cada direção e para cada eixo, onde quando não há um novo registro por mais de 5 segundos, a calibração é finalizada assumindo que estes valores são os novos parâmetros de calibração para o sensor. Durante o processo de calibração e medição dos valores é essencial que o veículo esteja rotacionando para que sejam medidos o maior número possível de pontos. Por fim, com os parâmetros medidos, a biblioteca utilizada para o sensor magnetômetro pode internamente já fazer a correção das influencias por ferro macio e ferro duro das quais já foram discutidas neste trabalho. De modo que não seja necessário realizar a calibração todas as vezes que o dispositivo é reiniciado, os parâmetros de calibração são armazenados na memória EEPROM do microcontrolador ESP32.

Para exemplificar os dados medidos durante o processo de calibração do sensor magnetômetro, foram tomadas algumas amostras por meio do cartão micro SD, onde com o auxílio do MATLAB, foram feitas algumas correções nos valores obtidos da mesma forma que a biblioteca faz a calibração dos valores. Na Figura 42 é mostrado em azul os dados amostrados, onde percebe-se que os dados em azul possuem um comportamento de centro deslocado da origem e formato ligeiramente elíptico.

Nos pontos em laranja, é feita apenas uma correção de amplitude dos valores máximos e mínimos, assim formando uma figura perfeitamente circular, porém ainda deslocada. Esta é a calibração da interferência de ferro macio. Já nos pontos em amarelo, é feita apenas a correção de deslocamento, ainda mantendo o formato de elipse nos pontos amostrados, sendo que este é o efeito da interferência de ferro duro. Por fim, em roxo são aplicadas as duas correções, onde a forma da curva demonstra um círculo perfeito (quando considerando os pontos máximos e mínimos de x e y) e ao mesmo tempo este círculo é centralizado na origem. Esta é a forma ideal para se calcular o angulo magnético em relação ao norte por meio do magnetômetro.

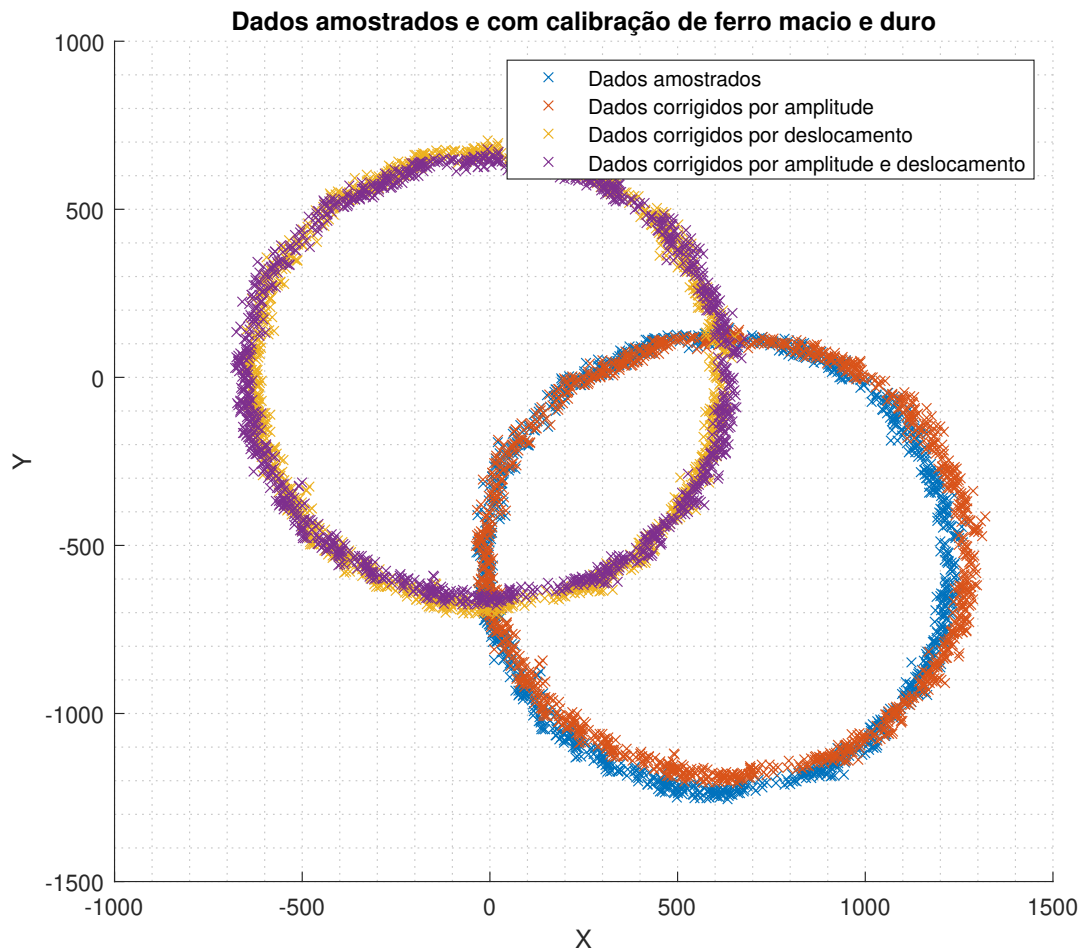
### 3.6.2 Cálculo de angulo com relação ao norte geográfico

Para realizar o cálculo da direção do norte geográfico, é feita uma amostragem dos valores de intensidades nos eixos x e y do magnetômetro, e então calculado o arco cuja tangente seja o angulo representado pelas duas parcelas medidas, ficando como mostrado na Equação 1, onde  $\theta$  é o angulo apontado em graus.

$$\theta = \arctan\left(\frac{y}{x}\right) \quad (1)$$

Com o valor do angulo calculado, basta subtrair o valor da declinação magnética, e também limitar o valor entre 0 e 360 graus para que seja representado um círculo mesmo após inserir a correção da declinação magnética. Assim é obtido o valor do ângulo apontado com

**Figura 42 – Curvas amostradas e corrigidas obtidas com magnetômetro.**



**Fonte: Do Autor (2023).**

relação ao polo geográfico da terra. Este valor é importante para ser usado como parâmetro de controle de direção.

### 3.6.3 Leitura de coordenadas do cartão micro SD

A leitura de coordenadas do cartão é feita de uma maneira bem simples, fazendo a separação de cada string por um caractere especial de ";", assim, o algoritmo entende que a cada aparição de ";" é a conclusão de uma instrução, além disso, cada uma dessas strings lidas é separada em seções por ",", onde cada um dos parâmetros de entrada é dividido dessa forma. Com isso, em cada linha de instrução que é passada pelo cartão micro SD contém parâmetros de coordenadas alvo compostas por latitude e longitude, comando para realizar a coleta de dados climáticos do ponto e comando para sinalizar que no ponto da linha atual deve ser feita ou não uma coleta de amostra de água.



O código foi preparado para suportar até 20 entradas de coordenadas diferentes com pontos de coleta de amostras e coleta de dados.

#### 3.6.4 Cálculo da direção entre a coordenada atual e coordenada alvo e a respectiva distância

Com a leitura das coordenadas atuais por meio da biblioteca *TinyGPSPlus* já discutida na subseção 3.1.3.2, ainda deve ser feito o cálculo da direção que deve ser apontado o veículo. Isso pode ser feito por meio da comparação entre as duas coordenadas conforme mostrado na Equação 2, onde  $\lambda$  é a coordenada de longitude,  $\phi$  é a coordenada de latitude, e  $\gamma$  é o ângulo que deve ser apontado em graus.

$$\gamma = \arctan \left( \frac{\sin(\Delta\lambda) * \cos(\phi_2)}{\cos(\phi_1) * \sin(\phi_2) - \sin(\phi_1) * \cos(\phi_2) * \cos(\Delta\lambda)} \right) \quad (2)$$

Com este valor, é possível se ter a referência de direção que pode ser usado como recurso para o controlador de direção. De forma orientativa para os testes, também é feito o cálculo da distância entre as duas coordenadas baseando-se no raio da terra. A fórmula usada para calcular esta distância é a fórmula de Haversine, que é mostrada na Equação 3, Equação 4, e Equação 5.

$$a = \sin^2 \left( \frac{\Delta\phi}{2} \right) + \cos(\phi_1) * \cos(\phi_2) * \sin^2 \left( \frac{\Delta\lambda}{2} \right) \quad (3)$$

$$c = 2 * \arctan \left( \frac{\sqrt{a}}{\sqrt{1-a}} \right) \quad (4)$$

$$d = R * c \quad (5)$$

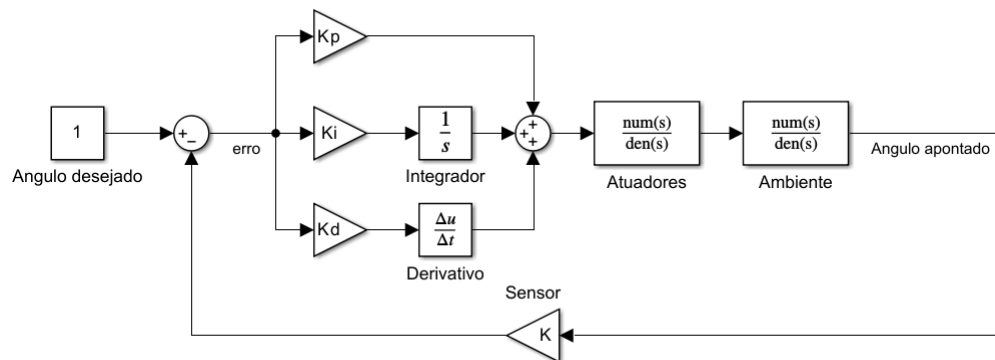
Aqui também  $\lambda$  é a coordenada de longitude,  $\phi$  é a coordenada de latitude,  $R$  é o raio da terra, que equivale a  $6.371km$ , e  $d$  é a distância calculada. Se  $R$  for inserido na fórmula em metros, o resultado de  $d$  será em metros também, conforme obtido na Equação 5.

#### 3.6.5 Controle de direção

A única planta de controle com resposta ajustada conforme erro é o controle de direção aplicado no projeto do qual baseia-se no cálculo da diferença entre o ângulo que deve ser apontado e o ângulo atual. O ângulo que deve ser apontado é calculado baseando-se em coordenadas esféricas geográficas, e o ângulo atual é corrigido com a declinação magnética no ponto em que é feito os testes. Sabe-se que na cidade em que os testes estão sendo realizados, cidade de Blumenau-SC, o valor da declinação magnética aproximada para o período do primeiro semestre de 2023 é de  $-19,85$  graus decimais. Sabendo como calcular o erro, e

considerando uma planta genérica com controle PID aplicado, pode-se aproximar o sistema pela planta mostrada na Figura 43, em que o sinal de referência sempre será o ângulo que deve ser apontado, a saída sendo o ângulo que o veículo aponta em relação ao norte geográfico e o erro sendo a diferença entre os dois ângulos, que quando inserido na planta de controle, faz a mudança de direção pela atuação dos motores propulsores.

**Figura 43 – Modelo de sistema semelhante ao encontrado no projeto.**



**Fonte: Do Autor (2023).**

O algoritmo PID utilizado no projeto trata-se de um código genérico que faz a aferição da componente integral, com base em um tempo de amostragem determinado, a aferição da componente derivativa com o mesmo tempo de amostragem determinado, e por fim a componente proporcional que baseia-se apenas no valor do erro multiplicado por uma constante de proporção.

Como resultado do controlador PID há uma saída que corresponde a apenas a correção de direção do veículo, que quando somada a uma velocidade de base fixada, faz o deslocamento do veículo entre as coordenadas desejadas juntamente com a correção de curso baseada na direção apontada e orientação entre a coordenada atual e a coordenada alvo.

Assim como o ajuste de velocidade, o controle de direção também deve ser convertido em um sinal de ângulo para servo motores, que ao ser transmitido para o controlador eletrônico de velocidade, faz o acionamento dos motores correspondentes. Como o tipo dos controladores utilizados é para aplicação náutica, estes possuem o recurso de reversão de direção, o que faz ser possível o veículo mudar de direção mesmo parado, sem necessariamente haver um deslocamento entre o plano da superfície do corpo d'água.

Resumindo, a saída de controle da planta corresponde a apenas uma variável que é aplicada somando a velocidade base em um dos motores e subtraindo no outro. Como o ângulo aplicada a saída servo para que os motores estejam parados é 90 (pode ser de 0 a 180), então o valor é aplicado de maneira simétrica na atuação dos dois controladores eletrônicos de velocidade. Devido aos motores serem com hélices espelhadas, a simples aplicação do controle de direção dessa forma somada a velocidade ajustada é suficiente para poder controlar o veículo de forma autônoma e confiável.

Devido a planta estudada ter uma certa inércia intrínseca, bem como ser um sistema relativamente lento, o tempo de amostragem foi escolhido em 50 milissegundos, sendo que este tempo precisa ser considerado principalmente ao levar em conta de que o microcontrolador gasta um certo tempo para fazer outras funções que estão nas rotinas de monitoramento de dados para análise, bem como manter o display com informações atualizadas. Sendo assim, o tempo de ciclo não pode ser muito pequeno, já que caso o microcontrolador demore tempo demasiado em outras atividades, a atuação do algoritmo controlador começa a ser prejudicada. Já que a atuação do controlador sempre depende de informações atualizadas de localização e também de leitura do magnetômetro, não foram usadas interrupções para o cálculo do controle PID, assim dando o tempo de coleta dos dados dos sensores antes que a operação de cálculo fosse executada. Esse tipo de problema acontece devido ao controlador ser uma aproximação digital de um controlador ideal, onde os tempos de processamento precisam ser levados em conta, inclusive os tempos de coleta de dados dos sensores.

Os parâmetros PID foram ajustados da forma mostrada a seguir:

- $K_p = 1$ ;
- $K_i = 0.005$ ;
- $K_d = 1$ ;

Onde ainda é ajustado pelo menu de interface do projeto o percentual usado em cima de cada um desses valores, permitindo um certo ajuste sem a necessidade de reprogramar o sistema todo.

### 3.6.6 Algoritmo para realização de atividades nos pontos escolhidos

Uma das sub rotinas mais importantes para o escopo do trabalho é a que executa as atividades em cada uma das coordenadas escolhidas. Esta sub rotina é realizada por meio de um algoritmo curto, que interpreta os comandos de cada ponto de alvo programado pelo cartão micro SD. No algoritmo em questão, o microcontrolador verifica se há o comando de se realizar uma leitura de dados na coordenada alvo, coletando as amostras de temperatura da água por meio do sensor DS18S20, e dados de temperatura do ar, umidade relativa, e pressão atmosférica por meio do BME280 preparando os dados coletados em uma string para ser escrita no cartão micro SD. Caso também haja o comando de coletar amostras de água, o algoritmo permite a coleta de até 4 amostras no mesmo percurso, armazenando as amostras em cada um dos 4 recipientes de amostragem que são alocados na estrutura do veículo. A coleta de água é feita por meio das bombas de água que são acionadas sequencialmente de 1 até 4 dependendo do ponto de amostragem. Uma vez que todos os 4 reservatórios já sejam enchidos, o comando de coletar amostras se torna inativo para caso haja comandos futuros para a amostragem. Ao final de cada amostragem é feito um ponto de controle sendo armazenado no cartão micro SD

as informações sobre esse ponto com a data, hora e coordenadas de amostragem, dados sobre a amostragem de água e também sobre a amostragem dos demais sensores. Este algoritmo é executado em cada uma das coordenadas alvo que são colocadas no cartão micro SD.

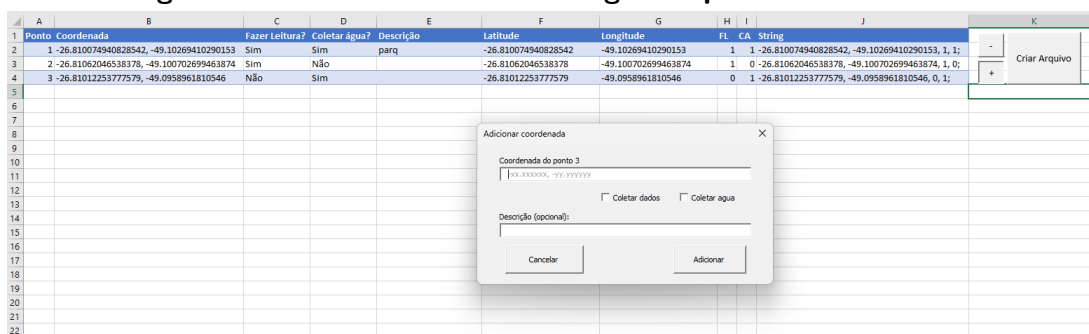
### 3.6.7 Algoritmo de navegação

O algoritmo que faz a navegação do veículo é o responsável pela maior parte da automatização de todo o processo de controle autônomo. Este em questão é quem comanda cada uma das instruções e etapas do processo de navegação do veículo dando as informações necessárias para a atuação dos controles e comportamentos vistos na navegação. Aqui o algoritmo se inicia ajustando a velocidade do veículo com base na velocidade ajustada em um dos parâmetros do menu principal. Isto é importante já que o veículo pode operar dentro deste algoritmo sem a necessidade de ter o deslocamento linear pelo corpo d'água, dando a oportunidade de se realizar testes com o veículo parado, porém nesta etapa, os motores ainda não são acionados, já que este passo só é feito quando chamada a função de controle de PID. Depois que a velocidade de deslocamento é ajustada, as coordenadas atuais são armazenadas depois do último ponto lido do cartão micro SD, e então o algoritmo circula por cada um dos pontos dentre todos os pontos lidos, onde para cada um desses pontos, há um loop que mantém o microcontrolador atualizando os dados de localização por GPS, atualizando os dados do magnetômetro, calculando a distância entre a coordenada atual e a coordenada alvo, ajustando o controle PID e ainda gerando um registro de dados no cartão micro SD. Este loop se mantém até que a distância do alvo seja menor do que a precisão ajustada, assim quando o veículo está perto do ponto de controle, os motores são parados, e são realizadas todas as atividades necessárias no ponto escolhido. Ao terminar as atividades de um ponto, o loop segue para o próximo ponto, refazendo todos os passos até que todos os pontos programados sejam finalizados, e ainda retornando ao ponto de origem, já que este foi colocado como último ponto da lista de alvos. Vale ressaltar que este algoritmo só inicia quando o GPS possui coordenadas válidas, sendo que isso é informado pela própria biblioteca *TinyGPSPlus*.

## 3.7 Programa criado em VBA e excel para a entrada de coordenadas

Para tornar mais fácil a entrada de coordenadas no arquivo txt que é colocado no cartão micro SD, foi criada uma pequena planilha de Excel juntamente com VBA (*Visual Basic for Applications*) para que a importação das coordenadas seja rápida e prática. A planilha mostrada na Figura 44 permite a inserção de dados de coordenadas clicando no botão "+", e assim colocando as coordenadas diretamente obtidas no *Google Maps*, como por exemplo "-23.551111088229337, -51.42749314246813" onde as coordenadas de latitude e longitude são separadas por vírgula respectivamente e o separador decimal usado é o ponto.

**Figura 44 – Planilha com interface gráfica por VBA construída.**



**Fonte: Do Autor (2023).**

Como o Excel entende a vírgula como separador decimal, precisaria ser feito um pequeno ajuste de valores para que as coordenadas fiquem no formato correto. Além disso, também é necessário agrupar as instruções de coleta de dados e de coleta de amostras de água respectivamente na linha de comando entendida pelo projeto. Ao final de cada linha de comando também é importante inserir um caractere ";" indicando que chegou ao fim da instrução, assim, se considerar que neste ponto mencionado como exemplo seja necessária a coleta de amostra de água, mas não coleta de dados climáticos, deve-se ficar na forma "-23.551111088229337, -51.42749314246813, 0, 1;" com um caractere de quebra de linha ao final. Esta operação é relativamente trabalhosa para se fazer manualmente todas as vezes, mas com o auxílio da planilha elaborada a tarefa se torna muito mais rápida. Com a mesma planilha é possível gerar um arquivo chamado "coordenadas.txt" onde este arquivo já é pronto para ser inserido no cartão micro SD.

## 4 RESULTADOS

Como resultado do trabalho, inicialmente há a necessidade de se adquirir todos os componentes usados no projeto, que estão listados na Tabela 1, sendo que a soma de todos os itens adquiridos ficou em R\$1020,01. A cotação de todos os itens foi feita no primeiro semestre de 2023, sendo que os valores são aplicados para cada unidade de cada item.

**Tabela 1 – Lista de materiais**

| Item | Qtd. | Qtd. Ref. | Descrição                               | Valor     |
|------|------|-----------|-----------------------------------------|-----------|
| 1    | 1    | un.       | ESP32 WROOM32 DOIT KIT V1               | R\$21,58  |
| 2    | 1    | un.       | Modulo GPS GY-NEO6MV2                   | R\$32,99  |
| 3    | 1    | un.       | Conversor buck ajustável                | R\$1,84   |
| 4    | 1    | un.       | Módulo BME280 3.3v                      | R\$31,86  |
| 5    | 1    | un.       | Módulo QMC5883L                         | R\$7,01   |
| 6    | 1    | un.       | Sensor DS18S20 inoxidável               | R\$3,04   |
| 7    | 1    | un.       | Módulo leitor de cartão micro SD        | R\$2,44   |
| 8    | 1    | un.       | Display OLED I2C 0.96" 128x64           | R\$13,63  |
| 9    | 2    | un.       | Suporte para 3 baterias 18650           | R\$8,82   |
| 10   | 6    | un.       | Bateria 18650 3,7V 2600mAh LGDB318650   | R\$153,92 |
| 11   | 2    | un.       | ESC ZTW Shark 40A SBEC G2               | R\$149,32 |
| 12   | 2    | un.       | Motor SkyArea BLDC 1450KV 17A 12-24V    | R\$170,56 |
| 13   | 4    | un.       | Bomba de água ZR370-05PM 12V            | R\$58,92  |
| 14   | 4    | un.       | Mosfet IRFZ46N                          | R\$16,00  |
| 15   | 4    | un.       | Optoacoplador 4N25                      | R\$5,32   |
| 16   | 4    | un.       | Soquete estampado 6 pinos               | R\$1,24   |
| 17   | 1    | un.       | Barra de pinos fêmea 2,54mm             | R\$1,72   |
| 18   | 1    | un.       | Barra de pinos macho 180 graus 2,54mm   | R\$0,78   |
| 19   | 14   | un.       | Resistores 1/4w                         | R\$0,84   |
| 20   | 4    | un.       | Diodo 1N4007                            | R\$0,64   |
| 21   | 2    | un.       | Capacitor cerâmico 100nF 100v           | R\$0,26   |
| 22   | 1    | un.       | Capacitor eletrolítico 10uF 16v         | R\$0,20   |
| 23   | 1    | un.       | Capacitor eletrolítico 100uF 16v        | R\$0,40   |
| 24   | 1    | un.       | Chave gangorra KCD1-101                 | R\$0,99   |
| 25   | 3    | un.       | Chave tátil KFC-A06 6x6x5mm             | R\$0,54   |
| 26   | 4    | un.       | Alojamento xh2.54 2 vias                | R\$0,32   |
| 27   | 2    | un.       | Terminal faston pré isolado 6.3mm       | R\$0,76   |
| 28   | 2    | un.       | Terminal faston macho para placa 6.3mm  | R\$0,54   |
| 29   | 2    | un.       | Placa fenolite virgem 10x20cm           | R\$25,88  |
| 30   | 27   | un.       | Terminal para alojamento xh2.54         | R\$1,29   |
| 31   | 5    | un.       | Conector para placa macho xh2.54 4 vias | R\$1,65   |
| 32   | 1    | un.       | Conector para placa macho xh2.54 3 vias | R\$0,31   |
| 33   | 4    | un.       | Conector para placa macho xh2.54 2 vias | R\$0,88   |
| 34   | 4    | un.       | Alojamento xh2.54 4 vias                | R\$0,52   |
| 35   | 1    | un.       | Alojamento xh2.54 3 vias                | R\$0,10   |
| 36   | 10   | mts.      | Fios e cabos                            | R\$21,00  |
| 37   | 3    | mts.      | Mangueira PVC 6x4mm                     | R\$30,00  |
| 38   | 4    | un.       | Garrafa PEAD 200ml                      | R\$10,00  |
| 39   | 2    | un.       | Flutuador espaguete polietileno 1,60m   | R\$18,00  |

(continua)

**Tabela 1 – Lista de materiais**  
(continuação)

| Item         | Qtd. | Qtd. Ref. | Descrição                         | Valor             |
|--------------|------|-----------|-----------------------------------|-------------------|
| 40           | 1    | un.       | Barra de rosca 8mm                | R\$9,00           |
| 41           | 54   | un.       | Parafuso M3x16                    | R\$12,96          |
| 42           | 8    | un.       | Parafuso M3x6                     | R\$1,36           |
| 43           | 6    | un.       | Parafuso M3x35                    | R\$2,10           |
| 44           | 24   | un.       | Parafuso M3x20                    | R\$5,88           |
| 45           | 92   | un.       | Inserto de latão M3x5x5mm         | R\$32,91          |
| 46           | 1    | un.       | Tubo de Cola Quente               | R\$1,00           |
| 47           | 1    | un.       | Bisnaga de silicone 50g           | R\$12,00          |
| 48           | 2    | kg        | Filamento ABS GTMAX3D 1,75mm      | R\$139,80         |
| 49           | 1    | pct.      | Abraçadeira de Nylon 2,5 x 150 mm | R\$6,90           |
| <b>Total</b> |      |           |                                   | <b>R\$1020,01</b> |

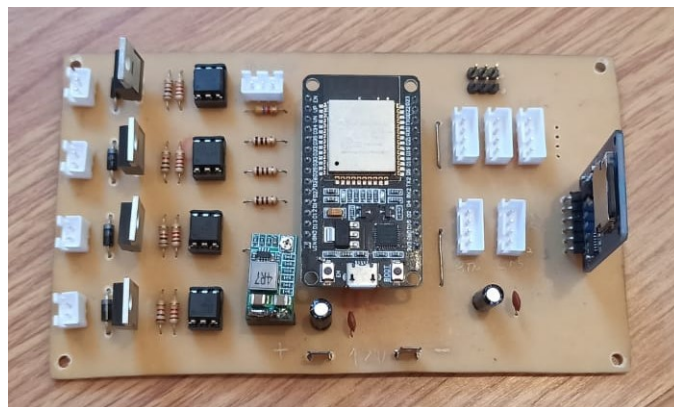
**Fonte: do Autor**

O valor totalizado representa a compra dos componentes para a construção de uma única peça do modelo projetado. Caso sejam produzidas unidades em lote, os valores tendem a ficar menores devido ao valor por atacado de cada componente.

#### 4.1 Montagem da placa

Com os desenhos da placa principal elaborados, foi confeccionada a placa de circuito impresso, perfurada e estanhada para ter maior durabilidade a oxidação. Após a limpeza das placas, todos os componentes foram soldados, apresentando um encaixe perfeito com o que foi projetado, sendo que foi minuciosamente respeitada toda e qualquer orientação dos componentes no projeto inicial. A vista superior da placa montada com os componentes pode ser vista na Figura 45. Nota-se que foi excepcional o projeto mecânico antes da montagem dos componentes devido ao posicionamento dos furos e principalmente o leitor de cartão.

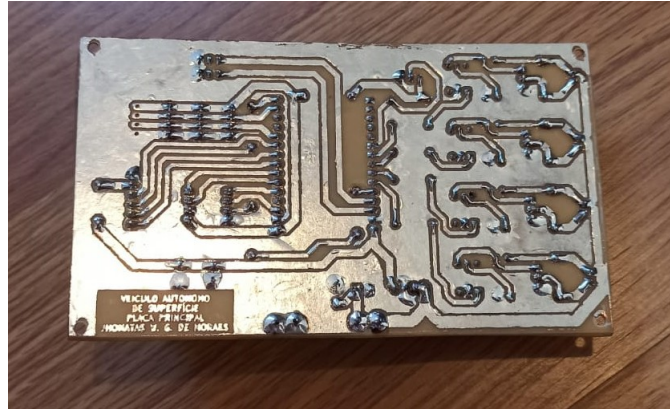
**Figura 45 – Vista superior da placa montada com os componentes.**



**Fonte: Do Autor (2023).**

Na Figura 46 pode ser observado de forma detalhada como que ficou o processo de solda na vista do verso da placa montada. Foi tomado o cuidado de verificar a existência de possíveis curtos indesejados antes da primeira energização, sendo que este é um fato que pode ocorrer devido ao processo ser artesanal e característico de prototipação.

**Figura 46 – Verso da placa montada com os componentes.**

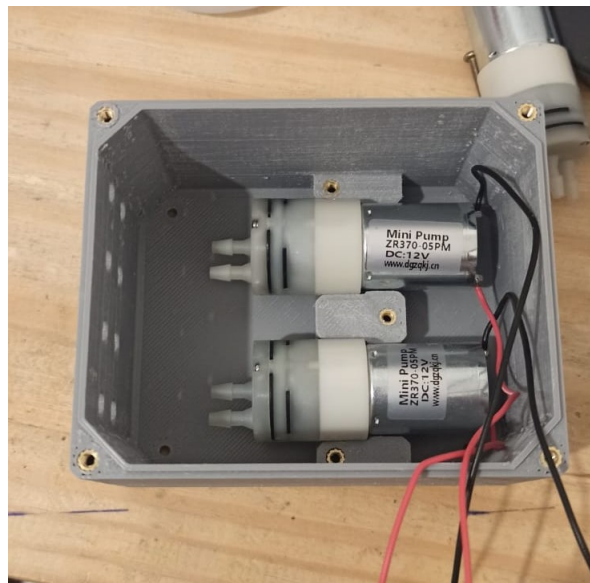


Fonte: Do Autor (2023).

## 4.2 Montagem das bombas

A montagem das bombas de água seguiu da forma que foi orientada no projeto. Inicialmente são posicionadas duas bombas na parte de baixo da caixa das bombas, onde as bombas ficam alojadas de forma encaixada na caixa, conforme mostrado na Figura 47.

**Figura 47 – Montagem das duas primeiras bombas na caixa.**



Fonte: Do Autor (2023).

Em seguida deve ser montada a trava central, fixando as duas bombas de baixo com parafusos, e em seguida são montadas as duas bombas superiores sobre a trava central, e



por fim é montada a trava superior, fixando as duas bombas superiores na trava central, que é fixada na base da caixa segurando as bombas inferiores como se fosse um sanduíche. A montagem das bombas superiores apenas com a trava superior e a trava central fora da caixa é mostrada na Figura 48.

**Figura 48 – Bombas montadas com a trava central e a trava superior.**



**Fonte: Do Autor (2023).**

Após a montagem das bombas em sua respectiva caixa, foram instaladas as mangueiras de PVC e também travadas com abraçadeiras de nylon de forma que não seja fácil de ocorrer alguma desconexão que possa causar vazamentos de água dentro da caixa. Além disso, como pode ser visto na Figura 49 os orifícios dos quais poderiam haver a entrada de água foram selados com silicone, focando principalmente na parte de entrada e saída das mangueiras de água, onde há a presença de mais furos, bem como a saída dos cabos, dos quais foram agrupados por meio de uma malha náutica, melhorando a sua estética.

Ao fim, a tampa da caixa também foi selada com silicone para evitar a entrada indesejada de água inclusive por cima da caixa.

**Figura 49 – Bombas montadas na caixa e seladas com silicone.**



**Fonte: Do Autor (2023).**

### **4.3 Montagem dos flutuadores e motores**

Para a montagem dos flutuadores de forma que fiquem firmes entre si, eles devem ser montados juntamente com os anéis de sustentação e com a estrutura central dos motores. Inicialmente os flutuadores são colocados apenas com um dos lados dos anéis para manter a sua posição fixa, como é mostrado de forma similar a organização dos respectivos flutuadores na Figura 50.

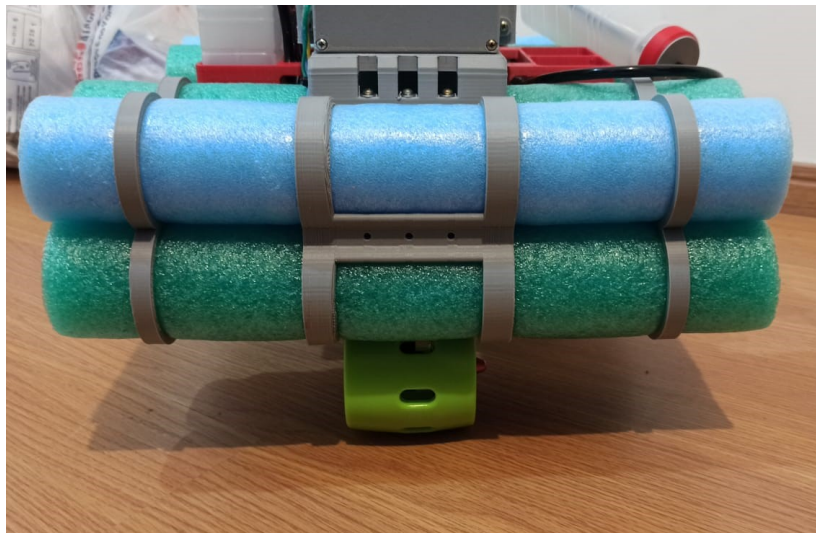
Assim, os flutuadores e os motores são montados com os anéis de sustentação e a estrutura de sustentação dos motores para cada um dos dois lados do veículo. Na Figura 51 é mostrado como que é o posicionamento destes itens por uma vista lateral. Nota-se que neste ponto a estrutura já possui bastante estabilidade, que é necessária para manter o veículo sem grandes dificuldades de locomoção.

**Figura 50 – Flutuadores montados nos anéis de sustentação.**



Fonte: Do Autor (2023).

**Figura 51 – Vista lateral destacando os flutuadores e um dos motores.**



Fonte: Do Autor (2023).

#### **4.4 Montagem dos suportes de garrafas**

Com o intuito de ter o lugar para armazenar as garrafas onde ficam armazenadas as amostras de água coletadas, são instalados os suportes de garrafas desenvolvidos juntamente na caixa das bombas, como pode ser mostrado na Figura 52.

Aqui as garrafas são posicionadas com a possibilidade de serem facilmente removidas pelo operador, assim, as mangueiras são instaladas nas tampas desenvolvidas, mantendo um respiro para que não haja problemas relacionados a pressão dentro das garrafas.

**Figura 52 – Suporte de garrafas montado.**



Fonte: Do Autor (2023).

**Figura 53 – Montagem do suporte das garrafas com as garrafas posicionadas.**



Fonte: Do Autor (2023).

#### **4.5 Montagem do suporte do magnetômetro e sensor de temperatura, umidade e pressão**

De forma a manter o sensor magnetômetro longe da interferência eletromagnética dos motores e ao mesmo tempo manter também o sensor BME280 longe de possíveis respingos

indesejados, foi montado o suporte destes dois sensores na parte superior do veículo, localizado no centro da tampa que coincide com o centro de giro do veículo também. Na Figura 54 pode ser visto o sensor de temperatura, umidade relativa e pressão localizado na parte superior e encapsulado de forma segura.

**Figura 54 – Vista interna do sensor BME280.**



**Fonte: Do Autor (2023).**

Como pode ser visto na Figura 55 é mostrada a montagem do suporte dos sensores na parte superior da tampa da caixa das placas. Nota-se que o posicionamento é estratégico e pode ser ajustada pela barra de rosca a direção em que é apontado os eixos x e y do magnetômetro.



**Figura 55 – Vista superior com o sensor posicionado e tampa montada.**

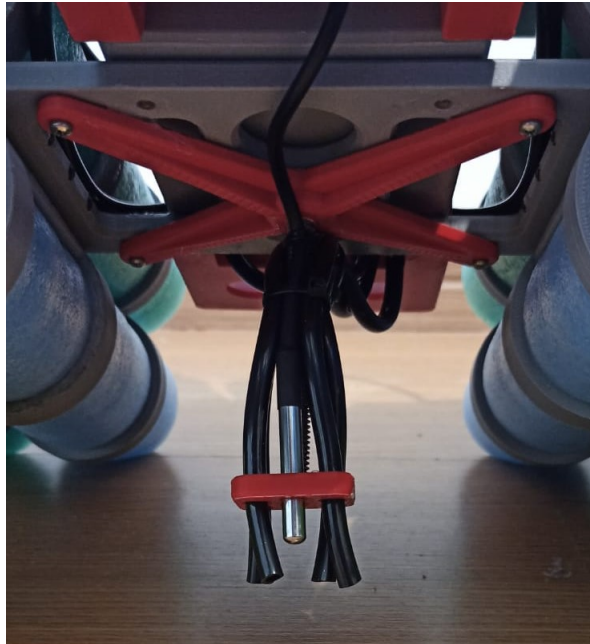


**Fonte: Do Autor (2023).**

#### **4.6 Montagem do suporte de mangueiras e sensor de temperatura**

Para manter as pontas das mangueiras e o sensor de temperatura da água sempre submersos, foi montado na parte inferior, juntamente com o suporte em X, a estrutura que segura estes componentes como visto na Figura 56. Aqui a altura da posição da ponta das mangueiras é aproximadamente no meio do eixo dos motores, garantindo a sua posição fixa na estrutura.

**Figura 56 – Vista do suporte de mangueiras e sensor montado.**

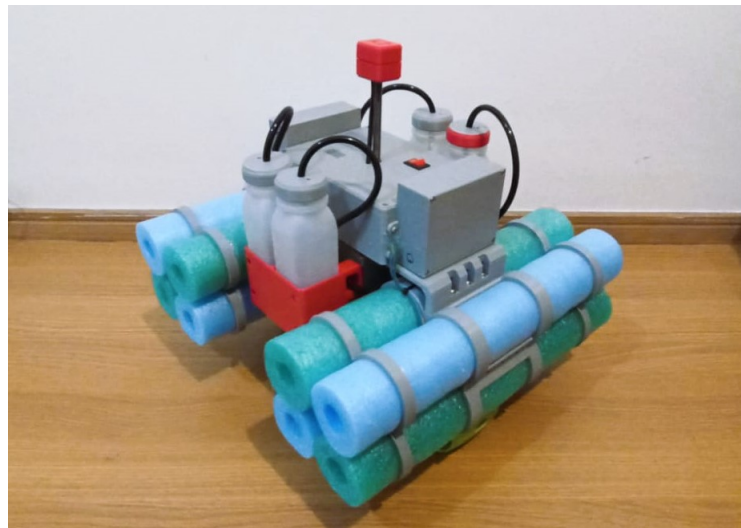


Fonte: Do Autor (2023).

#### **4.7 Montagem final**

Por fim, após a instalação de todos os componentes o projeto completamente montado é mostrado na Figura 57. Ao comparar esta figura com a Figura 41, nota-se que a montagem final do trabalho se assemelha muito ao que era esperado pelo desenho feito em software.

**Figura 57 – Vista isométrica do projeto montado.**



Fonte: Do Autor (2023).

Todas as medidas foram respeitadas de modo a não ter desvios consideráveis do projeto para o produto final. A partir deste ponto, ainda é necessário realizar os testes em campo e

verificar que todas as características das quais o projeto foi preparado estão condizentes com a sua operação.

#### 4.8 Testes em ambiente seguro (sem água)

Para testar a capacidade do veículo de atender os comportamentos esperados conforme a programação feita para o projeto, foram estipuladas algumas coordenadas em terra firme para que seja possível ver a atuação dos motores enquanto o veículo é transportado com as mãos pelos pontos desejados. Assim, a ação controle de orientação pode ser monitorada pela inspeção de algumas variáveis por meio de um registro de dados no cartão SD. O Veículo foi carregado pelo estacionamento de um condomínio, onde foram apontadas ações a serem realizadas em algumas coordenadas espalhadas pelo estacionamento, assim pode ser monitorado o comportamento veículo que dará uma visão prévia de quando o mesmo for colocado no ambiente final de testes.

Para este evento de testes, foram determinados 5 pontos de coleta de dados, porém somando com o ponto inicial que também é considerado, assim totalizando 6 pontos. Neste teste inicial foi programado para simular uma coleta de água nos pontos 1, 2, 4 e 5 conforme mostrado na Tabela 2.

**Tabela 2 – Pontos de alvo para o primeiro teste no estacionamento**

| Ponto | Latitude    | Longitude   | Fazer Leitura? | Coletar água? |
|-------|-------------|-------------|----------------|---------------|
| 1     | -26.8103623 | -49,1025558 | Sim            | Sim           |
| 2     | -26.8103994 | -49,1020046 | Sim            | Sim           |
| 3     | -26.8103946 | -49,1023211 | Sim            | Não           |
| 4     | -26.8106914 | -49,1023372 | Sim            | Sim           |
| 5     | -26.8103718 | -49,1023198 | Sim            | Sim           |

**Fonte: Do Autor (2023).**

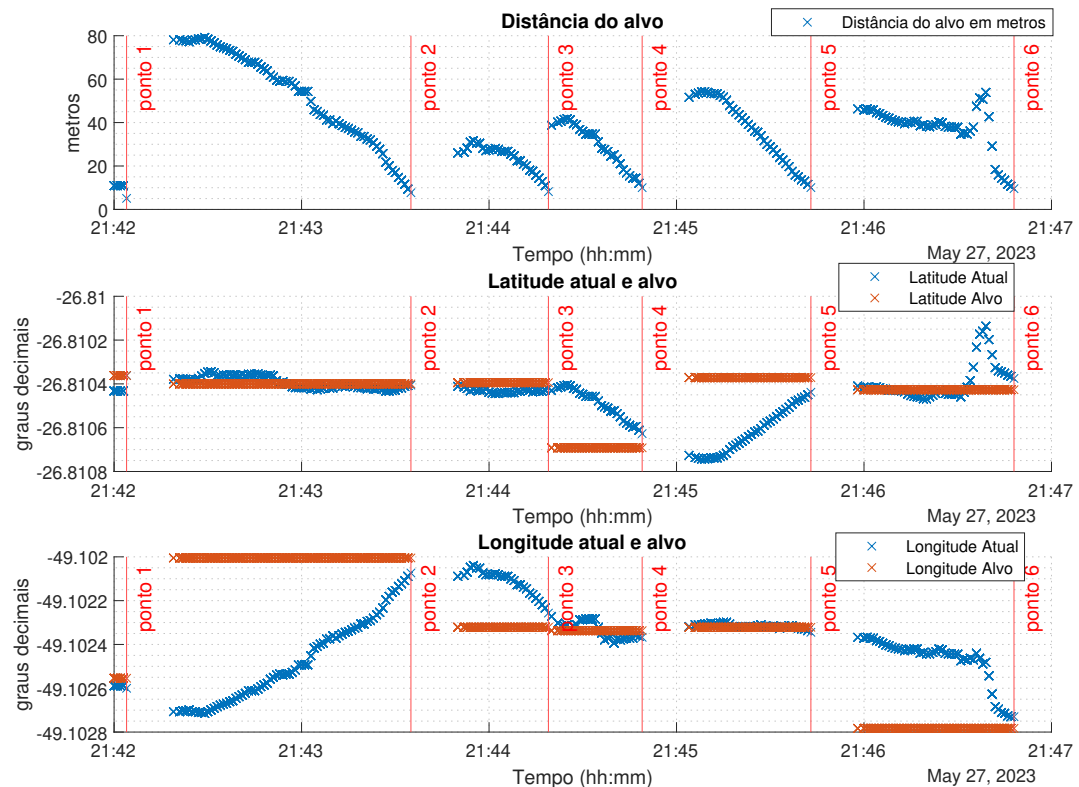
Os dados de latitude e longitude atual e alvo foram apontadas na Figura 58, onde aqui é indicada a variação de distância conforme há a locomoção para os alvos e as respectivas coordenadas.

Nota-se que nos dados obtidos é alcançada uma precisão relativamente boa, dado que os pontos coletados são bastante precisos, porém aqui percebe-se que se caso a precisão solicitada for menor que entre 10 a 7 metros, fica bastante difícil alcançar o alvo de forma certa. Outro fato a se notar é que há uma certa falha nos dados dos pontos 1, 2, 4 e 5 com um espaço de tempo sem dados, sendo provocada pelo tempo em que os motores das bombas de água estão ligados.

Na Figura 59 é mostrado um resumo de todas as coordenadas lidas durante este teste. Nota-se aqui que mesmo com uma taxa de precisão alta, o gps ainda pode indicar alguns erros de localização bastante consideráveis quando em espaços pequenos. Percebe-se também que durante o percurso, houve um certo desvio do ponto real e do ponto lido, onde durante



**Figura 58 – Dados obtidos de posição e distancia do alvo.**



**Fonte: Do Autor (2023).**

os testes este desvio já estava sendo percebido pela atuação dos motores indicando que a caminhada estava fora de curso.

Na Tabela 3 são dados os valores de temperatura da água e do ar em graus Celsius, a pressão em hPa, a umidade relativa em percentual, e por último qual reservatório foi enchido.

**Tabela 3 – Relatório de amostras no primeiro teste (estacionamento)**

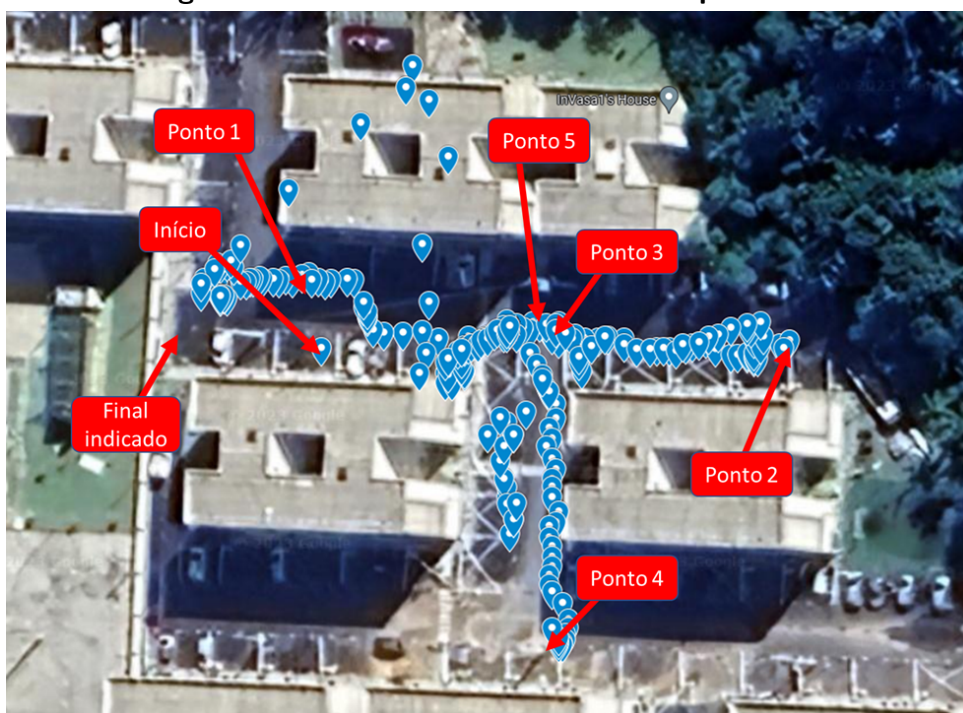
| Nº | Hora     | Latitude    | Longitude   | °C ar | UR%    | hPa     | °C Água | Res. |
|----|----------|-------------|-------------|-------|--------|---------|---------|------|
| 1  | 21:42:04 | -26.8103619 | -49.1026001 | 22.99 | 100,00 | 1014.73 | 22.50   | 1    |
| 2  | 21:43:35 | -26.8104057 | -49.1020737 | 22.83 | 100,00 | 1014.78 | 22.06   | 2    |
| 3  | 21:44:19 | -26.8104210 | -49.1022568 | 22.69 | 100,00 | 1014.79 | 22.00   | NC   |
| 4  | 21:44:49 | -26.8106270 | -49.1023674 | 22.60 | 100,00 | 1014.86 | 22.00   | 3    |
| 5  | 21:45:43 | -26.8104382 | -49.1023445 | 22.42 | 100,00 | 1014.81 | 21.94   | 4    |
| 6  | 21:46:48 | -26.8103752 | -49.1027298 | NC    | NC     | NC      | NC      | NC   |

**Fonte: Do Autor (2023).**

#### 4.8.1 Primeiras dificuldades encontradas

Durante o primeiro evento de testes foram percebidos alguns problemas em relação a complexabilidade do local de testes. Devido ao local de testes ser um estacionamento, existe a

**Figura 59 – Coordenadas amostradas e apontadas.**



**Fonte: Do Autor (2023).**

presença de edifícios altos que cercam o local de deslocamento do veículo que torna bastante difícil obter coordenadas válidas de geolocalização. Isso se dá pela visibilidade dos satélites ser apenas livre na direção vertical, assim dificultando bastante a triangularização das coordenadas. Isso pode ser percebido por um desvio mostrado na Figura 59, do qual indica que as coordenadas medidas estavam por cima de um dos edifícios. Por último, a presença de massas metálicas encontradas em grande volume próximo do local de testes também influencia na ação do magnetômetro, sendo que quanto mais próximo dos carros, maior a influência magnética dos metais é demonstrada. Todas essas condições não são esperadas em um ambiente de uso ideal, onde o veículo deve ser posicionado em ambiente apenas com água, geralmente onde não há a presença de grandes obstáculos metálicos ou edifícios altos próximos de sua localização.

#### **4.9 Testes fora da água para ajuste do tempo das bombas**

Como havia uma certa dificuldade de cálculo de tempo de sucção de cada bomba de água, onde não existem sensores de nível nas garrafas, bem como se torna complexo fato de o tempo de sucção ser diferente de quando as mangueiras estão cheias de ar ou cheias de líquido, o ajuste do tempo foi feito empiricamente. Para isso, foram feitos alguns ajustes no código para que ligasse as bombas por um pequeno tempo várias vezes até que fosse obtido um valor de tempo compatível com o volume de água de 200mL das garrafas. Após alguns testes com a sucção de água de recipientes controlados, o tempo de melhor ajuste foi de 14 segundos

em cada uma das garrafas. A Figura 60 mostra o esquema de posicionamento utilizado para verificação do tempo de enchimento das garrafas, onde em baixo do veículo foi posicionado um recipiente com água para a simulação das coletas.

**Figura 60 – Esquema de posicionamento do veículo para simulação de coleta de água.**



**Fonte: Do Autor (2023).**

#### **4.10 Testes em piscina pequena com água**

Para validar os cálculos de flutuação, e também verificar o controle de direção do veículo mediante pequenas perturbações com o ajuste de velocidade máxima em zero (apenas controle de direção sem deslocamento) o veículo foi colocado em uma pequena piscina com baixo nível de água para que fossem realizados os testes. Na Figura 61 é mostrada a piscina em que foram realizados os testes, sendo que esta é uma piscina infantil com tamanho aproximado de 5 metros de comprimento por 2 metros de largura. A piscina fica localizada no mesmo condomínio em que foram realizados os testes pelo estacionamento, e estava inoperante devido a um processo de reforma, porém ainda estava parcialmente cheia com aproximadamente 30cm de altura de água.

Na Figura 62 é mostrado o veículo já posicionado dentro da água, onde o veículo já demonstrava a estabilidade esperada devido ao formato de sua estrutura que foi discutido no Capítulo 3 deste trabalho. Mesmo causando algumas perturbações o veículo possui estabilidade em seu eixo de giro, bem como também estabilidade em seu balanço, sem que hajam inclinações consideráveis em sua estrutura.

Neste primeiro teste de flutuabilidade, nota-se que o veículo atende bem o que era esperado com relação as espumas utilizadas como flutuadores, onde a altura da água não chegou a tocar o compartimento das bombas de água na parte inferior da estrutura. Isto pode ser visto na Figura 63, onde é mostrada uma vista lateral indicando que a altura da água alcançada não chega a cobrir a base dos suportes de garrafas.



**Figura 61 – Piscina em que foi realizado o primeiro teste em água.**



**Fonte: Do Autor (2023).**

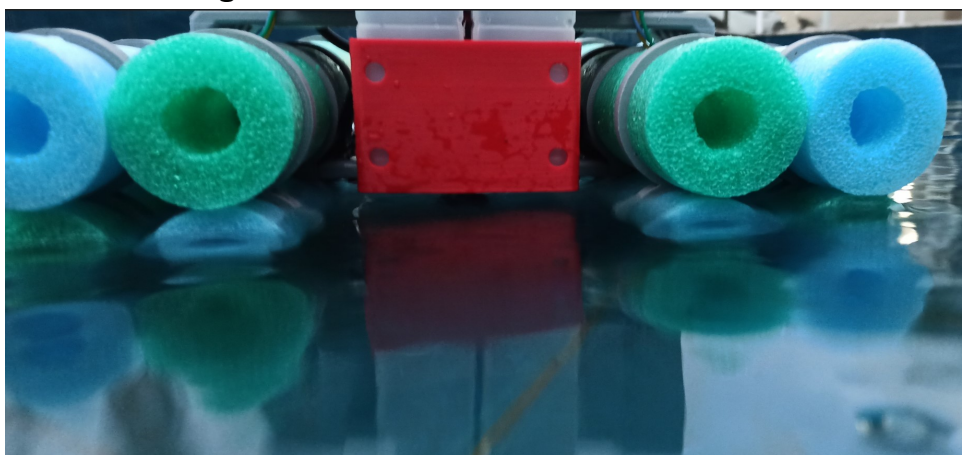
**Figura 62 – Veículo posicionado dentro da piscina.**



**Fonte: Do Autor (2023).**

Nos primeiros testes feitos em água, foi ajustado uma direção fixa para que o veículo aponte, assim, poderia ser verificado o esforço do veículo para corrigir a sua direção. Durante esse teste foram alterados manualmente os parâmetros PID até que se obtivesse um controle

**Figura 63 – Altura de submersão do veículo.**



**Fonte: Do Autor (2023).**

satisfatório sem grandes variações inesperadas, porém o ajuste foi apenas visual e empírico. Para os ensaios, inicialmente foi ajustado o parâmetro  $K_p$  em 10% do valor total, e assim foi possível ver que o sistema não seria perfeitamente controlado apenas com a ação proporcional, já que ficou com um comportamento oscilatório. A seguir foi incrementado o valor do controle derivativo, até 30%, o que fez diminuir consideravelmente a oscilação, porém ainda não era satisfatório, pois quando o veículo apontava para a direção próxima a desejada, os motores não exerciam esforço o suficiente para corrigir pequenos erros, que ao todo pode ser considerado um erro de regime. Por fim, foi adicionada a ação do controle integrador, fazendo com que o erro de regime ficasse bem pequeno, quase nulo, com apenas 4% do seu valor total para o parâmetro  $K_i$ . Após estes testes, foram feitos mais alguns ensaios de locomoção para verificar o funcionamento dos motores em água, e percebe-se que os motores são bastante potentes e atendem até acima das necessidades do veículo, o que cria a possibilidade para o uso do veículo em outras aplicações. Em relação a dinâmica do sistema quando em condições próximas a aplicação real foi possível verificar que a inércia do veículo é bem pequena nas curvas e no deslocamento, sendo que o veículo para quase imediatamente quando não há a atuação dos motores. Isso pode ser devido a frente dos flutuadores não serem perfeitamente hidrodinâmicas, o que no final das contas acaba sendo uma vantagem para o trabalho, já que torna um comportamento mais controlado. Com os dados obtidos neste teste, já seria possível colocar o veículo em condições mais complexas e reais para testes, que são as próximas etapas discutidas neste trabalho.

#### 4.10.1 Dificuldades encontradas o teste realizado em água dentro de piscina

Uma limitação nos testes realizados ficou por conta do espaço que a piscina possui, onde entre as duas pontas mais distantes da piscina, para fins de orientação de GPS não são considerados na prática como pontos diferentes, o que dificulta bastante quando necessita-se

ter um controle de posição em dispositivos geo localizados. Com isso, ao tentar inserir algumas coordenadas geográficas dentro da piscina, devido a tolerância estipulada de 8 metros, não foi possível fazer o teste de deslocamento de forma efetiva.

#### 4.11 Testes do veículo em condições reais

Para serem realizados os testes em condições reais, foi selecionado um lago cujas dimensões fossem suficientes para provocar a variação das coordenadas de GPS que são necessárias para a orientação e locomoção do veículo. O lago em questão fica dentro da propriedade do Recanto Fritzke em Blumenau, do qual possui área de aproximadamente  $2.256m^2$  sendo um lago bem tranquilo, sem grandes perturbações ou obstáculos que pudessem atrapalhar os testes de alguma forma. A vista de cima do lago é mostrada na Figura 64.

**Figura 64 – Vista do lago do Recanto Fritzke por satélite.**



**Fonte: Adaptado de (GOOGLE, 2023).**

##### 4.11.1 Primeiro teste no lago

Para fazer um teste inicial com algumas coordenadas dentro do lago, foi separado uma lista de pontos no roteiro do veículo para serem alcançados durante o percurso dele no lago. Estes pontos foram apontados inicialmente em um conjunto de possíveis pontos escolhidos antes da visita no local, visando evitar problemas que pudessem ser causados pela falta de conexão a internet no local de testes para a verificação das coordenadas no mapa. Os pontos que poderiam ser indicados são listados na Tabela 4. Nota-se que existem mais de 4 pontos para coleta de água, apontados propositalmente para verificar a limitação de coleta de água por meio das bombas do veículo, conforme previsto pelo algoritmo.

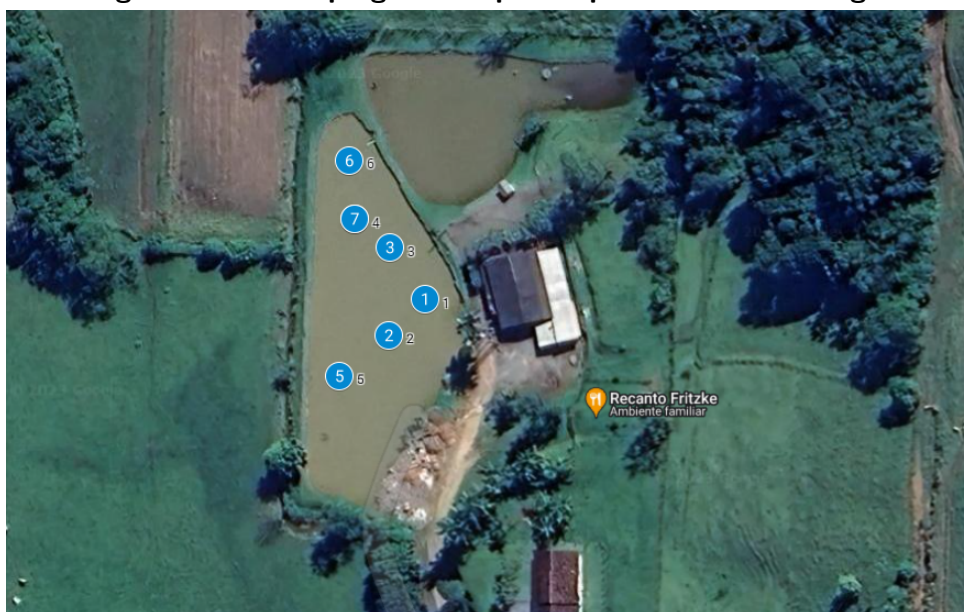


**Tabela 4 – Pontos de alvo para o primeiro teste no lago**

| Ponto | Latitude     | Longitude    | Fazer Leitura? | Coletar água? |
|-------|--------------|--------------|----------------|---------------|
| 1     | -26.74865051 | -49.09137955 | Sim            | Não           |
| 2     | -26.74872799 | -49.09146767 | Sim            | Sim           |
| 3     | -26.74853792 | -49.09146631 | Sim            | Não           |
| 4     | -26.74847618 | -49.09155037 | Sim            | Sim           |
| 5     | -26.74881757 | -49.09158833 | Sim            | Sim           |
| 6     | -26.74834785 | -49.09156392 | Sim            | Sim           |
| 7     | -26.74847618 | -49.09155037 | Sim            | Sim           |

**Fonte: Do Autor (2023).**

De forma a ilustrar onde ficam cada um desses pontos, foi gerada uma imagem no Google Maps que representasse a ordem dos destinos a serem alcançados. Isto é mostrado na Tabela 4, onde são numerados todos os pontos no mapa. Nota-se que quando é necessário o veículo apontar para o mesmo ponto mais de uma vez, o número é mostrado com um valor diferente respectivo da ordem do ponto ao lado do local indicado.

**Figura 65 – Rota programada para o primeiro teste no lago.**

**Fonte: Adaptado de (GOOGLE, 2023).**

Para que seja possível verificar o funcionamento do controle do veículo, o parâmetro de velocidade foi ajustado em 20% da velocidade máxima do veículo, o parâmetro de ajuste percentual de Kp foi ajustado em 20%, e os parâmetros percentuais de Ki e Kd foram ajustados em 4% e 30% respectivamente. Neste teste, a precisão ajustada para o alvo se manteve em 8 metros de distância para validar que o veículo está no ponto desejado. Ao colocar o veículo na água, uma vez que todos os sensores já estejam calibrados e o GPS conectado com os satélites, iniciou-se o processo de navegação como ilustrado pela fotografia na Figura 66.

Com estes parâmetros o veículo fez a navegação sequencial nos pontos marcados, apresentando o traçado mostrado na Figura 67. Aqui pode-se perceber que o veículo possui

**Figura 66 – Veículo durante o processo de navegação.**



**Fonte: Do Autor (2023).**

um deslocamento quase linear entre cada um dos pontos, tendo apenas uma pequena variação na direção que deixa os rastros de navegação um pouco curvos.

**Figura 67 – Rota do veículo durante a navegação do primeiro teste no lago.**

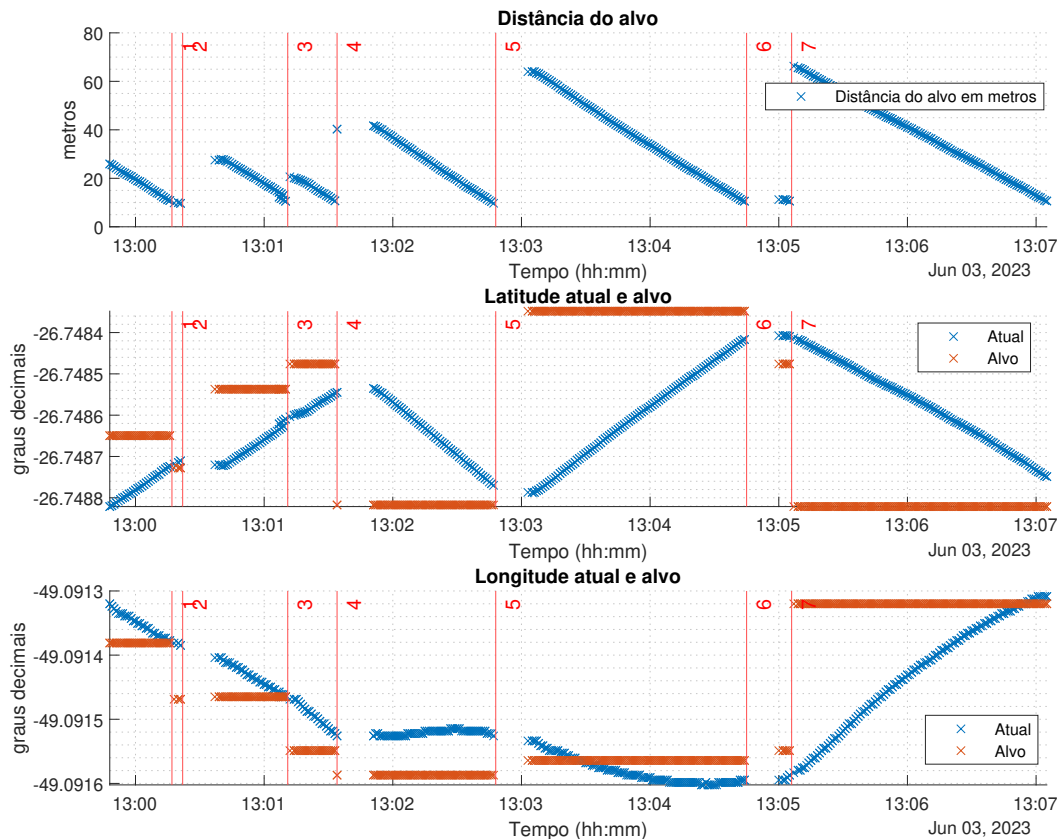


**Fonte: Adaptado de (GOOGLE, 2023).**

Como resultado, o veículo apresentou a variação de coordenadas e distância do alvo conforme mostrado na Figura 68, onde é demonstrado pelas curvas que o veículo de forma autônoma ajustou para os pontos de alvo, e realizou a locomoção até cada um dos pontos de forma autônoma.



**Figura 68 – Dados obtidos com a amostragem do primeiro teste no lago.**



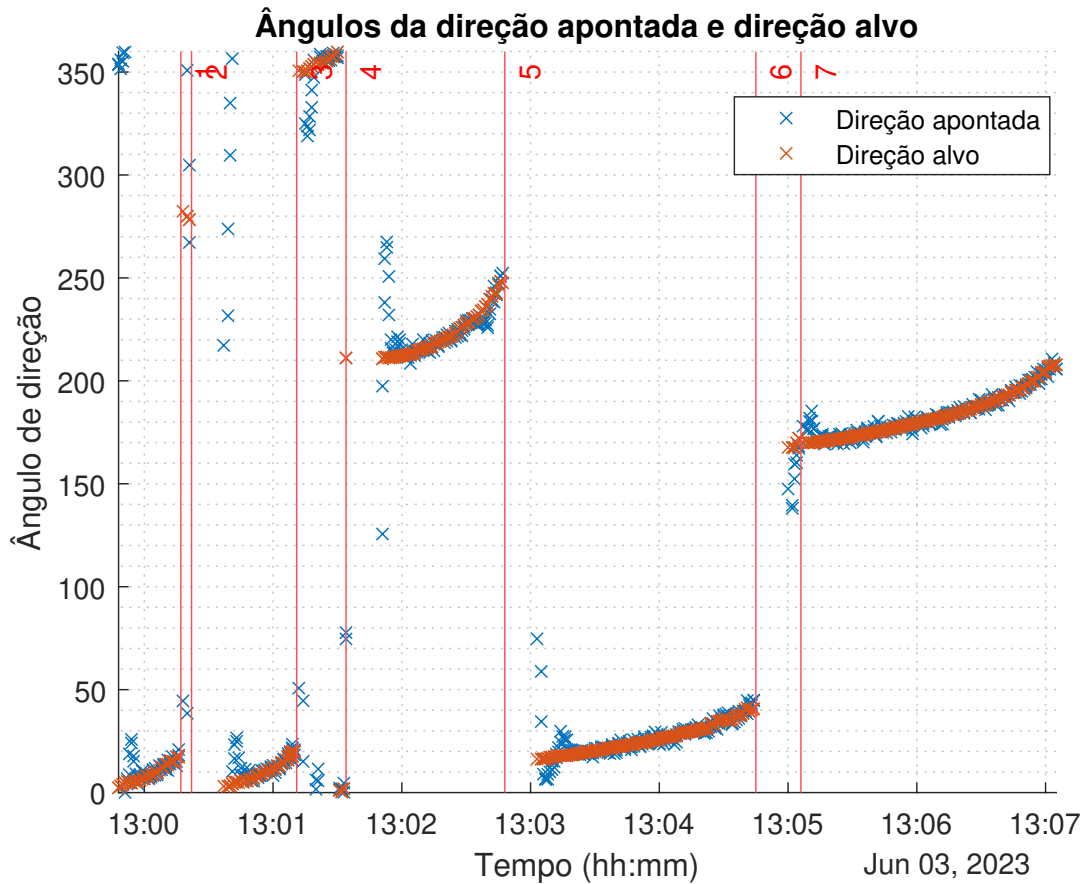
**Fonte: Do Autor (2023).**

Com os dados de variação de distância por tempo para os valores ajustados, é possível afirmar que quando a locomoção é linear, o veículo alcança uma velocidade de deslocamento aproximada de  $0,47m/s$ , onde ao somar todos os deslocamentos executados resulta em um valor total de aproximadamente 214 metros.

Durante o percurso, foi medida também a variação do ângulo apontado para o ângulo alvo, assim é possível observar a atuação do controle de direção para cada um dos pontos como pode ser visto na Figura 69.

Uma coisa a se notar neste gráfico é que pelas pequenas diferenças entre os dois motores, e pela ligeira assimetria da estrutura do veículo, e principalmente devido a um dano ocasionado na estrutura do sensor magnetômetro durante o transporte para o local de teste, os dados amostrados indicam que o veículo tende a apontar para um dos seus lados, porém, esta diferença é compensada pelo controlador, sendo corrigida em tempo real conforme o veículo se move sobre a superfície da água. Para indicar a variação do valor do erro, foram amostrados os valores de erro calculados pelo microcontrolador durante todo o período de navegação, como é apresentado na Figura 70. Com os dados obtidos, pode-se afirmar com bastante certeza de que a atuação do controle é bem eficiente para o veículo durante a navegação, já que o

**Figura 69 – Diferença dos ângulos apontados para os ângulos alvos durante o primeiro teste no lago.**



Fonte: Do Autor (2023).

tempo de estabelecimento é bastante pequeno, e ainda não é observada a oscilação do sistema durante a navegação.

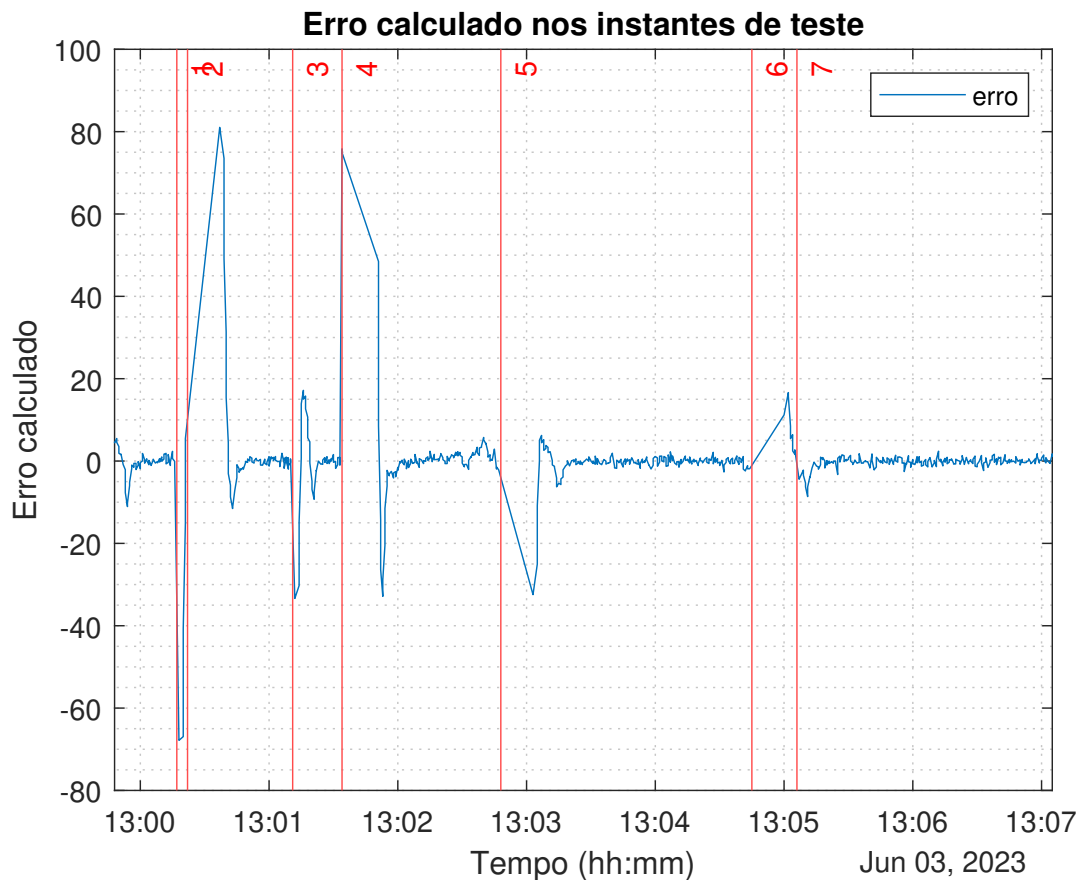
Por fim, na Tabela 5 são mostrados os dados coletados em cada um dos pontos visitados. Aqui, um detalhe a se observar é que no ponto 7 não foi coletada amostra de água, mesmo tendo o comando para coleta neste ponto do roteiro de navegação, porém há uma notificação no relatório de coleta com a mensagem "Todos os reservatórios cheios, nenhum enchido" indicando que este ponto era um ponto de coleta, porém não foi coletada nenhuma amostra de água.

Na Figura 71 é mostrado como que as amostras de água são trazidas de volta ao ponto de origem, sendo que cada amostra contém aproximadamente  $200mL$ .

#### 4.11.2 Segundo teste no lago

Para validar a construção do trabalho mais uma vez, foram separados mais alguns pontos para coleta, em que agora, os pontos são mais variados, permitindo uma navegação maior do veículo no decorrer do lago, assim demonstrando mais as suas capacidades. Os pontos

**Figura 70 – Amostragem dos valores de erro durante o primeiro teste no lago.**



Fonte: Do Autor (2023).

**Tabela 5 – Relatório de amostras do primeiro teste no lago.**

| Nº | Hora     | Latitude    | Longitude   | °C    | UR%   | hPa     | °C Água | Res. |
|----|----------|-------------|-------------|-------|-------|---------|---------|------|
| 1  | 13:00:17 | -26.7487202 | -49.0913811 | 20.88 | 97.09 | 1018.41 | 21.12   | NC   |
| 2  | 13:00:22 | -26.7487125 | -49.0913925 | 20.95 | 97.08 | 1018.45 | 21.12   | 1    |
| 3  | 13:01:11 | -26.7486057 | -49.0914650 | 21.79 | 94.51 | 1018.41 | 21.12   | NC   |
| 4  | 13:01:34 | -26.7485447 | -49.0915260 | 22.18 | 91.86 | 1018.38 | 21.12   | 2    |
| 5  | 13:02:48 | -26.7487736 | -49.0915260 | 23.09 | 89.85 | 1018.41 | 21.19   | 3    |
| 6  | 13:04:45 | -26.7484131 | -49.0915947 | 24.53 | 85.96 | 1018.39 | 21.00   | 4    |
| 7  | 13:05:06 | -26.7484131 | -49.0915833 | 24.65 | 82.34 | 1018.37 | 21.00   | NC   |
| 8  | 13:07:06 | -26.7487526 | -49.0913086 | NC    | NC    | NC      | NC      | NC   |

Fonte: Do Autor (2023).

escolhidos são mostrados na Tabela 6, onde o veículo pode até passar pelos mesmos pontos mais de uma vez com o intuito de otimizar as rotas para evitar os poucos obstáculos que estavam presentes na água durante os testes.

Dessa forma, os pontos escolhidos são apresentados na Figura 72, ordenados numericamente conforme indicado. Nessa segunda rodada de testes no lago buscou-se atingir pontos mais distantes que não foram visitados no primeiro teste, resultando numa quantidade de 12 pontos, incluindo o ponto inicial que deve ser o ponto de retorno.

**Figura 71 – Amostra de água coletada do lago.**

Fonte: Do Autor (2023).

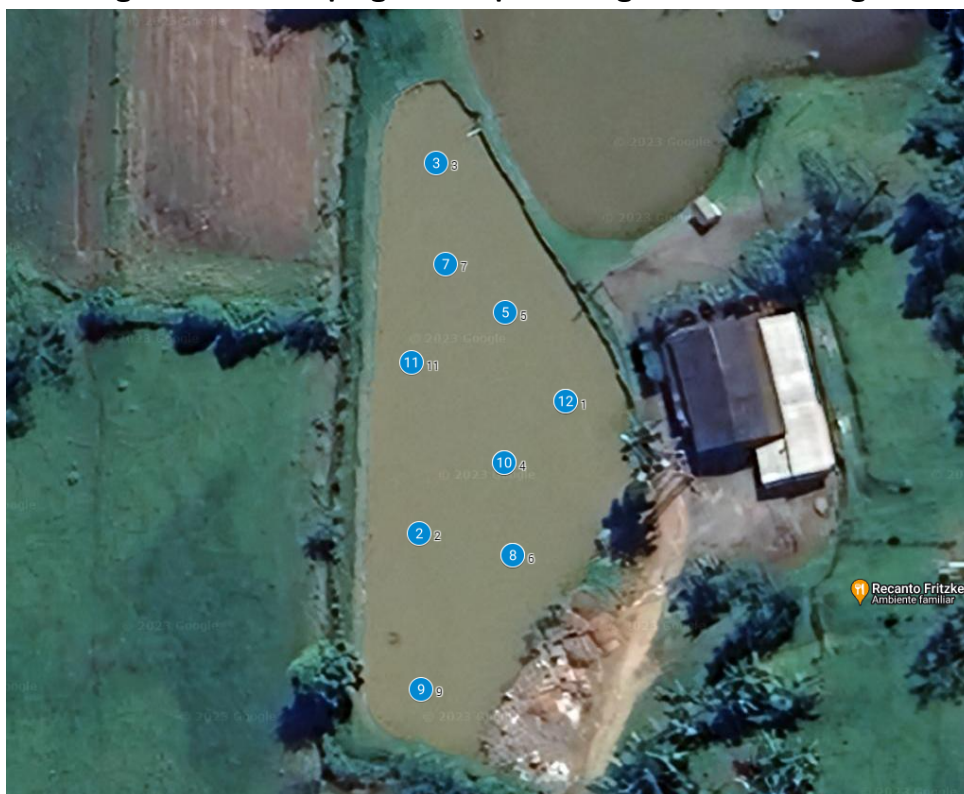
**Tabela 6 – Pontos de alvo para o segundo teste no lago**

| Ponto | Latitude     | Longitude    | Fazer Leitura? | Coletar água? |
|-------|--------------|--------------|----------------|---------------|
| 1     | -26.74865051 | -49.09137955 | Sim            | Sim           |
| 2     | -26.74881757 | -49.09158833 | Sim            | Não           |
| 3     | -26.74834785 | -49.09156392 | Sim            | Sim           |
| 4     | -26.74872799 | -49.09146767 | Sim            | Sim           |
| 5     | -26.74853792 | -49.09146631 | Sim            | Sim           |
| 6     | -26.74884542 | -49.09145547 | Sim            | Sim           |
| 7     | -26.74901491 | -49.09155037 | Sim            | Sim           |
| 8     | -26.74884542 | -49.09145547 | Sim            | Não           |
| 9     | -26.74901491 | -49.09158562 | Sim            | Não           |
| 10    | -26.74872799 | -49.09146767 | Sim            | Não           |
| 11    | -26.74860087 | -49.09159917 | Sim            | Não           |
| 12    | -26.74865051 | -49.09137955 | Sim            | Não           |

Fonte: Do Autor (2023).

Para o segundo teste no lago, os parâmetros de controle foram alterados para se obter uma ação menos agressiva do veículo, porém com velocidade de deslocamento maior, aumentando a sua velocidade de 20% para 30%, e os ajustes percentuais dos parâmetros  $K_p$ ,  $K_i$  e  $K_d$  para 20% , 4% e 40% respectivamente. Com este ajuste de parâmetros espera-se que o veículo demore mais para entrar em regime durante as curvas, mas também fazendo curvas mais suaves com raio maior. Uma das motivações para a alteração dos parâmetros é a necessidade de verificação de mudança de comportamento no controle do veículo, assim sendo possível observar pelos dados coletados. Outro fator ajustado na programação foi o valor da precisão necessária para validação de localização, onde antes o valor era de um raio de 8 metros, agora é de 3 metros, sendo uma condição de teste muito mais ousada, porém confiável devido a grande abertura de espaço do local.

**Figura 72 – Rota programada para o segundo teste no lago.**



**Fonte: Adaptado de (GOOGLE, 2023).**

Como resultado, o veículo demonstrou a rota ilustrada pela Figura 73, sendo agora um percurso muito maior também com uma variedade maior de locais visitados. Ao todo o percurso equivale a aproximadamente 408 metros de distância.

Na Tabela 7 é indicado o relatório de coleta de todos os 12 pontos especificados no roteiro, bem como também pode-se notar que foram feitas apenas as 4 amostragens de água, descartando as demais solicitações para esta atividade assim que todos os reservatórios estavam cheios.

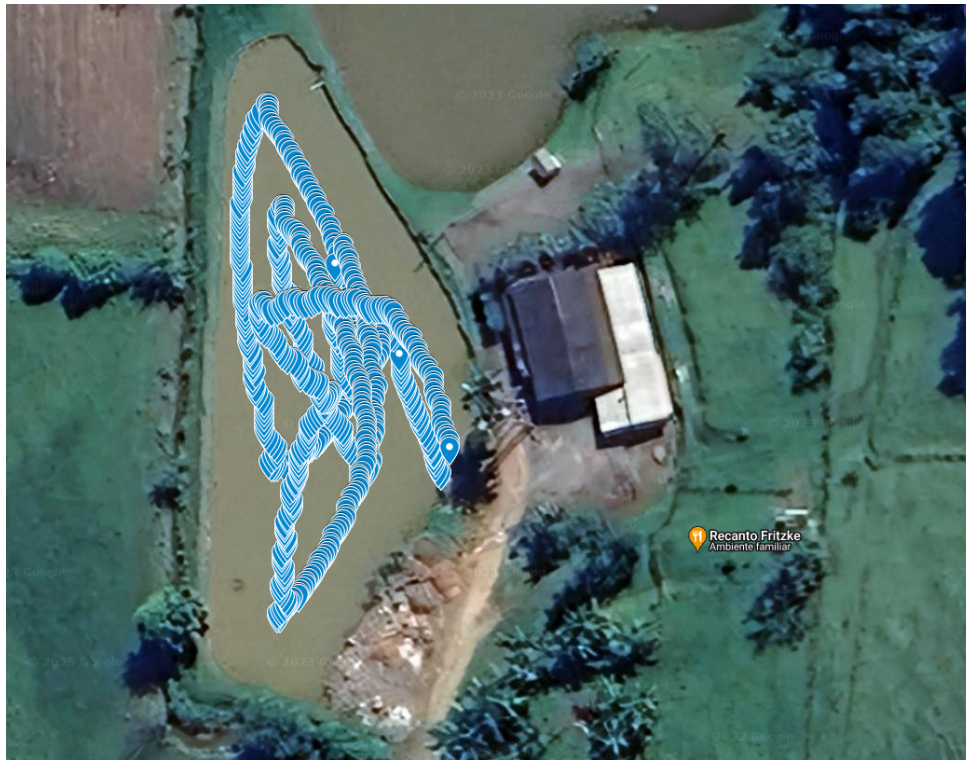
Com os dados coletados mostrados na Figura 74 já é possível afirmar que o reflexo na alteração de velocidade resultou em  $0,79m/s$ , quase dobrando a velocidade encontrada anteriormente. A não linearidade da alteração do parâmetro se dá pela dinâmica do sistema não ser linear, assim mesmo com uma alteração de 20% para 30% da velocidade máxima, o reflexo no comportamento do veículo é grande.

Outro fator que se pode observar pelas variações de latitude e longitude é que como a variação de latitude é maior entre cada um dos pontos, a variação de longitude aparenta ser mais suave, mas isso se dá apenas pela escala do gráfico apresentado. Outro ponto de atenção no gráfico é que no quesito de deslocamento, o ajuste dos parâmetros PID não fizeram diferença, já que não estão diretamente relacionados ao deslocamento linear do veículo.

Observando os resultados de variação angular do veículo pela Figura 75 pode-se ver com bastante destaque que a alteração nos parâmetros PID realmente apresentaram uma



**Figura 73 – Rota do veículo durante a navegação do segundo teste no lago.**



**Fonte: Adaptado de (GOOGLE, 2023).**

**Tabela 7 – Relatório de amostras do segundo teste no lago.**

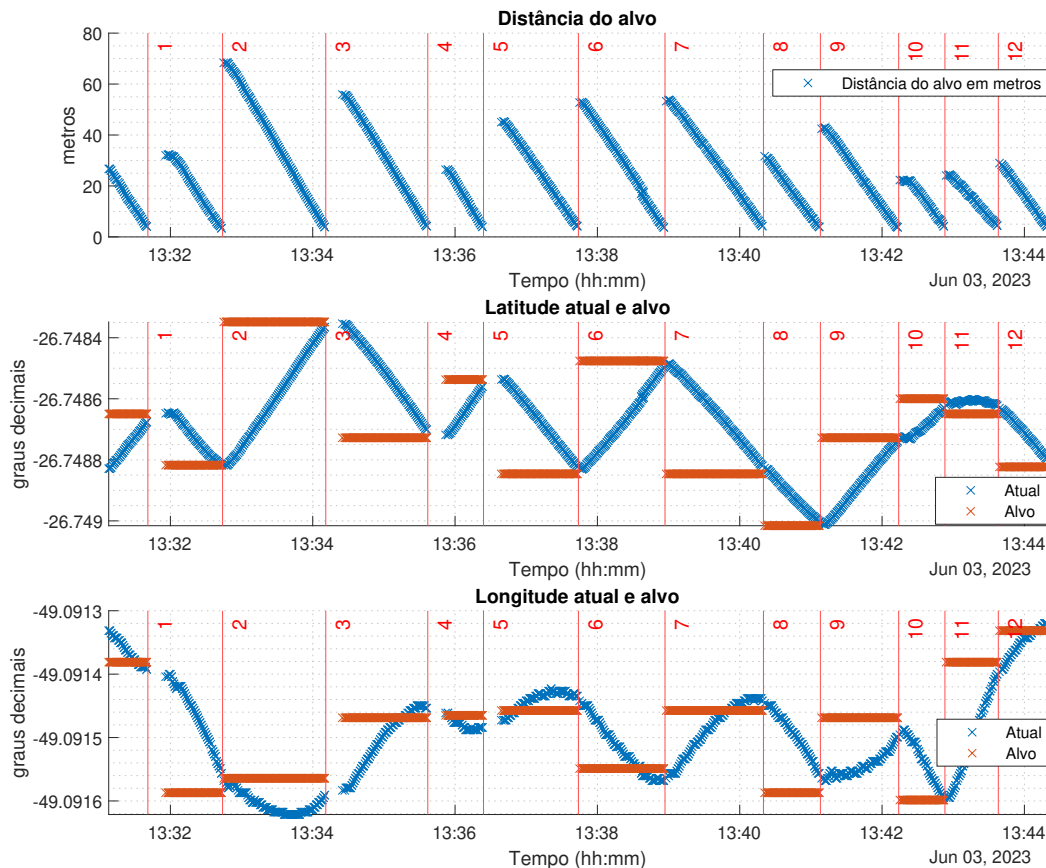
| Nº | Hora     | Latitude    | Longitude   | °C ar | UR%   | hPa     | °C Água | Res. |
|----|----------|-------------|-------------|-------|-------|---------|---------|------|
| 1  | 13:31:41 | -26.7486706 | -49.0913925 | 25.74 | 82.11 | 1018.40 | 21.31   | 1    |
| 2  | 13:32:44 | -26.7488174 | -49.0915604 | 26.53 | 73.61 | 1018.37 | 21.50   | NC   |
| 3  | 13:34:11 | -26.7483597 | -49.0915909 | 27.05 | 69.34 | 1018.38 | 21.62   | 2    |
| 4  | 13:35:37 | -26.7487068 | -49.0914536 | 27.10 | 70.13 | 1018.33 | 21.37   | 3    |
| 5  | 13:36:24 | -26.7485542 | -49.0914841 | 27.13 | 66.98 | 1018.34 | 21.50   | 4    |
| 6  | 13:37:44 | -26.7488270 | -49.0914383 | 26.79 | 68.54 | 1018.34 | 21.31   | NC   |
| 7  | 13:38:57 | -26.7484913 | -49.0915642 | 27.27 | 62.87 | 1018.37 | 21.69   | NC   |
| 8  | 13:40:20 | -26.7488251 | -49.0914459 | 27.05 | 66.02 | 1018.30 | 21.31   | NC   |
| 9  | 13:41:08 | -26.7490063 | -49.0915604 | 26.90 | 65.12 | 1018.29 | 21.87   | NC   |
| 10 | 13:42:14 | -26.7487335 | -49.0914955 | 27.34 | 67.23 | 1018.29 | 21.62   | NC   |
| 11 | 13:42:53 | -26.7486248 | -49.0915947 | 27.81 | 65.14 | 1018.27 | 21.75   | NC   |
| 12 | 13:43:38 | -26.7486286 | -49.0913963 | 27.96 | 64.37 | 1018.24 | 21.44   | NC   |
| 13 | 13:44:20 | -26.7488022 | -49.0913239 | NC    | NC    | NC      | NC      | NC   |

**Fonte: Do Autor (2023).**

grande diferença no ajuste de direção, sendo mostrado pelas oscilações entre as curvas de direção apontada e direção alvo. Esta variação pode inclusive ter forte relação com o aumento de velocidade do veículo, do qual o controle tem menos tempo para a correção dos valores antes que o veículo se desloque o suficiente para o erro ser evidenciado.

A Figura 76 demonstra que houve uma real variação nos valores de erro sendo é muito perceptível a resposta da alteração dos parâmetros de funcionamento do veículo. Porém é importante apontar que alguns picos iniciais dos valores de erro são devido a oposição da

**Figura 74 – Dados obtidos com a amostragem do segundo teste no lago.**



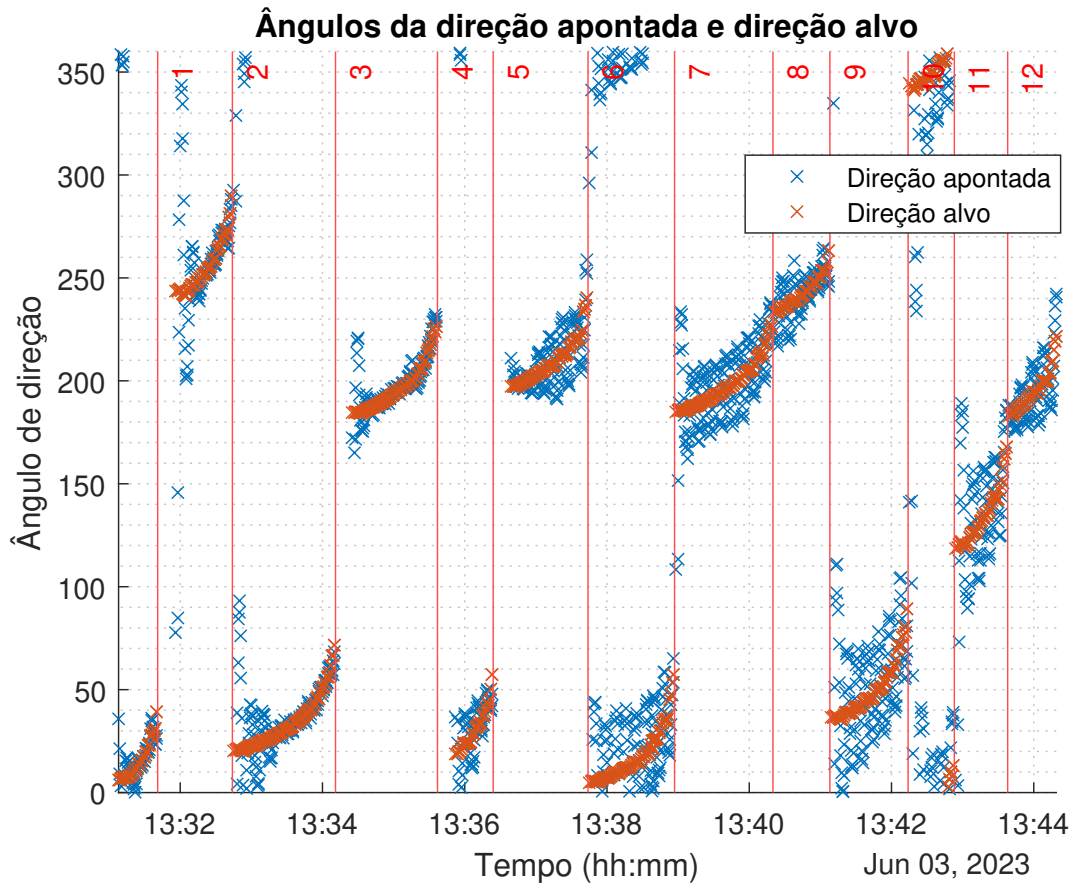
**Fonte: Do Autor (2023).**

sua direção apontada quando é determinada a nova direção alvo, sendo mais perceptível nos pontos em que não há coleta de água, já que o veículo faz apenas uma rápida amostra de dados, e subitamente gira para a direção alvo.

Parte da causa deste aumento de erro apresentando as oscilações é também a instabilidade do sensor magnetômetro devido ao dano ocasionado no transporte do veículo ao local de testes, onde nota-se que a oscilação aumenta mais próximo do final do percurso, o que pode indicar uma pequena variação da direção do sensor com relação a frente do veículo.

Um fato a se notar durante a interpretação do gráfico mostrado na Figura 76 bem como nos demais, é que há uma coleta de água que leva aproximadamente 14 segundos nos pontos 1, 3, 4 e 5, sendo que nos demais pontos indicados para coleta de água não há uma demora considerável entre os pontos. Isso é refletido ao gerar um gráfico de linha, em que os pontos mesmo sem dados são unidos pela linha conforme aqui visto, dando a falsa impressão de que o valor de erro se manteve alto por um longo período de tempo.

**Figura 75 – Diferença dos ângulos apontados para os ângulos alvos durante o segundo teste no lago.**



Fonte: Do Autor (2023).

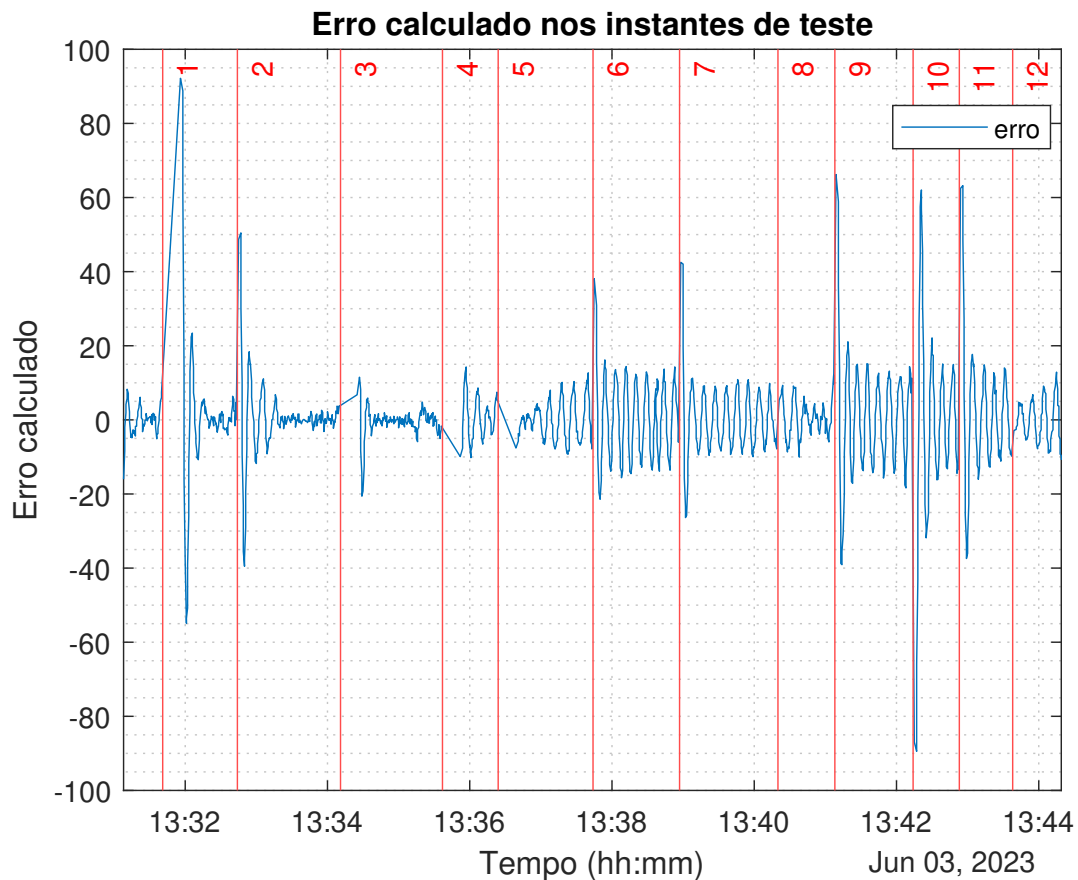
#### 4.11.3 Desempenho das baterias

Ao todo foram utilizadas 6 baterias de íon de lítio, onde a soma da capacidade total medida por um testador de baterias *Litokala Lii-500* equivale a 12607mAh. Ao fim do teste, as baterias retornaram com carga aproximada de 90% da capacidade nominal, sendo que os testes foram realizado por aproximadamente 40 minutos, incluindo os tempos para conexão de GPS, ajuste dos parâmetros de controle e navegação. Sendo assim a carga consumida das baterias foi de aproximadamente 1261mAh em uma tensão nominal de 3,7V resultando em um consumo total de 4,666Wh. Estima-se que o tempo de navegação aproximado total possa alcançar tranquilamente um tempo aproximado de 400 minutos. Os dados nominais das baterias são mostrados na 8.

Um fato observado durante os testes foi que quando eram usadas metade das baterias apenas (um dos compartimentos cheio), o veículo não desempenha o funcionamento dos motores perfeitamente como esperado, sendo de forma limitada e até mesmo reiniciando os controladores eletrônicos de velocidade por sub-tensão quando exigida a sua potência máxima.



**Figura 76 – Amostragem dos valores de erro durante o segundo teste no lago.**



Fonte: Do Autor (2023).

**Tabela 8 – Dados das células de baterias.**

| Nº | Capacidade nominal (mAh) | Resistência interna ( $m\Omega$ ) | Tensão nominal (V) |
|----|--------------------------|-----------------------------------|--------------------|
| 1  | 1910                     | 28                                | 3,7                |
| 2  | 1843                     | 31                                | 3,7                |
| 3  | 2283                     | 31                                | 3,7                |
| 4  | 2286                     | 67                                | 3,7                |
| 5  | 2175                     | 41                                | 3,7                |
| 6  | 2110                     | 31                                | 3,7                |

Fonte: Do Autor (2023).

#### 4.11.4 Dificuldades encontradas para os testes no lago

Um fato que não pode ser descartado é que devido ao meio de transporte utilizado para levar o veículo até o local de testes ser uma moto, e o veículo ter sido colocado dentro do baú da moto como mostrado na Figura 77 ocasionou a quebra do local em que o sensor magnetômetro fica alojado, assim não tendo mais a segurança e estabilidade da qual foi projetada para o sensor.

Na Figura 78 é mostrado o local em que o dano foi ocasionado. Com algumas adaptações realizadas no local de testes foi possível colocar o sensor em seu local aproximado,

**Figura 77 – Veículo armazenado dentro do baú da moto para transporte.**



**Fonte: Do Autor (2023).**

tornando possível a realização de todos os testes necessários, porém tendo em mente que este fato causaria uma certa variação no comportamento esperado do veículo.

**Figura 78 – Local em que houve o dano durante transporte.**



**Fonte: Do Autor (2023).**

Apesar do ocorrido, não foram encontrados grandes problemas na operação, já que o controle PID utilizado é bastante adaptável ao tipo de mudança causada no sistema, ainda

mantendo o veículo operante e capaz de realizar as operações planejadas. Uma técnica que poderia ser feita para que fosse evitado esse tipo de problema em uma segunda versão do projeto é deixar a barra de rosca que sustenta o suporte dos sensores um pouco mais curta, já que não há extrema necessidade de manter o sensor tão distante do restante da estrutura.

## 5 CONCLUSÃO

Retomando aos objetivos iniciais deste trabalho, é possível afirmar que aos poucos foram desenvolvidas etapas como a concepção da solução a ser aplicada, a elaboração do projeto mecânico do trabalho, o estudo das ferramentas e recursos necessários para atender as expectativas de execução, o desenvolvimento de algoritmos que criassem rotinas em que quando combinadas fossem capazes de dar as funções de automatização essenciais para fazer deste projeto algo possível, executável e aplicado.

Ao longo do trabalho foram encontrados desafios que provocaram em diversas vezes a oscilação da curva de aprendizado com recursos que estavam fora do escopo da formação da engenharia elétrica, porém que trazem um contexto e justificativa para a execução das atividades destacadas como objetivos do trabalho.

Ao fim deste trabalho foi possível observar que todo o desenvolvimento realizado na construção do projeto acabou por fornecer resultados satisfatórios, tanto do aspecto de construção, bem como do aspecto de desempenho e monitoramento de suas respectivas condições de funcionamento. A abordagem adotada combinou elementos de engenharia elétrica, utilização de componentes de baixo custo e testes experimentais em diferentes ambientes, tais como piscina e lago, além da realização de coletas de amostras com o auxílio humano. Através da integração de conhecimentos de diversas disciplinas da engenharia elétrica, como eletrônica, controle, computação e sistemas micro controlados, foi possível construir um veículo autônomo capaz de se movimentar de forma independente na superfície da água e coletar amostras para análise da qualidade da água. A utilização de componentes de baixo custo contribuiu para viabilizar a construção do veículo em larga escala, tornando-o acessível e economicamente viável, já que quando montado em maior escala de produção, tende a apresentar um custo reduzido, e ainda melhorando a qualidade do produto.

No contexto atual, em que a preservação e monitoramento dos recursos hídricos são fundamentais para a sustentabilidade e saúde do planeta, a contribuição deste trabalho é relevante, uma vez que oferece uma solução tecnológica viável e acessível para o monitoramento contínuo da qualidade da água em reservatórios. Espera-se que esse veículo autônomo de superfície possa ser aplicado em larga escala, auxiliando na identificação precoce de problemas ambientais e na tomada de decisões eficientes para a preservação dos recursos hídricos desde que os estudos relacionados ao projeto sejam estendidos em trabalhos futuros.

As conquistas alcançadas ao longo do projeto fornecem uma sólida base para futuras pesquisas e aprimoramentos nessa área, além de reforçar a importância da engenharia elétrica como uma ferramenta indispensável na solução de problemas complexos e na busca por soluções inovadoras. Apesar dos desafios encontrados ao longo do trabalho, foi possível encontrar uma solução viável e aplicável para cada um deles, mostrando que mesmo em grandes dificuldades, ainda podem ser realizadas pequenas adaptações que tornem as ideias compatíveis com a realidade.

## REFERÊNCIAS

- ADAFRUIT. **Adafruit bme280 library**. 2021. Disponível em: [https://github.com/adafruit/Adafruit\\_BME280\\_Library/](https://github.com/adafruit/Adafruit_BME280_Library/). Acesso em: 18 maio 2023.
- ADAFRUIT. **Adafruit unified sensor driver**. 2023. Disponível em: [https://github.com/adafruit/Adafruit\\_Sensor](https://github.com/adafruit/Adafruit_Sensor). Acesso em: 18 maio 2023.
- ALMEIDA, T. E. P. de. **Sistema de sensoriamento de orientação para um veículo aquático de superfície utilizando sensores de baixo custo**. 2014. Fevereiro 2014. 77 f. Dissertação (Mestrado em Ciências – Programa de Engenharia Elétrica) — Escola de Engenharia de São Carlos, São Carlos, 2014.
- AMRAN, I. Y. *et al.* Development of autonomous underwater vehicle for water quality measurement application. Springer Singapore, Singapore, p. 139–161, 2021.
- ANA, A. nacional de águas; CETESB companhia ambiental do estado de são paulo. **Guia Nacional de Coleta e Preservação de Amostras**. 2011. Disponível em: <https://cetesb.sp.gov.br/wp-content/uploads/2021/10/Guia-nacional-de-coleta-e-preservacao-de-amstras-2012.pdf>.
- BENNETT, J. K. **Servo library for esp32**. 2017. Disponível em: <https://github.com/jkb-git/ESP32Servo>. Acesso em: 18 maio 2023.
- BURTON, M. **Arduino library for maxim temperature integrated circuits**. 2020. Disponível em: <https://github.com/milesburton/Arduino-Temperature-Control-Library>. Acesso em: 18 maio 2023.
- CORPORATION, Q. **3-axis magnetic sensor qmc5883l**. 2016. Disponível em: [https://datasheet.lcsc.com/szlcsc/QST-QMC5883L-TR\\_C192585.pdf](https://datasheet.lcsc.com/szlcsc/QST-QMC5883L-TR_C192585.pdf). Acesso em: 12 abr. 2023.
- DORF, R. C.; BISHOP, R. H. **Modern control systems**. Harlow: Pearson, 2022.
- EL-RABBANY, A. **Introduction to gps: the global positioning system**. 1. ed. Boston: Artech House Mobile Communications Series, 2002.
- FADALI, M. S.; VISIOLI, A. **Digital control engineering analysis and design**. Oxford: Elsevier, 2013.
- GEOGRAPHY, G. **Magnetic north vs geographic (true) north pole**. 2022. Disponível em: <https://gisgeography.com/magnetic-north-vs-geographic-true-pole/>. Acesso em: 11 abr. 2023.
- GOOGLE. **Google maps**. 2023. Disponível em: <https://www.google.com.br/maps/>. Acesso em: 03 jun. 2023.
- GUTIERREZ, A. **Usina hidrelétrica de itaipu**. 1980. Disponível em: <https://www.andradegutierrez.com.br/Projetos/Itaipu.aspx#300>. Acesso em: 01 jun. 2023.
- HALLIDAY, D.; RESNICK, R.; WALKER, J. **Fundamentals of physics**. [S.l.]: John Wiley & Sons, 2013.
- HART, M. **TinyGPSplus**. 2013. Disponível em: <https://github.com/mikalhart/TinyGPSPlus>. Acesso em: 18 maio 2023.

JUNIOR, C. R. da S.; LIMA, W. da S.; OLIVEIRA, M. A. A. de. Barco-robô autônomo para monitoramento ambiental de reservatórios. **Seminário de Iniciação Científica - UFG**, 2012.

KHOSRAVI, K.; ALIZADEH, M. S.; POURMAHDIAN, H. Comparison of a designed scalar proton precession magnetometer with a scalar calibrated 3-axis fluxgate magnetometer. *In: 2020 28th Iranian Conference on Electrical Engineering (ICEE)*. [S.l.: s.n.], 2020. p. 1–4.

KRAUS, O. **U8g2: library for monochrome displays, version 2**. 2021. Disponível em: <https://github.com/olikraus/u8g2>. Acesso em: 18 maio 2023.

LIU, Z.; ZHU, M. Calibration and error compensation of magnetometer. *In: The 26th chinese control and decision conference (2014 CCDC)*. [S.l.: s.n.], 2014. p. 4122–4126.

MARTINS, L. R. **Construção de um protótipo de barco robô de baixo custo para análise de qualidade de água em rios e reservatórios**. 2020. Abril 2020. 76 f. Dissertação (Mestrado em Segurança de Barragem) — Universidade Federal Do Pará - Núcleo De Desenvolvimento Amazônico Em Engenharia, Tucuruí, 2020.

MPROGRAMS; NÄVEKE, C. **QMC5883L compass arduino library**. 2022. Disponível em: <https://github.com/mprograms/QMC5883LCompass>. Acesso em: 18 maio 2023.

OLIVEIRA, M. M. de. **Desenvolvimento de uma embarcação de baixo custo controlada a distância para monitoramento ambiental em corpos d'água rasos**. 2021. Junho 2021. 56 f. Dissertação (Mestrado em Engenharia Hídrica) — Universidade Federal De Itajubá Instituto De Recursos Naturais, Itajubá, 2021.

RASHID, M. *et al.* Design and development of an autonomous surface vehicle for water quality monitoring. **Research Gate**, 2022.

RENAUDIN, V.; AFZAL, M. H.; LACHAPELLE, G. Complete triaxis magnetometer calibration in the magnetic domain. **Journal of Sensors**, Hindawi, v. 2010, 2010. Disponível em: <https://www.hindawi.com/journals/js/2010/967245/>. Acesso em: 15 abr. 2023.

RINALDI, A. **Ford atuará em entrega com veículos autônomos**. 2021. Disponível em: <https://revistacarro.com.br/ford-atuara-em-entrega-com-veiculos-autonomos/>.

SAÚDE, M. da. **Portaria de consolidação nº 5, de 28 de setembro de 2017**. 2017. Disponível em: [http://portalsinan.saude.gov.br/images/documentos/Legislacoes/Portaria\\_Consolidacao\\_5\\_28\\_SETEMBRO\\_2017.pdf](http://portalsinan.saude.gov.br/images/documentos/Legislacoes/Portaria_Consolidacao_5_28_SETEMBRO_2017.pdf).

SCIENCE, S. Z. M.; TECHNOLOGY. **Shark programmable brushless esc instruction manual**. 2020. Disponível em: <https://www.ztwoem.com/wp-content/uploads/2020/01/ZTW-Shark-Series-boat-ESC-User-Manual.pdf>. Acesso em: 18 maio 2023.

SHAWNHYMEL. **How to calibrate a magnetometer**. 2022. Disponível em: <https://www.digikay.com/en/maker/projects/how-to-calibrate-a-magnetometer/50f6bc8f36454a03b664dca30cf33a8b>. Acesso em: 12 abr. 2023.

SILVEIRA, E. da. **Barco autônomo**. 2017. Disponível em: <https://revistapesquisa.fapesp.br/barco-autonomo/>.

SOUZA, F. **Como programar o esp32 na arduino ide?** 2022. Disponível em: <https://embarcados.com.br/como-programar-o-esp32-na-arduino-ide/>. Acesso em: 16 maio 2023.

SURVEY, B. G. **Models and compass variation**. 2020. Disponível em: [https://geomag.bgs.ac.uk/data\\_service/models\\_compass/home.html](https://geomag.bgs.ac.uk/data_service/models_compass/home.html). Acesso em: 11 abr. 2023.

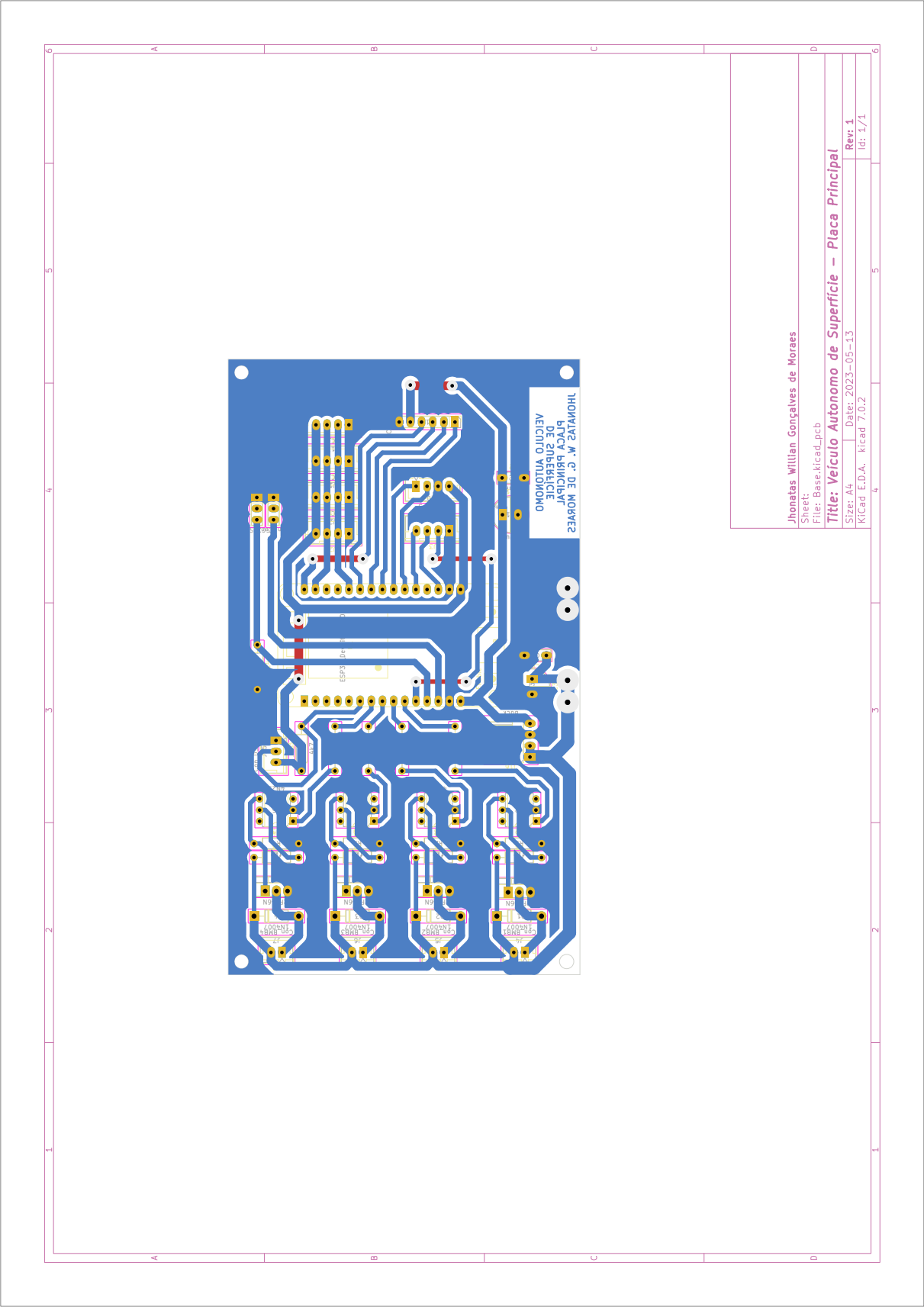
Sá, G. de. **Táxi autônomo será realidade nos eua ainda em 2021**. 2021. Disponível em: <https://revistacarro.com.br/taxi-autonomo-sera-realidade-nos-eua-ainda-em-2021/>.

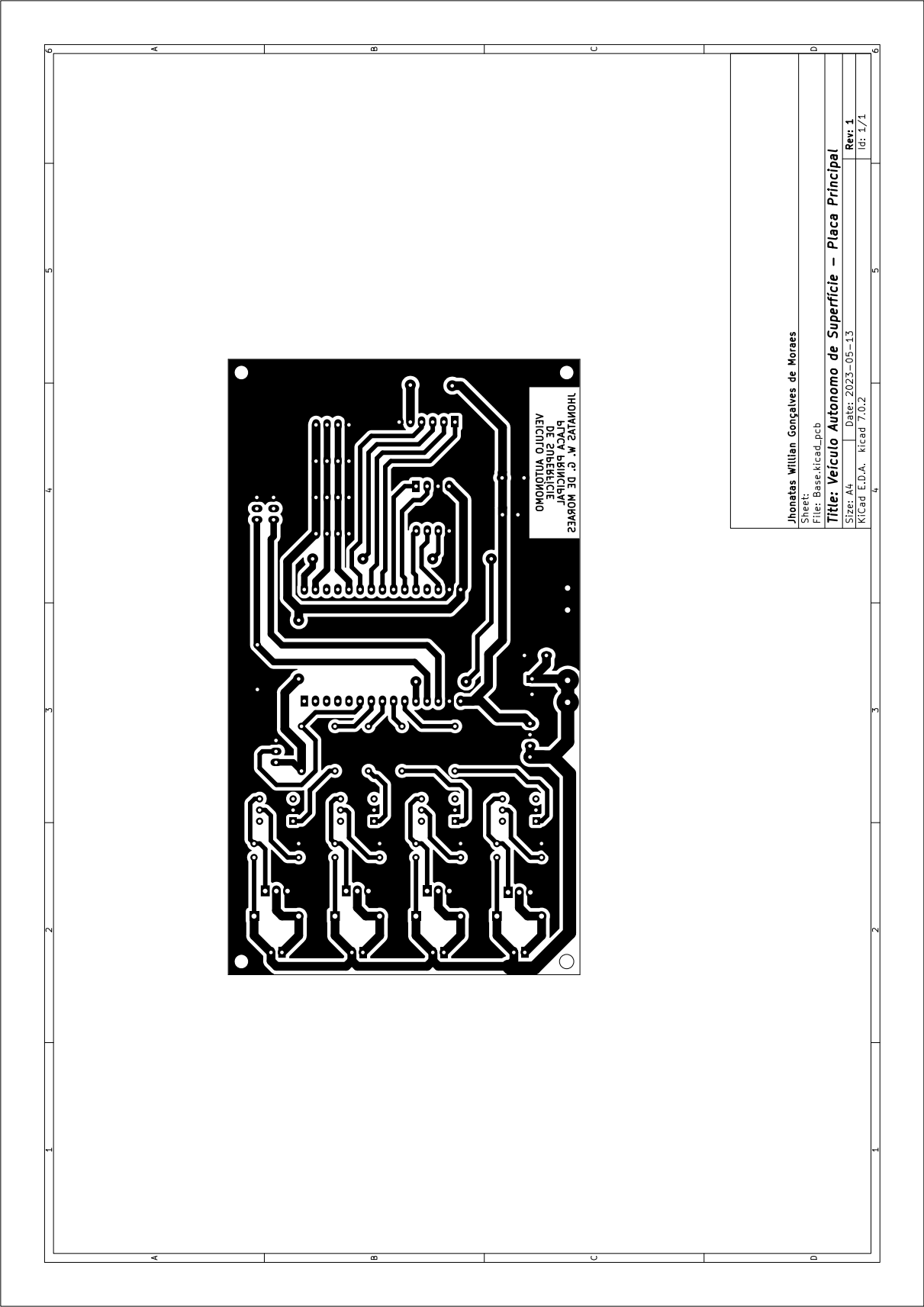
ZOGG, J.-M. **GPS essentials of satellite navigation**. 2009. Compendium. Disponível em: [https://content.u-blox.com/sites/default/files/gps\\_compendiumgps-x-02007.pdf](https://content.u-blox.com/sites/default/files/gps_compendiumgps-x-02007.pdf). Acesso em: 09 abr. 2023.

## **APÊNDICE A – Diagrama e Silk PCB da placa principal**

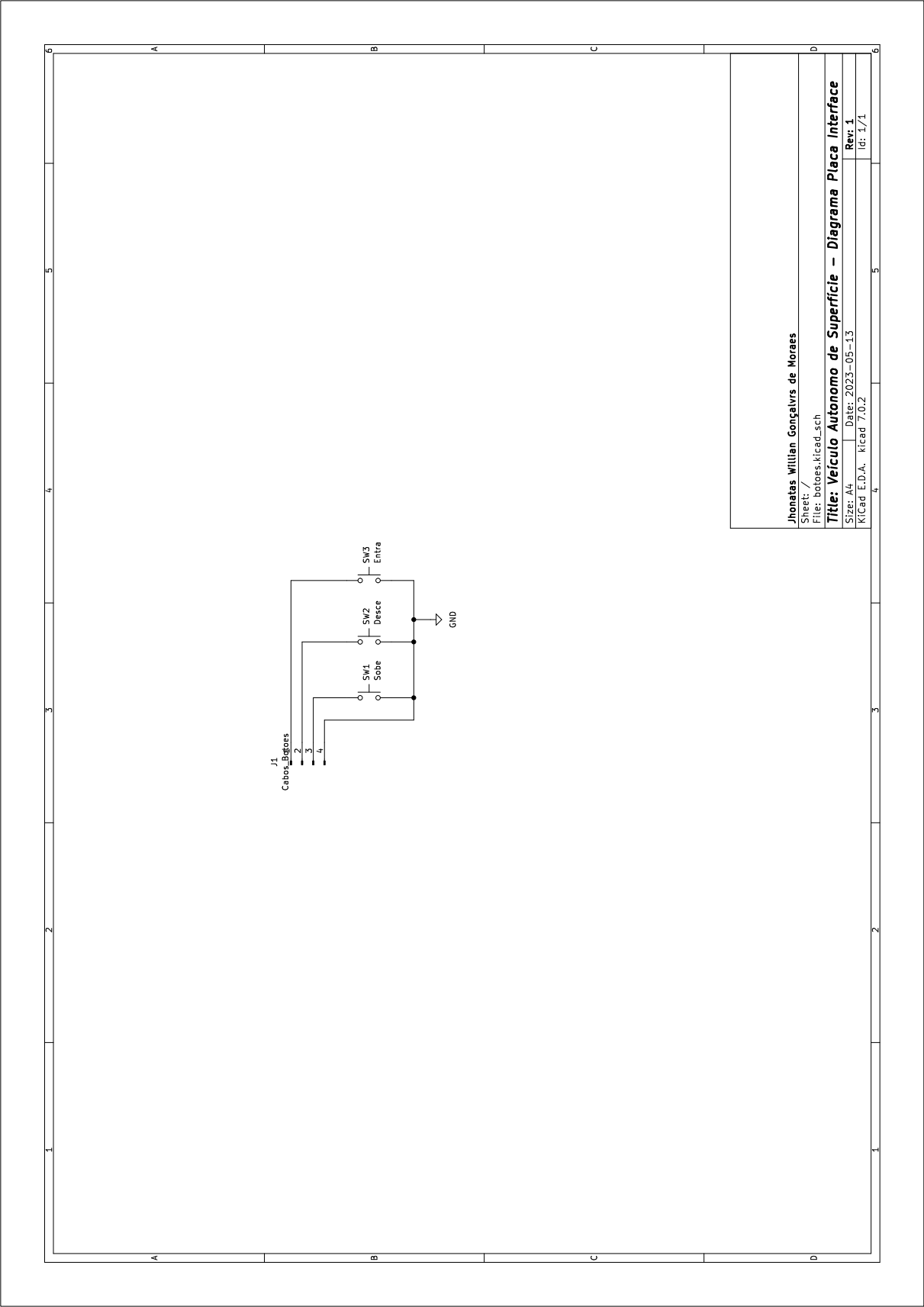


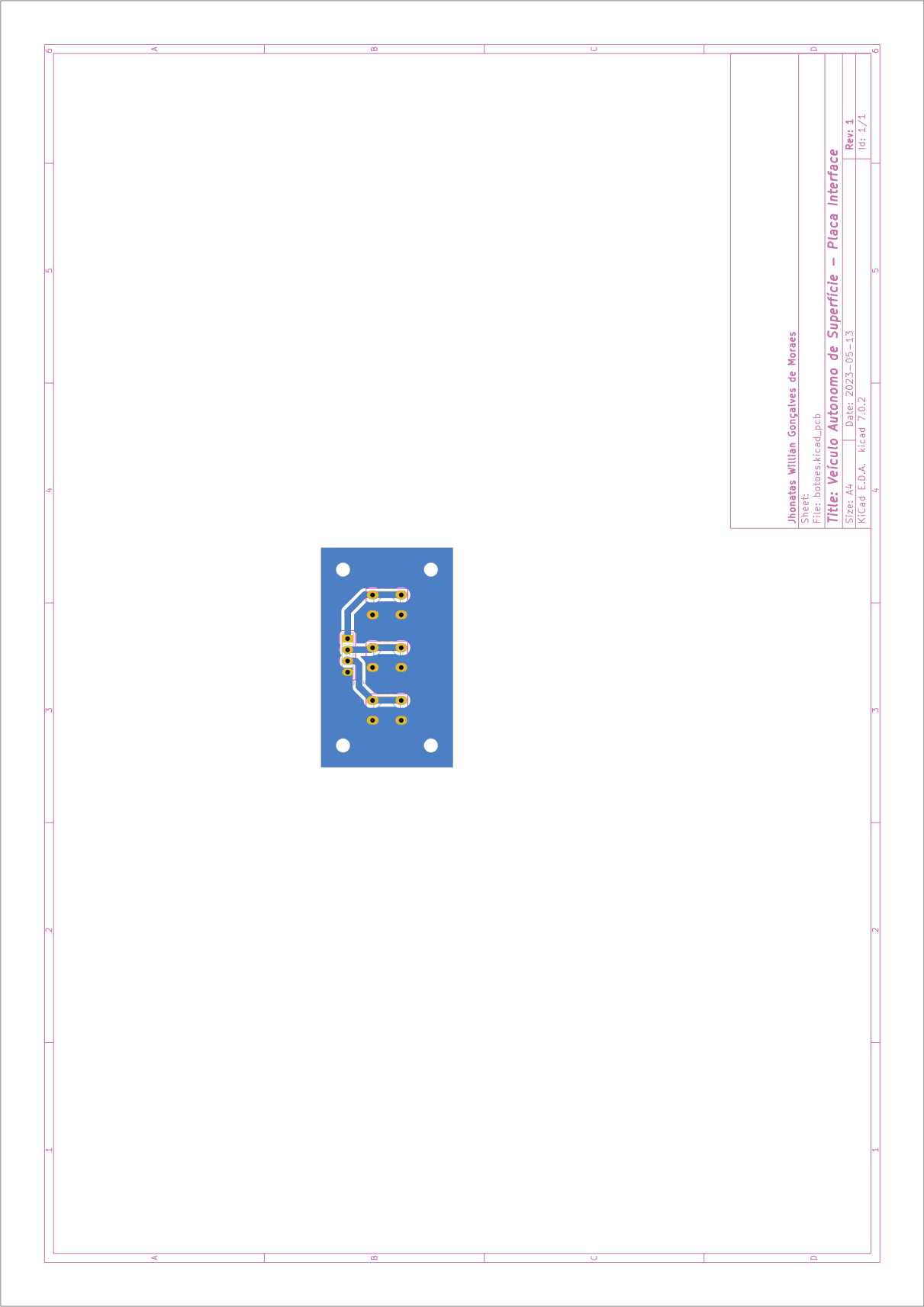


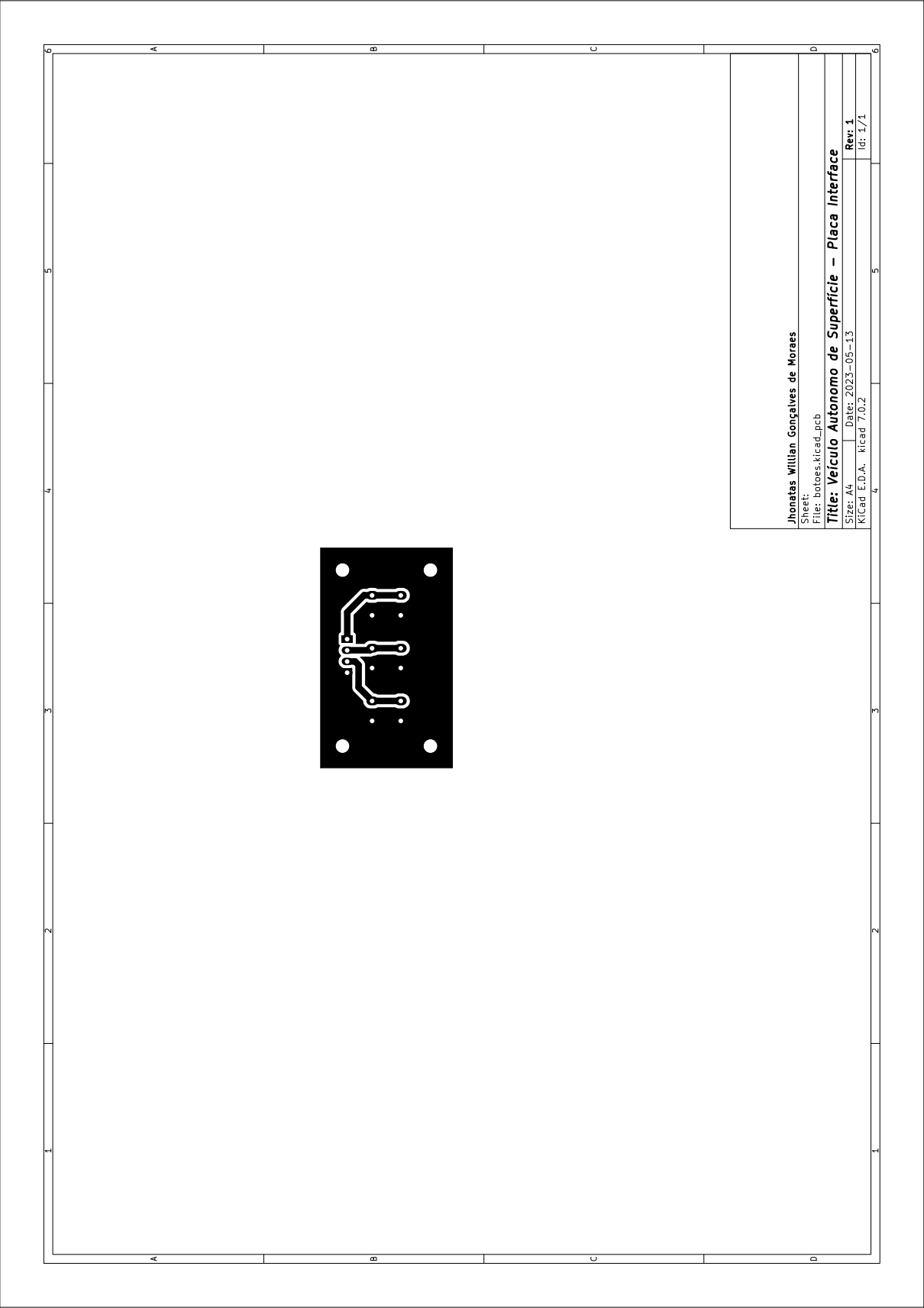




## **APÊNDICE B – Diagrama e Silk PCB da placa de interface**

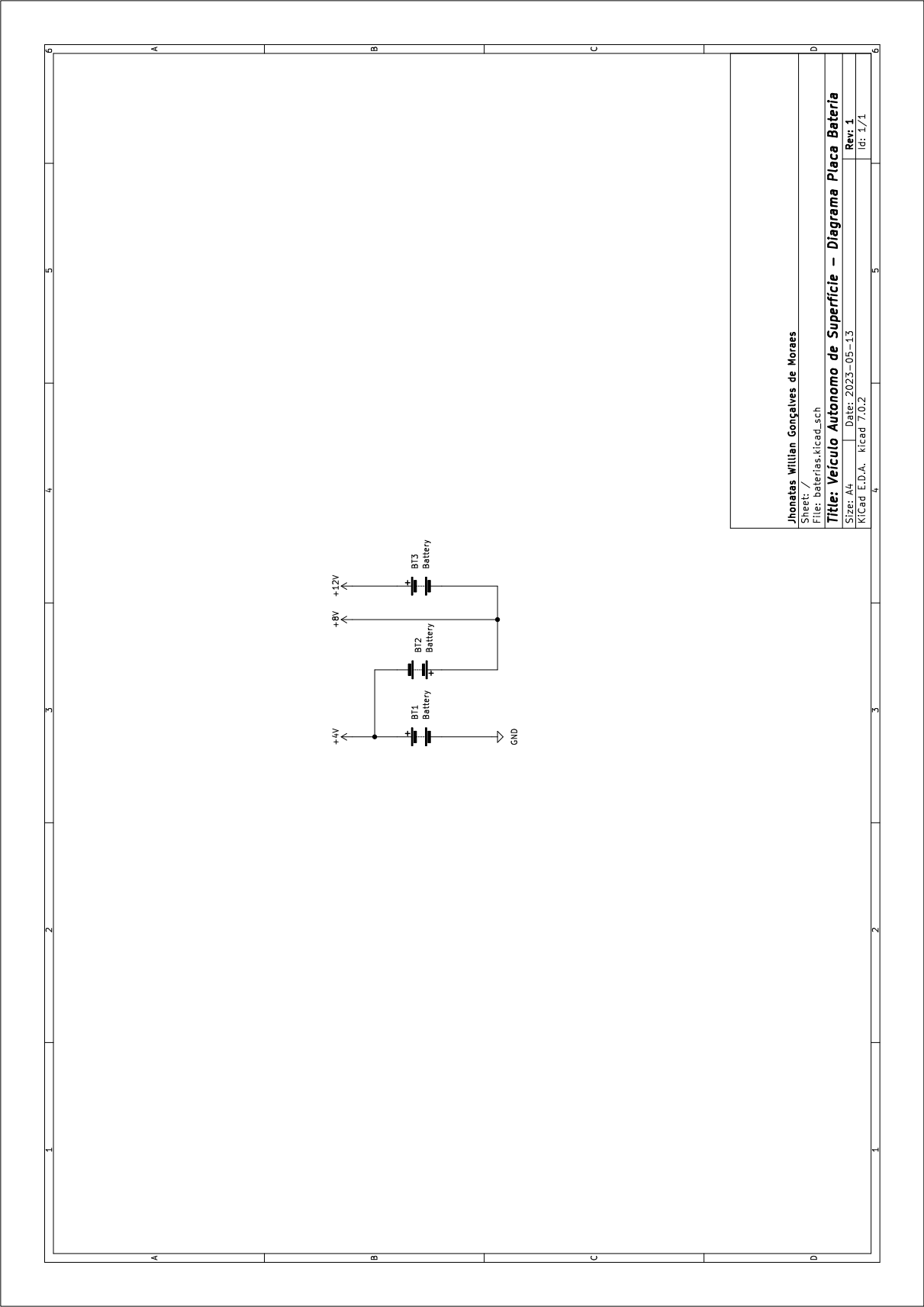


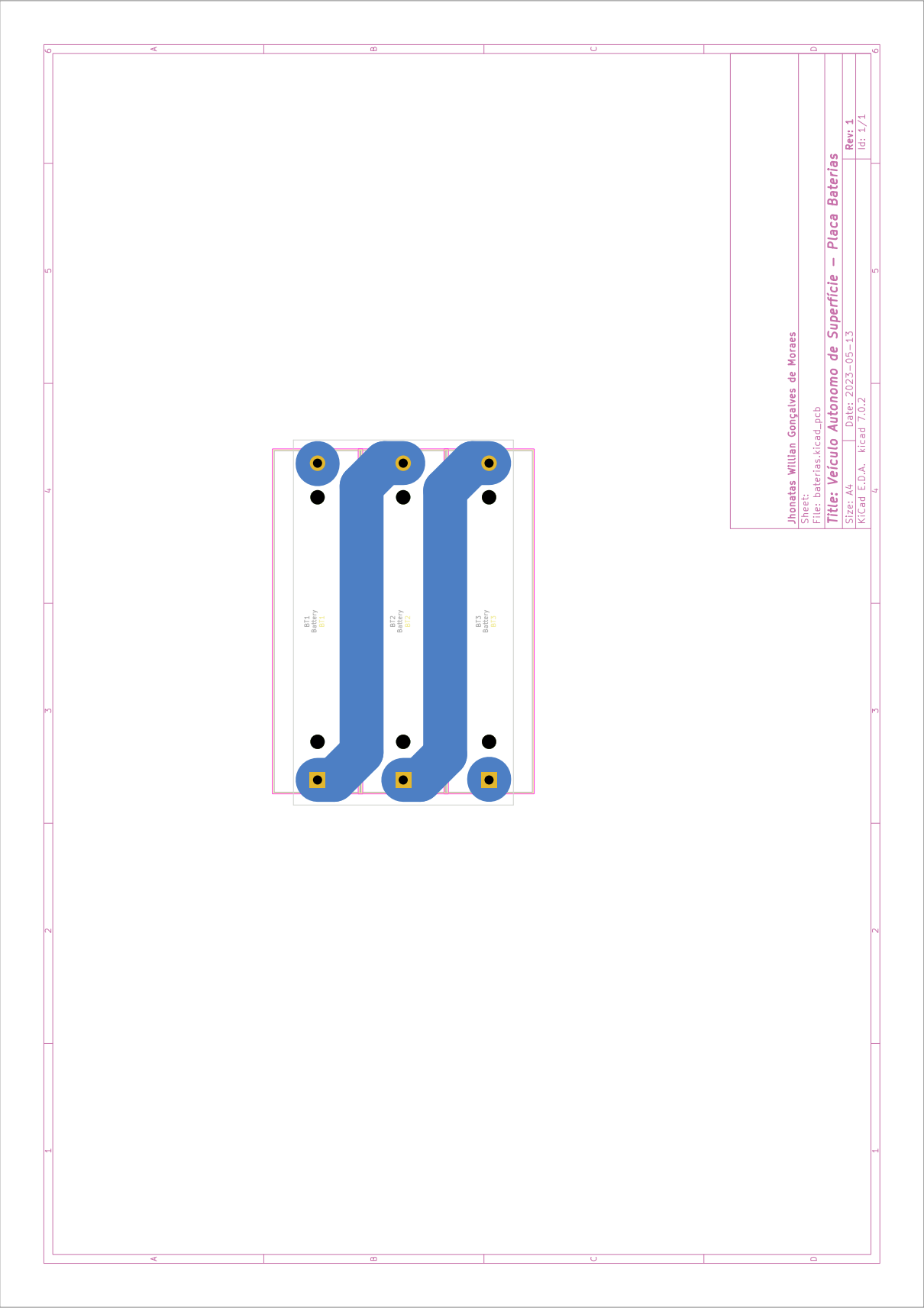


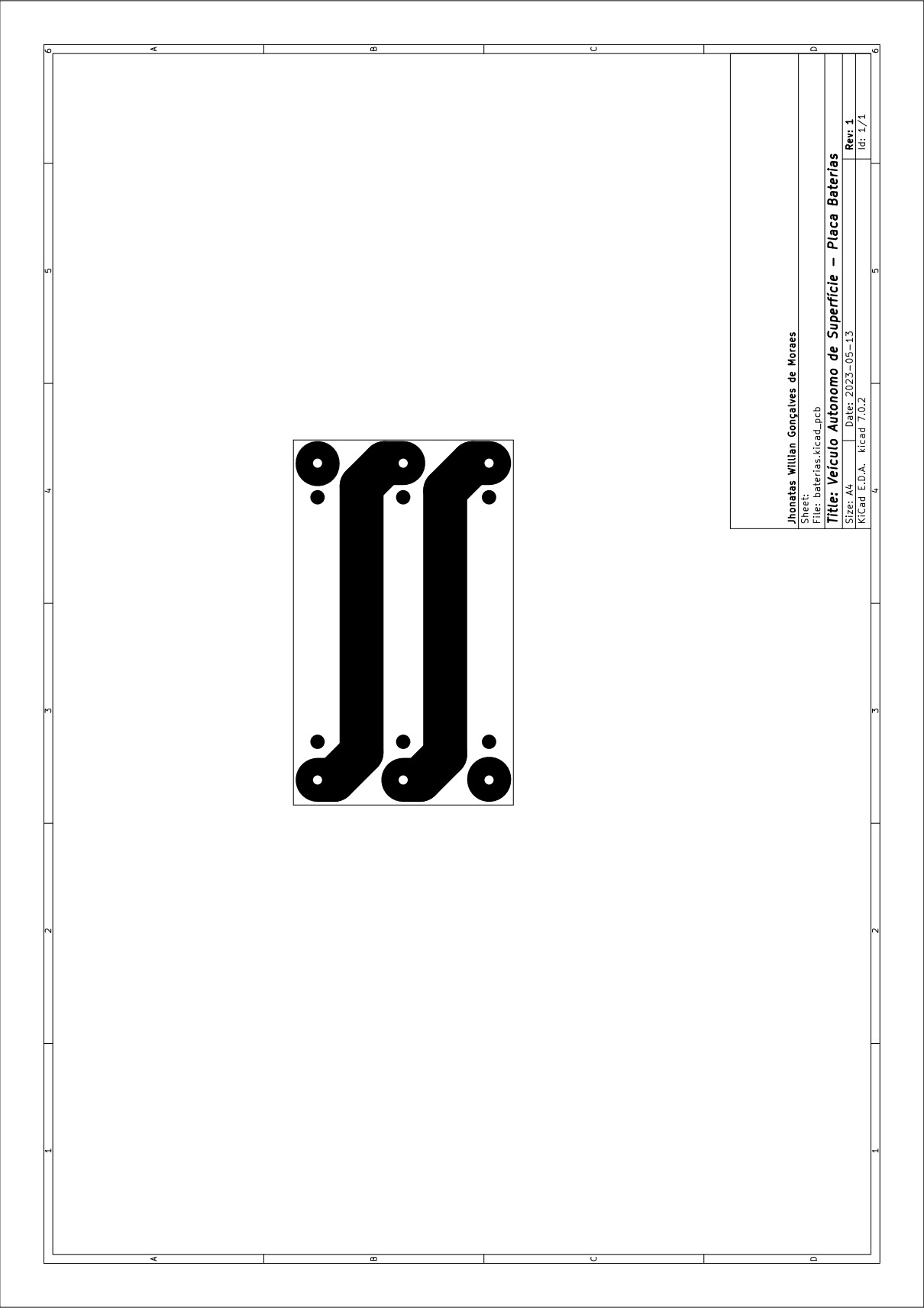


## **APÊNDICE C – Diagrama e Silk PCB da placa das baterias**









## **APÊNDICE D – Código escrito para o programa**

```

1  /*Codigo principal
2  Jhonatas Willian Gonçalves de Moraes
3  Trabalho de Conclusao de Curso
4  Universidade Tecnologica Federal do Parana
5  Campus Apucarana
6  Engenharia Eletrica*/
7  #include <Arduino.h>
8  #include <U8g2lib.h>
9  #include <Wire.h>
10 #include <TinyGPSPlus.h>
11 #include <QMC5883LCompass.h>
12 #include <telas_menu.h> //icones e telas
13 #include <math.h>
14 #include <EEPROM.h>
15 #include <ESP32Servo.h>
16 #include <SPI.h>
17 #include <SD.h>
18 #include <Adafruit_BME280.h>
19 #include <OneWire.h>
20 #include <DallasTemperature.h>
21
22 /*PINOS ESP32
23 Os pinos do ESP32 sao denominados da seguinte forma:
24
25 Pinos I2C
26   -> GPIO 21 --- SDA
27   -> GPIO 22 --- SCL
28 Pinos UART
29   -> GPIO 1 --- TXD0 (NC)
30   -> GPIO 3 --- RXD0 (NC)
31   -> GPIO 16 --- RXD2 (TX do GPS)
32   -> GPIO 17 --- TXD2 (RX do GPS)
33 Pinos SPI (Conectados ao leitor de cartao)
34   -> GPIO 5 --- SS
35   -> GPIO 18 --- SCK
36   -> GPIO 19 --- MISO
37   -> GPIO 23 --- MOSI
38 Pinos conectados aos botoes (Pull-up interno)
39   -> GPIO 4 --- BTN1
40   -> GPIO 14 --- BTN3 (Apresenta PWM durante boot)
41   -> GPIO 15 --- BTN2 (Apresenta PWM durante boot)
42 Pinos conectados as bombas
43   -> GPIO 25 --- BMB3
44   -> GPIO 26 --- BMB2
45   -> GPIO 27 --- BMB1
46   -> GPIO 33 --- BMB4
47 Pino conectado ao termometro DS18S20 (OneWire)
48   -> GPIO 32 --- DS18S20

```

```

49  Pinos conectados aos ESCs
50  -> GPIO 12 --- ESC1 ou ESCA (precisa estar com resistor de pull-down se conectado ao servo)
51  -> GPIO 13 --- ESC2 ou ESCB
52  Pinos nao conectados (abertos)
53  -> GPIO 2 --- NC (LED on board)
54  -> GPIX 34 --- NC
55  -> GPIX 35 --- NC
56  -> GPIX 36 --- NC (VP)
57  -> GPIX 39 --- NC (VN)
58  GPIOs nao disponiveis
59  -> GPIO 6 --- #N/D
60  -> GPIO 7 --- #N/D
61  -> GPIO 8 --- #N/D
62  -> GPIO 9 --- #N/D
63  -> GPIO 10 --- #N/D
64  -> GPIO 11 --- #N/D
65  -> GPIO 20 --- #N/D
66  -> GPIO 24 --- #N/D
67  -> GPIO 28 --- #N/D
68  -> GPIO 29 --- #N/D
69  -> GPIO 30 --- #N/D
70  -> GPIO 31 --- #N/D
71  -> GPIO 37 --- #N/D
72  */
73
74  //Inicializacao dos parametros das bibliotecas
75  // -> SD
76  File Arquivo;
77  const int CS = 5;
78  // -> Magnetometro QMC5883
79  QMC5883LCompass compass;
80  #define EEPROM_SIZE 16 //Define o tamanho da EEPROM para armazenamento dos parametros de calibracao do
    magnetometro (0-7), ajustes PID(8-10) e percentual Vmax (11)
81  // ->Display OLED
82  U8G2_SSD1306_128X64_NONAME_1_HW_I2C u8g2(U8G2_R2, /* reset=*/ U8X8_PIN_NONE); //U8G2_R2 faz o
    display girar 180graus
83  // -> Parametros de servo para os ESCs
84  Servo ESCA; //motor direito
85  Servo ESCB; //motor esquerdo
86  ESP32PWM pwm;
87  // -> GPS
88  TinyGPSPlus gps;
89  // ->Termohigrometro BME280
90  Adafruit_BME280 bme;
91
92  // ->Parametros para termometro DS18S20
93  const int oneWireBus = 32;
94  OneWire oneWire(oneWireBus);
95  DallasTemperature sensors(&oneWire);

```

```

96 //
97
98 const float declinacao = -19.85; //O ajuste deve ser negativo, assim o norte e apontado pela direcao X
99
100 //auxiliares
101 float alvo = 50;
102 bool StartScreen = true;
103
104
105 //Pinos GPIO
106 const int ESCA_Pin = 12;
107 const int ESCB_Pin = 13;
108 const int btn_1 = 15; //Botao anterior ou desce laranja btn2 se possivel, trocar no plug
109 const int btn_2 = 4; //Botao seguinte ou sobe vermelho btn1
110 const int btn_3 = 14; //Botao entra13 amarelo btn3
111 const int bomba1 = 33;
112 const int bomba2 = 25;
113 const int bomba3 = 26;
114 const int bomba4 = 27;
115
116 //Parametros Motores
117 int minUs = 1000; //tempo de pulso menor
118 int maxUs = 2000; //tempo de pulso maior
119 int ESCA_Vel = 90; //velocidade inicial
120 int ESCB_Vel = 90;
121 //O angulo do servo se da de 0 a 180, onde 90 e o motor parado.
122
123 //Parametros para magnetometro
124 int16_t max_x = -32768;
125 int16_t min_x = 32767;
126 int16_t max_y = -32768;
127 int16_t min_y = 32767;
128 int16_t x, y; //, z;
129
130 const int NUM_ITENS = 13; //Numero de itens do menu
131 char menu_items[NUM_ITENS][20] = { //Itens disponiveis
132     {"Cal. QMC5883"},
133     {"GPS"},
134     {"Direcao"},
135     {"Iniciar"},
136     {"Ler Dados"},
137     {"Turbina 1"},
138     {"Turbina 2"},
139     {"Teste"},
140     {"Info"},
141     {"Ajuste Vel."},
142     {"Ajuste P"},
143     {"Ajuste I"},
144     {"Ajuste D"}

```

```

145 };
146
147 int item_sel = 1; //Qual item do menu esta selecionado
148 int item_ant; //Item anterior ao selecionado
149 int item_pos; //Item posterior ao selecionado
150
151 int btn_1_clic=0;
152 int btn_2_clic=0;
153 int btn_3_clic=0;
154
155 int TelaAtual=0;
156 float direc = 0;
157 float Latitude = 0;
158 float Longitude = 0;
159
160 float dLat = 0;
161 float dLon = 0;
162 float lat1_rad = 0;
163 float lat2_rad = 0;
164 float yPoint = 0;
165 float xPoint = 0;
166 float bearing = 0;
167 float distancia = 0;
168 float raioTerra = 6371000;
169
170 int coordenada = 0;
171 const int QtdPontos = 20; //Capacidade determinada de ate 20 pontos
172 float LatAlvo[QtdPontos+1]; //Prepara as arrays para receberem ate 20 pontos +1 (origem)
173 float LonAlvo[QtdPontos+1];
174 bool PontoLeitura[QtdPontos]; //Separa os pontos em que deve ser feita a leitura dos valores
175 bool PontoColeta[QtdPontos]; //Separa os pontos em que devem ser coletados a agua
176 int PontosLidos = 0;
177
178 //Variaveis PID
179
180 float kp = 1; // Coeficiente proporcional
181 float ki = 0.005; // Coeficiente integral
182 float kd = 1; // Coeficiente derivativo
183 float saida = 0.0; // Saida do controlador
184 float erro = 0.0; // Erro atual
185 float ultimo_erro = 0.0; // ultimo erro
186 float proporcional = 0.0;
187 float integral = 0.0; // Soma do erro
188 float derivativo = 0.0; // Variacao do erro
189 unsigned long last_time = 0; // ultimo tempo de atualizacao
190 unsigned long dt = 50; // Intervalo de tempo em milissegundos
191 //unsigned long dt2 = 1; // Intervalo de tempo em milissegundos
192
193 //Variaveis BME

```



```

194 float bme_temperatura = 0;
195 float bme_umidade = 0;
196 float bme_pressao = 0;
197
198 //Variaveis bombas
199 bool bomba1_livre = true;
200 bool bomba2_livre = true;
201 bool bomba3_livre = true;
202 bool bomba4_livre = true;
203 int tempo_enchimento = 14000; //Tempo que a bomba fica ligada (ms)
204
205 //Variaveis de deslocamento
206 float velocidade_atual = 0; //−100 a 100, usar ate 50
207
208 //bearing = angulo que deve ser apontado
209 //direc = angulo atual
210
211 float erro2 = 0;
212
213 int8_t percentualP = 100;
214 int8_t percentualI = 100;
215 int8_t percentualD = 100;
216 int8_t percentualV = 50;
217 int8_t Vmax = 80;
218
219 bool percursoConcluido = false;
220 bool gpsvalid = false;
221 int satconectados = 0;
222 float precisao = 3; //metros
223 int p = 0; //auxiliar de contagem
224
225 uint32_t horaagora = 0; //Hora que e apontada pelo GPS
226 uint32_t datahoje = 0; //Data do GPS
227 String TextoLog; //Texto para gravacao do log
228 bool aux; //variavel auxiliar
229 unsigned long last_time_disp = 0; //variavel auxiliar para gravacao no display
230 unsigned long last_time_cartao = 0; //variavel auxiliar para gravacao no SD
231
232
233 //-----LOG de dados no SO
234
234 void LogSD(){
235     if(millis() - last_time_cartao > 500){
236         TextoLog = String(horaagora) + "; " + String(Latitude, 7) + "; " + String(Longitude, 7) + "; " + String(LatAlvo[p], 7) +
237             "; " + String(LonAlvo[p], 7) + "; " + String(direc) + "; " + String(bearing) + "; " + String(erro) + "; " + String(
238                 ESCA_Vel) + "; " + String(ESCB_Vel) + ";";
239         Arquivo = SD.open("/log.txt", FILE_APPEND,true);
240         if(Arquivo) {
241             Arquivo.println(TextoLog);

```

```

240     Arquivo.close();
241 }
242 last_time_cartao = millis();
243 }
244 }
245 //-----Le coordenadas
      -----
246 //Le ate 20 coordenadas disponibilizadas no arquivo "coordenadas.txt" dentro do cartao SD e armazena as instrucoes nas
      variaveis LatAlvo, LonAlvo, PontoLeitura e PontoColeta
247 bool LerCoordenadas(){
248
249     Arquivo = SD.open("/coordenadas.txt", FILE_READ);
250     if (Arquivo) {
251         while (Arquivo.available()) {
252             PontosLidos = 0;
253             for (int i=0; i<QtdPontos; i++){
254                 String coordenadas = Arquivo.readStringUntil(';');
255                 sscanf(coordenadas.c_str(), "%f, %f, %d, %d", &LatAlvo[i], &LonAlvo[i], &PontoLeitura[i], &PontoColeta[i]); //
                    Armazena em cada um dos elementos do array a coordenada alvo desejada
256                 if(LatAlvo[i] !=0 && LatAlvo[i]!=0){PontosLidos += 1;}
257                 Serial.print("Parametros[");
258                 Serial.print(i);
259                 Serial.print("]: ");
260                 Serial.print(LatAlvo[i],10);
261                 Serial.print(" ");
262                 Serial.print(LonAlvo[i],10);
263                 Serial.print(" ");
264                 Serial.print(PontoLeitura[i]);
265                 Serial.print(" ");
266                 Serial.println(PontoColeta[i]);
267             }
268         }
269         Arquivo.close();
270         return true;
271     }else{
272         Serial.println("Erro ao abrir o arquivo de coordenadas.");
273         PontosLidos = 0;
274         return false;
275     }
276 }
277 }
278 //-----Calcula o angulo que deve ser apontado e a distancia do alvo
      -----OK
279 void CalcDirecDist(float lat1, float lon1, float lat2, float lon2){ //Recebe coordenadas em graus decimais
280     dLat = (lat2 - lat1) * PI / 180; // Diferenca de latitude em radianos
281     dLon = (lon2 - lon1) * PI / 180; // Diferenca de longitude em radianos
282     lat1_rad = lat1 * PI / 180; // Latitude da primeira coordenada em radianos
283     lat2_rad = lat2 * PI / 180; // Latitude da segunda coordenada em radianos
284     yPoint = sin(dLon) * cos(lat2_rad);

```

```

285 xPoint = cos(lat1_rad) * sin(lat2_rad) - sin(lat1_rad) * cos(lat2_rad) * cos(dLon);
286 bearing = atan2(yPoint, xPoint) * 180 / PI - declinacao; // Calculo do angulo em graus
287 if(bearing < 0){bearing += 360;}
288 distancia = 2 * raioTerra * asin(sqrt(pow(sin(dLat/2),2) + cos(lat1_rad)*cos(lat2_rad)*pow(sin(dLon/2),2)));
289
290 }
291 //-----Algoritmo PID e controle dos motores
292
293 void AtuaESCs(float ajuste){ //Envia velocidade e ajuste de direcao aos motores (velocidade_atual)
294 //ajuste vai de 0 a 255
295 // -100 e gira para UM LADO =, 100 e gira PARA O OUTRO
296 // 0 nao faz correcao
297
298 ESCA_Vel = int(map(velocidade_atual,-100,100,0,180) + map(ajuste,-100,100,-90,90));
299 ESCB_Vel = int(map(velocidade_atual,-100,100,0,180) - map(ajuste,-100,100,-90,90));
300
301 if(ESCA_Vel > 180){ESCA_Vel = 180;}else if(ESCA_Vel < 0){ESCA_Vel = 0;}
302 if(ESCB_Vel > 180){ESCB_Vel = 180;}else if(ESCB_Vel < 0){ESCB_Vel = 0;}
303
304 //ESCA_Vel = map(ESCA_Vel,0,180,180,0);
305 //ESCB_Vel = map(ESCB_Vel,0,180,180,0); //inverte a rotacao dos motores
306 ESCA.write(180-ESCA_Vel);
307 ESCB.write(180-ESCB_Vel);
308 }
309
310 void CalculaPIDDirec(){ //Algoritmo generico de controle PID com tempo de intervalo de correcao "dt"
311 //O objetivo do erro e ser 0, entao os valores precisam ir de -255 a 255
312
313 if((millis() - last_time) > dt){
314 erro = bearing - direc; //Compara a direcao apontada com a direcao desejada
315 if(erro > 180.00){
316 erro -= 360.0;
317 }else if(erro < -180.0){
318 erro += 360.0;
319 }
320 erro = erro/1.80; //mapeia de 0 a 180 para 0 a 100
321 proporcional = (float(percentualP)/100) * kp * erro;
322 integral += (float(percentualI)/100) * ki * erro * float(dt);
323 derivativo = (float(percentualD)/100) * kd * (erro - ultimo_erro) / float(dt);
324 saida = proporcional + integral + derivativo;
325 if(saida > 100){saida = 100;}else if(saida < -100){saida = -100;} // Limita a saida do PID de -100 a 100 (%)
326 ultimo_erro = erro;
327 last_time = millis();
328 AtuaESCs(saida);
329 }
330
331 void parar_motores(){ //Funcao auxiliar para parar os motores
332 ESCA.write(90);

```

```

333   ESCB.write(90);
334   velocidade_atual = 0;
335 }
336
337 //-----Le GPS
      -----
338 void leGPS(){
339   while (Serial2.available()){gps.encode(Serial2.read());} //Le os dados da porta serial
340   if(gps.location.isValid()){
341     Latitude = gps.location.lat(); // float ->aponta os dados atuais de latitude e longitude
342     Longitude = gps.location.lng(); // float
343     gpsvalid = true;
344     satconectados = gps.satellites.value();
345     horaagora = gps.time.value();
346     datahoje = gps.date.value();
347   }else{gpsvalid = false;}
348 }
349 //-----Calibra magnetometro
      -----
350 void calibrar_qmc5883(){
351   compass.setCalibration(-32768, 32767, -32768, 32767, -32768, 32767);
352   int tempo_final = millis() + 5000; //marca o instante de final da calibracao
353   max_x = -32768;
354   min_x = 32767;
355   max_y = -32768;
356   min_y = 32767;
357   //int max_z = -32768; //Nao e calibrado z, pois nao e usado
358   //int min_z = 32767;
359   bool novo_v = false;
360   bool pronto = false;
361   x = y = 0; //, z;
362
363   while(!pronto){
364     compass.read();
365     x = compass.getX();
366     y = compass.getY();
367     //z = compass.getZ();
368
369     novo_v = false;
370     if (x > max_x){max_x = x; novo_v = true;}
371     if (x < min_x){min_x = x; novo_v = true;}
372     if (y > max_y){max_y = y; novo_v = true;}
373     if (y < min_y){min_y = y; novo_v = true;}
374
375     Arquivo = SD.open("/dados_calib.txt", FILE_APPEND,true);
376     //Se abriu, acrescenta no arquivo
377     if(Arquivo) {
378       //Latitude; Longitude; LatAlvo[p]; lonAlvo[p]; direc; bearing; erro; ESCA_Vel; ESCB_Vel;
379       Arquivo.print(x);

```

```

380     Arquivo.print(" ");
381     Arquivo.print(y);
382     Arquivo.println(" ");
383     Arquivo.close();
384 }
385 //if (z > max_z) max_z = z; novo_v = true;
386 //if (z < min_z) min_z = z; novo_v = true;
387 if (novo_v == true && pronto == false){
388     tempo_final = millis() + 5000;
389     novo_v = false;
390 }
391
392 if (millis() > tempo_final){
393     pronto = true;
394     compass.setCalibration(min_x, max_x, min_y, max_y, -32768, 32767);
395     EEPROM.put(0, min_x >> 8); //Armazena valores de calibracao no EEPROM
396     EEPROM.put(1, min_x & 0x00FF);
397     EEPROM.put(2, max_x >> 8);
398     EEPROM.put(3, max_x & 0x00FF);
399     EEPROM.put(4, min_y >> 8);
400     EEPROM.put(5, min_y & 0x00FF);
401     EEPROM.put(6, max_y >> 8);
402     EEPROM.put(7, max_y & 0x00FF);
403     EEPROM.commit();
404
405     Arquivo = SD.open("/dados_calib.txt", FILE_APPEND,true);
406     if(Arquivo) {
407         Arquivo.println("Pronto!");
408         Arquivo.close();
409     }
410 }
411 }
412 }
413 //-----Calcula graus
414
414 void CalculaGraus(){ //Calcula a quantos graus esta do norte geografico (direcao que esta apontando)
415     compass.read();
416     x = compass.getX();
417     y = compass.getY();
418     direc = atan2(y,x) * 180 / 3.141592653;
419     direc = direc - declinacao;
420     if (direc < 0) {
421         direc += 360;
422     }
423 }
424 //-----Fazer atividade no ponto
425
425 void AtividadePonto(int ponto){
426

```

```

427 String texto_bomba;
428 String texto_bme;
429 String texto_DS18S20;
430 String texto_GPS;
431
432 if (PontoLeitura[ponto]){
433     bme_temperatura = bme.readTemperature();
434     bme_umidade = bme.readHumidity();
435     bme_pressao = bme.readPressure() / 100.0F;
436     sensors.requestTemperatures();
437     texto_bme = "Temperatura = " + String(bme_temperatura) + " *C, " + "Umidade = " + String(bme_umidade) + " %,
        " + "Pressao = " + String(bme_pressao) + " hPa";
438     texto_DS18S20 = "Temperatura da agua = " + String(sensors.getTempCByIndex(0)) + " *C";
439
440 }else{
441     bme_temperatura = 0;
442     bme_umidade = 0;
443     bme_pressao = 0;
444     texto_bme = "Dados nao lidos";
445     texto_DS18S20 = "Dados nao lidos";
446 }
447
448 if(PontoColeta[ponto]){
449     if(bomba1_livre){//Bomba 1 livre
450         digitalWrite(bomba1, HIGH);
451         delay(tempo_enchimento);
452         digitalWrite(bomba1, LOW);
453         bomba1_livre = false;
454         texto_bomba = "Reservatorio 1 enchido";
455     }else if(bomba2_livre){//Bomba 2 livre
456         digitalWrite(bomba2, HIGH);
457         delay(tempo_enchimento-500);//desconta 500ms pois a bomba 2 e um pouco mais forte
458         digitalWrite(bomba2, LOW);
459         bomba2_livre = false;
460         texto_bomba = "Reservatorio 2 enchido";
461     }else if(bomba3_livre){//Bomba 3 livre
462         digitalWrite(bomba3, HIGH);
463         delay(tempo_enchimento);
464         digitalWrite(bomba3, LOW);
465         bomba3_livre = false;
466         texto_bomba = "Reservatorio 3 enchido";
467     }else if(bomba4_livre){//Bomba 4 livre
468         digitalWrite(bomba4, HIGH);
469         delay(tempo_enchimento);
470         digitalWrite(bomba4, LOW);
471         bomba4_livre = false;
472         texto_bomba = "Reservatorio 4 enchido";
473     }else{
474         texto_bomba = "Todos os reservatorios cheios, nenhum enchido";

```

```

475 }}else{
476     texto_bomba = "Amostra nao coletada";
477 }
478 leGPS();
479 texto_GPS = "Latitude, Longitude: " + String(Latitude,7) + ", " + String(Longitude,7);
480 Arquivo = SD.open("/resultados.txt", FILE_APPEND,true);
481 //Se abriu, acrescenta no arquivo
482 if(Arquivo) {
483     Arquivo.println("Ponto de controle");
484     Arquivo.print(datahoje);
485     Arquivo.print(" ");
486     Arquivo.print(horaagora);
487     Arquivo.println(" ");
488     Arquivo.println(texto_GPS);
489     Arquivo.println(texto_bme);
490     Arquivo.println(texto_DS18S20);
491     Arquivo.println(texto_bomba);
492     Arquivo.println("
-----"
);
493     Arquivo.close();
494
495
496     u8g2.firstPage();
497     do{
498         u8g2.setFont(u8g2_font_7x14B_tf);
499         u8g2.drawStr(1,15,"Dados armazenados");
500         u8g2.drawStr(29,30," no cartao");
501
502     }while(u8g2.nextPage());
503     delay(2000);
504
505 }else{
506
507     u8g2.firstPage();
508     do{
509         u8g2.setFont(u8g2_font_7x14B_tf);
510         u8g2.drawStr(4,15,"Erro ao gravar dados");
511         u8g2.drawStr(4,30,"Verifique o cartao");
512
513     }while(u8g2.nextPage());
514     delay(2000);
515 }
516 }
517 //-----Funcao Navegar
-----
518 void Navegar(){
519
520 leGPS();

```

```

521 if(gpsvalid){
522
523   LatAlvo[PontosLidos] = Latitude; //Armazena o ponto atual como ultimo ponto (verificar depois de deixar como opcional)
524   LonAlvo[PontosLidos] = Longitude;
525   velocidade_atual = 100 * percentualV/100;
526
527   for (p = 0; p <= PontosLidos;p++){
528     velocidade_atual = 100 * percentualV/100;
529     do{
530
531       leGPS();
532       CalculaGraus();
533       CalcDirecDist(Latitude,Longitude, LatAlvo[p], LonAlvo[p]); //retorna distancia e angulo do alvo
534       CalculaPIDDirec();
535       LogSD();
536       if(digitalRead(btn_3) == LOW){
537         parar_motores();
538         TelaAtual=0;
539         percursoConcluido = true;
540         distancia = precisao-1;
541         p = PontosLidos+1;
542       }
543       if(millis() - last_time_disp > 300){//roda a cada 300ms
544         u8g2.firstPage();
545         do{//mostra status no display
546           u8g2.setFont(u8g2_font_7x14B_tf);
547           u8g2.drawStr(16,15,"Atual | Alvo");
548           u8g2.setCursor(110, 15);
549           u8g2.print(p);//Indica qual ponto esta como alvo
550           u8g2.setFont(u8g2_font_squeezed_r6_tr);
551           u8g2.setCursor(1, 23);
552           u8g2.print(Latitude,8); //Coordenada atual Lat
553           u8g2.setCursor(60, 23);
554           u8g2.print("|");
555           u8g2.setCursor(65, 23);
556           u8g2.print(LatAlvo[p],8); //coordenada alvo Lat
557           u8g2.setCursor(1, 31);
558           u8g2.print(Longitude,8); //Coordenada atual Lon
559           u8g2.setCursor(60, 31);
560           u8g2.print("|");
561           u8g2.setCursor(65, 31);
562           u8g2.print(LonAlvo[p],8); //coordenada alvo Lon
563           u8g2.setCursor(1, 38);
564           u8g2.print(direc,3); //mostra direcao apontada
565           u8g2.setCursor(60, 38);
566           u8g2.print("|");
567           u8g2.setCursor(65, 38);
568           u8g2.print(bearing,3); //mostra direcao que deve apontar
569           u8g2.setFont(u8g2_font_7x14_tf);

```



```

570     u8g2.setCursor(1, 53);
571     u8g2.print(distancia,1); //mostra distancia em metros
572     u8g2.setCursor(86, 53);
573     u8g2.print("metros");
574     u8g2.setFont(u8g2_font_squeezed_r6_tr);
575     u8g2.setCursor(10, 64);
576     u8g2.print(ESCB_Vel);
577     u8g2.setCursor(86, 64);
578     u8g2.print(ESCA_Vel);
579   }while(u8g2.nextPage());
580   last_time_disp = millis();
581   }
582 }while(distancia>precisao); //Fazer enquanto a distancia do ponto for maior que 10m
583 parar_motores();
584 AtividadePonto(p);
585 if(p >= PontosLidos){
586   percursoConcluido = true; //Sinaliza que a navegacao terminou
587 }
588 }
589
590 }else{
591   u8g2.firstPage();
592   do{
593     u8g2.setFont(u8g2_font_7x14B_tf);
594     u8g2.drawStr(1,15,"GPS invalido");
595     u8g2.drawStr(1,30,"Sats conectados");
596     u8g2.setCursor(110,30);
597     u8g2.print(satconectados);
598   }while(u8g2.nextPage());
599 }
600 }
601
602 //-----Mostra Tela
603
604 void MostraTela(){
605   switch (TelaAtual){
606     case 0: //Tela inicial (menu) OK
607       u8g2.setBitmapMode(1); //faz os bitmaps transparentes
608       u8g2.firstPage();
609       do{
610         //Item Anterior
611         u8g2.setFont(u8g2_font_7x14_tf);
612         u8g2.drawStr(26,15,menu_items[item_ant]);
613         u8g2.drawBitmap(4,2,16/8,16, Bitmap_Icons[item_ant]); //Desenha Icone (0)
614         //Item Selecionado
615         u8g2.setFont(u8g2_font_7x14B_tf);
616         u8g2.drawStr(26,37,menu_items[item_sel]);
617         u8g2.drawBitmap(4,24,16/8,16, Bitmap_Icons[item_sel]); //Desenha Icone (1)
618         //Item Posterior

```

```

618     u8g2.setFont(u8g2_font_7x14_tf);
619     u8g2.drawStr(26,59,menu_items[item_pos]);
620     u8g2.drawBitmap(4,46,16/8,16, Bitmap_Icons[item_pos]); //Desenha Icone (2)
621     u8g2.drawBitmap(0,22,128/8,21, epd_bitmap_BordaSelec); //Desenha selecionado
622     //u8g2.drawBitmap(120,0,8/8,64, epd_bitmap_BarraRolagem); //Desenha Barra de rolagem (Nao implementado)
623 }while(u8g2.nextPage());
624
625 break;
626
627 case 1: //Calibra Magnetometro
628     u8g2.firstPage();
629     do{
630         u8g2.setFont(u8g2_font_7x14_tf);
631         u8g2.drawStr(4,15,"A calibrar em");
632         u8g2.drawStr(4,37,"5 segundos");
633     }while(u8g2.nextPage());
634     delay(5000);
635     u8g2.firstPage();
636     do{
637         u8g2.drawStr(4,15,"Calibrando");
638     }while(u8g2.nextPage());
639     calibrar_qmc5883();//Chama a funcao para calibrar
640     u8g2.firstPage();
641     do{
642         u8g2.drawStr(4,15,"Calibrando");
643         u8g2.drawStr(4,37,"Pronto!");
644     }while(u8g2.nextPage());
645     delay(2000);
646     TelaAtual = 0; //volta para o menu principal
647     break;
648
649 case 2: //GPS
650     leGPS();
651     CalcDirecDist(Latitude,Longitude, LatAlvo[coordenada], LonAlvo[coordenada]);
652     u8g2.firstPage();
653     do{
654         u8g2.setFont(u8g2_font_squeezed_r6_tr);
655         u8g2.drawStr(1,8,"Latitude / Longitude Atual");
656         u8g2.setCursor(0, 16);
657         u8g2.print(Latitude,6);
658         u8g2.setCursor(52, 16);
659         u8g2.print("/");
660         u8g2.setCursor(60, 16);
661         u8g2.print(Longitude,6);
662         u8g2.drawStr(1,24,"Latitude / Longitude Alvo");
663         u8g2.setCursor(0,32);
664         u8g2.print(LatAlvo[coordenada],6);
665         u8g2.setCursor(52, 32);
666         u8g2.print("/");

```



```

716     }while(u8g2.nextPage());
717 }}
718     break;
719
720     case 5: //Tela ler coordenadas do cartao SD
721         u8g2.firstPage();
722         do{
723             u8g2.setFont(u8g2_font_7x14B_tf);
724             u8g2.drawStr(4,15,"Lendo...");
725         }while(u8g2.nextPage());
726
727         delay(500);
728
729         if(LerCoordenadas()){
730             u8g2.firstPage();
731             do{
732                 u8g2.setFont(u8g2_font_7x14B_tf);
733                 u8g2.drawStr(4,15,"Coordenadas lidas");
734                 u8g2.setCursor(4, 37);
735                 u8g2.print(PontosLidos);
736                 u8g2.print(" pontos");
737             }while(u8g2.nextPage());
738         }else{
739             u8g2.firstPage();
740             do{
741                 u8g2.setFont(u8g2_font_7x14B_tf);
742                 u8g2.drawStr(1,15,"Falha na leitura");
743                 u8g2.drawStr(1,30,"Verifique o cartao");
744             }while(u8g2.nextPage());
745         }
746         TelaAtual = 0;
747         delay(2000);
748         break;
749
750     case 6: //Ajuste manual motor 1
751         u8g2.setBitmapMode(1); //faz os bitmaps transparentes
752         u8g2.firstPage();
753         do{
754             u8g2.drawBitmap(4,2,16/8,16, Bitmap_Icons[5]); //Desenha Icone turbina 1
755             u8g2.setFont(u8g2_font_7x14B_tf);
756             u8g2.drawStr(26,15,"Vel Motor 1");
757             u8g2.setCursor(50, 37);
758             u8g2.print(ESCA_Vel);
759             u8g2.drawBitmap(0,43,128/8,21, epd_bitmap_BordaSelec);
760             u8g2.drawBox(4,45,map(ESCA_Vel, 0, 180, 0,114),16);
761         }while(u8g2.nextPage());
762         break;
763
764     case 7: //Ajuste manual motor 2

```

```

765     u8g2.setBitmapMode(1); //faz os bitmaps transparentes
766     u8g2.firstPage();
767     do{
768         u8g2.drawBitmap(4,2,16/8,16, Bitmap_Icons[6]); //Desenha Icone turbina 2
769         u8g2.setFont(u8g2_font_7x14B_tf);
770         u8g2.drawStr(26,15,"Vel Motor 2");
771         u8g2.setCursor(50, 37);
772         u8g2.print(ESCB_Vel);
773         u8g2.drawBitmap(0,43,128/8,21, epd_bitmap_BordaSelec);
774         u8g2.drawBox(4,45,map(ESCB_Vel, 0, 180, 0,114),16);
775     }while(u8g2.nextPage());
776     break;
777
778 case 8: //nave
779     bearing = 50;
780     if(millis() - last_time_disp > 300){//roda a cada 300ms
781         u8g2.firstPage();
782         do{
783             u8g2.setFont(u8g2_font_7x14B_tf);
784             u8g2.drawStr(1,15,"Menu teste");
785             u8g2.setCursor(1, 30);
786             u8g2.print(bearing);
787             u8g2.setCursor(48, 30);
788             u8g2.print(direc);
789             u8g2.setCursor(4, 45);
790             u8g2.print(ESCA_Vel);
791             u8g2.setCursor(35, 45);
792             u8g2.print(ESCB_Vel);
793             u8g2.setCursor(1, 60);
794             u8g2.print("V");
795             u8g2.setCursor(8, 60);
796             u8g2.print(percentualV);
797             u8g2.setCursor(32, 60);
798             u8g2.print("PID");
799             u8g2.setCursor(56, 60);
800             u8g2.print(percentualP);
801             u8g2.setCursor(80, 60);
802             u8g2.print(percentuall);
803             u8g2.setCursor(104, 60);
804             u8g2.print(percentualD);
805         }while(u8g2.nextPage());
806         last_time_disp = millis();
807     }
808     //Escreve os valores nos ESCs
809     CalculaGaus();
810     CalculaPIDDirec();
811     if(aux){

```

```

812     TextoLog = String(millis()) + " " + String(proporcional) + " " + String(integral) + " " + String(derivativo) + " " +
+ String(saida) + " " + String(direc) + " " + String(bearing) + " " + String(erro) + " " + String(ESCA_Vel) + " " +
" + String(ESCB_Vel) + " ";
813     Arquivo = SD.open("/logparado.txt", FILE_APPEND,true);
814     if(Arquivo) {
815         Arquivo.println(TextoLog);
816         Arquivo.close();
817     }}
818     aux = !aux;
819
820     break;
821
822 case 9: //Tela de informacoes
823     u8g2.firstPage();
824     do{
825         u8g2.setFont(u8g2_font_7x14B_tf);
826         u8g2.drawStr(1,15,"Veiculo Autonomo");
827         u8g2.drawStr(1,30,"de Superficie");
828         u8g2.setFont(u8g2_font_7x14B_tf);
829         u8g2.drawStr(1,45,"Trab Conc Curso");
830         u8g2.drawStr(1,60,"Jhonatas WG Moraes");
831     }while(u8g2.nextPage());
832     break;
833
834 case 10: //Ajuste V
835     u8g2.setBitmapMode(1); //faz os bitmaps transparentes
836     u8g2.firstPage();
837     do{
838         u8g2.drawBitmap(4,2,16/8,16, Bitmap_Icons[9]);
839         u8g2.setFont(u8g2_font_7x14B_tf);
840         u8g2.drawStr(26,15,"Ajuste Vel.");
841         u8g2.setCursor(50, 37);
842         u8g2.print(percentualV);
843         u8g2.drawBitmap(0,43,128/8,21, epd_bitmap_BordaSelec);
844         u8g2.drawRect(4,45,map(percentualV, 0, 100, 0,114),16);
845     }while(u8g2.nextPage());
846     break;
847
848 case 11: //Ajuste P
849     u8g2.setBitmapMode(1); //faz os bitmaps transparentes
850     u8g2.firstPage();
851     do{
852         u8g2.drawBitmap(4,2,16/8,16, Bitmap_Icons[10]);
853         u8g2.setFont(u8g2_font_7x14B_tf);
854         u8g2.drawStr(26,15,"Ajuste P");
855         u8g2.setCursor(50, 37);
856         u8g2.print(percentualP);
857         u8g2.drawBitmap(0,43,128/8,21, epd_bitmap_BordaSelec);
858         u8g2.drawRect(4,45,map(percentualP, 0, 100, 0,114),16);

```

```

859     }while(u8g2.nextPage());
860     break;
861
862     case 12: //Ajuste I
863         u8g2.setBitmapMode(1); //faz os bitmaps transparentes
864         u8g2.firstPage();
865         do{
866             u8g2.drawBitmap(4,2,16/8,16, Bitmap_Icons[11]);
867             u8g2.setFont(u8g2_font_7x14B_tf);
868             u8g2.drawStr(26,15,"Ajuste I");
869             u8g2.setCursor(50, 37);
870             u8g2.print(percentualI);
871             u8g2.drawBitmap(0,43,128/8,21, epd_bitmap_BordaSelec);
872             u8g2.drawRect(4,45,map(percentualI, 0, 100, 0,114),16);
873         }while(u8g2.nextPage());
874         break;
875
876     case 13: //Ajuste D
877         u8g2.setBitmapMode(1); //faz os bitmaps transparentes
878         u8g2.firstPage();
879         do{
880             u8g2.drawBitmap(4,2,16/8,16, Bitmap_Icons[12]);
881             u8g2.setFont(u8g2_font_7x14B_tf);
882             u8g2.drawStr(26,15,"Ajuste D");
883             u8g2.setCursor(50, 37);
884             u8g2.print(percentualD);
885             u8g2.drawBitmap(0,43,128/8,21, epd_bitmap_BordaSelec);
886             u8g2.drawRect(4,45,map(percentualD, 0, 100, 0,114),16);
887         }while(u8g2.nextPage());
888         break;
889
890
891     default://Caso alguma tela ainda nao implementada
892         TelaAtual = 0;
893         break;
894     }
895 }
896
897 //-----Setup
898
899 void setup(void) {
900     //Inicializa diplay, bme e cartao SD
901     u8g2.begin();
902     bme.begin(0x76);
903     SD.begin(10);
904     //Inicializa portas seriais (Serial0 = USB; Serial2 = GPS)
905     Serial.begin(115200);
906     Serial2.begin(9600);

```

```

907 //Inicializa EEPROM e carrega os parametros de calibracao do magnetometro
908 EEPROM.begin(EEPROM_SIZE);
909 min_x=EEPROM.read(0)<<8|EEPROM.read(1);
910 max_x=EEPROM.read(2)<<8|EEPROM.read(3);
911 min_y=EEPROM.read(4)<<8|EEPROM.read(5);
912 max_y=EEPROM.read(6)<<8|EEPROM.read(7);
913 percentualP = EEPROM.read(8); //Carrega valores P
914 percentualI = EEPROM.read(9); //Carrega valores I
915 percentualD = EEPROM.read(10); //Carrega valores D
916 percentualV = EEPROM.read(11); //Carrega valores Velocidade
917 //Inicializa o magnetometro e aplica os parametros de calibracao carregados da EEPROM (Apenas valores de x e y sao
    importantes)
918 compass.init();
919 compass.setCalibration(min_x, max_x, min_y, max_y, -32768, 32767);
920 //Prepara os pinos de entrada para os botoes
921 pinMode(btn_1, INPUT_PULLUP);
922 pinMode(btn_2, INPUT_PULLUP);
923 pinMode(btn_3, INPUT_PULLUP);
924 pinMode(bomba1, OUTPUT);
925 pinMode(bomba2, OUTPUT);
926 pinMode(bomba3, OUTPUT);
927 pinMode(bomba4, OUTPUT);
928 TelaAtual = 0;
929
930 //Prepara os paramtros dos ESCs conectados
931 ESP32PWM::allocateTimer(0);
932 ESP32PWM::allocateTimer(1);
933 ESCA.setPeriodHertz(50); // Standard 50hz servo
934 ESCB.setPeriodHertz(50); // Standard 50hz servo
935 ESCA.attach(ESCA_Pin, minUs, maxUs);
936 ESCB.attach(ESCB_Pin, minUs, maxUs);
937
938 delay(1000);
939 ESCA_Vel = 90; //velocidade inicial dos ESCs (ponto "zero", meio do curso)
940 ESCB_Vel = 90;
941 ESCA.write(ESCA_Vel);
942 ESCB.write(ESCB_Vel);
943
944 //Inicializa cartao SD
945 Serial.println("Inicializando cartao SD...");
946 if (!SD.begin(CS)) {
947     Serial.println("Falha na inicializacao! Verificar os cabos e o cartao.");
948     return;
949 }
950
951 Serial.println("Inicializado.");
952 }
953 //-----Loop

```



```

954 void loop(void) {
955     //Splash Screen
956     if(StartScreen){
957         u8g2.firstPage();
958         do{
959             u8g2.drawBitmap(0,0,128/8,64, epd_bitmap_StartScreen); //Desenha selecionado
960         }while(u8g2.nextPage());
961         delay(2000);
962         StartScreen = false;
963     }
964
965     //Comandos para botao 1 (desce)
966     if((digitalRead(btn_1) == LOW)&&(btn_1_clic == 0)){
967         if(TelaAtual == 0){
968             item_sel = item_sel-1;
969             if(item_sel<0){item_sel = NUM_ITENS-1;}
970         }else if(TelaAtual == 6){
971             ESCA_Vel-= 10;//10
972             if(ESCA_Vel<0){ESCA_Vel = 0;}
973             ESCA.write(180-ESCA_Vel);
974         }else if(TelaAtual == 7){
975             ESCB_Vel-= 10;//10
976             if(ESCB_Vel<0){ESCB_Vel = 0;}
977
978             ESCB.write(180-ESCB_Vel);
979         }else if(TelaAtual == 2){
980             coordenada = coordenada-1;
981             if(coordenada<0){coordenada = PontosLidos;}
982         }else if(TelaAtual == 10){//Ajusta V
983             percentualV-= 1;
984             if(percentualV<0){percentualV = 0;}
985         }else if(TelaAtual == 11){//Ajusta P
986             percentualP-= 1;
987             if(percentualP<0){percentualP = 0;}
988         }else if(TelaAtual == 12){//Ajusta P
989             percentuall-= 1;
990             if(percentuall<0){percentuall = 0;}
991         }else if(TelaAtual == 13){//Ajusta P
992             percentualD-= 1;
993             if(percentualD<0){percentualD = 0;}
994         }
995         btn_1_clic = 1;
996     }
997     //Comandos para botao 2 (sobe)
998     if((digitalRead(btn_2) == LOW)&&(btn_2_clic == 0)){
999         if(TelaAtual == 0){
1000             item_sel = item_sel+1;
1001             if(item_sel>= NUM_ITENS){item_sel = 0;}
1002         }else if(TelaAtual == 6){

```

```

1003     ESCA_Vel += 10; //10
1004     if(ESCA_Vel > 180){ESCA_Vel = 180;}
1005     ESCA.write(180-ESCA_Vel);
1006 }else if(TelaAtual == 7){
1007     ESCB_Vel += 10;
1008     if(ESCB_Vel > 180){ESCB_Vel = 180;}
1009     ESCB.write(180-ESCB_Vel);
1010 }else if(TelaAtual == 2){
1011     coordenada = coordenada+1;
1012     if(coordenada >= PontosLidos){coordenada = 0;}
1013 }else if(TelaAtual == 10){//Ajusta Vmax
1014     percentualV += 1;
1015     if(percentualV > Vmax){percentualV = Vmax;}
1016 }else if(TelaAtual == 11){//Ajusta P
1017     percentualP += 1;
1018     if(percentualP > 100){percentualP = 100;}
1019 }else if(TelaAtual == 12){//Ajusta I
1020     percentualI += 1;
1021     if(percentualI > 100){percentualI = 100;}
1022 }else if(TelaAtual == 13){//Ajusta D
1023     percentualD += 1;
1024     if(percentualD > 100){percentualD = 100;}
1025 }
1026
1027     btn_2_clic = 1;
1028 }
1029 //Comandos para botao 3 (entra)
1030 if((digitalRead(btn_3) == LOW)&&(btn_3_clic == 0)){
1031
1032     if(TelaAtual == 0){
1033         switch (item_sel)
1034         {
1035             case 0:
1036                 TelaAtual = 1;
1037                 break;
1038
1039             case 1:
1040                 TelaAtual = 2;
1041                 break;
1042
1043             case 2:
1044                 TelaAtual = 3;
1045                 break;
1046
1047             case 3:
1048                 TelaAtual = 4;
1049                 break;
1050
1051             case 4:

```

```

1052     TelaAtual = 5;
1053     break;
1054
1055 case 5:
1056     TelaAtual = 6;
1057     break;
1058
1059 case 6:
1060     TelaAtual = 7;
1061     break;
1062 case 7:
1063     TelaAtual = 8;
1064     break;
1065 case 8:
1066     TelaAtual = 9;
1067     break;
1068
1069 case 9:
1070     TelaAtual = 10;
1071     break;
1072
1073 case 10: //Ajuste P
1074     TelaAtual = 11;
1075     break;
1076
1077 case 11: //Ajuste I
1078     TelaAtual = 12;
1079     break;
1080
1081 case 12: //Ajuste D
1082     TelaAtual = 13;
1083     break;
1084
1085 default:
1086     TelaAtual = 0;
1087     break;
1088 }
1089 }else if(TelaAtual == 4 || TelaAtual == 8){
1090     parar_motores();
1091     u8g2.firstPage();
1092     do{
1093         u8g2.setFont(u8g2_font_7x14B_tf);
1094         u8g2.drawStr(1,15,"Motores parados");
1095     }while(u8g2.nextPage());
1096     delay(2000);
1097     TelaAtual = 0;
1098 }else if(TelaAtual == 10){
1099     EEPROM.put(11, percentualV);
1100     EEPROM.commit();

```

```

1101     u8g2.firstPage();
1102     do{
1103         u8g2.setFont(u8g2_font_7x14B_tf);
1104         u8g2.drawStr(1,15,"Ajuste Vmax armazenado");
1105     }while(u8g2.nextPage());
1106     delay(2000);
1107     TelaAtual = 0;
1108 }else if(TelaAtual == 11){
1109     EEPROM.put(8, percentualP);
1110     EEPROM.commit();
1111     u8g2.firstPage();
1112     do{
1113         u8g2.setFont(u8g2_font_7x14B_tf);
1114         u8g2.drawStr(1,15,"Ajuste P armazenado");
1115     }while(u8g2.nextPage());
1116     delay(2000);
1117     TelaAtual = 0;
1118 }else if(TelaAtual == 12){
1119     EEPROM.put(9, percentualI);
1120     EEPROM.commit();
1121     u8g2.firstPage();
1122     do{
1123         u8g2.setFont(u8g2_font_7x14B_tf);
1124         u8g2.drawStr(1,15,"Ajuste I armazenado");
1125     }while(u8g2.nextPage());
1126     delay(2000);
1127     TelaAtual = 0;
1128 }else if(TelaAtual == 13){
1129     EEPROM.put(10, percentualD);
1130     EEPROM.commit();
1131     u8g2.firstPage();
1132     do{
1133         u8g2.setFont(u8g2_font_7x14B_tf);
1134         u8g2.drawStr(1,15,"Ajuste D armazenado");
1135     }while(u8g2.nextPage());
1136     delay(2000);
1137     TelaAtual = 0;
1138 }else{
1139     TelaAtual = 0;
1140 }
1141 btn_3_clic = 1;
1142 }
1143
1144 //Debounce dos botoes
1145 if((digitalRead(btn_1) == HIGH)&&(btn_1_clic == 1)){btn_1_clic = 0;}
1146 if((digitalRead(btn_2) == HIGH)&&(btn_2_clic == 1)){btn_2_clic = 0;}
1147 if((digitalRead(btn_3) == HIGH)&&(btn_3_clic == 1)){btn_3_clic = 0;}
1148 //Corrige os limites da tela inicial
1149 item_ant = item_sel-1;

```

```

1150  if(item_ant<0){item_ant = NUM_ITENS-1;}
1151  item_pos = item_sel+1;
1152  if(item_pos >= NUM_ITENS){item_pos = 0;}
1153
1154  MostraTela(); //usa a funcao para mostrar dados no display, e as funcoes do display para executar as operacoes
1155  }

```

```

1  /*Codigo auxiliar dos bitmaps
2  Jhonatas Willian Gonçalves de Moraes
3  Trabalho de Conclusao de Curso
4  Universidade Tecnologica Federal do Parana
5  Campus Apucarana
6  Engenharia Eletrica*/
7
8  // 'StartScreen', 128x64px
9  const unsigned char epd_bitmap_StartScreen [] PROGMEM = {
10  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
11  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
12  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
13  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
14  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
15  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0xc0, 0x00, 0x00, 0x00, 0x00,
16  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1c, 0xc0, 0x03, 0x80, 0x00, 0x01, 0xc0, 0x00,
17  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1c, 0xc0, 0x00, 0x00, 0x00, 0x01, 0xc0, 0x00,
18  0x00, 0x00, 0x07, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1c, 0xc7, 0xc3, 0x87, 0xc6, 0x71, 0xc3, 0xe0,
19  0x00, 0x00, 0x0b, 0x00, 0x00, 0x00, 0x00, 0x00, 0x1c, 0xce, 0xe3, 0x8e, 0xe6, 0x71, 0xc6, 0x70,
20  0x00, 0x00, 0x0b, 0x80, 0x00, 0x00, 0x00, 0x00, 0x1c, 0xce, 0x63, 0x8c, 0x66, 0x71, 0xc6, 0x70,
21  0x00, 0x00, 0x0b, 0x80, 0x00, 0x00, 0x00, 0x00, 0x1c, 0xcf, 0xe3, 0x8c, 0x06, 0x71, 0xc6, 0x70,
22  0x00, 0x00, 0x0f, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x1c, 0xcf, 0xe3, 0x8c, 0x06, 0x71, 0xc6, 0x70,
23  0x00, 0x00, 0x17, 0x20, 0x00, 0x00, 0x00, 0x00, 0x1d, 0xce, 0x03, 0x8c, 0x06, 0x71, 0xc6, 0x70,
24  0x00, 0x00, 0x67, 0x38, 0x00, 0x00, 0x00, 0x00, 0x0f, 0x8e, 0x63, 0x8c, 0xe6, 0x71, 0xc6, 0x70,
25  0x00, 0x01, 0x87, 0x70, 0x00, 0x00, 0x00, 0x00, 0x07, 0x07, 0xc3, 0x87, 0xc3, 0xe0, 0xe3, 0xe0,
26  0x00, 0x06, 0x1d, 0x60, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
27  0x00, 0x08, 0x6d, 0x90, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
28  0x00, 0x30, 0xc7, 0xd8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
29  0x00, 0xc1, 0x0d, 0x54, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x38, 0x00, 0x00, 0x00, 0x00, 0x00,
30  0x01, 0x07, 0xd5, 0x32, 0x10, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7c, 0x00, 0x00, 0x00, 0x00, 0x00,
31  0x03, 0x18, 0x64, 0x8d, 0x6c, 0x00, 0x01, 0xf0, 0x00, 0x00, 0xce, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
32  0x0e, 0x23, 0xb0, 0x02, 0x86, 0x00, 0x03, 0x98, 0x00, 0x70, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
33  0x09, 0xc2, 0x18, 0x03, 0x99, 0x00, 0x03, 0x99, 0xcc, 0xfc, 0x7c, 0x7e, 0x1f, 0x1f, 0xc3, 0xe0,
34  0x0b, 0xce, 0x06, 0x0c, 0x21, 0x80, 0x03, 0x99, 0xcc, 0x70, 0xee, 0x77, 0x3b, 0x9f, 0xe7, 0x70,
35  0x07, 0x52, 0xc9, 0xd0, 0xc3, 0x00, 0x03, 0x99, 0xcc, 0x70, 0xce, 0x67, 0x33, 0x9b, 0x67, 0x30,
36  0x09, 0xe6, 0x30, 0x61, 0x04, 0x80, 0x03, 0xf9, 0xcc, 0x70, 0xce, 0x67, 0x33, 0x9b, 0x67, 0x30,
37  0x08, 0xaf, 0x20, 0x86, 0x08, 0x80, 0x03, 0xf9, 0xcc, 0x70, 0xce, 0x67, 0x33, 0x9b, 0x67, 0x30,
38  0x09, 0x7e, 0xc3, 0x08, 0x33, 0x00, 0x03, 0x99, 0xcc, 0x70, 0xce, 0x67, 0x33, 0x9b, 0x67, 0x30,
39  0x07, 0xe0, 0x2c, 0x30, 0x44, 0x00, 0x03, 0x99, 0xcc, 0x70, 0xce, 0x67, 0x33, 0x98, 0x67, 0x30,
40  0x00, 0x00, 0x18, 0x41, 0x88, 0x00, 0x03, 0x98, 0xf8, 0x3c, 0x7c, 0x67, 0x1f, 0x18, 0x63, 0xe0,
41  0x00, 0x00, 0x61, 0x82, 0x30, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
42  0x00, 0x00, 0x82, 0x0c, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

```

```

43 0x00, 0x01, 0x6c, 0x11, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
44 0x00, 0x01, 0x30, 0x22, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x60, 0x00,
45 0x00, 0x01, 0x38, 0xcc, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x60, 0x00,
46 0x00, 0x01, 0xaf, 0x10, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0xe3, 0xe0,
47 0x00, 0x01, 0xc6, 0x20, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0e, 0xe7, 0x70,
48 0x00, 0x01, 0x24, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0e, 0x67, 0x30,
49 0x00, 0x01, 0xfd, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0e, 0x67, 0xf0,
50 0x00, 0x00, 0x27, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0e, 0x67, 0xf0,
51 0x00, 0x00, 0x3c, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0e, 0x67, 0x00,
52 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0e, 0x67, 0x30,
53 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0xe3, 0xe0,
54 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
55 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
56 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x60, 0x00, 0x00, 0x00,
57 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x70, 0xe0, 0x00, 0xc0, 0x00,
58 0x00, 0x00, 0x00, 0x00, 0x00, 0x1f, 0x00, 0x00, 0x00, 0x00, 0x00, 0x78, 0xc0, 0x01, 0xc0, 0x00,
59 0x00, 0x00, 0x00, 0x00, 0x00, 0x33, 0x80, 0x00, 0x00, 0x00, 0x00, 0xc0, 0x00, 0x00, 0x00, 0x00,
60 0x00, 0x00, 0x00, 0x00, 0x00, 0x30, 0x19, 0xcf, 0xc3, 0xe3, 0x38, 0xc0, 0xc3, 0xe1, 0xc3, 0xe0,
61 0x00, 0x00, 0x00, 0x00, 0x00, 0x30, 0x19, 0xcc, 0xe6, 0x73, 0x79, 0xf8, 0xc7, 0x31, 0xc7, 0x30,
62 0x00, 0x00, 0x00, 0x00, 0x00, 0x1f, 0x19, 0xcc, 0xe6, 0x73, 0x80, 0xc0, 0xc7, 0x01, 0xc7, 0x30,
63 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0x99, 0xcc, 0xe7, 0xf3, 0x00, 0xc0, 0xc7, 0x01, 0xc7, 0xf0,
64 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0x99, 0xcc, 0xe6, 0x03, 0x00, 0xc0, 0xc7, 0x01, 0xc7, 0x00,
65 0x00, 0x00, 0x00, 0x00, 0x00, 0x33, 0x99, 0xcc, 0xe6, 0x63, 0x00, 0xc0, 0xc7, 0x31, 0xc7, 0x30,
66 0x00, 0x00, 0x00, 0x00, 0x00, 0x33, 0x9f, 0x8f, 0xc7, 0xe3, 0x00, 0xc0, 0xc3, 0xf1, 0xc7, 0xf0,
67 0x00, 0x00, 0x00, 0x00, 0x00, 0x1f, 0x0f, 0x8f, 0xc3, 0xe3, 0x00, 0xc0, 0xc3, 0xe1, 0xc3, 0xe0,
68 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0c, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
69 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0c, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
70 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
71 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
72 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
73 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
74 };
75
76
77
78
79
80 // 'Bussolalcon', 16x16px
81 const unsigned char epd_bitmap_Bussolalcon [] PROGMEM = {
82 0x07, 0xe0, 0x18, 0x98, 0x20, 0x9c, 0x50, 0x3a, 0x48, 0x6a, 0x81, 0xd1, 0x82, 0xa1, 0x87, 0xc7,
83 0xe5, 0x41, 0x8b, 0x81, 0x97, 0x01, 0x6c, 0x0a, 0x78, 0x06, 0x30, 0x84, 0x18, 0x98, 0x07, 0xe0
84 };
85 // 'GPSIcon', 16x16px
86 const unsigned char epd_bitmap_GPSIcon [] PROGMEM = {
87 0x40, 0x00, 0x20, 0x00, 0xfc, 0x00, 0x87, 0xc0, 0x7b, 0x70, 0x16, 0x10, 0x0c, 0x18, 0x04, 0x34,
88 0x3c, 0x47, 0x23, 0xd8, 0x3c, 0xbb, 0x0c, 0x86, 0x1a, 0x81, 0x03, 0x80, 0x00, 0x00, 0x00, 0x00
89 };
90 // 'Itemlco', 16x16px
91 const unsigned char epd_bitmap_Itemlco [] PROGMEM = {

```

```

92 0x00, 0x00, 0x00, 0x00, 0x30, 0x00, 0x30, 0x00, 0x20, 0x70, 0x1f, 0x90, 0x17, 0x90, 0x17, 0x88,
93 0x15, 0x88, 0x17, 0x88, 0x16, 0x08, 0x3f, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
94 };
95
96 // 'Bandeiralcon', 16x16px
97 const unsigned char epd_bitmap_Bandeiralcon [] PROGMEM = {
98 0x00, 0x3c, 0xf8, 0xc3, 0x67, 0x01, 0x60, 0x01, 0x60, 0x01, 0x60, 0x01, 0x60, 0x01, 0x60, 0x01,
99 0x60, 0x3d, 0x78, 0xc3, 0x67, 0x00, 0x60, 0x00, 0x60, 0x00, 0x60, 0x00, 0x60, 0x00, 0x60, 0x00
100 };
101 // 'Cartaolcon', 16x16px
102 const unsigned char epd_bitmap_Cartaolcon [] PROGMEM = {
103 0x0f, 0xfe, 0x1a, 0xaa, 0x2a, 0xaa, 0x6a, 0xaa, 0xa0, 0x03, 0x03, 0x80, 0x02, 0x40, 0x02,
104 0x80, 0x02, 0x80, 0x02, 0x80, 0x02, 0x80, 0x02, 0x80, 0x02, 0x80, 0x02, 0x80, 0x02, 0xff, 0xfe
105 };
106 // 'Turbinalcon1', 16x16px
107 const unsigned char epd_bitmap_Turbinalcon1 [] PROGMEM = {
108 0x07, 0xe0, 0x1b, 0x98, 0x22, 0x44, 0x42, 0x22, 0x42, 0x22, 0x82, 0x61, 0x82, 0x41, 0x81, 0x81,
109 0x9e, 0x61, 0xa1, 0xff, 0xc2, 0xc0, 0x64, 0x4c, 0x58, 0x44, 0x20, 0x44, 0x18, 0x44, 0x07, 0xc4
110 };
111 // 'Turbinalcon2', 16x16px
112 const unsigned char epd_bitmap_Turbinalcon2 [] PROGMEM = {
113 0x07, 0xe0, 0x1b, 0x98, 0x22, 0x44, 0x42, 0x22, 0x42, 0x22, 0x82, 0x61, 0x82, 0x41, 0x81, 0x81,
114 0x9e, 0x61, 0xa1, 0xff, 0xc2, 0xc0, 0x64, 0x4e, 0x58, 0x42, 0x20, 0x44, 0x18, 0x48, 0x07, 0xce
115 };
116
117 // 'Navelcon', 16x16px
118 const unsigned char epd_bitmap_Navelcon [] PROGMEM = {
119 0x01, 0x80, 0x01, 0x80, 0x01, 0x80, 0x02, 0x40, 0x02, 0x40, 0x02, 0x40, 0x02, 0x40, 0x04, 0x20,
120 0x85, 0xa1, 0xc9, 0x93, 0xb0, 0x0d, 0x83, 0xc1, 0x80, 0x01, 0x90, 0x09, 0x9e, 0x79, 0xf3, 0xcf
121 };
122 // 'Infolcon', 16x16px
123 const unsigned char epd_bitmap_Infolcon [] PROGMEM = {
124 0x1f, 0xf8, 0x10, 0x08, 0x10, 0x08, 0x1c, 0x38, 0x04, 0x20, 0x04, 0x20, 0x04, 0x20, 0x04, 0x20,
125 0x04, 0x20, 0x04, 0x20, 0x04, 0x20, 0x04, 0x20, 0x1c, 0x38, 0x10, 0x08, 0x10, 0x08, 0x1f, 0xf8
126 };
127
128 // 'engrenagem', 16x16px
129 const unsigned char epd_bitmap_engrenagem [] PROGMEM = {
130 0x01, 0x80, 0x03, 0xc0, 0x1c, 0x38, 0x2f, 0xf4, 0x32, 0x4c, 0x34, 0x2c, 0x59, 0x9a, 0xd2, 0x4b,
131 0xd2, 0x4b, 0x59, 0x9a, 0x34, 0x2c, 0x32, 0x4c, 0x2f, 0xf4, 0x1c, 0x38, 0x03, 0xc0, 0x01, 0x80
132 };
133
134
135
136 // 'P', 16x16px
137 const unsigned char epd_bitmap_P [] PROGMEM = {
138 0x01, 0x80, 0x03, 0xc0, 0x1c, 0x38, 0x2f, 0xf4, 0x32, 0x4c, 0x34, 0x2c, 0x59, 0x9a, 0xd2, 0x4b,
139 0xd2, 0x4b, 0x59, 0xff, 0x34, 0x40, 0x32, 0x5e, 0x2f, 0xd2, 0x1c, 0x5e, 0x03, 0xd0, 0x01, 0xd0
140 };

```





