



UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO  
CENTRO MULTIDISCIPLINAR DE CARAÚBAS  
DEPARTAMENTO DE ENGENHARIAS  
CURSO DE ENGENHARIA ELÉTRICA

DULCIEL MARCOS ABEL DA SILVA

**PROJETO E DESENVOLVIMENTO DE UM SISTEMA ELETRÔNICO DE  
MONITORAMENTO PARA VEÍCULO *OFF-ROAD* TIPO BAJA**

CARAÚBAS – RN

2024

DULCIEL MARCOS ABEL DA SILVA

**PROJETO E DESENVOLVIMENTO DE UM SISTEMA ELETRÔNICO DE  
MONITORAMENTO PARA VEÍCULO *OFF-ROAD* TIPO BAJA**

Monografia apresentada à Universidade Federal Rural do Semi-Árido como requisito para obtenção do título de Bacharel em Engenharia Elétrica.

Orientador: Walber Medeiros Lima, Prof. Dr.

Coorientador: Diego David Silva Diniz, Prof. Dr.

CARAÚBAS – RN

2024

© Todos os direitos estão reservados a Universidade Federal Rural do Semi-Árido. O conteúdo desta obra é de inteira responsabilidade do (a) autor (a), sendo o mesmo, passível de sanções administrativas ou penais, caso sejam infringidas as leis que regulamentam a Propriedade Intelectual, respectivamente, Patentes: Lei nº 9.279/1996 e Direitos Autorais: Lei nº 9.610/1998. O conteúdo desta obra tomar-se-á de domínio público após a data de defesa e homologação da sua respectiva ata. A mesma poderá servir de base literária para novas pesquisas, desde que a obra e seu (a) respectivo (a) autor (a) sejam devidamente citados e mencionados os seus créditos bibliográficos.

S586p Silva, Dulciel Marcos Abel da.  
Projeto e desenvolvimento de um sistema eletrônico de monitoramento para veículo off-road tipo baja / Dulciel Marcos Abel da Silva. - 2024.  
67 f. : il.

Orientador: Walber Medeiros Lima.  
Coorientador: Diego David Silva Diniz.  
Monografia (graduação) - Universidade Federal Rural do Semi-árido, Curso de , 2024.

1. baja SAE. 2. veículos off-road. 3. sistemas embarcados. 4. arduino. 5. comunicação I2C. I. Lima, Walber Medeiros, orient. II. Diniz, Diego David Silva, co-orient. III. Título.

Ficha catalográfica elaborada por sistema gerador automático em conformidade com AACR2 e os dados fornecidos pelo(a) autor(a).  
Biblioteca Campus Caraúbas  
Bibliotecária: Dalvanira Brito Rodrigues  
CRB: 15/700

O serviço de Geração Automática de Ficha Catalográfica para Trabalhos de Conclusão de Curso (TCC's) foi desenvolvido pelo Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (USP) e gentilmente cedido para o Sistema de Bibliotecas da Universidade Federal Rural do Semi-Árido (SISBI-UFERSA), sendo customizado pela Superintendência de Tecnologia da Informação e Comunicação (SUTIC) sob orientação dos bibliotecários da instituição para ser adaptado às necessidades dos alunos dos Cursos de Graduação e Programas de Pós-Graduação da Universidade.

DULCIEL MARCOS ABEL DA SILVA

**PROJETO E DESENVOLVIMENTO DE UM SISTEMA ELETRÔNICO DE  
MONITORAMENTO PARA VEÍCULO *OFF-ROAD* TIPO BAJA**

Monografia apresentada à Universidade  
Federal Rural do Semi-Árido como requisito  
para obtenção do título de Bacharel em  
Engenharia Elétrica.

Defendida em: 24/ 10 / 2024.

**BANCA EXAMINADORA**

---

Prof. Dr. Walber Medeiros Lima (UFERSA)  
Presidente

---

Prof. Dr. Diego David Silva Diniz (UFERSA)  
Membro Examinador

---

Eng. Me. Allyson Arilson Lima Filgueira (UFERSA)  
Membro Examinador

*Dedico aos meus pais, Dulcinea Abel e Manoel Silva, à minha tia Maria do Socorro Abel, à minha namorada Francisca Gizele Bezerra e ao meu avô Cícero Abel, pelo amor, apoio e valores que me foram ensinados ao longo dos anos.*

## **AGRADECIMENTOS**

Em primeiro lugar, agradeço a Deus por suas graças derramadas sobre minha família e por me dar força e coragem durante toda a minha caminhada.

Agradeço aos meus pais, Dulcinea Maria Abel e Manoel Marcos da Silva, à minha irmã Dávila Emanuely Abel da Silva, à minha namorada, melhor amiga e companheira Francisca Gizele Bezerra. Também agradeço aos meus amigos Lino Alessandro Moraes Alves, Erick Aragão Gadelha de Oliveira, Luiz Paulo Gomes da Silva, Werly Renan Fernandes Lucena, Adriel Batista Soares e Ednardo Édson Lucena e Lima, que estiveram comigo durante todos esses anos, ajudaram-me e possibilitaram que eu chegasse a essa nova etapa da minha vida.

Agradeço ao meu avô Cícero Abel, que Deus o tenha, que infelizmente não pôde ver a pessoa que me tornei, mas que contribuiu muito para que eu me tornasse quem sou. Meu avô foi como um pai para mim, e carrego com todo o carinho os ensinamentos e lições de vida que me foram passados.

Agradeço aos meus professores e orientadores Walber Medeiros Lima, Diego David Silva Diniz, pela orientação, aconselhamento e disposição para a conclusão deste trabalho.

Agradeço a todos os professores que tive ao longo da minha vida acadêmica, que repassaram conhecimentos e compartilharam ensinamentos, contribuindo para a formação da pessoa que sou atualmente.

Agradeço também a todos que, de forma direta ou indireta, contribuíram para a conclusão deste trabalho.

*“A procura é a natureza da  
sabedoria.”*

*(Percy Jackson e os Olimpianos)*

## RESUMO

O projeto Baja SAE é uma competição universitária internacional que desafia estudantes de engenharia a projetar, construir e testar um veículo *off-road*, integrando sistemas mecânicos e elétricos. Com o objetivo de aplicar na prática conceitos teóricos e desenvolver habilidades em gestão de projetos e trabalho em equipe, este trabalho foca na implementação de um sistema de monitoramento eletrônico utilizando placas Arduino. O sistema coleta e analisa, em tempo real, dados de parâmetros como o acionamento do sistema de tração 4x4, temperatura, velocidade angular da transmissão continuamente variável (CVT) e velocidade do veículo, visando à otimização do desempenho, à segurança do veículo e do piloto. Foram desenvolvidas placas de circuito impresso (PCI) específicas, integrando componentes do sistema de monitoramento e utilizando placas Arduino, visando organizar e otimizar os circuitos eletrônicos. A metodologia abrangeu o desenvolvimento dos circuitos, a programação das placas para coleta e análise de dados, além de ensaios experimentais para validar o sistema. Os testes de temperatura e velocidade angular realizados com o sistema proposto comprovaram sua eficácia, garantindo a precisão na leitura dos dados e uma resposta rápida às variações dos parâmetros monitorados. Os resultados mostram que o sistema é capaz de melhorar a segurança e o desempenho do veículo baja SAE, tornando-o uma ferramenta valiosa para futuras competições e aprimoramentos no projeto.

**Palavras-chave:** baja SAE, veículos *off-road*, sistemas embarcados, arduino, comunicação I2C.



## **ABSTRACT**

The Baja SAE project is an international university competition that challenges engineering students to design, build and test an off-road vehicle, integrating mechanical and electrical systems. With the aim of applying theoretical concepts into practice and developing skills in project management and teamwork, this work focuses on the implementation of an electronic monitoring system using Arduino boards. The system collects and analyzes, in real time, data on parameters such as the activation of the 4x4 traction system, temperature, angular speed of the continuously variable transmission (CVT) and vehicle speed, aiming to optimize performance, vehicle and vehicle safety. Specific printed circuit boards (PCB) were developed, integrating components of the monitoring system and using Arduino boards, aiming to organize and optimize electronic circuits. The methodology covered the development of the circuits, the programming of the boards for data collection and analysis, as well as experimental tests to validate the system. Temperature and angular velocity tests carried out with the proposed system proved its effectiveness, ensuring accuracy in data reading and a quick response to variations in monitored parameters. The results show that the system is capable of improving the safety and performance of the SAE baja vehicle, making it a valuable tool for future competitions and design improvements.

**Keywords:** baja SAE, off-road vehicles, embedded systems, arduino, I2C communication.

## LISTA DE FIGURAS

|                                                                                                                                         |    |
|-----------------------------------------------------------------------------------------------------------------------------------------|----|
| <b>Figura 1</b> - Exemplos de placas de prototipagem para sistemas embarcados.....                                                      | 19 |
| <b>Figura 2</b> - Esquema das partes de um microcontrolador utilizado para controle de elevadores<br>.....                              | 20 |
| <b>Figura 3</b> - Microcontroladores disponíveis comercialmente .....                                                                   | 21 |
| <b>Figura 4</b> - Placa Arduino Mega 2560 Rev3.....                                                                                     | 23 |
| <b>Figura 5</b> - Topologia do Arduino Mega 2560 Rev3 .....                                                                             | 24 |
| <b>Figura 6</b> - Pinagem Arduino Mega 2560 Rev3 parte 1/2 .....                                                                        | 25 |
| <b>Figura 7</b> - Pinagem Arduino Mega 2560 Rev3 parte 2/2 .....                                                                        | 25 |
| <b>Figura 8</b> - Arduino Nano .....                                                                                                    | 26 |
| <b>Figura 9</b> - Pinagem do Arduino Nano .....                                                                                         | 28 |
| <b>Figura 10</b> - Página de download do Arduino IDE.....                                                                               | 29 |
| <b>Figura 11</b> - Interface do usuário do <i>software</i> Arduino IDE versão 2.3.2 .....                                               | 29 |
| <b>Figura 12</b> - Ferramentas mais utilizadas no Arduino IDE.....                                                                      | 30 |
| <b>Figura 13</b> - Esquema de conexão I2C entre Arduinos.....                                                                           | 32 |
| <b>Figura 14</b> – (a) Esquema de conexão do sensor indutivo com Arduino Mega 2560 Rev3, (b)<br>Sensor indutivo modelo LJ12A3-4-Z ..... | 33 |
| <b>Figura 15</b> - Esquema do par termoeletrico e o termopar tipo K.....                                                                | 34 |
| <b>Figura 16</b> - Módulo Max6675 .....                                                                                                 | 35 |
| <b>Figura 17</b> - Conexão entre Arduino Nano, módulo Max6675 e termopar tipo K .....                                                   | 36 |
| <b>Figura 18</b> - Tela LCD TFT .....                                                                                                   | 37 |
| <b>Figura 19</b> - Push button .....                                                                                                    | 38 |
| <b>Figura 20</b> - Esquema de ligação do botão com resistor de pull-down.....                                                           | 38 |
| <b>Figura 21</b> - Esquema de ligação do botão com resistor de pull-up .....                                                            | 39 |
| <b>Figura 22</b> - Placa de circuito impresso de fase simples.....                                                                      | 40 |
| <b>Figura 23</b> - Placa fenolite .....                                                                                                 | 40 |
| <b>Figura 24</b> - Composição do sistema de monitoramento eletrônico .....                                                              | 41 |
| <b>Figura 25</b> - Projeto do circuito regulador de tensão para alimentação do Arduino.....                                             | 42 |
| <b>Figura 26</b> - Esquema de conexão entre os elementos da placa secundária do sistema de<br>monitoramento.....                        | 43 |
| <b>Figura 27</b> - Esquema de conexão entre os elementos da placa principal do sistema de<br>monitoramento.....                         | 44 |

|                                                                                                                |    |
|----------------------------------------------------------------------------------------------------------------|----|
| <b>Figura 28</b> - Fluxograma da programação referente a placa secundaria.....                                 | 45 |
| <b>Figura 29</b> - Fluxograma da programação referente a placa principal.....                                  | 49 |
| <b>Figura 30</b> - Sistema para ensaio de temperatura.....                                                     | 51 |
| <b>Figura 31</b> - Montagem do sistema para realização do ensaio de velocidade angular.....                    | 52 |
| <b>Figura 32</b> - Teste de envio e recebimento de dados por comunicação I2C.....                              | 59 |
| <b>Figura 33</b> - Modelo de exibição utilizado.....                                                           | 59 |
| <b>Figura 34:</b> (a) <i>Layout</i> da placa principal, (b) <i>Layout</i> da placa secundária .....            | 61 |
| <b>Figura 35</b> - Visualização 2D e 3D do <i>layout</i> da placa secundária do sistema de monitoramento ..... | 61 |
| <b>Figura 36</b> - Visualização 2D e 3D do <i>layout</i> da placa principal do sistema de monitoramento .....  | 62 |
| <b>Figura 37</b> - Placa de circuito impresso para o sistema de monitoramento secundário.....                  | 63 |
| <b>Figura 38</b> - Placa de circuito impresso para o sistema de monitoramento principal.....                   | 63 |

## LISTA DE TABELAS

|                                                                                                      |    |
|------------------------------------------------------------------------------------------------------|----|
| <b>Tabela 1</b> - Condições de operação recomendadas para o Arduino Mega 2560 Rev3 .....             | 24 |
| <b>Tabela 2</b> - Condições de operação recomendadas para o Arduino Nano .....                       | 27 |
| <b>Tabela 3</b> - Características NA e NF do push button .....                                       | 38 |
| <b>Tabela 4</b> - Faixas de RPM .....                                                                | 53 |
| <b>Tabela 5</b> - Faixas de temperaturas .....                                                       | 54 |
| <b>Tabela 6</b> - Especificações técnicas das placas de circuito impresso principal e secundária.... | 54 |
| <b>Tabela 7</b> - Dados obtidos durante o ensaio de velocidade angular.....                          | 57 |
| <b>Tabela 8</b> - Valores de Erro para o ensaio de RPM .....                                         | 57 |

## LISTA DE GRÁFICOS

|                                                                                                                                                                          |    |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| <b>Gráfico 1</b> - Comparação entre as curvas de aquecimento: banho ultratermostático, sistema de medição e termômetro MT-450.....                                       | 55 |
| <b>Gráfico 2</b> - Variação do erro de temperatura do sistema de Monitoramento e termômetro MT-450 em relação aos valores de referência do banho ultratermostático ..... | 56 |
| <b>Gráfico 3</b> - Variação do erro de RPM em função da quantidade de ímãs para diferentes velocidades de rotação.....                                                   | 58 |

## LISTA DE ABREVIATURAS E SIGLAS

|       |                                                      |
|-------|------------------------------------------------------|
| RISC  | – <i>Reduced Instruction Set Computing</i>           |
| SAE   | – <i>Society of Automotive Engineers</i>             |
| RPM   | – Rotações por minuto                                |
| LED   | – Diodo emissor de luz                               |
| CI    | – Circuito integrado                                 |
| PIC   | – <i>Programmable Interface Controller</i>           |
| CPU   | – <i>Central processing unit</i>                     |
| RAM   | – <i>Random Access Memory</i>                        |
| ROM   | – <i>Read only memory</i>                            |
| AVR   | – <i>Advanced virtual RISC</i>                       |
| ARM   | – <i>Advanced RISC Machine</i>                       |
| IDE   | – <i>Integrated Development Environment</i>          |
| UARTs | – <i>Universal Asynchronous Receiver/Transmitter</i> |
| ICSP  | – <i>In-Circuit Serial Programming</i>               |
| GND   | – <i>Ground</i>                                      |
| SCL   | – <i>Serial clock line</i>                           |
| SDA   | – <i>Serial data line</i>                            |
| ADC   | – Conversor analógico-digital                        |
| I2C   | – Inter Integrated Circuit                           |
| NA    | – Normalmente aberto                                 |
| NF    | – Normalmente fechado                                |
| LCD   | – <i>Liquid Crystal Display</i>                      |
| TFT   | – <i>Thin Film Transistor</i>                        |
| PCI   | – Placa de circuito impresso                         |
| Km/h  | – Quilômetros por hora                               |
| CVT   | – Transmissão contínua variável                      |
| I/O   | – Entrada/Saída                                      |

## SUMÁRIO

|                                                                                                              |           |
|--------------------------------------------------------------------------------------------------------------|-----------|
| <b>1. INTRODUÇÃO .....</b>                                                                                   | <b>16</b> |
| <b>2. OBJETIVOS .....</b>                                                                                    | <b>18</b> |
| <b>2.1 Objetivos Geral .....</b>                                                                             | <b>18</b> |
| <b>2.2 Objetivos Específicos .....</b>                                                                       | <b>18</b> |
| <b>3. REFERENCIAL TEÓRICO .....</b>                                                                          | <b>19</b> |
| <b>3.1 Sistemas embarcados e prototipagem .....</b>                                                          | <b>19</b> |
| <b>3.2 Microcontroladores .....</b>                                                                          | <b>20</b> |
| <b>3.3 Arduino .....</b>                                                                                     | <b>21</b> |
| 3.3.1 Arduino Mega 2560 Rev3 .....                                                                           | 22        |
| 3.3.2 Arduino Nano .....                                                                                     | 26        |
| 3.3.3 Arduino IDE .....                                                                                      | 28        |
| <b>3.4 Comunicação I2C.....</b>                                                                              | <b>31</b> |
| <b>3.5 Sensores.....</b>                                                                                     | <b>32</b> |
| 3.5.1 Sensor indutivo LJ12A3-4-Z.....                                                                        | 33        |
| 3.5.2 Termopar Tipo K e módulo Max6675.....                                                                  | 34        |
| <b>3.6 Módulo de tela LCD TFT.....</b>                                                                       | <b>36</b> |
| <b>3.7 <i>Push button</i> na configuração <i>pull-down</i> ou <i>pull-up</i> .....</b>                       | <b>37</b> |
| <b>3.8 Placa de circuito impresso.....</b>                                                                   | <b>39</b> |
| <b>4. METODOLOGIA .....</b>                                                                                  | <b>41</b> |
| <b>4.1 Sistema eletrônico de monitoramento .....</b>                                                         | <b>41</b> |
| 4.1.1 Projeto e desenvolvimento do <i>layout</i> da placa secundária .....                                   | 42        |
| 4.1.2 Projeto e desenvolvimento do <i>layout</i> da placa principal .....                                    | 43        |
| <b>4.2 Programação da placa secundária .....</b>                                                             | <b>44</b> |
| <b>4.3 Programação da placa principal do sistema de monitoramento .....</b>                                  | <b>48</b> |
| <b>4.4 Ensaio de temperatura .....</b>                                                                       | <b>51</b> |
| <b>4.5 Ensaio de velocidade angular.....</b>                                                                 | <b>52</b> |
| <b>4.6 Teste de envio de dados por conexão I2C .....</b>                                                     | <b>53</b> |
| <b>4.7 Características técnicas escolhidas para a confecção das placas do sistema de monitoramento .....</b> | <b>54</b> |
| <b>5. RESULTADOS E DISCUSSÕES .....</b>                                                                      | <b>55</b> |
| <b>5.1 Resultado do ensaio de temperatura .....</b>                                                          | <b>55</b> |
| <b>5.2 Resultados de velocidade angular .....</b>                                                            | <b>56</b> |

|     |                                                                              |    |
|-----|------------------------------------------------------------------------------|----|
| 5.3 | Teste de envio de dados por conexão I2C .....                                | 58 |
| 5.4 | Modelo de exibição do módulo <i>display</i> LCD TFT .....                    | 59 |
| 5.5 | Confecção das placas primaria e secundaria do sistema de monitoramento ..... | 60 |
| 6.  | CONSIDERAÇÕES FINAIS .....                                                   | 64 |
|     | REFERÊNCIAS BIBLIOGRÁFICAS .....                                             | 65 |



## 1. INTRODUÇÃO

A tecnologia está presente em todos os setores industriais da sociedade atual, como nas indústrias metalúrgica, agropecuária, petroquímica e automobilística. Veículos modernos possuem inúmeros sensores (como os de velocidade, pressão dos pneus e nível de combustível) e tecnologias avançadas (como conectividade com a internet, sistemas de propulsão, iluminação inteligente e assistência ao estacionamento) que ajudam o motorista a gerenciar o veículo corretamente e a garantir maior segurança.

Todos os recursos tecnológicos de um veículo são controlados por um sistema embarcado, que é um sistema computacional projetado para executar tarefas específicas com alta precisão e velocidade. Esses sistemas utilizam a arquitetura de processadores do tipo RISC (*Reduced Instruction Set Computing*), que se destaca pelo desempenho eficiente, baixo consumo de energia e um conjunto simplificado de instruções (Toniolo, 2018). A importância dos sistemas embarcados também se aplica a projetos acadêmicos e competições automotivas, como o Baja SAE, que visa preparar estudantes para o desenvolvimento de tecnologias voltadas para o setor automotivo.

O Projeto Baja SAE (*Society of Automotive Engineers*) foi criado na Universidade da Carolina do Sul, nos Estados Unidos, e teve sua primeira competição no ano de 1976. Em 1994, teve-se início ao projeto SEA Brasil, seguido pela primeira competição nacional em 1995, realizada na cidade de São Paulo. Atualmente, as competições se dividem em regionais (Nordeste, Sudeste e Sul), uma nacional que reúne todas as instituições que possuem o projeto Baja SAE no Brasil e uma competição internacional (Baja SAE Brasil, 2022). Ao participar do programa Baja SAE, os alunos que fazem parte da equipe devem desenvolver, construir e testar um veículo *off-road* (Baja SAE Brasil, 2022).

Além de todos os sistemas mecânicos, como direção, suspensão, chassi, transmissão e freio, é crucial desenvolver, construir e testar um sistema eletrônico de monitoramento para garantir o desempenho adequado e a segurança do veículo *off-road* tipo baja, assim como de seu piloto. Esse sistema eletrônico é responsável por coletar e fornecer informações vitais sobre o funcionamento do veículo durante as corridas e atividades de campo em tempo real. Isso inclui monitorar a temperatura em graus celsius, e a velocidade angular em rotações por minuto (RPM) da transmissão continuamente variável (CVT) e a velocidade do carro em quilômetros por hora (Km/h), entre outros parâmetros essenciais.

Este trabalho apresenta o projeto e desenvolvimento de um sistema eletrônico de monitoramento para um veículo *off-road* do tipo baja. O sistema utiliza sensores para medir a velocidade linear do veículo, a velocidade angular e a temperatura do CVT, além de monitorar o acionamento do sistema 4x4. As plataformas de prototipagem Arduino Mega 2560 Rev3 e Arduino Nano foram empregadas para gerenciar os dados e sensores. Para garantir as conexões elétricas dos sensores com essas plataformas, foram projetadas e confeccionadas placas de circuito impresso (PCI) específicas.

## 2. OBJETIVOS

### 2.1 Objetivos Geral

Desenvolver um sistema eletrônico eficiente de monitoramento para veículos *off-road* do tipo baja, capaz de medir a velocidade do veículo, monitorar o acionamento do 4x4, e medir a velocidade angular e a temperatura da CVT.

### 2.2 Objetivos Específicos

- Realizar um levantamento bibliográfico e elaborar de um referencial teórico sobre o tema;
- Desenvolver a programação adequada para o sistema eletrônico de monitoramento;
- Projetar o *layout* da placa de circuito impresso de modo a garantir uma conexão adequada entre os sensores e as placas Arduino empregadas;
- Confeccionar a placa de circuito impresso desenvolvida;
- Monitorar o sistema de tração 4x4 e sinalizar sua ativação;
- Realizar testes de temperatura, velocidade angular e velocidade do veículo para validar tanto o sistema eletrônico quanto a programação desenvolvida.

### 3. REFERENCIAL TEÓRICO

#### 3.1 Sistemas embarcados e prototipagem

Os sistemas embarcados são sistemas computacionais dedicados, encapsulados em dispositivos físicos específicos, projetados para executar funções determinadas ou controlar operações em tempo real. Esses sistemas são desenvolvidos para serem eficientes em termos de consumo de energia, mas possuem recursos limitados de processamento e memória (Chase, 2017).

Algumas das plataformas de prototipagem mais utilizadas na criação de sistemas embarcados incluem Raspberry Pi, BeagleBone, Arduino e ESP32. No entanto, é importante ressaltar que essas plataformas são mais adequadas para sistemas embarcados simples ou de média complexidade, como automação residencial e dispositivos IoT. Para aplicações que exigem alta confiabilidade e segurança, como em sistemas complexos de aviões e trens, utilizam-se soluções específicas e certificadas, alinhadas às rigorosas normas desses setores. A Figura 1 ilustra algumas das plataformas de prototipagem comumente empregadas em sistemas embarcados de menor complexidade.

**Figura 1** - Exemplos de placas de prototipagem para sistemas embarcados



**Fonte:** Adaptado de Santos, 2021.

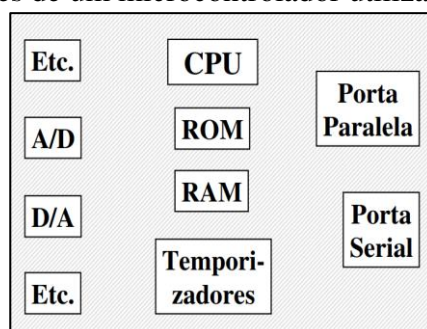
A prototipagem é fundamental na criação de sistemas embarcados, facilitando a integração de microcontroladores e sensores em dispositivos físicos dedicados. Este processo permite desenvolver protótipos funcionais que demonstram as principais características do produto final, sem precisar construir o sistema completo. Além de possibilitar a validação rápida de ideias e simulações de uso real, os protótipos permitem ajustes ágeis durante o desenvolvimento (Melo, 2007). Assim, as placas de prototipagem aceleram a criação de

sistemas embarcados, assegurando que os produtos finais sejam robustos, eficientes e alinhados às necessidades dos usuários.

### 3.2 Microcontroladores

Um microcontrolador é um circuito integrado (CI) que contém um microprocessador simples e dispositivos essenciais para seu funcionamento, necessitando apenas de uma fonte de alimentação externa (Kerschbaumer, 2013). A Figura 2 ilustra os componentes de um microcontrolador utilizado no controle de um elevador, incluindo a CPU (Unidade Central de Processamento), a memória RAM (Memória de Acesso Aleatório), a memória ROM (Memória Somente de Leitura) e conversores de sinais analógicos para digitais (A/D) e digitais para analógicos (D/A). Essa integração oferece diversas vantagens, como a redução do tamanho do microcontrolador, menor consumo de energia, diminuição do tempo de desenvolvimento, maior confiabilidade e facilidade de manutenção (Zelenovsky e Mendonça, 2013).

**Figura 2** - Esquema das partes de um microcontrolador utilizado para controle de elevadores

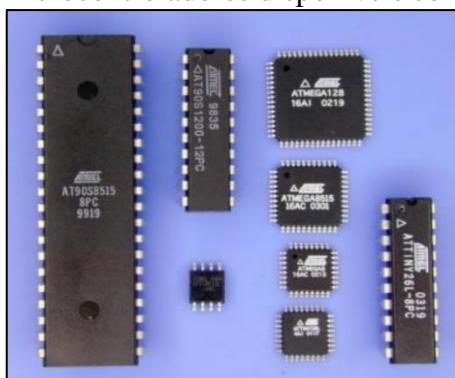


Fonte: Zelenovsky e Mendonça, 2013.

É importante ressaltar que os dispositivos integrados ao microcontrolador variam conforme a aplicação específica. Na Figura 3 apresenta alguns microcontroladores desenvolvidos pela Atmel Corporation.

Os microprocessadores presentes nos microcontroladores não são projetados para operações sofisticadas, como as de um computador pessoal ou *notebook*. Em vez disso, são construídos com a arquitetura RISC, que simplifica o conjunto de instruções, priorizando operações básicas e diretas. Isso resulta em maior eficiência e velocidade na execução das operações. Em contrapartida, os microprocessadores em computadores e notebooks utilizam a arquitetura CISC, que possui um conjunto de instruções mais complexo e diversificado, mas tende a ser mais lenta.

**Figura 3 -** Microcontroladores disponíveis comercialmente



**Fonte:** Kerschbaumer, 2013.

### 3.3 Arduino

O Arduino é uma plataforma de prototipagem eletrônica de código aberto que utiliza um *software* de fácil manuseio, permitindo a criação de diversos projetos interativos. Desenvolvida em 2005 no Ivrea Interaction Design Institute, localizado no norte da Itália, a plataforma foi idealizada pelos pesquisadores Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino e David Mellis (Arduino, 2018).

O Arduino foi inicialmente concebido como uma ferramenta para rápida prototipagem, com o objetivo de atender principalmente pessoas sem formação prévia em eletrônica e programação. Essa plataforma de prototipagem eletrônica é composta por uma placa de circuito impresso que integra um microcontrolador, geralmente utilizando tecnologias AVR (*Advanced Virtual RISC*) ou ARM (*Advanced RISC Machine*). A plataforma de prototipagem eletrônica disponibiliza uma ampla gama de pinos digitais e analógicos, além de suportar protocolos de comunicação serial e outras funcionalidades.

A plataforma Arduino oferece uma variedade de placas, como Arduino Uno, Nano e Mega 2560, que variam de modelos básicos a versões mais avançadas com maior capacidade de processamento e conectividade (Neri, 2014). Além disso, o Arduino é compatível com diversos sensores, atuadores e componentes eletrônicos, permitindo o desenvolvimento de projetos que vão desde o simples piscar de um LED (Diodo emissor de luz) até sistemas automatizados complexos. Segundo Carvalho, Santos e Santos (2023), as principais vantagens do Arduino incluem:

- **Preço:** O preço acessível é um grande atrativo ao escolher o Arduino como plataforma de prototipagem. Em 2024, por exemplo, é possível adquirir um Arduino Uno por menos de R\$ 50.
- **Software Arduino IDE:** O Arduino IDE (*Integrated Development Environment*) é um ambiente de desenvolvimento multiplataforma, baseado na linguagem C/C++, que pode ser instalado em sistemas operacionais Windows, macOS e Linux.
- **Ambiente de programação simples:** O *software* Arduino IDE possui um ambiente de programação intuitivo e de fácil entendimento, ideal para iniciantes que desejam desenvolver seus primeiros projetos.
- **Software de código aberto e extensível:** O Arduino IDE é licenciado em código aberto, permitindo modificações e personalizações. A linguagem C/C++ pode ser expandida por meio da adição de bibliotecas ao Arduino IDE. Novas bibliotecas surgem constantemente, facilitando o uso de sensores e atuadores específicos (Arduino, 2018).
- **Hardware extensível e de código aberto:** Assim como o *software*, o *hardware* Arduino também é licenciado em código aberto, permitindo que projetistas experientes ou iniciantes façam alterações no design físico da placa, adequando-a às necessidades específicas de seus projetos.

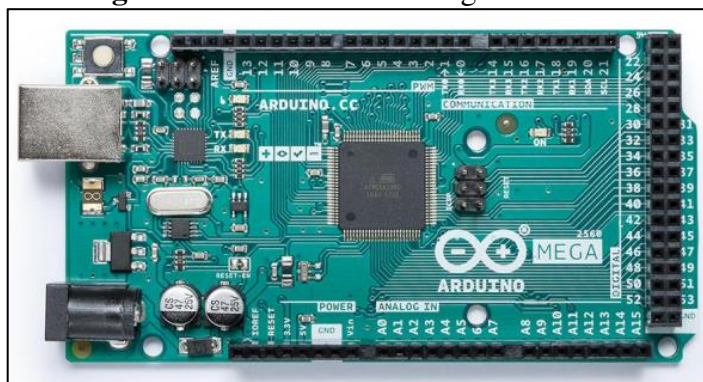
Essas características tornam o Arduino uma plataforma amplamente utilizada em projetos de prototipagem, destacando-se por sua acessibilidade e flexibilidade tanto em *software* quanto em *hardware*.

### 3.3.1 Arduino Mega 2560 Rev3

O Arduino Mega 2560 Rev3 é uma placa de desenvolvimento projetada para aplicações que exigem um número elevado de pinos de entrada/saída (I/O) (*Datasheet* Arduino Mega 2560 Rev3, 2024). Equipado com o microcontrolador ATmega2560, que opera a 16 MHz, essa placa possui 54 pinos digitais e 16 pinos analógicas, tornando-se a placa da família Arduino com o maior número de pinos. Em termos de conectividade, o Arduino Mega 2560 Rev3 conta com 4

UARTs (*Universal Asynchronous Receiver/Transmitter*) para comunicação serial, uma conexão USB, um conector de alimentação, um conector ICSP (*In-Circuit Serial Programming*) e um botão de reinicialização (*Datasheet Arduino Mega 2560 Rev3*, 2024). A Figura 4 ilustra a placa do Arduino Mega 2560.

**Figura 4 - Placa Arduino Mega 2560 Rev3**



**Fonte:** *Datasheet Arduino Mega 2560 Rev3*, 2024.

O ATmega2560 possui uma memória RAM consideravelmente maior em comparação com outros microcontroladores da família Arduino (*Datasheet Arduino Mega 2560 Rev3*, 2024). Além disso, o Arduino Mega 2560 Rev3 é equipado com um processador secundário, o ATmega16U2, que gerencia o processamento da comunicação serial USB. Esse processador atua como interface entre os sinais de entrada USB e o processador principal, aumentando a flexibilidade na conexão de periféricos à placa. A seguir, estão listadas as características do ATmega2560 e do ATmega16U2.

#### 1. Características do ATmega2560

- Taxa de transferência de até 16 MIPS a 16 MHz;
- 256k bytes (dos quais 8k são usados para o *bootloader*);
- EEPROM de 4k bytes;
- 8k bytes SRAM interna;
- Registros de uso geral  $32 \times 8$ ;
- Contador em tempo real com oscilador separado;
- Quatro canais PWM de 8 bits;
- Quatro USART seriais programáveis;
- Interface serial SPI controlador/periférico.

#### 2. Características do ATmega16U2



- Taxa de transferência de até 16 MIPS a 16 MHz;
- Memória Flash ISP de 16k bytes;
- EEPROM de 512 bytes;
- SRAM de 512 bytes;
- USART com modo somente mestre SPI e controle de fluxo de hardware (RTS/CTS);
- Interface serial SPI mestre/escravo.

A Tabela 1 apresenta as condições de operação mínimas, típicas e máximas para tensões e temperaturas suportadas pela placa do Arduino Mega. É importante observar que a tensão de alimentação varia conforme a forma de conexão da placa, seja via USB ou DC Jack. A ultrapassagem dos valores máximos indicados na tabela pode resultar em danos à placa.

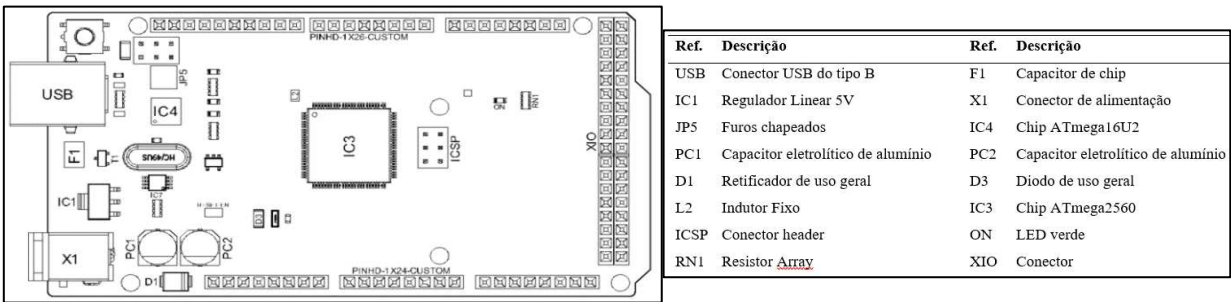
**Tabela 1** - Condições de operação recomendadas para o Arduino Mega 2560 Rev3

| Símbolo   | Descrição                                    | Mínimo | Típico | Máximo | Unidade |
|-----------|----------------------------------------------|--------|--------|--------|---------|
| $V_{IN}$  | Tensão de entrada<br>utilizando o pino Jack. | 7      | 7,0    | 12     | V       |
| $V_{USB}$ | Tensão de entrada do<br>conector USB         | 4,8    | 5,0    | 5,5    | V       |
| $T_{OP}$  | Temperatura de<br>operação                   | -40    | 25     | 85     | °C      |

Fonte: Adaptado de *Datasheet* Arduino Mega 2560 Rev3, 2024.

A Figura 5 ilustra a topologia da placa Arduino Mega 2560 Rev3, destacando os principais componentes. Entre os componentes estão o conector USB, o X1 (pino Jack de alimentação), o IC1 (regulador de tensão responsável por fornecer 5V a pinos específicos) e o LED ON (indicador de funcionamento da placa). Além disso, as marcações IC3 e IC4 identificam, respectivamente, as localizações do ATmega2560 e do ATmega16U2.

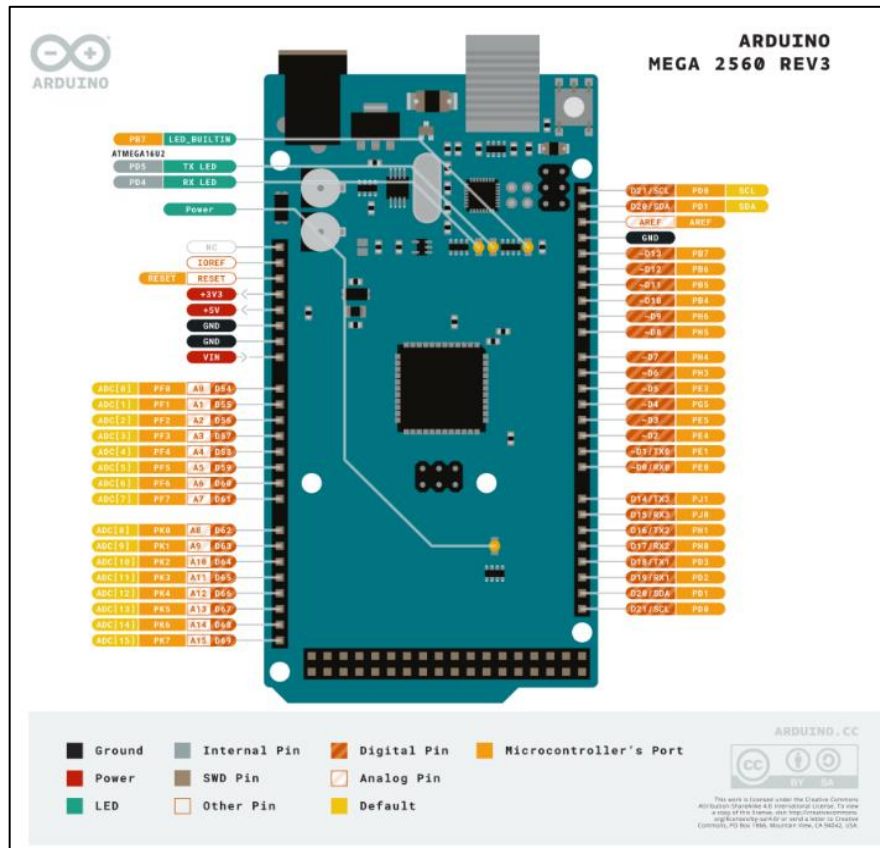
**Figura 5** - Topologia do Arduino Mega 2560 Rev3



Fonte: Adaptado de *Datasheet* Arduino Mega 2560 Rev3, 2024.

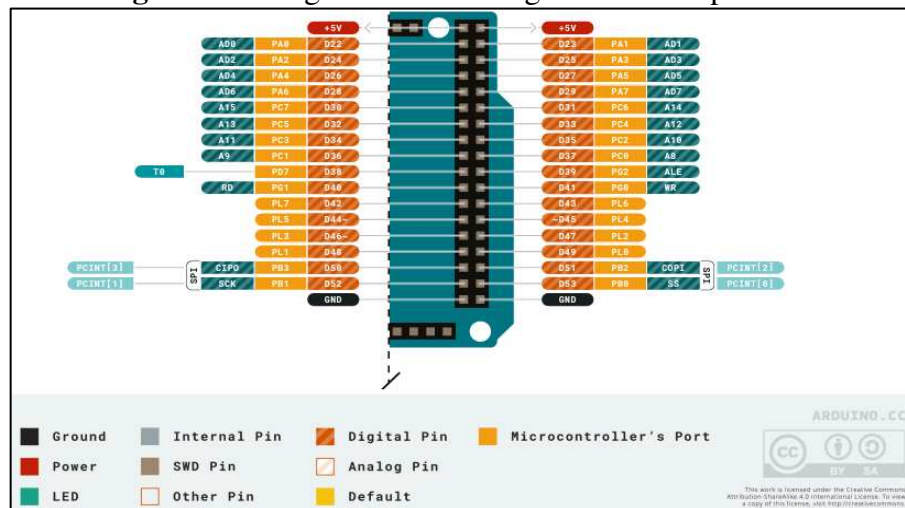
As Figuras 6 e 7 mostram a pinagem completa do Arduino Mega 2560 Rev3. Os pinos digitais podem ser configurados como entradas ou saídas e operam em dois estados distintos: alto (*High*) e baixo (*Low*). O estado alto corresponde a uma tensão próxima de 5V (ou 3,3V, dependendo do modelo), enquanto o estado baixo corresponde a aproximadamente 0V.

**Figura 6 - Pinagem Arduino Mega 2560 Rev3 parte 1/2**



Fonte: Datasheet Arduino Mega 2560 Rev3, 2024.

**Figura 7 - Pinagem Arduino Mega 2560 Rev3 parte 2/2**



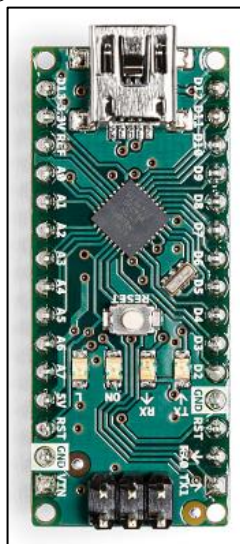
Fonte: Adaptado de Datasheet Arduino Mega 2560 Rev3, 2024.

Os pinos analógicos do Arduino são utilizados principalmente como entradas para ler valores de tensão contínua, que podem variar ao longo do tempo. Esses pinos possuem um conversor analógico-digital (ADC), responsável por transformar a tensão analógica em um valor digital, permitindo que o microcontrolador processe e interprete os dados. A resolução do ADC no Arduino é de 10 bits, o que possibilita representar a tensão de entrada com valores entre 0 e 1023.

### 3.3.2 Arduino Nano

De acordo com o *Datasheet* do Arduino Nano (2024), essa placa é projetada para a construção de protótipos eletrônicos mais rápidos e compactos, oferecendo interfaces adequadas para diversas aplicações. O microcontrolador que gerencia as entradas digitais, entradas analógicas e saídas digitais da placa é o ATmega328, que opera a uma frequência de 16 MHz. O Arduino Nano possui 20 pinos de I/O digitais e 8 pinos analógicos, além de uma porta mini-USB, utilizada para alimentação e transferência de códigos por meio do software Arduino IDE. A Figura 8 ilustra a placa do Arduino Nano.

**Figura 8 - Arduino Nano**



**Fonte:** *Datasheet* Arduino Nano, 2024.

O microcontrolador ATmega328 é um dispositivo de 8 bits que se destaca pelo seu alto desempenho e baixo consumo de energia. O ATmega328 é capaz de se comunicar com diversos dispositivos externos através de protocolos de comunicação serial, como UART TTL (5V), I2C (TWI) e SPI (*Datasheet* Arduino Nano, 2024). O Arduino Nano pode ser programado utilizando

o *software* Arduino IDE, o que facilita a introdução para novos usuários. A seguir, estão listadas algumas das características mais relevantes do ATmega328.

#### 1. Características do ATmega328

- Processador de 8 bits de alto desempenho e baixo consumo;
- Atinge até 16 MIPS para uma frequência de *clock* de 16 MHz;
- 32 KB dos quais 2 KB são usados pelo *bootloader*;
- 2 KB de SRAM interna;
- 1 KB de EEPROM;
- 32 x 8 Registradores de uso geral;
- Contador em tempo real com oscilador separado;
- Seis canais PWM;
- USART serial programável;
- Interface Serial SPI mestre/escravo.

A Tabela 2 apresenta as condições de operação mínimas, típicas e máximas para as tensões e temperaturas suportadas pela placa do Arduino Nano. Assim como no Arduino Mega 2560 Rev3, a tensão de alimentação depende da fonte utilizada, que pode ser o Mini-USB ou o pino de alimentação VIN disponível na placa. É importante ressaltar que ultrapassar os valores máximos especificados na tabela pode causar danos à placa.

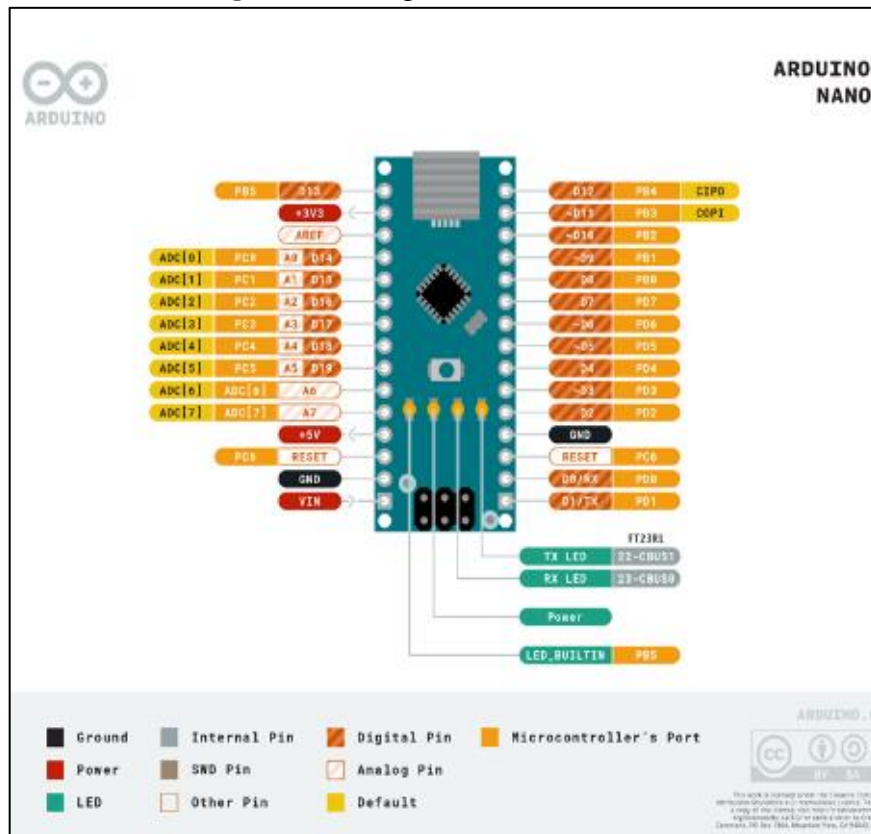
| <b>Tabela 2 - Condições de operação recomendadas para o Arduino Nano</b> |                                              |               |               |               |                |
|--------------------------------------------------------------------------|----------------------------------------------|---------------|---------------|---------------|----------------|
| <b>Símbolo</b>                                                           | <b>Descrição</b>                             | <b>Mínimo</b> | <b>Típico</b> | <b>Máximo</b> | <b>Unidade</b> |
| V <sub>IN</sub>                                                          | Tensão de entrada<br>utilizando o pino Jack. | 7             | --            | 15            | V              |
| V <sub>USB</sub>                                                         | Tensão de entrada do<br>conector USB         | --            | 5V            | --            | V              |
| T <sub>OP</sub>                                                          | Temperatura de<br>operação                   | -40           | --            | 85            | °C             |

**Fonte:** Adaptado de *Datasheet* Arduino Nano, 2024.

A pinagem completa do Arduino Nano pode ser visualizada na Figura 9. Os pinos analógicos são numerados de A0 a A7, enquanto os pinos digitais são numerados de D0 a D12. Os pinos de alimentação fornecem tensões de 5V e 3,3V. O pino VIN é destinado à tensão de entrada (alimentação), que, conforme a Tabela 2, pode variar entre 7 e 15 V. No entanto, testes

realizados em laboratório demonstraram que a placa também funciona adequadamente com uma tensão de entrada de 5V.

**Figura 9 - Pinagem do Arduino Nano**



Fonte: Adaptado *Datasheet* Arduino Nano, 2024.

### 3.3.3 Arduino IDE

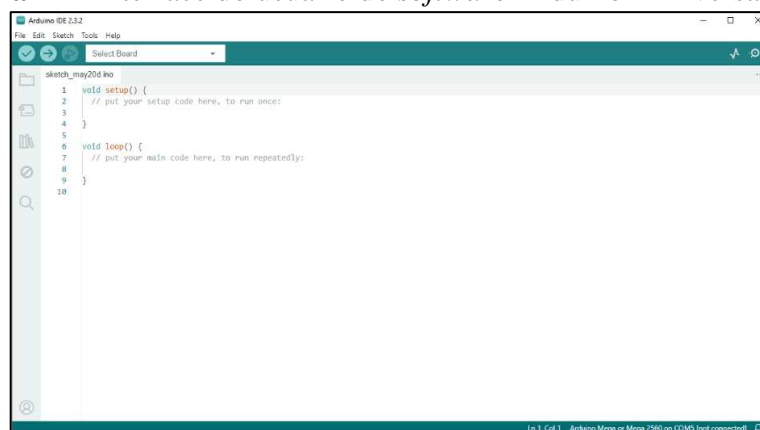
O *software* Arduino IDE é um ambiente de desenvolvimento integrado projetado para programar microcontroladores da plataforma Arduino, mas também pode ser utilizado com outras plataformas. Ele oferece uma interface simples, o que facilita o uso por iniciantes que desejam construir seus projetos. Ao mesmo tempo, é uma ferramenta completa que atende às necessidades de programadores mais experientes. Por ser um *software* de código aberto, o Arduino IDE pode ser modificado para se adaptar às necessidades específicas do usuário.

O Arduino IDE, mostrado na Figura 10, pode ser baixado diretamente do site oficial da Arduino (<https://www.arduino.cc/en/software>). Este *software* está disponível em versões para Windows, Linux e macOS.

**Figura 10 -** Página de download do Arduino IDE

**Fonte:** Autoria própria, 2024.

A Figura 11 ilustra a interface do usuário do Arduino IDE na versão 2.3.2. A programação é realizada no espaço numerado, onde estão as funções fundamentais que compõem qualquer *sketch* (programa) no Arduino IDE, como *void setup* e *void loop*. A função *void setup* é executada uma única vez e é utilizada para realizar as configurações iniciais do programa, como a definição de comunicação, configuração de pinos digitais e pinos analógicos, entre outros. Há uma ampla gama de códigos que podem ser inseridos no *void setup*. Em contrapartida, o *void loop* contém o programa principal, que deve ser executado continuamente. Após cada ciclo de execução, essa função é reiniciada, e esse processo se repete enquanto a placa Arduino estiver ligada. Para o desenvolvimento dos *sketches*, utiliza-se a linguagem C/C++ com o *software* Arduino IDE.

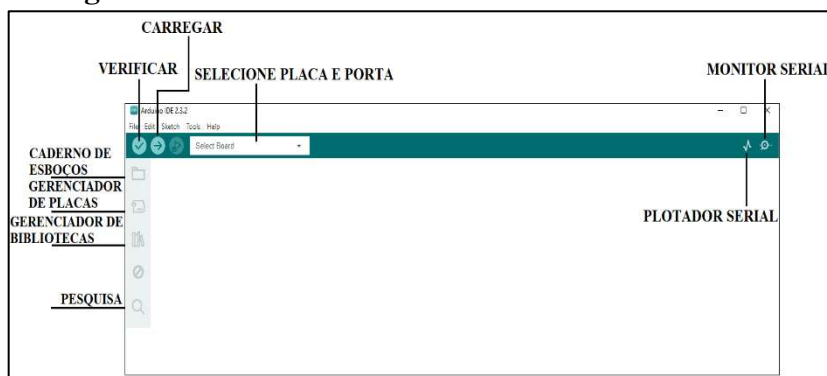
**Figura 11 -** Interface do usuário do *software* Arduino IDE versão 2.3.2

**Fonte:** Autoria própria, 2024.

A Figura 12 destaca alguns atalhos de ferramentas importantes disponíveis na interface do usuário do Arduino IDE. Entre esses atalhos, estão: Verificar, Carregar, Selecionar Placa e

Porta, Caderno de Esboços, Gerenciador de Placas, Gerenciador de Bibliotecas, Pesquisa, Monitor Serial e Plotador Serial.

**Figura 12 - Ferramentas mais utilizadas no Arduino IDE**



**Fonte:** Autoria própria, 2024.

A ferramenta “Verificar” serve para compilar o código, analisando todas as linhas em busca de erros de sintaxe. Se algum erro for encontrado, a ferramenta indica sua localização no código e sugere correções. Caso não haja erros, o programa estará pronto para ser carregado no Arduino (Söderby e Hylén, 2024c). Por sua vez, a ferramenta “Carregar” é utilizada para transferir o código que será executado pelo microcontrolador presente na placa Arduino (Söderby e Hylén, 2024c).

A ferramenta “Selecione Placa e Porta” é essencial para o funcionamento adequado do Arduino IDE. Antes de compilar e carregar o programa, é necessário identificar qual modelo de placa Arduino está sendo utilizado, uma vez que a pinagem pode variar entre os diferentes modelos. Além disso, é fundamental informar a porta de conexão USB do computador à qual a placa está conectada, pois isso permite que a transferência do código escrito seja realizada corretamente utilizando a ferramenta de carregamento (Söderby e Hylén, 2024c).

A ferramenta “Caderno de Esboços” possibilita o acesso a todos os códigos armazenados localmente no computador. Além disso, a ferramenta permite a sincronização com o Arduino *Cloud*, facilitando a obtenção dos códigos no ambiente online (Söderby e Hylén, 2024c).

A ferramenta “Gerenciador de Placas” é utilizada para instalar diferentes pacotes de placas disponíveis no computador. Cada pacote é projetado para microcontroladores específicos. O Arduino oferece uma variedade de placas, cada uma com diferentes tipos de microcontroladores. Embora esses microcontroladores executem tarefas de maneiras



semelhantes, a compilação do código pode variar, pois os pinos são mapeados de forma diferente e os recursos disponíveis podem ser distintos (Söderby e Hylén, 2024c).

A ferramenta “Gerenciador de Bibliotecas” oferece uma das principais vantagens de programar em C/C++: a inclusão de bibliotecas. Essas bibliotecas facilitam a utilização de sensores, atuadores e displays de maneira simples. O Gerenciador de Bibliotecas permite buscar, baixar e instalar diversas bibliotecas para complementar os projetos (Söderby e Hylén, 2024c).

A ferramenta “Pesquisa” permite realizar buscas no código escrito na área de programação (Söderby e Hylén, 2024c). Por sua vez, a ferramenta “Monitor Serial” é uma interface textual que possibilita a comunicação direta entre o computador e o Arduino. Ela é utilizada para enviar e receber dados de forma serial através da porta USB que conecta o Arduino ao computador (Söderby, 2024a).

A ferramenta “Plotador Serial” é uma interface gráfica que possibilita a visualização em tempo real dos dados transmitidos pelo Arduino. Essa ferramenta é especialmente útil para interpretar variáveis que mudam ao longo do tempo, como leituras de sensores, sinais analógicos e outros dados dinâmicos (Söderby, 2024b).

### 3.4 Comunicação I2C

A comunicação I2C ou  $I^2C$  (*Inter Integrated Circuit*) é um protocolo desenvolvido pela empresa NXP *Semiconductors* (anteriormente conhecida como Philips *Semiconductors*) na década de 1980. Trata-se de uma comunicação serial síncrona, que permite a interconexão de diversos dispositivos em um único barramento. Com esse protocolo, é possível conectar múltiplos sensores aos mesmos pinos de uma placa Arduino. O único requisito para garantir uma comunicação eficaz é que cada sensor conectado possua um endereço único.

Além disso, a comunicação I2C permite conectar várias placas Arduino diferentes, possibilitando que uma delas envie dados para uma placa específica. Essa configuração é conhecida como mestre e escravo (Dannegård, 2022).

O protocolo I2C utiliza dois pinos para comunicação em um Arduino: o pino SDA (*Serial Data Line*) e o pino SCL (*Serial Clock Line*). Ambos os pinos oferecem comunicação bidirecional, permitindo que enviem e recebam dados de todos os dispositivos conectados ao barramento. O pino SDA transporta os dados a serem enviados, enquanto o pino SCL é responsável por transmitir os pulsos de clock, que garantem a sincronização dos dados, assegurando que cheguem na ordem correta (NPX Semiconductors, 2021).

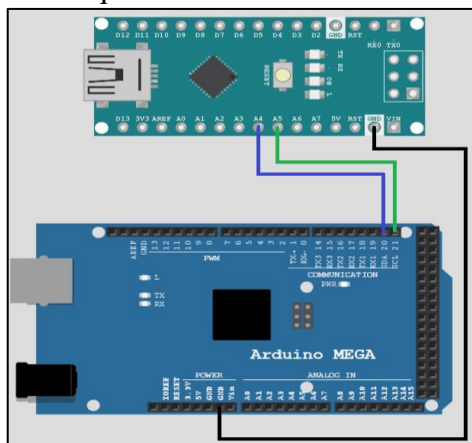


Quando duas placas Arduino são conectadas ao barramento I2C, uma delas é classificada como mestre e a outra como escravo. O dispositivo mestre é responsável por gerar os pulsos de *clock* e iniciar a comunicação entre os dispositivos, solicitando os dados que devem ser enviados a ele. Por sua vez, o dispositivo escravo responde aos comandos do mestre, enviando os conjuntos de dados solicitados.

Nessa configuração, é comum utilizar apenas um Arduino como mestre, embora seja possível ter múltiplos Arduinos como escravos. Para isso, é essencial definir corretamente o endereço de cada escravo na programação de todos os Arduinos envolvidos no projeto. A Figura 13 ilustra a conexão I2C entre um Arduino Mega 2560 Rev3 e um Arduino Nano.

Para estabelecer a comunicação I2C entre um Arduino Mega 2560 Rev3 e um Arduino Nano, deve-se conectar os pinos SDA e SCL de um Arduino aos pinos correspondentes do outro. No caso do Arduino Mega 2560 Rev3, os pinos SDA e SCL são, respectivamente, os pinos digitais 20 e 21. Já no Arduino Nano, esses pinos correspondem aos pinos analógicos A4 e A5. Além disso, é crucial conectar os terminais GND de ambos os Arduinos, pois referências de tensão diferentes podem resultar em transferências de dados não confiáveis.

**Figura 13** - Esquema de conexão I2C entre Arduinos



Fonte: Autoria Própria, 2024.

### 3.5 Sensores

Os sensores são componentes essenciais e amplamente integrados na sociedade contemporânea, encontrados em diversos dispositivos, como balanças, termômetros, automóveis, sistemas de iluminação, computadores e celulares. Esses dispositivos desempenham um papel crucial ao medir e detectar variações em diferentes grandezas físicas dentro de um ambiente ou sistema. A capacidade dos sensores de transformar essas mudanças

detectadas em um sinal elétrico de saída proporcional à alteração é fundamental para o funcionamento eficiente dos equipamentos modernos, permitindo a coleta e o processamento de dados de forma analógica e digital em tempo real.

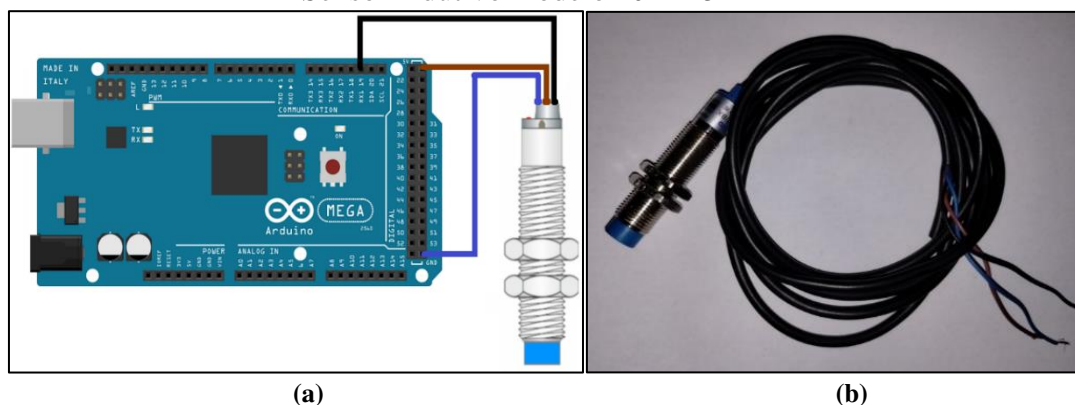
Existem diferentes tipos de sensores, sendo os analógicos e digitais os mais comuns. Os sensores analógicos geram um sinal de saída contínuo, capaz de assumir qualquer valor dentro de um intervalo específico. Isso permite uma representação mais detalhada das variações, sendo especialmente útil em aplicações que requerem precisão. Por outro lado, os sensores digitais operam com níveis de tensão pré-determinados e utilizam conceitos de lógica binária, convertendo as informações em dígitos que representam estados discretos.

Essa distinção entre os tipos de sensores é fundamental, pois determina a forma como os dados são interpretados e utilizados em diversas aplicações tecnológicas, desde o controle de processos industriais até o uso cotidiano em dispositivos eletrônicos (Zeymer et al., 2021). Essa dualidade nos sensores ilustra não apenas a complexidade dos sistemas modernos, mas também a importância de cada tipo na adaptação às necessidades específicas da sociedade atual.

### 3.5.1 Sensor indutivo LJ12A3-4-Z

O sensor indutivo, ilustrado na Figura 14 (b), possui três fios de ligação: um marrom (VCC), um azul (GND) e um preto (sinal). Para utilizá-lo com uma placa Arduino, deve-se conectar o cabo azul ao pino GND, o cabo marrom ao pino de 5V e o cabo preto ao pino digital desejado. Embora o sensor indutivo opere dentro da faixa de 6V a 36V, testes realizados em laboratório demonstraram que ele funciona corretamente quando conectado ao pino de 5V da placa Arduino.

**Figura 14** – (a) Esquema de conexão do sensor indutivo com Arduino Mega 2560 Rev3, (b) Sensor indutivo modelo LJ12A3-4-Z



Fonte: Autoria própria, 2024.

A Figura 14 (a) ilustra o esquema de conexão entre o sensor indutivo e a placa Arduino Mega 2560 Rev3. Neste esquema, o sensor é alimentado com 5V, e a conexão de sinal está ligada ao pino digital 19. Quando não há presença de metal em seu alcance (de 2 a 4 milímetros), o sensor envia um sinal de nível alto para o pino digital 19. Assim que um objeto metálico (ou ímã) entra em seu alcance, o sensor emite um nível lógico baixo para a porta digital.

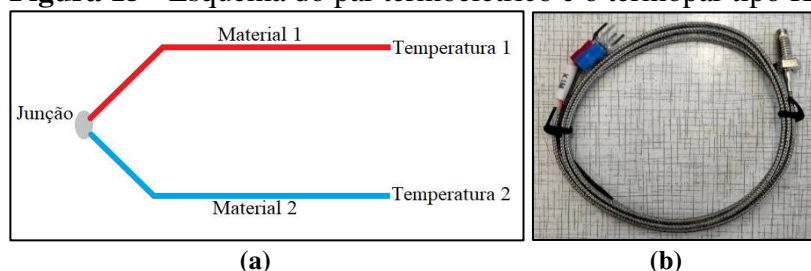
### 3.5.2 Termopar Tipo K e módulo Max6675

A temperatura é uma grandeza física que mensura o grau de agitação térmica das partículas de um corpo ou sistema. De maneira mais simples, a temperatura é utilizada para avaliar o quão quente ou frio algo está. A temperatura pode ser medida em várias escalas, sendo as mais comuns as escalas Celsius, Fahrenheit e Kelvin. A escala Kelvin é considerada a escala absoluta, pois não apresenta temperaturas negativas (Sobrinho e Souza, 2006).

Em computadores, celulares, circuitos embarcados e em muitos outros dispositivos, a temperatura de operação é um fator crítico que afeta diretamente o desempenho, a eficiência e a vida útil de cada um deles. Portanto, o monitoramento da temperatura é fundamental. Existem diversos dispositivos capazes de medir a temperatura de um corpo, objeto ou ambiente, como termômetros, pirômetros, termistores e termopares. A escolha do dispositivo adequado depende, no entanto, das características específicas do projeto.

O sensor termopar pode ser utilizado juntamente com a plataforma Arduino. Ao unir as extremidades de dois materiais condutores ou semicondutores distintos, forma-se uma conexão conhecida como par termoelétrico, conforme ilustrado na Figura 15 (a). Quando o termopar é aquecido, uma diferença de potencial elétrico (DDP) surge nas extremidades não conectadas, resultando da diferença de temperatura entre os materiais que compõem o termopar (Neto, 2018). Essa relação entre a diferença de temperatura de dois materiais e a intensidade da diferença de potencial gerada é conhecida como efeito Seebeck.

**Figura 15** - Esquema do par termoelétrico e o termopar tipo K

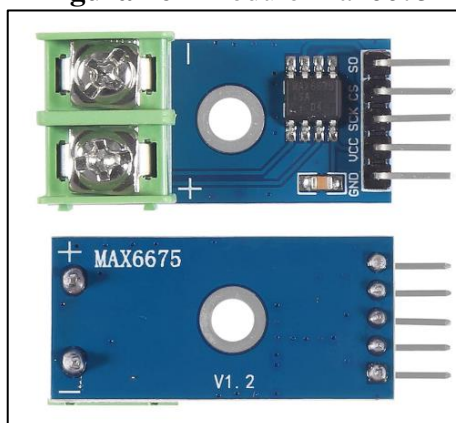


(a) (b)  
Fonte: Autoria Própria, 2024.

Existem diferentes modelos de termopar, e um deles é o tipo K. Este sensor é formado por dois materiais distintos: o Cromel (90% níquel e 10% cromo), que atua como eletrodo positivo, e o Alumel (95% níquel e 5% alumínio), que funciona como eletrodo negativo. O termopar tipo K, ilustrado na Figura 15(b), apresenta uma ampla faixa de operação, elevada resistência à corrosão e pode ser utilizado em diversos ambientes.

A utilização do termopar tipo K deve ser acompanhada por um dispositivo capaz de converter a diferença de potencial gerada pela variação de temperatura em um valor correspondente em graus Celsius. Por essa razão, o termopar é frequentemente utilizado em conjunto com o módulo MAX6675 em projetos que envolvem placas Arduino, conforme ilustrado na Figura 16.

**Figura 16 - Módulo Max6675**



**Fonte:** Circuito fácil, 2019.

De acordo com o *datasheet* da Maxim Integrated Products (2021), o módulo MAX6675 é capaz de medir temperaturas que variam de 0°C a 1024°C utilizando um termopar tipo K. O módulo opera eficientemente em uma faixa de temperatura de -20°C a 85°C, sendo sensível a ambientes fora dessa faixa, o que pode resultar em danos permanentes. A tensão de alimentação requerida varia de 3V a 5,5V.

O módulo MAX6675 possui cinco terminais de conexão: GND, VCC, SCK, CS e SO, além de dois terminais adicionais, um positivo e um negativo, para a conexão do termopar.

O terminal "VCC" é destinado à alimentação do módulo e deve ser conectado a uma fonte de energia com tensão entre 3V e 5,5V. O terminal "GND" é o pino de terra, responsável por estabelecer a referência de potencial do circuito.

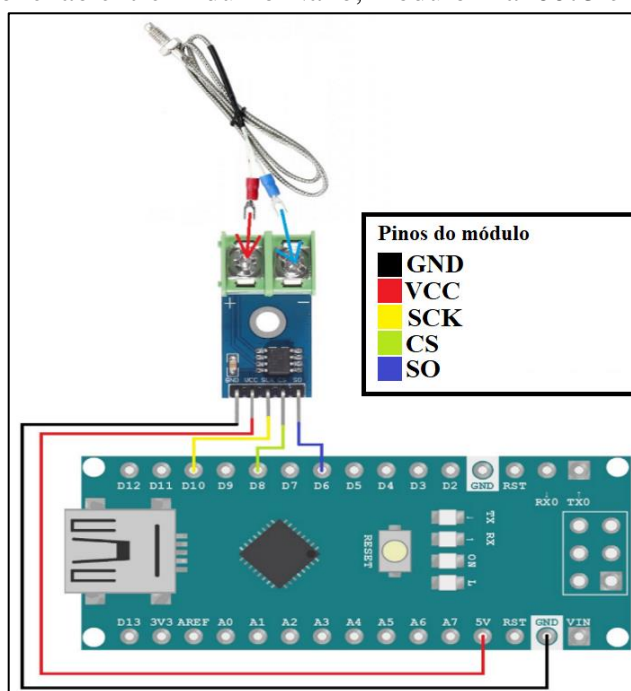
O terminal "SO" (*Serial Out*) é a linha de saída de dados, pela qual o MAX6675 envia as leituras de temperatura para o microcontrolador. O terminal "CS" (*Chip Select*) é utilizado

para seleccionar o dispositivo no barramento SPI, controlando a comunicação entre o módulo e o microcontrolador. Por sua vez, o terminal “SCK” (*Serial Clock*) é responsável por fornecer o sinal de clock, sincronizando a comunicação, e é gerado pelo microcontrolador.

Os terminais "positivo" e "negativo" do módulo são destinados à conexão dos terminais correspondentes do termopar tipo K. O terminal positivo do termopar deve ser conectado ao terminal positivo do módulo, enquanto o terminal negativo deve ser ligado ao terminal negativo. Essa conexão correta é fundamental para garantir leituras precisas da temperatura.

Os pinos SCK, CS e SO do módulo MAX6675 devem ser conectados aos pinos digitais da placa Arduino. A Figura 17 ilustra a conexão entre o módulo MAX6675, o termopar tipo K e a placa Arduino Nano. Para que o sistema funcione corretamente, é necessário baixar e instalar a biblioteca max6675.h no Arduino IDE.

**Figura 17** - Conexão entre Arduino Nano, módulo Max6675 e termopar tipo K



Fonte: Autoria própria, 2024.

### 3.6 Módulo de tela LCD TFT

Em diversos projetos, é essencial visualizar os valores das grandezas monitoradas. No Arduino, essa visualização pode ser feita por meio do *software* Arduino IDE utilizando as ferramentas “Monitor Serial” ou “Plotador Serial”. No entanto, em muitas aplicações, não é

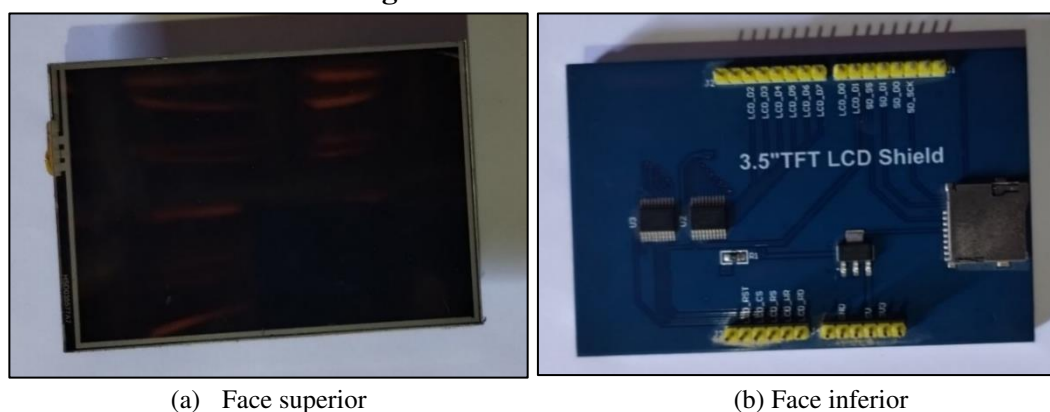
viável manter um computador permanentemente conectado à placa Arduino. Portanto, é necessário utilizar algum periférico de saída para exibir os dados monitorados.

O módulo de tela LCD (*Liquid Crystal Display*) TFT (*Thin Film Transistor*) é amplamente utilizado como um monitor para exibir ao usuário os dados medidos, como temperatura, pressão, velocidade e mensagens, entre outros.

A Figura 18 ilustra a face superior e inferior de um módulo de tela LCD TFT de 3,5 polegadas, ele possui um *display* com resolução de 480 x 320. O módulo suporta alta definição (HD) e possui um *slot* para cartão de memória, permitindo o armazenamento de dados e imagens para exibição posterior.

Os pinos de conexão estão localizados na face inferior do módulo, que opera com uma tensão de 5V. Esse módulo foi desenvolvido especificamente para a placa Arduino Mega 2560 Rev3, e sua conexão deve ser realizada com atenção, alinhando corretamente os pinos de alimentação (5V e 3,3V) com os pinos correspondentes da placa Arduino.

**Figura 18 - Tela LCD TFT**



(a) Face superior

(b) Face inferior

*Fonte: Autoria própria, 2024.*

### 3.7 *Push button* na configuração *pull-down* ou *pull-up*

Um botão funciona como uma chave mecânica em circuitos eletrônicos, permitindo ou bloqueando a passagem da corrente elétrica. O *push button* (botão de apertar) pode ser do tipo normalmente aberto (NA), onde seus contatos ficam abertos quando não pressionado e se fecham ao serem acionados. Alternativamente, pode ser do tipo normalmente fechado (NF), que mantém os contatos fechados até ser pressionado, momento em que se abrem. Amplamente utilizado em automação e projetos com Arduino, o *push button* possibilita iniciar ou interromper processos, acionar alarmes e executar funções pré-programadas. A Figura 19 ilustra um botão

*push button*, enquanto a Tabela 3 apresenta as características de funcionamento dos modelos NA e NF.

Figura 19 - *Push button*



Fonte: Autoria própria, 2024.

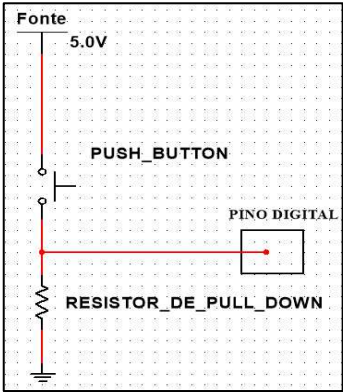
Tabela 3 - Características NA e NF do *push button*

| <i>Push button</i> | Não pressionado | Pressionado |
|--------------------|-----------------|-------------|
| NA                 |                 |             |
| NF                 |                 |             |

Fonte: Autoria própria, 2024.

Para usar um *push button* em um Arduino, é necessário definir o tipo de ligação do botão, ou seja, se ele será conectado a um resistor de *pull-down* ou *pull-up*, escolha que depende da aplicação. Um resistor de *pull-down*, por exemplo, é ligado entre o pino do botão e o GND do Arduino, garantindo que o pino digital permaneça em nível baixo (0V) enquanto o botão (NA) não é pressionado, pois toda a corrente flui para o GND através do resistor. Quando o botão é pressionado, o pino do botão se conecta diretamente aos 5V do Arduino, levando o pino digital ao nível alto (5V), como ilustrado na Figura 20.

Figura 20 - Esquema de ligação do botão com resistor de *pull-down*

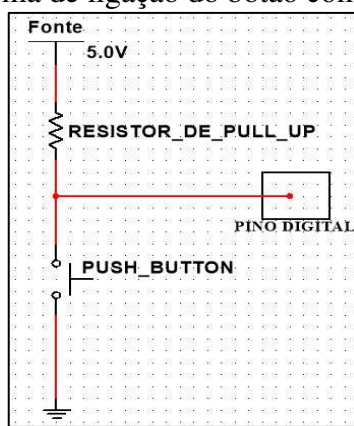


Fonte: Autoria própria, 2024.



Na configuração com resistor *pull-up*, o resistor é ligado entre o pino do botão NF e o pino de 5V do Arduino, conforme ilustrado na Figura 21. Dessa forma, quando o botão NA não está pressionado, o pino do botão e o pino digital conectado a ele permanecem em nível alto (5V), pois a corrente flui da fonte de alimentação para o pino digital do Arduino. Ao pressionar o botão, ele conecta o pino do botão diretamente ao GND do Arduino, direcionando toda a corrente para o GND. Esse desvio faz com que a entrada digital do Arduino receba um nível lógico baixo (0V).

**Figura 21** - Esquema de ligação do botão com resistor de *pull-up*



Fonte: Autoria própria, 2024.

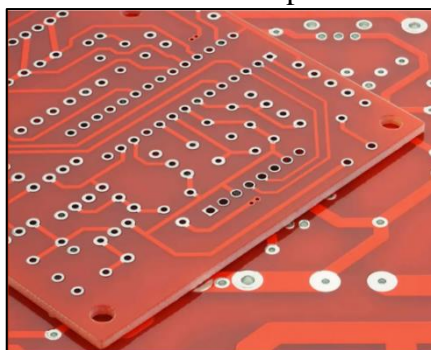
### 3.8 Placa de circuito impresso

A placa de circuito impresso (PCI) é um componente crucial na eletrônica, servindo como base para trilhas de cobre que conectam diversos componentes, como resistores, LEDs, diodos, capacitores, indutores e transistores. A placa atua como um material dielétrico, proporcionando o isolamento necessário entre as conexões. O desempenho elétrico do circuito depende do correto dimensionamento das trilhas, considerando sua largura, espessura e espaçamento (Melo, Rios e Gutierrez, 2001).

Inicialmente, as placas de circuito impresso eram fabricadas de forma simples, com trilhas de cobre em uma única face e componentes montados no lado oposto. Os terminais dos componentes atravessavam a placa por furos para se conectar às trilhas, e a junção elétrica era garantida por soldagem (Melo, Rios e Gutierrez, 2001). A Figura 22 mostra uma PCI de face simples. Com o avanço tecnológico, atualmente utiliza-se ambos os lados da placa para trilhas e montagem de componentes, obtendo-se assim uma PCI dupla face.



**Figura 22** - Placa de circuito impresso de fase simples



**Fonte:** Li, 2019.

As PCIs podem ser fabricadas com materiais diversos, escolhidos conforme a aplicação específica. A produção pode ser manual ou automatizada, sendo esta última mais segura e confiável. A manufatura automatizada também otimiza o uso dos materiais, aumentando a eficiência na construção das placas.

As PCIs podem ser classificadas conforme o tipo de laminação empregada na construção dos circuitos. Elas se dividem em duas categorias principais: circuitos rígidos convencionais e circuitos rígidos profissionais (Melo, Rios e Gutierrez, 2001). Os circuitos convencionais são fabricados com laminados de papel aglomerado impregnado com resina fenólica, como ilustrado na Figura 23. “Esses circuitos são amplamente utilizados em produtos eletrônicos de consumo, incluindo eletrodomésticos, brinquedos e aparelhos de áudio e vídeo” (Melo, Rios e Gutierrez, 2001, p. 121). Em contraste, os circuitos profissionais empregam laminados de fibra de vidro, como mostrado na Figura 22, sendo aplicados em áreas que exigem maior durabilidade e resistência, como informática, telecomunicações, automação e eletrônica embarcada (Melo, Rios e Gutierrez, 2001).

**Figura 23** - Placa fenolite



**Fonte:** Autoria própria, 2024.

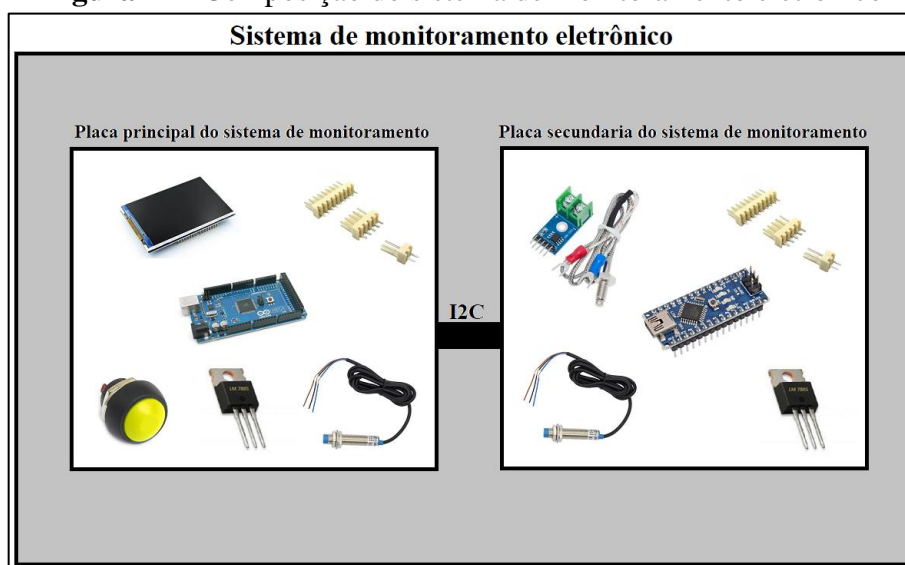
## 4. METODOLOGIA

Este trabalho tem como objetivo desenvolver um sistema eletrônico de monitoramento para o veículo *off-road* tipo baja, criado pela equipe do projeto de extensão Caraubaja, da UFERSA - Campus Caraúbas. O sistema monitora em tempo real a velocidade linear do veículo, a velocidade angular e a temperatura da CVT, além de monitorar o acionamento do sistema de tração 4x4 por meio de um botão. O desenvolvimento e a validação do sistema ocorreram nos laboratórios de Engenharia Elétrica e Mecânica da UFERSA, visando aprimorar a eficiência e a segurança operacional do veículo.

### 4.1 Sistema eletrônico de monitoramento

Foi projetado, inicialmente, um sistema de monitoramento eletrônico composto por duas placas de circuito impresso – uma principal e uma secundária – que se comunicam utilizando o protocolo I2C. A placa de monitoramento secundária é encarregada do cálculo da velocidade angular e da medição da temperatura da CVT. Por sua vez, a placa principal é responsável pelo cálculo da velocidade linear do veículo, pela supervisão do acionamento do sistema de tração 4x4 e pela exibição de todos os dados monitorados por ambas as placas. A Figura 24 ilustra o sistema de monitoramento eletrônico, destacando as placas de monitoramento principal e secundária e o canal de comunicação estabelecido via protocolo I2C.

**Figura 24** - Composição do sistema de monitoramento eletrônico



**Fonte:** Autoria própria, 2024.

#### 4.1.1 Projeto e desenvolvimento do *layout* da placa secundária

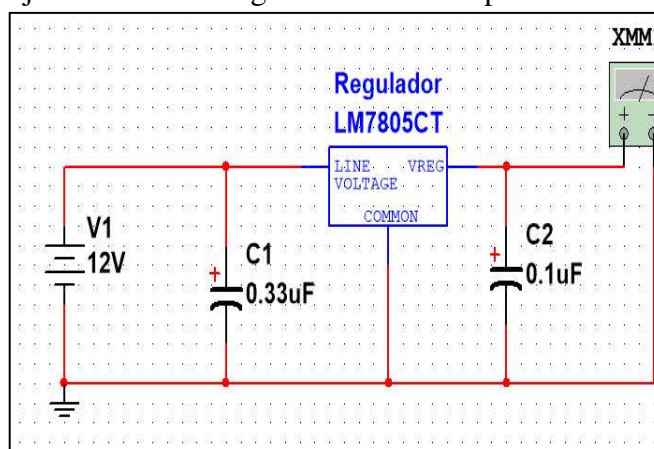
Para o desenvolvimento do *layout* da placa secundária do sistema de monitoramento, foi utilizado o *software* EasyEDA. Esta ferramenta de projeto de circuitos eletrônicos permite a criação, simulação e compartilhamento de esquemas de PCIs e de circuitos eletrônicos.

Todos os componentes necessários para o *layout* da placa secundária foram inseridos na área de trabalho do EasyEDA. O microcontrolador ATmega328p, integrado na placa de prototipagem Arduino Nano, fará o gerenciamento de todos os dados e sensores, essa placa foi selecionada devido ao seu tamanho compacto, capacidade de processamento e quantidade de pinos I/O digitais e analógicos, adequados às necessidades do projeto. Incluiu-se também o módulo MAX6675, que, junto ao sensor termopar tipo K, proporciona medições de temperatura precisas, sendo compatível com microcontroladores (Predolin, 2017).

Conectores KF2510 foram utilizados para permitir a conexão de sensores e atuadores aos pinos analógicos e digitais da placa Arduino Nano. Além disso, foram inseridos os componentes necessários para implementar um circuito regulador de tensão que converte 12V (provenientes da bateria do veículo) para 5V (tensão recomendada para placas Arduino, alimentadas pelo pino VIN), usando o regulador linear LM7805.

O LM7805 recebe uma entrada de 12V e a regula para uma saída estável de 5V. Esse circuito inclui dois capacitores: um capacitor de  $0,33\mu\text{F}$  na entrada, que estabiliza a tensão e filtra ruídos da fonte de alimentação, e um capacitor de  $0,1\mu\text{F}$  na saída, que garante uma saída de tensão estável. A Figura 25 ilustra o projeto do circuito regulador de tensão.

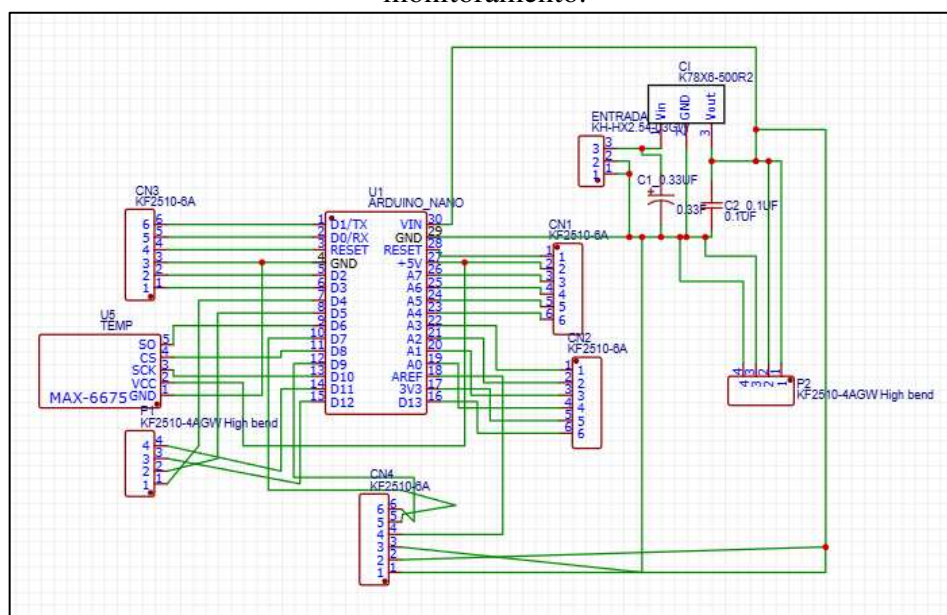
**Figura 25** - Projeto do circuito regulador de tensão para alimentação do Arduino



Fonte: Autoria própria, 2024.

Após a adição de todos os componentes necessários para o *layout* da placa secundária do sistema de monitoramento na área de trabalho do EasyEDA, todos os elementos foram interconectados de forma adequada. A Figura 26 ilustra o esquema de ligação, que foi então convertido para o formato de placa de circuito impresso, permitindo assim o desenvolvimento do *design* da placa secundária de monitoramento.

**Figura 26** - Esquema de conexão entre os elementos da placa secundária do sistema de monitoramento.



Fonte: Autoria própria, 2024.

#### 4.1.2 Projeto e desenvolvimento do *layout* da placa principal

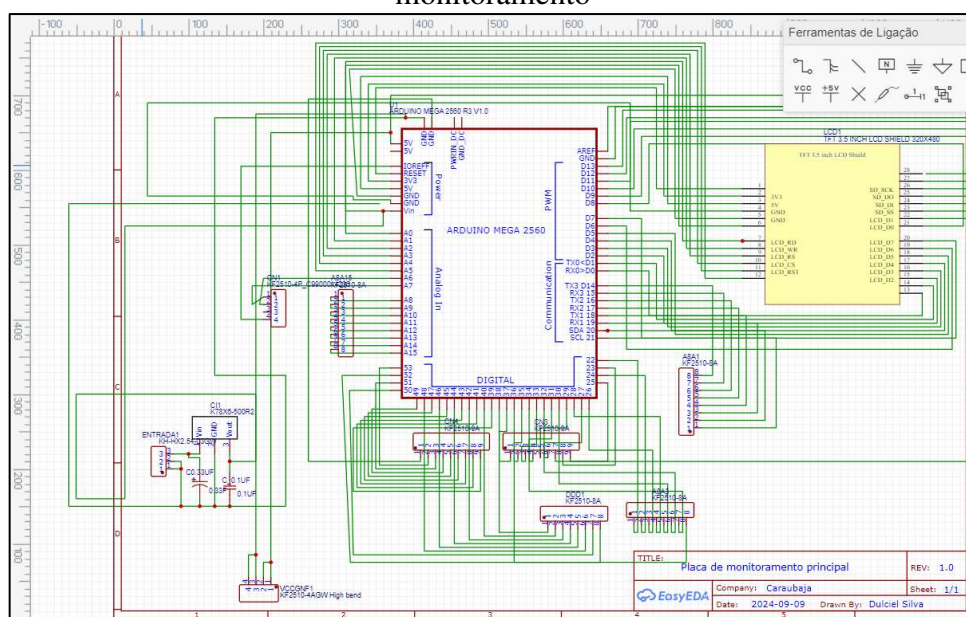
Para gerenciar os dados, sensores e o módulo de exibição, optou-se pela placa de prototipagem Arduino Mega 2560 Rev3, que possui o microcontrolador ATmega2560. Essa escolha foi fundamentada em sua robustez de processamento, ampla disponibilidade de pinos de entrada e saída analógicos e digitais, além da compatibilidade com diversos módulos LCD TFT.

O módulo LCD TFT de 3,5 polegadas, que possui resolução HD e é compatível com o Arduino Mega, também foi integrado ao *software* EasyEDA. Este módulo será utilizado para exibir informações como a velocidade linear do veículo, velocidade angular e temperatura da CVT, além de indicar o acionamento do sistema de tração 4x4.

Além disso, foram incluídos os componentes essenciais para o circuito regulador de tensão. Os conectores KF2510 foram adicionados para garantir o acesso a todos os pinos

analógicos e digitais do Arduino Mega. Esses conectores foram escolhidos por seu baixo custo, confiabilidade nas conexões e facilidade de montagem. A Figura 27 ilustra o esquema de conexão dos elementos na placa principal do sistema de monitoramento.

**Figura 27** - Esquema de conexão entre os elementos da placa principal do sistema de monitoramento



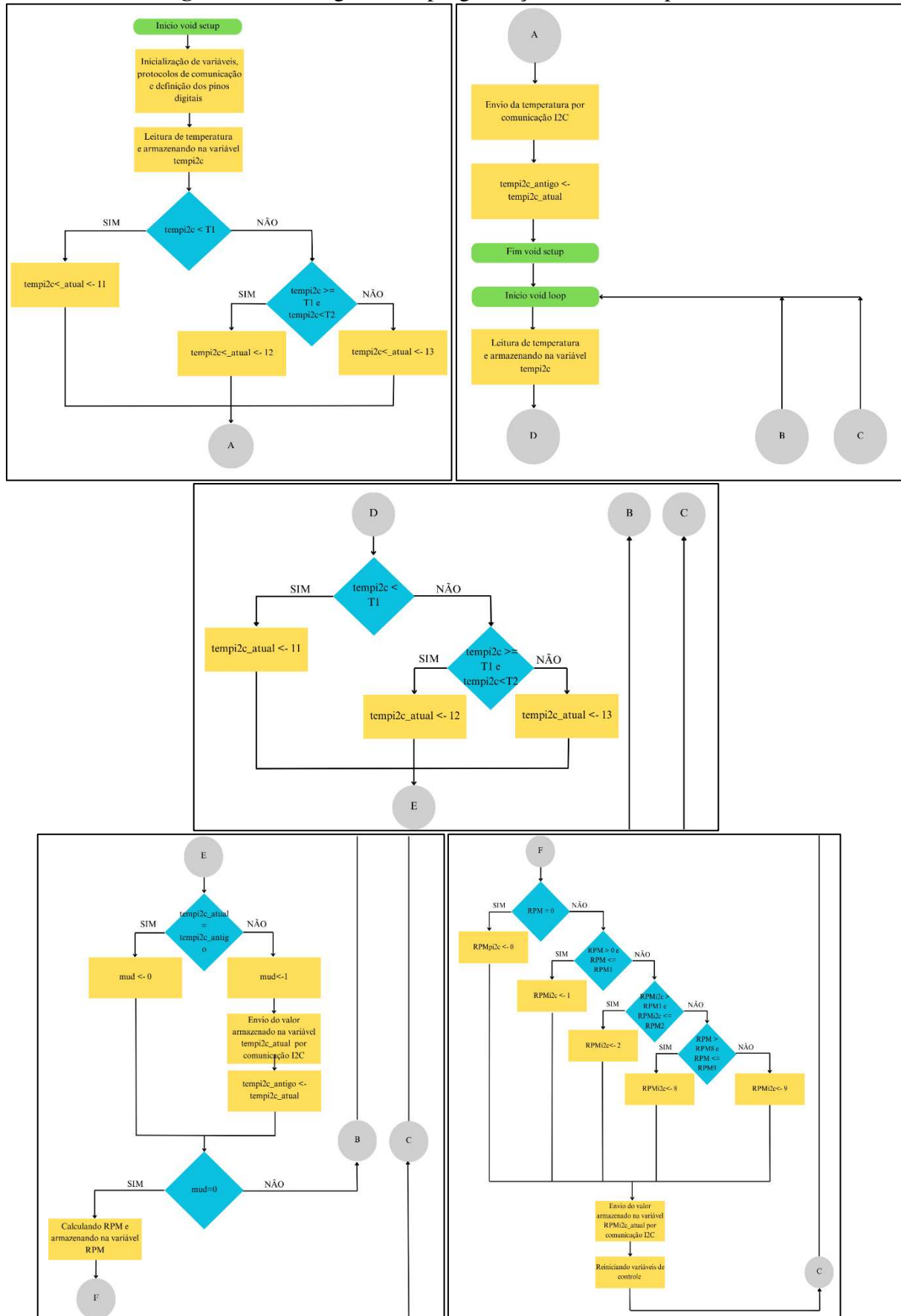
Fonte: Autoria própria, 2024.

## 4.2 Programação da placa secundária

Um código em Arduino é estruturado em duas funções principais. A primeira delas é a *void setup*, que é responsável pela inicialização do sistema, definição dos pinos e protocolos a serem utilizados. Essa função é executada apenas uma vez, durante a inicialização da placa de prototipagem Arduino. Nessa etapa, os pinos são configurados como entrada ou saída, e bibliotecas ou dispositivos são inicializados, preparando o ambiente para o funcionamento do programa.

A Figura 28 ilustra o fluxograma da programação desenvolvida no *software* Arduino IDE para a placa secundária do sistema de monitoramento. No início, é descrita a função *void setup*, onde todas as variáveis, constantes e protocolos de comunicação necessários para o funcionamento do sistema são configurados. Em seguida, o programa realiza a leitura da temperatura da CVT, armazenando o valor obtido na variável *tempi2c*.



**Figura 28 - Fluxograma da programação referente a placa secundaria**

Fonte: Autoria própria, 2024.

Com o valor da temperatura obtido, realiza-se uma comparação entre essa temperatura e os níveis de referência, T1 e T2. Se a temperatura for inferior a T1, considera-se que a temperatura da CVT está em um nível normal, e a variável *tempi2c\_atual* recebe o valor 11. Se a temperatura estiver entre T1 e T2 (ou seja, maior ou igual a T1 e menor que T2), a variável *tempi2c\_atual* assume o valor 12, indicando uma faixa de atenção, sinalizando que a CVT está próxima da sua temperatura máxima de operação. Por fim, se a temperatura for superior a T2, a variável *tempi2c\_atual* recebe o valor 13, indicando um estado de superaquecimento da CVT, com temperatura acima da máxima permitida. Nesse caso, é sugerido ao piloto que desligue o veículo para evitar danos ao sistema.

Após determinar a faixa de temperatura (11, 12 ou 13), o valor armazenado na variável *tempi2c\_atual* é transmitido para a placa principal do sistema de monitoramento via protocolo de comunicação I2C, com o intuito de ser exibido. Em seguida, esse valor é copiado para a variável *tempi2c\_antigo*, permitindo o armazenamento da última de faixa de temperatura. Assim, conclui-se a execução da função *void setup*.

Após a inicialização da função *void loop*, que, ao contrário da *void setup*, é executada continuamente enquanto a placa Arduino estiver ligada, o sistema realiza a leitura da temperatura da CVT e classifica o valor obtido na faixa de temperatura apropriada. Em seguida, calcula-se a velocidade angular, utilizando a Equação 1, e o valor resultante é armazenado na variável RPM. Esse valor é então comparado e classificado em uma faixa numérica de 0 a 9, sendo transferido para a variável *RPMi2c*. Por fim, a variável *RPMi2c* é enviada para a placa principal do sistema de monitoramento via comunicação I2C, para ser exibida juntamente com os demais dados do sistema.

$$RPM = \frac{1}{\frac{Q_{imans}}{T1}} \cdot contador \cdot 60000 \text{ [RPM]} \quad (1)$$

Neste sistema, a variável RPM armazena o valor da velocidade angular calculada, enquanto *Qiman* representa a quantidade de ímãs ou dentes de uma engrenagem metálica que o sensor detectará. A variável T1 define o tempo de amostragem e o contador registra a sequência de pulsos gerados pelo sensor indutivo ao identificar um objeto metálico ou ímã durante o intervalo de amostragem.

Considerando as especificações da comunicação I2C, as características dos sensores e a capacidade de processamento das placas Arduino, optou-se por enviar um único dado por ciclo

da função *void loop*. Como a temperatura varia lentamente, o valor tende a se manter dentro da mesma faixa por vários ciclos. Para evitar a transmissão repetitiva do mesmo dado, foi criada a variável `tempi2c_antigo`, que armazena a faixa de temperatura enviada no ciclo anterior. Também se introduziu a variável `mud` (mudança) para indicar o dado a ser transmitido: quando `mud` é igual a 0, transmite-se a faixa de velocidade angular; quando `mud` é igual a 1, envia-se a faixa de temperatura.

Quando os valores de `tempi2c_atual` e `tempi2c_antigo` diferem, isso indica uma mudança na faixa de temperatura. Nessa condição, a variável `mud` recebe o valor 1, sinalizando que ocorreu uma atualização na faixa. O valor atualizado de `tempi2c_atual` é então transmitido para a placa de monitoramento principal, garantindo que alterações das faixas de temperatura sejam comunicadas imediatamente ao sistema.

A seguir, verifica-se se `mud` é igual a 0. Caso `mud` seja 1, essa condição impede a continuidade do fluxo de execução, que incluiria o cálculo da velocidade angular, o armazenamento na variável `RPM`, a seleção da faixa e o envio desse dado para a placa principal. Dessa forma, garante-se que apenas um dado seja transmitido por vez, atendendo aos limites de processamento da comunicação I2C e à prioridade da informação de mudança.

Quando os valores de `tempi2c_atual` e `tempi2c_antigo` são iguais, indica que não houve alteração na faixa de temperatura, tornando desnecessário o reenvio dessa informação para a placa principal. Nesse cenário, a variável `mud` é ajustada para 0, permitindo que o programa prossiga para o cálculo da velocidade angular. O valor calculado é alocado na faixa correspondente e armazenado na variável `RPMi2c`, que é enviada para a placa de monitoramento principal. Dessa forma, garante-se o envio apenas da informação mais relevante no ciclo atual de execução.

Para medir a velocidade angular da CVT, utilizou-se um sensor indutivo, que detecta objetos metálicos ou ímãs próximos. Cada vez que um dente da engrenagem ou um ímã fixado na CVT passa pelo sensor, um pulso é gerado. No projeto da equipe Caraubaja, ímãs foram fixados no eixo traseiro para calcular a velocidade linear do veículo, enquanto as engrenagens da CVT são utilizadas para medir a velocidade angular.

A placa secundária de monitoramento utiliza uma interrupção para contar esses pulsos dentro de um intervalo de tempo específico ( $T_1$ ), geralmente de um segundo. O comando de interrupção direciona o fluxo do código para uma rotina que incrementa uma variável chamada contador a cada detecção do sensor indutivo. Com os valores das variáveis contador, `Qimans` e  $T_1$ , a velocidade angular da CVT é calculada conforme a Equação 1, apresentada anteriormente.



O sistema define dez faixas de velocidade angular, denominadas de RPM0 a RPM9. Na primeira faixa, onde a CVT está desligada ( $RPM = 0$ ), a variável RPMi2c recebe o valor 0. Para as demais faixas, valores sequenciais de 1 a 9 são atribuídos conforme a velocidade angular aumenta. Esse valor em RPMi2c controla a quantidade de retângulos exibidos no LCD TFT, que sinalizam visualmente o nível de velocidade angular.

Por exemplo, quando RPMi2c é igual a 3, o LCD TFT exibe dois retângulos verdes e um amarelo, indicando uma velocidade intermediária. Quando RPMi2c alcança 9, todos os retângulos são exibidos dois verdes, cinco amarelos e dois vermelhos, representando a velocidade angular máxima da CVT.

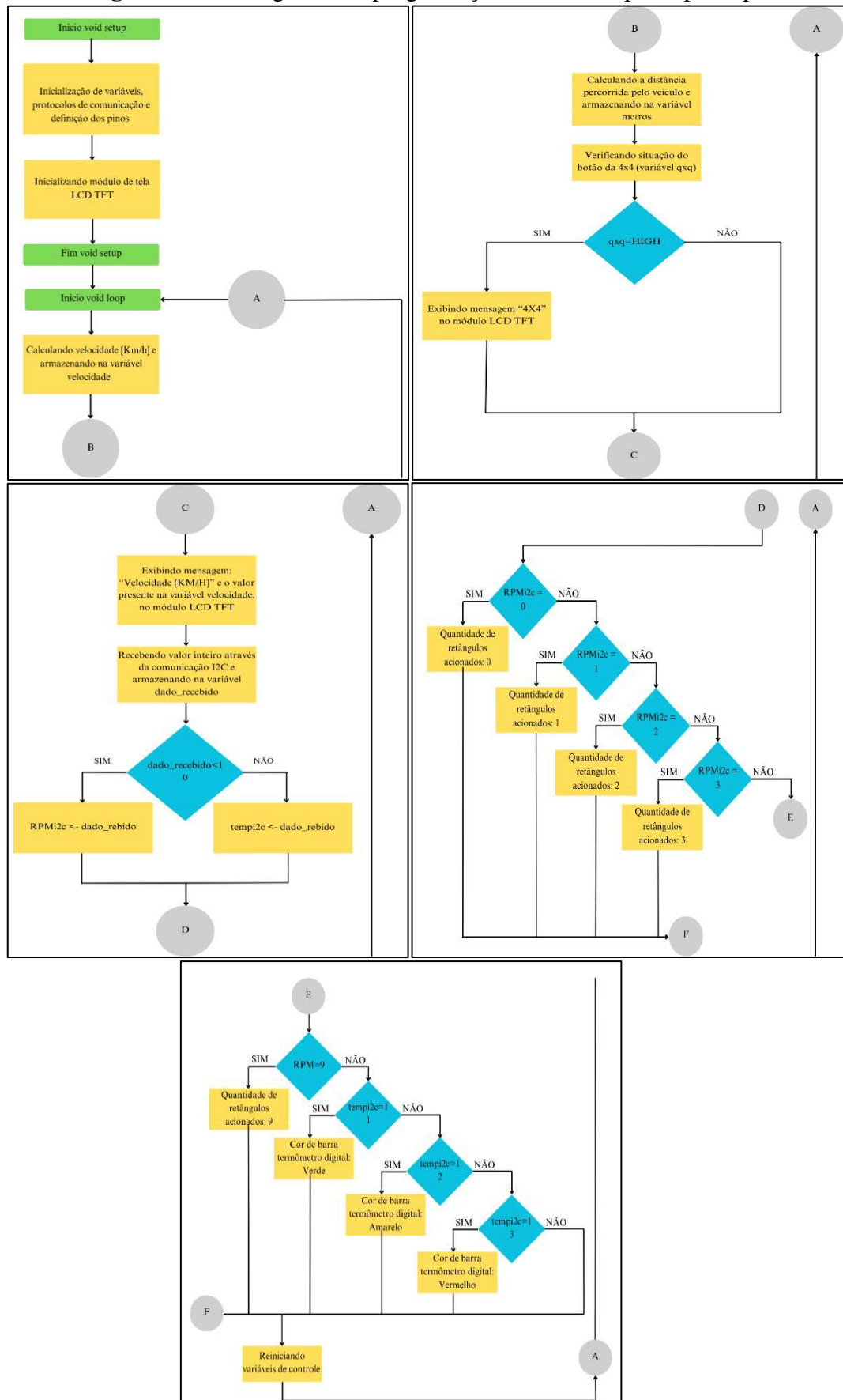
Após a definição da faixa de velocidade angular, o valor contido na variável RPMi2c é transmitido para o sistema de monitoramento principal, que exibe no módulo LCD TFT a quantidade correspondente de retângulos correspondente ao dado enviado, conforme a faixa de RPM. Em seguida, todas as variáveis de controle são reiniciadas para garantir a correta contagem e análise no próximo ciclo de leitura.

As faixas de temperatura (T1 e T2) e de velocidade angular (RPM1 a RPM9) são definidas como constantes no código. Essa abordagem não só otimiza o uso da memória do microcontrolador, mas também assegura maior eficiência no processamento, uma vez que evita a redefinição contínua desses parâmetros a cada execução, garantindo estabilidade e confiabilidade no monitoramento.

A comunicação I2C nas placas Arduino limita a transmissão a valores inteiros de 0 a 255, impossibilitando o envio direto de valores de velocidade angular superiores, como os da CVT, que podem ultrapassar 3.000 RPM. Para resolver isso, foi decidido exibir no módulo LCD TFT faixas que indicam a intensidade do RPM e da temperatura, já que o piloto não necessita do valor exato. Essa abordagem facilita a interpretação das informações, permitindo decisões informadas sem depender de dados numéricos precisos.

### **4.3 Programação da placa principal do sistema de monitoramento**

A placa principal do sistema de monitoramento é responsável por medir a velocidade linear do veículo e verificar se o sistema de tração 4x4 está ativado. Além disso, a placa exibe os dados monitorados, como as faixas de temperatura e a velocidade angular da CVT, que são recebidos da placa secundária via comunicação I2C. O fluxograma da programação criada no *software* Arduino IDE para a placa principal pode ser visualizado na Figura 29.

**Figura 29** - Fluxograma da programação referente a placa principal

Fonte: Autoria própria, 2024.

Na função *void setup*, são inicializadas as variáveis, configurados os protocolos de comunicação (serial e I2C) e definidos os pinos a serem utilizados, além de inicializar o módulo de tela LCD TFT para exibição de informações. A função *void loop* calcula a velocidade linear do veículo a partir da velocidade angular do eixo da roda traseira, armazenando o resultado na variável RPM. Em seguida, aplica-se a Equação 2 para converter a velocidade angular (RPM) em velocidade linear (Km/h), armazenando o valor na variável velocidade.

$$velocidade = \frac{RPM \cdot 2\pi R \cdot 3.6}{60} [\text{Km/h}] \quad (2)$$

O valor de RPM indica a velocidade angular da roda traseira em rotações por minuto. O termo  $2\pi R$  representa o comprimento da circunferência da roda, enquanto o fator  $3,6/60$  converte a velocidade de metros por segundo para quilômetros por hora. Conhecendo o comprimento da circunferência das rodas, é possível calcular a distância percorrida em um intervalo de tempo T1 utilizando a Equação 3, e o resultado é armazenado na variável metrosp.

$$metrosp = \frac{contador \cdot 2\pi R}{Qimans} \quad (3)$$

Após calcular a velocidade e a distância percorrida, o programa verifica o estado do botão de sinalização do sistema de tração 4x4. Esse botão está conectado ao pino de 5V do Arduino e ao pino digital 47, utilizando um resistor *pull-down*. Quando o botão é pressionado, um sinal lógico de nível alto (*HIGH*) é enviado ao pino digital 47, indicando que o sistema de tração 4x4 está ativo. Nesse caso, a mensagem “4x4” é exibida no canto inferior direito da tela, alertando o piloto sobre a ativação do sistema. Se o botão não for pressionado, um sinal lógico baixo é enviado ao pino 47 e a mensagem não aparece.

Após a verificação do botão, o programa exibe o valor da velocidade no *display* do módulo LCD TFT. Em seguida, realiza-se a verificação da comunicação I2C, analisando os dados recebidos da placa secundária de monitoramento, que incluem as faixas de velocidade angular (variando de 0 a 9) e de temperatura (11 a 13) da CVT. Para identificar a natureza dos dados recebidos, verifica-se se o valor é menor que 10, indicando uma faixa de velocidade angular, ou maior que 10, referindo-se à temperatura. Se o dado recebido for uma faixa de velocidade angular, o valor é armazenado na variável RPMi2c, caso contrário, se for uma faixa de temperatura, o dado é armazenado na variável tempi2c. Em seguida, são realizadas comparações para determinar quantos retângulos devem ser exibidos, representando a

intensidade da velocidade angular da CVT. Por exemplo, se a variável RPMi2c for igual a 3, serão exibidos 3 retângulos na parte superior do display do módulo LCD TFT, se for igual a 5, serão exibidos 5 retângulos.

A temperatura é exibida em um termômetro digital com barras verticais localizadas na lateral esquerda do display do módulo LCD TFT, identificadas pelo rótulo "TEMP". Quando o valor da temperatura é igual a 11, uma barra verde é exibida, indicando que a temperatura está dentro da faixa normal de operação. Se o valor for igual a 12, uma barra amarela é adicionada, sinalizando que a temperatura da CVT está próxima do limite máximo. Por outro lado, se o valor for igual a 13, uma barra vermelha é adicionada acima da barra amarela, indicando que a temperatura excedeu os limites operacionais permitidos para a CVT. Após essa exibição, todas as variáveis de controle são reinicializadas, preparando o sistema para o próximo ciclo de operação.

#### 4.4 Ensaio de temperatura

Foi realizado um ensaio de temperatura para avaliar a precisão do sensor termopar tipo K e da programação da placa secundária do sistema de monitoramento, conforme ilustrado na Figura 30. Para garantir a confiabilidade dos dados, utilizou-se o banho ultratermostático SL 152, equipado com um sensor integrado que mede a temperatura do líquido e exibe os valores em tempo real em um display frontal, utilizando água destilada como meio de condução térmica. Adicionalmente, foi empregado um termômetro digital MT-450, da marca Minipa, para complementar a precisão das leituras e fornecer uma comparação adicional.

**Figura 30 - Sistema para ensaio de temperatura**



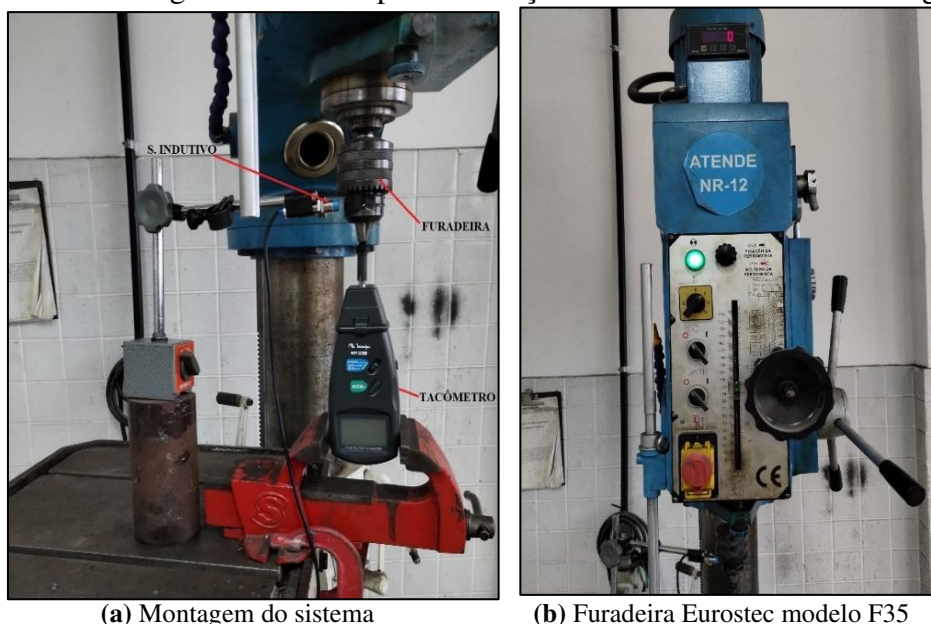
**Fonte:** Autoria própria, 2024.

A temperatura inicial do ensaio foi ajustada para 25°C. Após a estabilização da água destilada, o banho ultratermostático foi configurado para atingir 80°C. Durante o aumento gradual da temperatura do líquido, os valores medidos por todos os equipamentos de monitoramento foram registrados a cada 30 segundos, possibilitando uma análise detalhada do comportamento térmico da água destilada ao longo do ensaio.

#### 4.5 Ensaio de velocidade angular

Realizou-se o ensaio de velocidade angular, no laboratório de Engenharia Mecânica, para validar a programação e a escolha do sensor indutivo. Esse sensor mede a velocidade angular da CVT na placa secundária e do eixo traseiro na placa de monitoramento principal, convertendo-a em velocidade linear em km/h, de acordo com a Equação 2. Como referência, utilizou-se um tacômetro digital Minipa, modelo MDT-2238B, projetado para medir velocidade angular. Ambos os dispositivos foram utilizados para medir a velocidade angular de uma furadeira de coluna, modelo F35, da marca Eurostec ilustrada na Figura 31(b).

**Figura 31** - Montagem do sistema para realização do ensaio de velocidade angular



(a) Montagem do sistema

(b) Furadeira Eurostec modelo F35

Fonte: Autoria própria, 2024.

A furadeira elétrica possui seis níveis de velocidade selecionáveis: CE, AE, BE, CD, AD e BD, cada um com duas variações controladas por um botão. Na primeira variação (botão na posição 1), a velocidade é inferior à da segunda variação (botão na posição 2). O ensaio foi

realizado acoplando o tacômetro e o sensor indutivo à parte rotatória da furadeira, conforme mostrado na Figura 31 (a). O tacômetro foi instalado verticalmente para medir a velocidade angular, enquanto o sistema de monitoramento foi posicionado horizontalmente, com ímãs fixados na furadeira para permitir a leitura pelo sensor. Os ensaios foram conduzidos com um, dois, três e quatro ímãs em todos os níveis de velocidade mencionados, com o botão ajustado na segunda posição. A velocidade angular foi calculada utilizando a Equação 1.

#### 4.6 Teste de envio de dados por conexão I2C

Para estabelecer a comunicação I2C, os pinos SDA e SCL da placa Arduino Mega 2560 Rev3 são conectados aos pinos correspondentes da Arduino Nano, configurando-a como mestre e a Nano como escravo. Durante os testes, a velocidade angular foi medida e armazenada na variável RPM na placa secundária do sistema de monitoramento. Os valores foram organizados em faixas conforme as Tabelas 4 e 5, que definem o valor a ser enviado de acordo com a faixa em que o valor se encaixa. Vale ressaltar que essas faixas não correspondem às utilizadas pela equipe Caraubaja, mas foram selecionadas para facilitar o envio de dados sem equipamentos especializados.

**Tabela 4** - Faixas de RPM

| Faixas de RPM | Valor enviado via comunicação I2C |
|---------------|-----------------------------------|
| 12 < RPM ≤ 24 | 1                                 |
| 24 < RPM ≤ 29 | 2                                 |
| 29 < RPM ≤ 30 | 3                                 |
| 30 < RPM ≤ 50 | 4                                 |
| 50 < RPM ≤ 70 | 5                                 |
| 70 < RPM ≤ 80 | 6                                 |
| 80 < RPM ≤ 85 | 7                                 |
| 85 < RPM ≤ 90 | 8                                 |
| RPM > 90      | 9                                 |

**Fonte:** Autoria própria, 2024.

**Tabela 5 - Faixas de temperaturas**

| Faixas de temperatura em °C     | Valor enviado via comunicação I2C |
|---------------------------------|-----------------------------------|
| Temperatura lida < 29°C         | 11                                |
| 29°C >= Temperatura lida < 31°C | 12                                |
| Temperatura lida >= 31°C        | 13                                |

**Fonte:** Autoria própria, 2024.

#### 4.7 Características técnicas escolhidas para a confecção das placas do sistema de monitoramento

As placas utilizadas no sistema de monitoramento foram confeccionadas pela empresa TEC-CI Circuitos Impressos. A plataforma da empresa oferece opções para a personalização completa das PCIs, permitindo a escolha de características essenciais, como o formato (retangular ou redondo), o tipo de laminado, o número de faces com trilhas, a espessura do cobre, entre outros. No caso do sistema de monitoramento, foram selecionadas PCIs com laminado de vidro, espessura de 1,6 mm e cobre de 35  $\mu\text{m}$  (1 Oz/ft<sup>2</sup>). Além disso, as placas possuem máscara de solda verde aplicada em ambos os lados e legendas. Todas as especificações técnicas para as placas que compõem o sistema de monitoramento podem ser vistas na Tabela 6.

**Tabela 6 - Especificações técnicas das placas de circuito impresso principal e secundária**

| Características                                    | PCI Principal                            | PCI Secundária                           |
|----------------------------------------------------|------------------------------------------|------------------------------------------|
| Largura x Comprimento                              | 130 mm x 150 mm                          | 60 mm x 90 mm                            |
| Número de faces                                    | 2                                        | 2                                        |
| Laminado                                           | Vidro                                    | Vidro                                    |
| Espessura da sua placa                             | 1.6 mm                                   | 1.6 mm                                   |
| A espessura do cobre                               | 35 $\mu\text{m}$ (1 Oz/ft <sup>2</sup> ) | 35 $\mu\text{m}$ (1 Oz/ft <sup>2</sup> ) |
| Máscara de solda aplicada em um lado da PCB        | Não                                      | Não                                      |
| Máscara de solda aplicada em ambos os lados da PCB | Sim                                      | Sim                                      |
| Cor da máscara anti solda                          | Verde                                    | Verde                                    |
| Legendas em uma face                               | Não                                      | Não                                      |
| Legendas em ambas as faces                         | Sim                                      | Sim                                      |

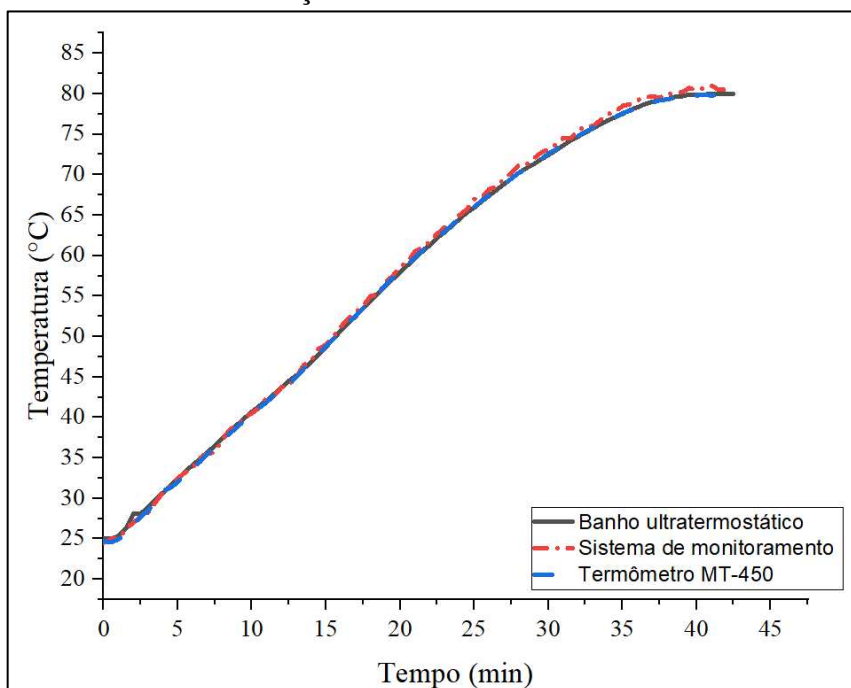
**Fonte:** Autoria própria, 2024.

## 5. RESULTADOS E DISCUSSÕES

### 5.1 Resultado do ensaio de temperatura

A análise dos dados coletados para os diferentes equipamentos utilizados na medição da temperatura da água destilada no banho ultratermostático possibilitou a elaboração das curvas apresentadas no Gráfico 1. Os resultados demonstram que as curvas obtidas pelo termômetro digital e pelo sistema de monitoramento apresentam um comportamento muito semelhante à curva de referência do banho ultratermostático. Essa congruência destaca a eficácia do sistema de monitoramento em realizar medições de temperatura com precisão e confiabilidade.

**Gráfico 1** - Comparação entre as curvas de aquecimento: banho ultratermostático, sistema de medição e termômetro MT-450



Fonte: Autoria própria, 2024.

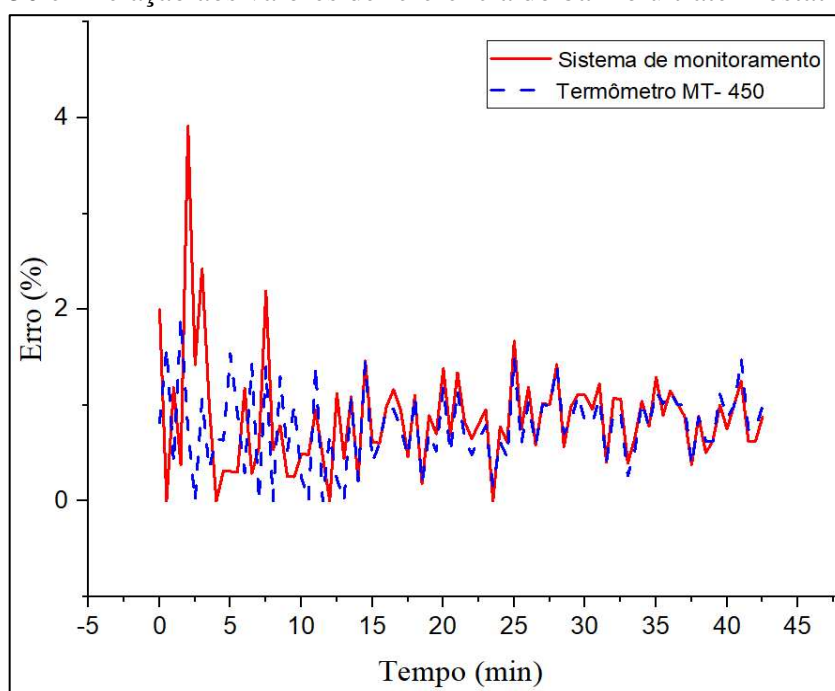
A diferença observada entre as medições do sistema de monitoramento e aquelas registradas pelo banho ultratermostático foi de aproximadamente  $\pm 0,75^{\circ}\text{C}$ . Esse pequeno desvio indica que o sensor utilizado está funcionando adequadamente, garantindo um bom nível de precisão nas medições. A consistência entre os dados dos diferentes instrumentos não apenas valida a qualidade do sistema de monitoramento, mas também reforça a credibilidade das conclusões que podem ser tiradas desse conjunto de dados.



O Gráfico 2 apresenta o erro relativo entre as medições do sistema de monitoramento, do termômetro digital e os valores de referência do banho ultratermostático. O erro de temperatura atinge um pico de 4,5% nos primeiros minutos, considerado aceitável, possivelmente devido ao tempo de estabilização do sistema e do termômetro MT-450, além de eventuais leituras incorretas do sensor do banho.

No Gráfico 1, observa-se um aumento abrupto na temperatura de referência seguido de uma diminuição aos 2,5 minutos, coincidente com o pico de erro mencionado anteriormente, logo o erro de 4,5% pode ter sido causado devido a uma medição equivocada do sensor presente no banho ultratermostático. À medida que a temperatura aumenta, ambos os sistemas exibem menor oscilação e mantêm o erro em uma faixa mais estável.

**Gráfico 2** - Variação do erro de temperatura do sistema de Monitoramento e termômetro MT-450 em relação aos valores de referência do banho ultratermostático



Fonte: Autoria própria, 2024.

O termômetro MT-450 apresenta uma variação de erro ligeiramente menor e mais estável do que o sistema de monitoramento nos primeiros minutos do ensaio. No entanto, à medida que o tempo avança e a temperatura do líquido aumenta, o erro do sistema de monitoramento tende a diminuir. Ao final do ensaio, com a temperatura próxima de 80°C, o erro do sistema de monitoramento se torna inferior ao erro registrado pelo termômetro digital.

## 5.2 Resultados de velocidade angular

As medições da velocidade angular foram primeiramente realizadas utilizando um ímã em todas as configurações de velocidade do nível 2 da furadeira. Em seguida, o processo foi repetido com a adição de dois, três e quatro ímãs, conforme especificado na Tabela 7, que apresenta os dados coletados para cada configuração.

**Tabela 7** - Dados obtidos durante o ensaio de velocidade angular

| <b>Velocidade furadeira</b> | <b>Velocidade angular (RPM)</b> |                                 |               |               |               |
|-----------------------------|---------------------------------|---------------------------------|---------------|---------------|---------------|
| <b>Posição do botão = 2</b> | <b>Tacômetro</b>                | <b>Sistema de monitoramento</b> |               |               |               |
|                             |                                 | <b>1 íman</b>                   | <b>2 ímãs</b> | <b>3 ímãs</b> | <b>4 ímãs</b> |
| CE                          | 124,6                           | 180                             | 150           | 140           | 135           |
| AE                          | 263,6                           | 300                             | 270           | 280           | 270           |
| BE                          | 566,6                           | 600                             | 570           | 580           | 570           |
| CD                          | 722,8                           | 780                             | 749           | 740           | 734           |
| AD                          | 1529                            | 1560                            | 1530          | 1540          | 1530          |
| BD                          | 3286                            | 3300                            | 3300          | 3290          | 3290          |

**Fonte:** Autoria própria, 2024.

Considerando os valores do tacômetro como referência, observa-se que o erro nas medições do sistema de monitoramento diminui com o aumento da velocidade angular e/ou do número de ímãs. Isso ocorre porque o sensor indutivo realiza leituras em intervalos fixos de um segundo, permitindo que, com velocidades mais altas ou mais ímãs, o número de leituras aumente, tornando a medição da velocidade angular mais precisa. A Tabela 8 detalha o erro relativo dessas medições em comparação aos valores de referência apresentados na Tabela 7.

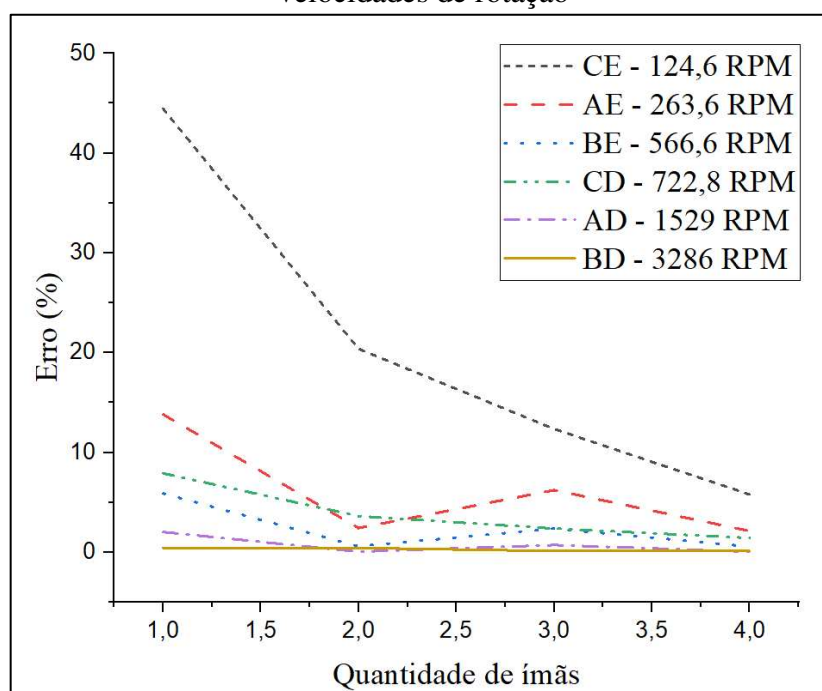
**Tabela 8** - Valores de Erro para o ensaio de RPM

| <b>Tacômetro</b>              | <b>Sistema de monitoramento</b> |               |               |               |
|-------------------------------|---------------------------------|---------------|---------------|---------------|
| <b>Valor referência (RPM)</b> | <b>Erro (%)</b>                 |               |               |               |
|                               | <b>1 íman</b>                   | <b>2 ímãs</b> | <b>3 ímãs</b> | <b>4 ímãs</b> |
| 124,6                         | 44,46                           | 20,38         | 12,36         | 5,78          |
| 263,6                         | 13,81                           | 2,43          | 6,22          | 2,13          |
| 566,6                         | 5,89                            | 0,60          | 2,36          | 0,57          |
| 722,8                         | 7,91                            | 3,62          | 2,38          | 1,44          |
| 1529                          | 2,027                           | 0,06          | 0,72          | 0,06          |
| 3286                          | 0,43                            | 0,43          | 0,12          | 0,12          |

**Fonte:** Autoria própria, 2024.

O Gráfico 3 apresenta a curva do erro de velocidade angular (%) em função da quantidade de ímãs, com base nos dados da Tabela 8. Cada linha do gráfico representa uma velocidade angular de referência medida pelo tacômetro durante o ensaio. Nota-se que o erro tende a se aproximar de zero à medida que tanto a velocidade de rotação da furadeira quanto o número de ímãs aumentam, conforme explicado anteriormente.

**Gráfico 3** - Variação do erro de RPM em função da quantidade de ímãs para diferentes velocidades de rotação

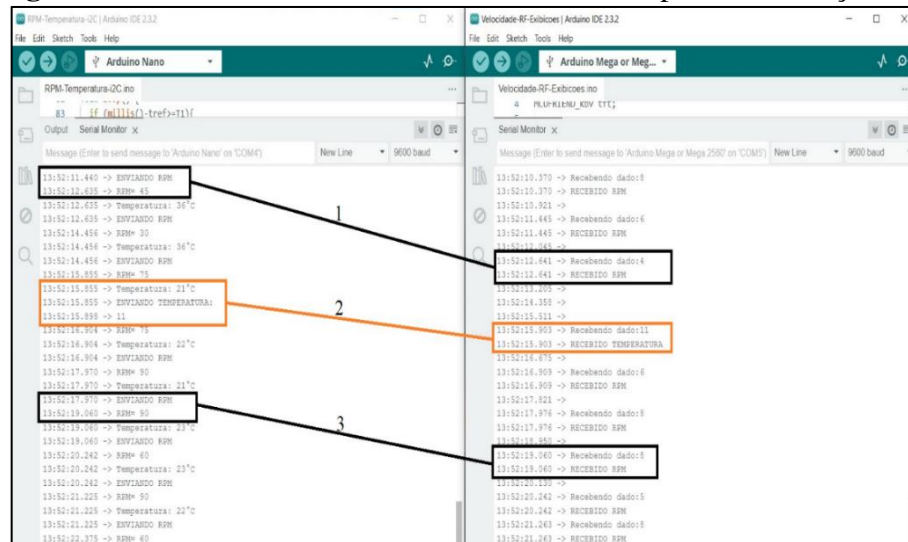


Fonte: Autoria própria, 2024.

### 5.3 Teste de envio de dados por conexão I2C

A Figura 32 ilustra os testes de transmissão e recepção de dados via comunicação I2C. No primeiro teste, o dado transmitido é uma velocidade angular de 45 RPM, classificada na quarta faixa de RPM conforme a Tabela 4, resultando no envio do número 4. No segundo teste, o dado corresponde à temperatura, com uma leitura de 21°C, sendo então enviado o número 11, conforme a faixa de temperatura da Tabela 5. O terceiro teste transmite uma velocidade angular de 90 RPM, resultando no envio do número 8.

**Figura 32 -** Teste de envio e recebimento de dados por comunicação I2C



Fonte: Autoria própria, 2024.

A estrutura lógica empregada para identificar e transmitir via I2C as faixas correspondentes aos dados dos sensores, visando à exibição na placa principal de monitoramento, demonstrou-se eficaz. Vale destacar que as faixas de temperatura e velocidade angular usadas nos testes foram ajustadas para facilitar ensaios rápidos, com valores inferiores aos aplicados ao veículo real. Isso ocorreu porque reproduzir condições extremas, como 3000 RPM ou 200°C, demandaria equipamentos especializados, inviáveis neste contexto experimental.

#### 5.4 Modelo de exibição do módulo *display* LCD TFT

A Figura 33 apresenta o modelo de exibição do display LCD TFT, que fornece ao piloto informações essenciais do sistema de monitoramento, tais como a velocidade linear do veículo, a velocidade angular, a temperatura da CVT e o status do sistema de tração 4x4.

**Figura 33 -** Modelo de exibição utilizado



Fonte: Autoria própria, 2024.

Na parte superior do display, estão dispostos nove retângulos, organizados em dois verdes, cinco amarelos e dois vermelhos, que representam diferentes faixas de velocidade angular da CVT. Quando todos os retângulos estão apagados (0 RPM), isso indica que a CVT está desligada. Ao ativar a CVT, a velocidade angular inicial é sinalizada pelo primeiro retângulo verde. Conforme a velocidade aumenta, outros retângulos aparecem até que se alcance o último retângulo vermelho, que indica o limite máximo da intensidade da velocidade angular.

A temperatura da CVT é exibida por meio de uma barra vertical rotulada como "TEMP". Essa barra é segmentada em três cores: verde, amarelo e vermelho. A faixa verde indica que a temperatura está dentro de uma faixa segura de operação, enquanto a faixa amarela sugere atenção, indicando que a temperatura está próxima do limite máximo. Já a faixa vermelha alerta que a CVT ultrapassou a temperatura máxima permitida.

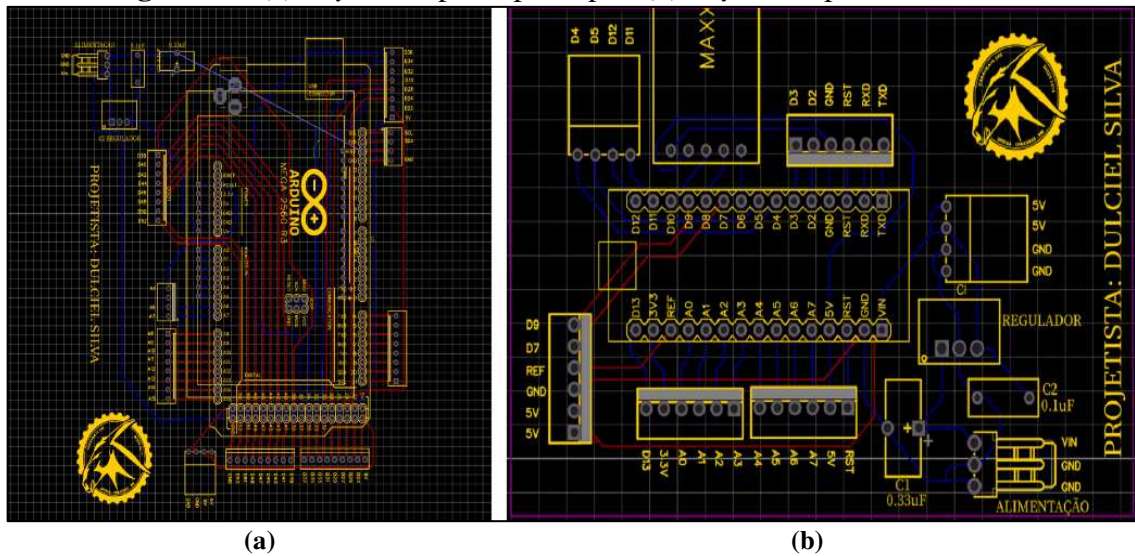
Na parte inferior direita do LCD, encontra-se o indicador do sistema de tração 4x4. Esse indicador exibe a mensagem "4x4" abaixo do termômetro quando o sistema está ativado, conforme ilustrado na Figura 33. Caso o sistema não esteja ativo, a mensagem não é exibida.

A velocidade angular e a temperatura da CVT são apresentadas em faixas, pois essas informações são mais orientativas e não exigem valores exatos para que o piloto possa avaliar a situação atual do veículo e tomar decisões. Em contraste, a velocidade linear do veículo é uma informação crucial para o piloto, que necessita de um valor preciso. Portanto, o valor real da velocidade linear é exibido em km/h, ocupando a maior área do *display* devido à sua relevância.

## 5.5 Confecção das placas primaria e secundaria do sistema de monitoramento

As placas primária e secundária do sistema de monitoramento foram fabricadas com precisão e confiabilidade, utilizando equipamentos especializados. O *layout*, criado no *software* EasyEDA é ilustrado na Figura 34, possui trilhas cobreadas em azul, indicando as trilhas presente na parte inferior da placa, já as trilhas em vermelho são as trilhas localizadas na parte superior da placa, todas as trilhas possuem uma largura de 0,5 mm, suportando correntes de até 1 A, sem sofrerem danificações. O design também incorpora o símbolo da equipe Caraubaja, rótulos dos componentes utilizados e o nome do projetista, posicionados em camadas sem cobre para uma identificação clara dos elementos presentes nas placas.

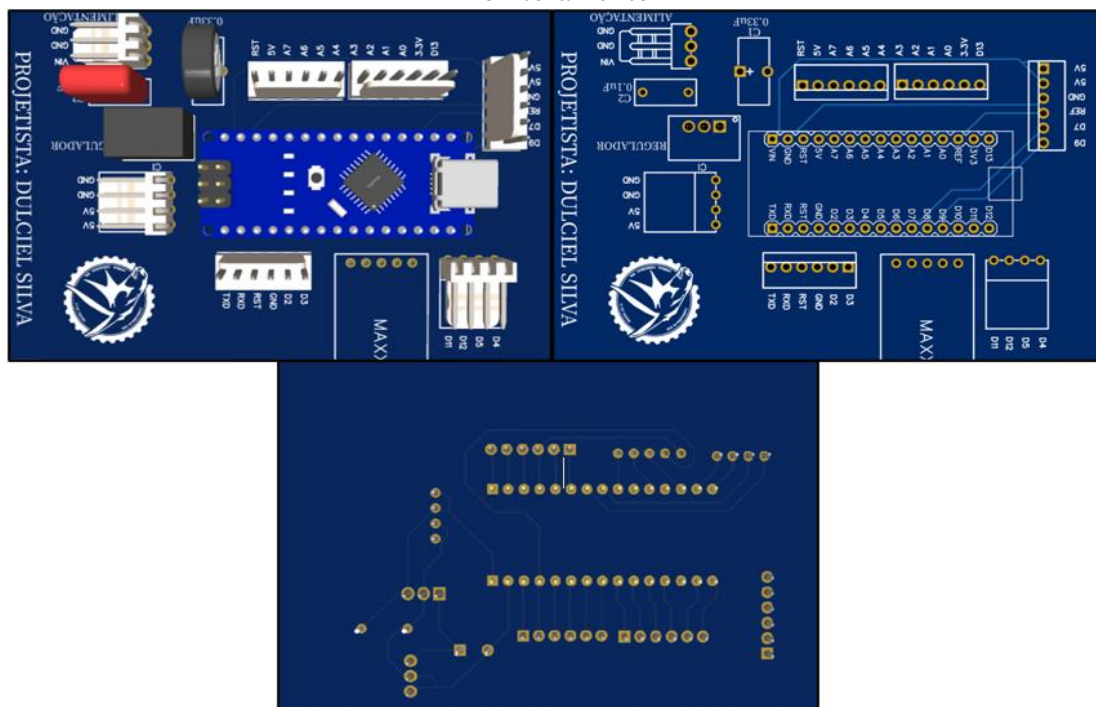
**Figura 34:** (a) *Layout* da placa principal, (b) *Layout* da placa secundária



Fonte: Autoria própria, 2024.

O EasyEDA oferece a funcionalidade de visualização em 3D dos projetos desenvolvidos, permitindo ao projetista visualizar como será o produto final após a confecção. A Figura 35 apresenta a visualização em 3D da placa secundária do sistema de monitoramento, enquanto a placa principal é ilustrada na Figura 36.

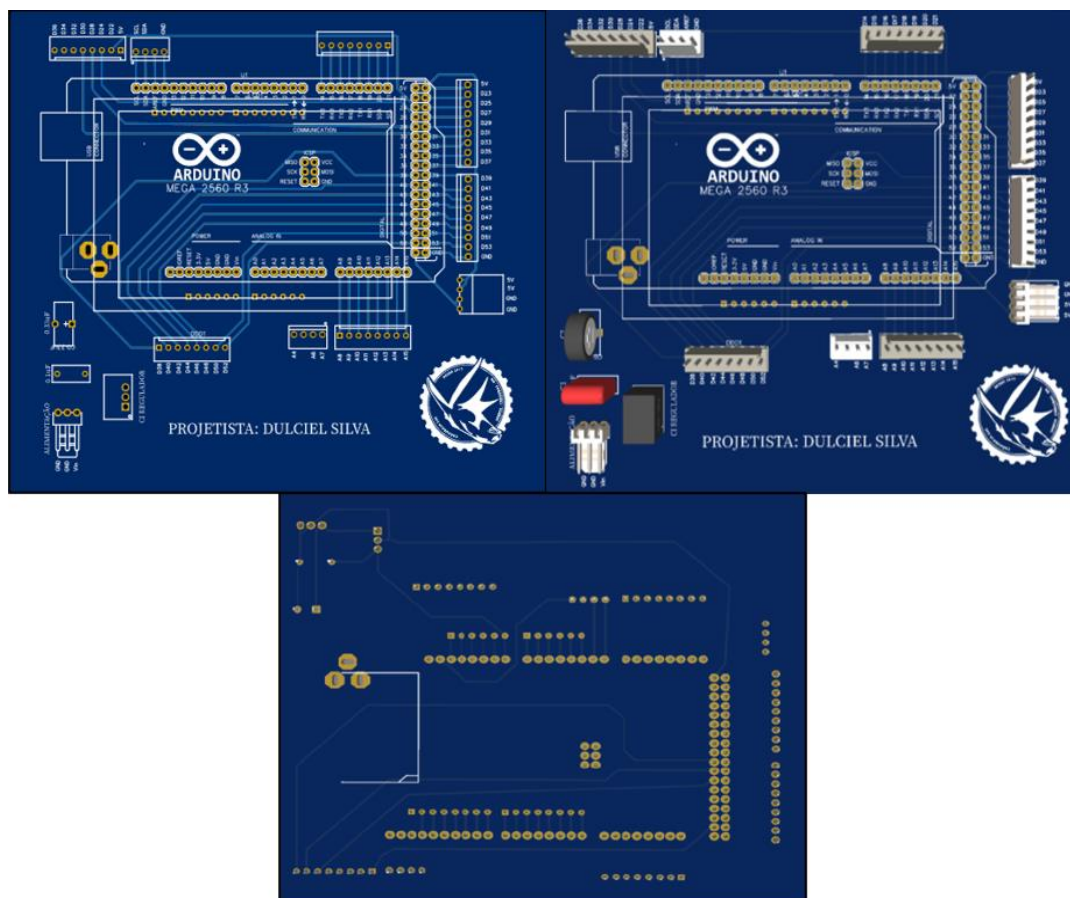
**Figura 35 -** Visualização 2D e 3D do *layout* da placa secundária do sistema de monitoramento



Fonte: Autoria própria, 2024.



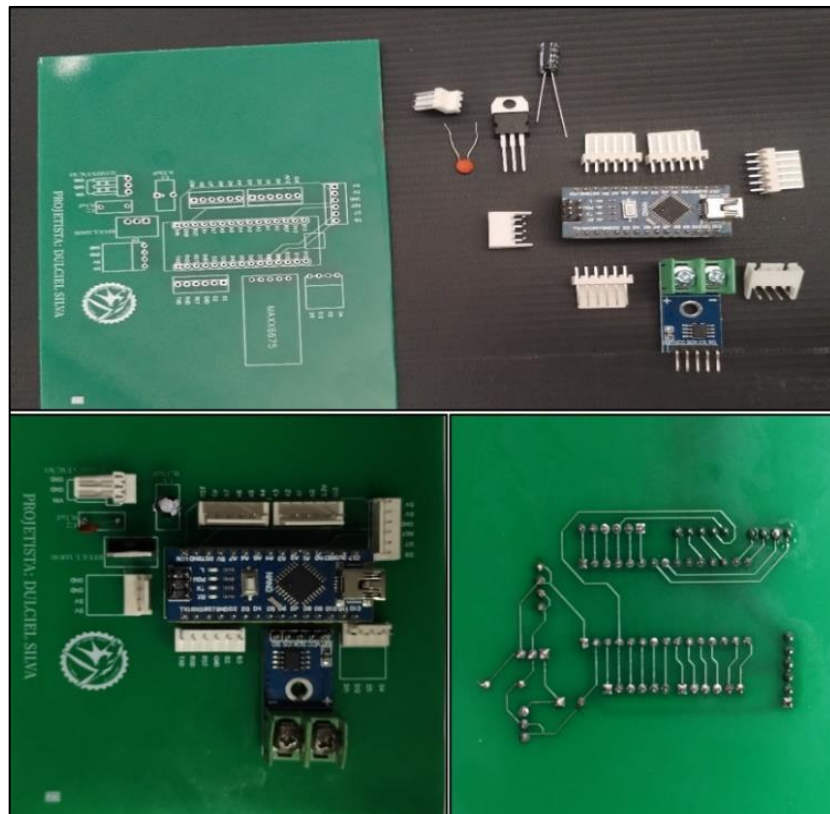
**Figura 36** - Visualização 2D e 3D do *layout* da placa principal do sistema de monitoramento



Fonte: Autoria própria, 2024.

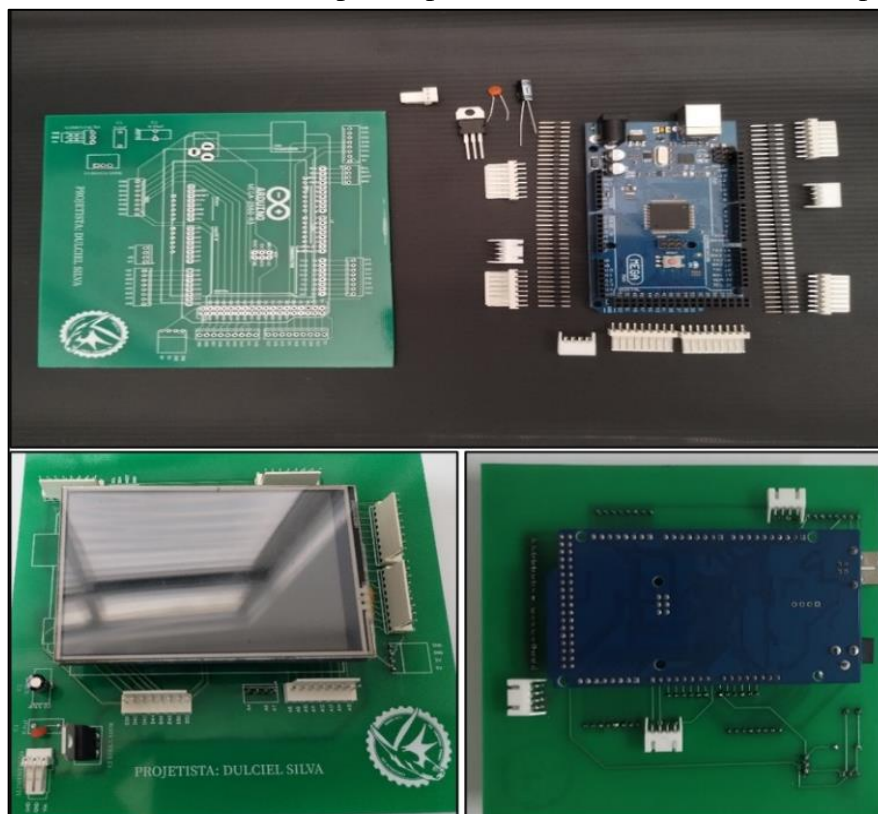
As placas primária e secundária do sistema de monitoramento foram confeccionadas com alto nível de precisão e confiabilidade, utilizando equipamentos especializados para garantir a qualidade e durabilidade do processo, todo o processo de confecção foi realizado pela empresa TEC-CI Circuitos Impressos, especializada nesse tipo de projeto. As Figuras 37 e 38 apresentam, respectivamente, as placas de circuito impresso, para as placas secundária e principal do sistema de monitoramento eletrônico desenvolvido para veículos *off-road* tipo baja.

**Figura 37** - Placa de circuito impresso para o sistema de monitoramento secundário



Fonte: Autoria própria, 2024.

**Figura 38** - Placa de circuito impresso para o sistema de monitoramento principal



Fonte: Autoria própria, 2024.



## 6. CONSIDERAÇÕES FINAIS

A implementação do sistema de monitoramento em um veículo *off-road* do tipo baja demonstrou ser uma solução eficaz. Diversos testes foram realizados para assegurar a calibração dos sensores e a comunicação entre a placa principal e a placa secundária, utilizando o protocolo de comunicação I2C.

A temperatura medida pelo termopar tipo K apresentou um erro desprezível de aproximadamente  $0,75^{\circ}\text{C}$  em relação ao sensor de temperatura do banho ultratermostático, resultando em um erro de 0,94% em relação à temperatura máxima de  $80^{\circ}\text{C}$  utilizada no ensaio. Esses resultados atestam a eficiência do sistema de medição.

Os resultados do ensaio de RPM indicam que aumentar a quantidade de ímãs no sistema de monitoramento reduz significativamente o erro nas medições de velocidade angular da CVT e da velocidade linear do veículo. Essa melhoria ocorre porque o sensor indutivo realiza mais leituras em altas velocidades. A comparação com os dados do tacômetro digital válida a eficácia desse método de medição.

A comunicação I2C entre as placas Arduino Mega 2560 Rev3 e Arduino Nano foi eficaz, permitindo o envio preciso dos dados de temperatura e RPM de acordo com as faixas definidas. A estrutura lógica para separar e enviar os pacotes de dados funcionou como esperado, comprovando a eficiência do método.

Nesse contexto, as placas Arduino se destacaram como uma ferramenta versátil e essencial para o avanço tecnológico no desenvolvimento de veículos de competição, oferecendo uma combinação de baixo custo e elevada capacidade de processamento.

## REFERÊNCIAS BIBLIOGRÁFICAS

ARDUINO, Arduino. **Trademark & Copyright**. [S. l.], 2020. Disponível em: <https://www.arduino.cc/en/trademark>. Acesso em: 16 maio 2024.

ARDUINO, Arduino. **What is Arduino?**. [S. l.], 2018. Disponível em: <https://www.arduino.cc/en/Guide/Introduction>. Acesso em: 16 maio 2024.

ARDUINO, Software. **Downloads**. [S. l.], 2024. Disponível em: <https://www.arduino.cc/en/software>. Acesso em: 20 maio 2024.

CARVALHO, Gilson Amorim; SANTOS, José Vicente Cardoso; SANTOS, Raul. ARDUINO NA AUTOMAÇÃO INDUSTRIAL: VANTAGENS, LIMITAÇÕES E PERSPECTIVAS FUTURAS. **Apoena Revista Eletrônica**, Salvador, p. 1-11, 1 jul. 2023. Disponível em: <https://transformauj.com.br/wp-content/uploads/2023/06/22.-Arduino-na-Automacao-Industrial-Vantagens-Limitacoes-e-Perspectivas-Futuras.pdf>. Acesso em: 26 out. 2024.

CIRCUITO FÁCIL, Empresa. **Módulo Sensor de Temperatura Tipo K MAX6675**. [S. l.], 2019. Disponível em: <https://www.circuitofacil.com.br/produto/modulo-sensor-de-temperatura-tipo-k-max6675/>. Acesso em: 27 jun. 2024.

CHASE, Otavio; ALMEIDA, F. J. Sistemas embarcados. **Mídia Eletrônica. Página na internet:** < [www.sbjovem.org/chase](http://www.sbjovem.org/chase) >, capturado em, v. 10, n. 11, p. 13, 2007.

DANNEGÅRD, Benjamin. **Connecting Two Nano 33 BLE Sense Boards Through I2C**. [S. l.], 14 jun. 2022. Disponível em: <https://docs.arduino.cc/tutorials/nano-33-ble-sense/i2c/>. Acesso em: 8 jun. 2024.

DATASHEETS ARDUINO MEGA 2560 REV3, Arduino. **Datasshet Arduino® MEGA 2560 Rev3**. [S. l.: s. n.], 2024. 17 p. Disponível em: <https://docs.arduino.cc/resources/datasheets/A000067-datasheet.pdf>. Acesso em: 17 maio 2024.

DATASHEETS ARDUINO NANO, Arduino. **Datasshet Arduino® Nano**. [S. l.: s. n.], 2024. 26 p. Disponível em: <https://docs.arduino.cc/hardware/nano/>. Acesso em: 16 maio 2024.

KERSCHBAUMER, Ricardo. **Microcontroladores**. [S. l.: s. n.], 2013. Disponível em: <https://professor.luzerna.ifc.edu.br/ricardo-kerschbaumer/wp-content/uploads/sites/43/2018/02/Apostila-Microcontroladores.pdf>. Acesso em: 14 maio 2024.

LI, Will. **O que é PCB de face única?**. [S. l.], 2019. Disponível em: <https://www.mokotechnology.com/pt/what-is-single-sided-pcb/>. Acesso em: 2 jul. 2024.

MAXIM INTEGRATED PRODUCTS, Incorporated. **MAX6675: Cold-Junction-Compensated K-Thermocouple- to-Digital Converter (0°C to +1024°C)**. [S. l.: s. n.], 2021. Disponível em: <https://www.analog.com/media/en/technical-documentation/datasheets/max6675.pdf>. Acesso em: 27 jun. 2024.

MELO, Leonimer Flávio de. **Proposta de Simulador Virtual para Sistema de Navegação de Robôs Móveis Utilizando Conceitos de Prototipagem Rápida**. 2007. 293 p. Tese (Mestrado em engenharia mecânica) - Universidade Estadual de Campinas, São Paulo, 2007. Disponível

em: [https://bdtd.ibict.br/vufind/Record/UNICAMP-30\\_bcdc77d24ad5263c4399dd69cbdf53ff](https://bdtd.ibict.br/vufind/Record/UNICAMP-30_bcdc77d24ad5263c4399dd69cbdf53ff). Acesso em: 11 out. 2024.

MELO, Paulo Roberto de Sousa; RIOS, Evaristo Carlos Silva Duarte; GUTIERREZ, Regina Maria Vinhais. **PLACAS DE CIRCUITO IMPRESSO: MERCADO ATUAL E PERSPECTIVAS**. Rio de Janeiro: Banco Nacional de Desenvolvimento Econômico e Social, 2001. Disponível em: <https://web.bndes.gov.br/bib/jspui/handle/1408/13440>. Acesso em: 2 jul. 2024.

NERI, Hermes Gustavo Fernandes. **UTILIZAÇÃO DA PLATAFORMA ARDUINO PARA CONTROLE DE EXPERIMENTOS REMOTOS DE FÍSICA**. Orientador: Eduardo Kojy Takahashi. 2014. Trabalho de conclusão de curso (Licenciatura em física) - Universidade Federal de Uberlândia, Uberlândia, 2014. Disponível em: <https://repositorio.ufu.br/bitstream/123456789/26520/1/UtilizacaoPlataformaArduino.pdf>. Acesso em: 26 out. 2024.

NETO, José Barroso de Carvalho. **APLICAÇÃO DO EFEITO SEEBECK NA CONVERSÃO DA ENERGIA TÉRMICA EM ELÉTRICA**. Orientador: Rudson de Souza Lima. 2018. 30 p. Trabalho de conclusão de curso (Bacharelado em Ciência e Tecnologia) - Universidade Federal Rural do Semi-Árido, Caraúbas, 2018. Disponível em: <https://repositorio.ufersa.edu.br/server/api/core/bitstreams/721e0628-11de-4830-bda5-0b7fd5f111db/content>. Acesso em: 25 jun. 2024.

NPX SEMICONDUCTORS, User Manual. **UM10204: I2C-bus specification and user manual**. [S. l.], 1 out. 2021. Disponível em: <chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.nxp.com/docs/en/user-guide/UM10204.pdf>. Acesso em: 8 jun. 2024.

PREDOLIN, R. E. (2017). **Desenvolvimento de um sistema de aquisição de dados usando plataforma aberta**. Dissertação (Mestrado em Engenharia Mecânica) - Programa de Pós-graduação em Engenharia Mecânica da Universidade Estadual Paulista "Júlio de Mesquita Filho", São Paulo, Brasil.

SAE BRASIL, Baja. **Histórico**. [S. l.], 2022. Disponível em: <https://saebrasil.org.br/programas-estudantis/baja-sae-brasil/>. Acesso em: 8 maio 2024.

SANTOS, Geise. **Prototipação: Plataformas de hardware e software para sistemas embarcados**. [S. l.], 2021. Disponível em: <https://embarcados.com.br/prototipacao-plataformas-de-hardware-e-software-para-sistemas-embarcados/>. Acesso em: 17 jul. 2024.

SOBRINHO, Antônio Araújo; SOUZA, Gilberto Morel de Paula e. **Física Termica Teorica e Experimental**. [S. l.]: CEFET-RN, 2006. 152 p. Disponível em: <https://memoria.ifrn.edu.br/bitstream/handle/1044/1038/Fisica%20Termica%20Teorica%20e%20Experimental%20-%20Ebook.pdf?sequence=1>. Acesso em: 26 out. 2024.

SÖDERBY, Karl. **Using the Serial Monitor tool**. [S. l.], 2024a. Disponível em: <https://docs.arduino.cc/software/ide-v2/tutorials/ide-v2-serial-monitor/>. Acesso em: 20 maio 2024.

SÖDERBY, Karl. **Using the Serial Plotter Tool**. [S. l.], 2024b. Disponível em: <https://docs.arduino.cc/software/ide-v2/tutorials/ide-v2-serial-plotter/>. Acesso em: 20 maio 2024.

SÖDERBY, Karl; HYLÉN, Jacob. **Getting Started with Arduino IDE 2c**. [S. l.], 2024. Disponível em: <https://docs.arduino.cc/software/ide-v2/tutorials/getting-started-ide-v2/>. Acesso em: 20 maio 2024.

TONIOLO, Cristiano Marçal. **Sistemas Embarcados**. 1. ed. [S. l.]: Editora e Distribuidora Educacional S.A., 2018. 196 p.

ZELENOVSKY, Ricardo; MENDONÇA, Alexandre. **Microcontroladores: Programação e Projeto com a Família 8051**. [S. l.]: MZ Editora, 2013. 448 p.

ZEYMER, Juliana Soares; SILVA, Carlos Henrique Osório; SCHNEIDER, Murilo Gehrmann; AGUIAR, Eduardo de; SELINGER, Tarcísio Cardoso. **Comparação entre a Exatidão de Sensores Analógicos e Digitais Utilizados em Sistemas de Termometria**. [S. l.], 17 fev. 2021. Disponível em: <https://ojs.brazilianjournals.com.br/ojs/index.php/BJAER/article/view/24749>. Acesso em: 10 jun. 2024.