

TXAI VIEIRA GARCIA

**Comunicação wireless para controle embarcado de  
reômetro**

São Paulo  
2015

TXAI VIEIRA GARCIA

**Comunicação wireless para controle embarcado de  
reômetro**

Trabalho de formatura apresentado à Escola  
Politécnica da Universidade de São Paulo  
para obtenção do título de Bacharel em En-  
genharia

Orientador: Prof. Dr. Gustavo Pamplona Reh-  
der

Coorientador: Prof. Dr. Rafael Giuliano Pi-  
leggi

São Paulo  
2015

TXAI VIEIRA GARCIA

**Comunicação wireless para controle embarcado de  
reômetro**

Trabalho de formatura apresentado à Escola  
Politécnica da Universidade de São Paulo  
para obtenção do título de Bacharel em En-  
genharia

Área de Concentração:Sistemas Eletrônicos  
Orientador: Prof. Dr. Gustavo Pamplona Reh-  
der

Coorientador: Prof. Dr. Rafael Giuliano Pi-  
leggi

São Paulo  
2015

Catalogação-na-publicação

m\_2015 H

Garcia, Txai Vieira

Comunicação wireless para controle embarcado de reômetro / T. V. Garcia  
- São Paulo, 2015.  
78 p.

Trabalho de Formatura - Escola Politécnica da Universidade de São  
Paulo. Departamento de Engenharia de Sistemas Eletrônicos.

1. microcontrolador 2. sistema embarcado 3. reômetro I. Universidade de  
São Paulo. Escola Politécnica. Departamento de Engenharia de Sistemas  
Eletrônicos II.t.

*A meus pais, por me apoiarem e me ensinarem o valor do trabalho duro.*

*Aos meus irmãos, por estarem sempre ao meu lado.*

*À minha noiva, por ser meu porto seguro nos momentos realmente difíceis*

# Agradecimentos

Agradeço principalmente a Deus pela disposição e inteligência e pela oportunidade de realizar este trabalho. Agradeço também à minha família, pela compreensão nos momentos difíceis e pelo amor incondicional que me motivou a seguir adiante. Agradeço à minha noiva por estar sempre ao meu lado nas pequenas conquistas e nos momentos em que tudo parecia perdido.

Agradecimentos especiais aos meus orientadores, pela oportunidade de poder participar de um projeto tão ambicioso e fascinante.

Agradeço também à Control iD, por todas as lições ali aprendidas, e pela ajuda inestimável com conselhos, materiais e equipamentos.

*"Eu não creio que exista algo mais emocionante para o coração humano do que a emoção sentida pelo inventor quando ele vê alguma criação da mente se tornando algo de sucesso. Essas emoções fazem o homem esquecer comida, sono, amigos, amor, tudo.*

*(Nikola Tesla)*

## Resumo

O objetivo deste projeto é desenvolver um sistema embarcado de controle de reômetro, utilizando comunicação sem fio, visando o aperfeiçoamento do projeto de reômetro portátil de baixo custo, desenvolvido pelo Laboratório de Microestruturas e ecoeficiência da Escola Politécnica da Universidade de São Paulo. Procurou-se projetar um sistema que fosse autônomo e robusto e que realizasse o controle dos experimentos programados, a aquisição e armazenamento de dados, além do envio destes dados à usuários que pudessem conectar-se ao sistema por meio de dispositivos dotados de comunicação wifi. O projeto engloba diversas tecnologias tais como microcontroladores, memórias não voláteis, transceptores wifi, sistemas analógicos de aquisição de dados e geração de sinais e sistemas operacionais de tempo real. Neste sistema, o microcontrolador utiliza um sistema operacional de tempo real para receber, decodificar e responder requisições HTTP através do módulo wifi e controlar um servomotor e um motor DC que compõem o reômetro. Através de duas interfaces analógicas os sinais de referência são gerados e enviados ao servomotor, e os sinais de torque e velocidade do servomotor são adquiridos, armazenados na memória não volátil e enviados para o usuário através da interface wifi. O sistema ainda conta com comunicação através de cabo USB e comunicação através de interface serial RS232 para programar o servomotor. Foram realizados testes de validação do hardware projetado e da arquitetura de software, obtendo-se resultados satisfatórios de desempenho e confiabilidade, comprovando que os requisitos de engenharia estabelecidos ao longo do projeto foram atendidos. Por fim, é apresentada uma conclusão elucidando sugestões de melhoria e limitações do projeto.

**Palavras-chave:** reômetro, embarcado, wireless.

# Abstract

This project proposes an embedded rheometer control system, using wireless communication, aiming to improve the portable low cost rheometer project, developed by the Laboratório de microestruturas e ecoeficiência [Laboratory of microstructures and eco-efficiency] of Escola Politécnica da Universidade de São Paulo. We tried to design an autonomous and reliable system that could control scheduled experiments, acquiring and storing data, besides sending these data to users connected to the system through devices that are capable of communicating through wifi. The project employs multiple technologies as microcontrollers, non volatile memories, wifi transceivers, analog signal generation and acquisition systems and real time operational systems. In this system, the microcontroller use a real time operational system to receive, parse and answer HTTP requests through the wifi module and control a servo-motor and a DC motor that make up the rheometer. Through two analog interfaces the reference signals are generated and send to the servo-motor, and the signals proportional to torque and speed are acquired, stored in memory and sent to the user through wifi. Also, the system has USB communication and RS232 communication to program the servo-motor. We made validation tests for the hardware and the software architecture, obtaining satisfying results of performance and reliability, showing that the engineering requirements established along the project were met. Lastly, we show a conclusion elucidating improvement suggestions and limitations of the project.

**Keywords:** rheometer, embedded, wireless.

# **Lista de ilustrações**

Figura 1 – Reômetro tipicamente utilizado em ensaios reométricos . . . . .	14
Figura 2 – Árvore de objetivos . . . . .	21
Figura 3 – Exemplo de comunicação via HTTP . . . . .	26
Figura 4 – Mensagem de requisição HTTP (GUIDE, 2005a) . . . . .	26
Figura 5 – Mensagem de resposta HTTP (GUIDE, 2005b) . . . . .	27
Figura 6 – Interconexão de dispositivos na interface SPI (PARADIGM, 2015) . .	28
Figura 7 – Interconexão de dispositivos na interface SPI (PARADIGM, 2015) . .	29
Figura 8 – Circuito de conversão de amplitude para o ADC interno . . . . .	34
Figura 9 – Circuito de conversão de amplitude para o DAC interno . . . . .	35
Figura 10 – Diagrama de blocos de nível 0 . . . . .	39
Figura 11 – Diagrama de blocos de nível 1 . . . . .	40
Figura 12 – Diagrama de fluxo de nível 1 . . . . .	41
Figura 13 – Diagrama de fluxo da inicialização do sistema . . . . .	43
Figura 14 – Diagrama de fluxo do modo de testes . . . . .	43
Figura 15 – FRDM-KL25Z: placa de desenvolvimento do KL25Z128 . . . . .	48
Figura 16 – Placa de alimentação do módulo wifi: vista inferior . . . . .	49
Figura 17 – Placa de alimentação do módulo wifi: vista superior . . . . .	49
Figura 18 – Vista do conjunto montado e operando . . . . .	49
Figura 19 – Tela do WebServer aberta no navegador Web Firefox . . . . .	50
Figura 20 – Foto da placa, lado TOP . . . . .	52
Figura 21 – Foto da placa, lado BOTTOM . . . . .	52
Figura 22 – Conexão da placa com a programadora . . . . .	53
Figura 23 – Conexão com a placa controladora do motor de plataforma . . . . .	54
Figura 24 – Servomotor conectado à placa controladora . . . . .	55
Figura 25 – Conexão analógica . . . . .	56
Figura 26 – Conexão serial RS232 . . . . .	57
Figura 27 – Exemplo de objeto em C++ (header) . . . . .	58
Figura 28 – Exemplo de objeto em C++ (source) . . . . .	58
Figura 29 – Seleção de modo de programação . . . . .	60
Figura 30 – Modo de programação . . . . .	60
Figura 31 – Página html sendo servida pela placa controladora . . . . .	61

## **Lista de tabelas**

Tabela 1 – Tabela de pesos de nível 0 . . . . .	21
Tabela 2 – Tabela de pesos das funcionalidades de autonomia . . . . .	22
Tabela 3 – Tabela de pesos das funcionalidades de comunicação . . . . .	22
Tabela 4 – Tabela de pesos das funcionalidades de confiabilidade . . . . .	22
Tabela 5 – Requisitos de engenharia e suas respectivas justificativas . . . . .	25
Tabela 6 – Benchmark competitivo . . . . .	30
Tabela 7 – Opções para requisitos de engenharia de hardware . . . . .	31
Tabela 8 – Opções para requisitos de engenharia de comunicação . . . . .	31
Tabela 9 – Opções para requisitos de engenharia de software . . . . .	32
Tabela 10 – Análise de forças/fraquezas para a comunicação sem fio . . . . .	33
Tabela 11 – Análise de forças/fraquezas para a geração de sinais . . . . .	35
Tabela 12 – Análise de forças/fraquezas para a fonte simétrica . . . . .	37
Tabela 13 – Lista de materiais . . . . .	47

## **Lista de abreviaturas e siglas**

ADC	Analog Digital Converter
DAC	Digital Analog Converter
HTTP	Hyper Text Transfer Protocol
SPI	Serial Peripheral Interface
USB	Universal Serial Bus

# Sumário

<b>Lista de ilustrações . . . . .</b>	<b>9</b>
<b>Introdução . . . . .</b>	<b>14</b>
<b>1 FORMULAÇÃO DO PROBLEMA . . . . .</b>	<b>15</b>
<b>1.1 Identificação do problema . . . . .</b>	<b>15</b>
<b>1.2 Identificação das necessidades . . . . .</b>	<b>15</b>
<b>1.3 Declaração das necessidades . . . . .</b>	<b>16</b>
<b>1.4 Declaração dos objetivos . . . . .</b>	<b>16</b>
<b>1.5 Pesquisa e levantamento da situação . . . . .</b>	<b>17</b>
<b>1.5.1 Visão geral . . . . .</b>	<b>17</b>
<b>1.5.2 Tecnologias relevantes . . . . .</b>	<b>19</b>
<b>1.5.3 Árvore de objetivos . . . . .</b>	<b>20</b>
<b>2 ESPECIFICAÇÃO DE REQUISITOS . . . . .</b>	<b>23</b>
<b>2.1 Requisitos de marketing . . . . .</b>	<b>23</b>
<b>2.2 Requisitos de engenharia . . . . .</b>	<b>23</b>
<b>2.3 Restrições . . . . .</b>	<b>25</b>
<b>2.4 Padrões técnicos . . . . .</b>	<b>26</b>
<b>2.5 Benchmark competitivo . . . . .</b>	<b>29</b>
<b>3 GERAÇÃO DE CONCEITO . . . . .</b>	<b>31</b>
<b>4 DECOMPOSIÇÃO FUNCIONAL . . . . .</b>	<b>39</b>
<b>4.1 Nível 0 . . . . .</b>	<b>39</b>
<b>4.2 Nível 1 . . . . .</b>	<b>39</b>
<b>4.3 Nível 2 . . . . .</b>	<b>42</b>
<b>5 GERENCIAMENTO DE PROJETO . . . . .</b>	<b>46</b>
<b>5.1 Cronograma . . . . .</b>	<b>46</b>
<b>5.2 Principais riscos . . . . .</b>	<b>46</b>
<b>5.3 Custos . . . . .</b>	<b>46</b>
<b>6 PROVA DE CONCEITO . . . . .</b>	<b>48</b>
<b>7 PROTÓTIPO . . . . .</b>	<b>51</b>
<b>7.1 Hardware . . . . .</b>	<b>51</b>
<b>7.2 Software . . . . .</b>	<b>57</b>

<b>8</b>	<b>TESTES E VALIDAÇÃO .....</b>	<b>61</b>
<b>9</b>	<b>CONCLUSÃO .....</b>	<b>63</b>
	<b>REFERÊNCIAS .....</b>	<b>64</b>
	<b>ANEXOS</b>	<b>66</b>

# Introdução

O projeto apresentado neste documento tem por objetivo atender uma solicitação do Laboratório de Microestrutura e Eficiência do Departamento de Engenharia Civil da Escola Politécnica da Universidade de São Paulo, o qual desenvolve como parte de suas atividades de pesquisa, ensaios reométricos de diversas composições de cimento, material amplamente utilizado na construção civil.

Estes ensaios consistem na análise das propriedades reológicas, ou seja, a relação entre a tensão e taxa de cisalhamento dos materiais ensaiados, e são realizadas por meio de um aparelho denominado reômetro, atualmente em desenvolvimento através de uma parceria entre o referido Laboratório e o Departamento de Sistemas Eletrônicos da Escola Politécnica.

O reômetro consiste numa plataforma elevatória, onde é apoiada uma cuba com a amostra de cimento, e de um braço rotatório, responsável por misturar o cimento. Tanto a plataforma quanto o braço são controlados por motores servomecânicos de tensão alternada. Os motores também possuem sensores responsáveis por gerar os dados relativos ao ensaio reométrico (velocidade de rotação e torque do motor).

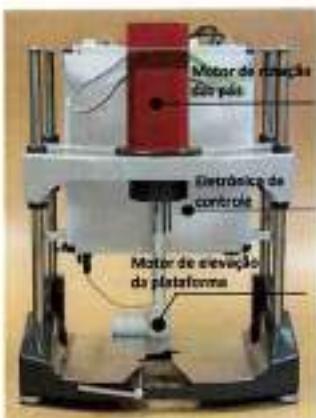


Figura 1 – Reômetro tipicamente utilizado em ensaios reométricos

Atualmente o controle deste dispositivo é feito exclusivamente por um computador ligado a uma placa conversora Analógico-Digital através de um cabo USB. A placa, por sua vez, comunica-se com os motores por meio de interface serial RS232 ou sinais digitais. Esta configuração, apesar de funcional, apresenta uma série e dificuldades e limitações que serão elencadas a seguir, e que devem ser sanadas ou diminuídas através do sistema aqui proposto.

# 1 Formulação do problema

## 1.1 Identificação do problema

O controle do reômetro, da forma como é feita hoje, apresenta quatro problemas principais:

1. a utilização do cabo USB para comunicação entre reômetro e computador limita a distância que este último pode ficar da máquina, expondo-o à poeira e à água presentes no ambiente do ensaio, o que pode danificá-lo permanentemente;
2. a placa conversora analógica-digital não oferece uma interface suficientemente imune à ruídos de natureza eletromagnética gerados pelos motores ou por equipamentos próximos. Isto torna a comunicação serial não confiável, afetando a confiabilidade do ensaio;
3. a necessidade de cabos de comunicação entre o computador e o reômetro não permite que haja uma completa isolação entre o compartimento onde estão localizados circuitos eletrônicos e o ambiente externo. O reômetro pode então ser danificado quando exposto à água e resíduos sólidos;
4. ensaios deste tipo podem levar de 15h a 24h e caso haja uma falha de comunicação entre o computador e o equipamento o ensaio precisa ser interrompido e reiniciado o que gera um custo enorme de tempo.

## 1.2 Identificação das necessidades

As necessidades do cliente podem ser agrupadas basicamente em três grupos distintos:

### Comunicação

- O sistema deve possuir comunicação sem fio entre o computador e o reômetro.
- Deve haver a possibilidade de uma conexão cabeada caso a comunicação sem fio não seja possível.
- O computador deve ficar a, no mínimo, 5 metros de distância do local do ensaio.

## Interface

- a interface entre o controlador e os motores deve ser imune a ruído eletromagnético.
- as interfaces de comunicação antigas devem ser mantidas por questões de retrocompatibilidade.

## Autonomia

- o sistema deve ser capaz de continuar operando sem o computador.
- o sistema deve apresentar os dados colhidos ao computador, tão logo haja comunicação entre os dois.

### 1.3 Declaração das necessidades

O reômetro controlado pelo computador através de comunicação cabecada possui vulnerabilidade ao meio externo que põe em risco tanto a operação da máquina quanto do computador, oferecendo um risco financeiro de aproximadamente R\$7000,00, segundo estimativa do custo médio de um computador, dos motores e da placa de conversão acoplados ao reômetro. Além de dificultar a pesquisa devido aos ensaios de 15h a 24h que eventualmente são perdidos devido à interferência de ruído ou falha de comunicação. Sendo assim, deve-se projetar um sistema de controle que ofereça comunicação sem fio, imunidade a ruído e autonomia de modo a garantir a confiabilidade do sistema. Soluções parecidas com a que pretende-se adotar são utilizadas em outros reômetros existentes no mercado, tais como os da Anton-Paar que oferecem controle embarcado dos equipamentos e interface via rede de computadores. (PAAR, 2015)

### 1.4 Declaração dos objetivos

O objetivo deste projeto é projetar um sistema embarcado autônomo que possa se comunicar com um computador por uma interface sem fio e que ofereça proteção a ruído eletromagnético. O controle principal será executado por um microcontrolador, dotado de memória externa de modo a poder armazenar os dados do ensaio e controlá-lo automaticamente, sem a necessidade do computador. Também será dotado de comunicação em alta frequência, para que possa receber e enviar dados para o computador. Por fim, terá interfaces analógicas para controle dos motores, com conexão BNC para a utilização de cabos blindados contra interferência eletromagnética. O sistema de comunicação também deverá ser protegido contra ruído eletromagnético. O software embarcado de controle

será executado em sistema operacional multithreading e terá um servidor web integrado, responsável pela comunicação com o computador, através do modelo cliente-servidor.

## 1.5 Pesquisa e levantamento da situação

### 1.5.1 Visão geral

As análises reológicas têm como objetivo analisar os comportamentos mecânicos das substâncias e estabelecer as respectivas leis de comportamento. Ela engloba numerosas disciplinas fundamentais, como a resistência dos materiais, a mecânica dos fluidos, a plasticidade, etc, e constituem um ramo muito importante para a construção civil e para a pesquisa de novos materiais de construção.

Fundamentalmente, a reologia estuda o fluxo e a deformação da matéria, avaliando as relações entre a tensão de cisalhamento aplicada, e a deformação em determinado período de tempo, estabelecendo a relação entre o fluxo ou deformação da matéria e as forças aplicadas sobre ela. (JOHN, 2006)

O equipamento utilizado para realizar ensaios reológicos é denominado reômetro. O reômetro é o equipamento dedicado à determinação de parâmetros como viscosidade e escoamento de argamassas e concretos. Atuam através de dois princípios básicos de funcionamento: ou um torque controlado é aplicado ao fluido e é medida a tensão de cisalhamento resultante; ou é aplicada uma tensão de cisalhamento controlada e é medido o torque resultante. (JOHN, 2006)

Sendo assim, o reômetro possui como elemento básico um motor que possa ter velocidade (relativa à tensão de cisalhamento) ou torque controlados e elementos de medição capazes de medir estas duas grandezas. Além disso, é necessário uma unidade computacional para receber e analisar os dados, a qual, geralmente consiste de um computador do tipo desktop ou notebook, ligados à máquina por cabo serial RS232 e, mais recentemente, por cabo USB.

A estrutura de um reômetro então é definida por:

- uma plataforma, geralmente móvel, com altura controlada, onde fica apoiada a amostra a ser submetida ao ensaio reológico;
- um motor com velocidade e torques controláveis;
- pás de metal para realizar a mistura;
- um dispositivo de interface analógico-digital que permita a aquisição de dados para o computador;

- um computador ligado ao dispositivo de interface analógico-digital;

Esta montagem é adequada ao ambiente laboratorial, onde as análises são realizadas em locais fixos, geralmente em bancadas. Com o avanço das técnicas de construção civil, entretanto, sobretudo durante a década de 90, houve uma demanda por realizar-se análises reológicas *in loco*, com a finalidade de prevenir falhas na composição da argamassa e concreto, evitando acidentes futuros. Sendo assim, há uma demanda, ainda não totalmente suprida, de utilizar-se reômetros portáteis, que possam ser deslocados conforme a necessidade em diferentes locais como canteiros de obra, construções de barragens, etc. As soluções atuais envolvem importação de materiais de alto custo. Assim, pesquisas dirigidas no sentido de desenvolver equipamentos portáteis de baixo custo e nacionais foram empreendidas a partir do ano 2000, entre elas está a pesquisa empreendida pelo Laboratório de Microestruturas e ecoeficiência da EPUSP.

De encontro à este paradigma, vêm os avanços na área de sistemas embarcados e o advento da internet das coisas. O uso de sistemas microcontrolados vêm crescendo nos últimos anos estando presente em uma infinidade de aparelhos. Segundo pesquisa realizada pela Universidade do Paraná, uma pessoa entra em contato, em média, com cerca de 20 sistemas embarcados por dia. Sendo que esse número deve chegar a 350 até 2020. (ESC, 2015)

A versatilidade e o baixo custo (atualmente é possível encontrar microcontroladores ao custo de US\$3,00), fazem do microcontrolador uma escolha ideal para o controle de equipamentos de laboratório como o reômetro.

Um sistema embarcado, comporta-se como um minicomputador, possuindo uma unidade central de processamento, memórias e dispositivos eletrônicos auxiliares que permitem a aquisição de dados, produção de sinais, comunicação com dispositivos externos, tais como computadores ou outros sistemas embarcados, entre outros. A unidade de processamento é geralmente programável e executa todo o processo de aquisição de dados e controle dos dispositivos auxiliares. A popularização das memórias flash, notadamente na década de 90, permitiu também que os sistemas embarcados tivessem acesso a uma tecnologia barata, de grande capacidade e tamanho reduzido de armazenamento permanente de dados. Esta característica é altamente desejável, pois permite o armazenamento de grande volume de dados, oriundos dos experimentos realizados.

Por fim, há a crescente demanda para que esses sistemas estejam conectados remotamente a uma rede de computadores. Assim, para executar a comunicação sem fio entre o reômetro e o computador, uma escolha natural pela tecnologia Wi-fi permite que o equipamento esteja inserido no campo da Internet das Coisas, possibilitando que futuramente os dados da análise possam ser disponibilizados na Internet em tempo real para outros pesquisadores envolvidos com a análise.

Este projeto visa unir estas três tecnologias, produzindo um sistema capaz de atender não só a necessidade imediata de comunicação sem fio e autonomia, como também tornar a máquina escalável para a rápida adoção de novas tecnologias e modos de utilização.

### 1.5.2 Tecnologias relevantes

#### Microcontrolador

Um microcontrolador é um SoC ("System on a Chip" ou "Sistema contido num chip") composto por um processador programável, memória ROM, memória RAM e periféricos tais como interface USB, comunicação SPI, I<sup>2</sup>C, conversores analógico digitais e digital analógicos, entre outros. O tipo de periféricos e a quantidade destes variam de acordo com cada microcontrolador, dependendo do fabricante, modelo, etc. Por serem programáveis são muito utilizados em sistemas embarcados para aplicações em controle, internet das coisas, etc. (ELETRONICS, 2015)

#### Memória Flash

Memória Flash é uma memória não-volátil, programável eletronicamente (EEPROM). Isto significa que ela pode armazenar dados de forma permanente, mesmo que não haja energia elétrica alimentando o circuito. Além disso, o fato de ela ser programada eletronicamente permite que ela seja usada em conjunto com um microcontrolador para armazenar dados processados pelo processador de maneira permanente.

A memória flash é dividida em páginas e blocos, sendo cada bloco um conjunto de páginas e cada página um conjunto de 2048 bits em geral. A informação armazenada pode ser lida e gravada por página. No entanto é necessário apagar os dados antigos antes de programar novas informações em uma página, e o apagamento só pode ser feito por blocos. (WIKIPEDIA, 2015b)

#### ADC/DAC

Conversores analógico digitais (ADC) e Conversores digitais analógicos (DAC) são dispositivos eletrônicos que permitem que sinais analógicos sejam convertidos em sinais digitais para que estes sejam processados por um microcontrolador e também que sinais digitais sejam convertidos em sinais analógicos para serem enviados para outros equipamentos. Os chips disponíveis no mercado permitem que sinais de grande magnitude sejam convertidos e transportados através de comunicação SPI ou I<sup>2</sup>C, permitindo o controle de equipamentos de grande porte, tal como motores, e a medição de fontes de alimentação.(WIKIPEDIA, 2015a)

### Wi-fi

Wi-Fi ou Wifi é uma tecnologia sem fio que permite a um dispositivo conectar-se à uma rede local de computadores (LAN) através de uma banda de 2,4GHz UHF, permitindo acesso à Internet e a troca de informações entre os equipamentos conectados à rede, através do protocolo TCP/IP, o qual é responsável por gerenciar as conexões entre computadores em uma rede. (WIKIPEDIA, 2015c)

Devido à forte presença em tablets, smartphones e computadores, a tecnologia Wi-fi vem sendo adotada para realizar a comunicação entre estes dispositivos e sistemas embarcados, eliminando, assim, a necessidade do desenvolvimento de drivers de comunicação e da adaptação de algum hardware específico à cada equipamento com a qual se deseja comunicar-se. Esta forma de comunicação é conhecida como Internet das Coisas e permite que qualquer sistema embarcado possa obter acesso à Internet e comunicar-se com computadores.

Ainda, a tecnologia Wi-fi permite que se realizem comunicações criptografadas, utilizando o protocolo HTTPS, através de SSL, tornando a conexão segura e robusta.

### 1.5.3 Árvore de objetivos

A figura 2 apresenta a árvore de objetivos do projeto. As necessidades do cliente foram agrupadas em três necessidades principais e foram desdobradas em requisitos que devem compor o sistema.

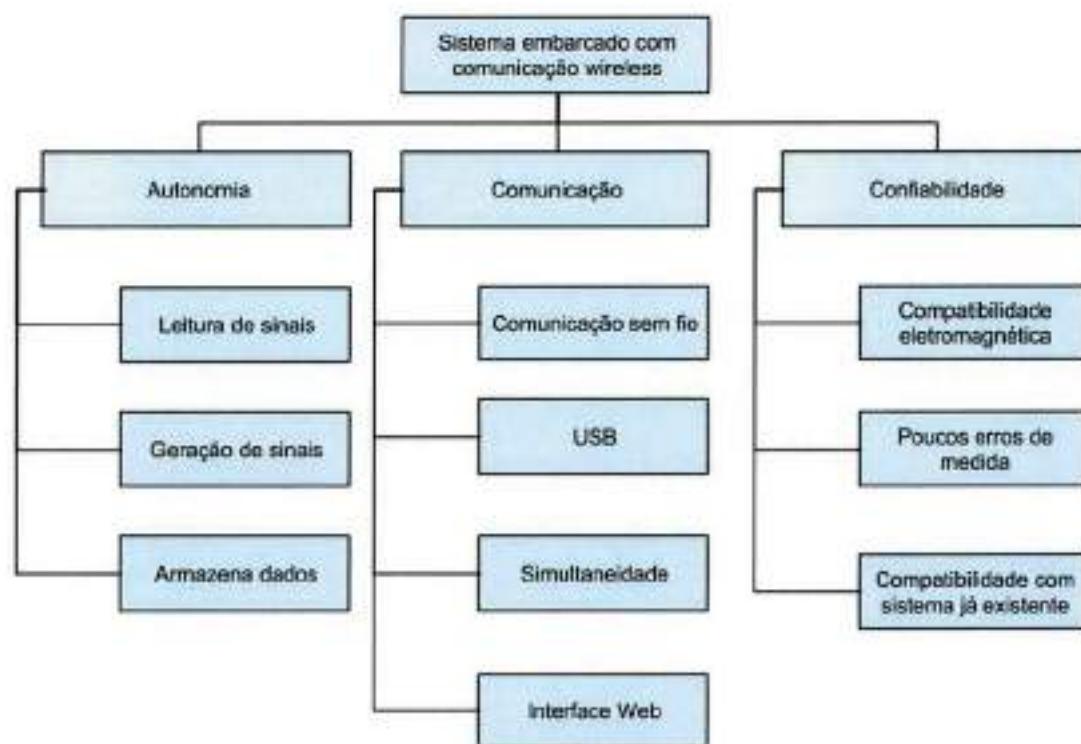


Figura 2 – Árvore de objetivos

As tabelas 1, 2, 3 e 4 apresentam os pesos relativos de cada item da árvore de objetivo

	Autonomia	Comunicação	Confiabilidade	$m_g$	$\omega$
Autonomia	1	2	3	1.82	0.54
Comunicação	1/2	1	2	1	0.29
Confiabilidade	1/3	1/2	1	0.55	0.16

Tabela 1 – Tabela de pesos de nível 0

	Leitura de sinais	Geração de sinais	Armazena dados	mg	$\omega$
Leitura de sinais	1	1	1/2	0.79	0.25
Geração de sinais	1	1	1/2	0.79	0.25
Armazena dados	2	2	1	1.58	0.50

Tabela 2 – Tabela de pesos das funcionalidades de autonomia

	Comunicação sem fio	USB	Simultaneidade	Interface Web	mg	$\omega$
Comunicação sem fio	1	3	3	3	2.28	0.49
USB	1/3	1	2	2	1.07	0.23
Simultaneidade	1/3	1/2	1	1/4	0.45	0.09
Interface Web	1/3	1/2	4	1	0.90	0.18

Tabela 3 – Tabela de pesos das funcionalidades de comunicação

	EMC	Poucos erros	Compatibilidade	mg	$\omega$
EMC	1	1/3	1/3	0.48	0.14
Poucos erros	3	1	2	1.81	0.53
Compatibilidade	3	1/2	1	1.14	0.33

Tabela 4 – Tabela de pesos das funcionalidades de confiabilidade

## 2 Especificação de requisitos

### 2.1 Requisitos de marketing

Os requisitos de marketing, conforme levantados com o Laboratório de Microestrutura e Eco eficiência são os seguintes:

1. O usuário deve ser capaz de conectar-se ao reômetro através de um computador ou tablet sem a necessidade de cabos ou hardware adicional.
2. Vários usuários devem conseguir conectar-se ao sistema simultaneamente.
3. Deve armazenar os dados de até 10 ensaios.
4. Deve comunicar-se com os motores através das interfaces disponíveis.
5. Ler e enviar os dados de velocidade e torque do motor para o usuário.
6. Deve ler os dados com a menor taxa de erro possível.
7. Deve oferecer uma fácil integração com o software de controle já existente.
8. Deve ser imune ao ruído elétrico proveniente da própria operação do reômetro.
9. Deve oferecer interface de comunicação cabeada em caso de falha da comunicação sem fio.

### 2.2 Requisitos de engenharia

A tabela 5 apresenta os requisitos de engenharia, quais requisitos de marketing eles atendem e suas devidas justificativas.

Requisitos de marketing	Requisito de engenharia	Justificativa
1,2	Possuir comunicação sem fio com alcance máximo de, no mínimo, 10 metros.	A distância de 10 metros é a necessária para proteger os equipamentos do usuário da poeira e umidade do ensaio.

3	Possuir memória de, no mínimo, 128MB	Essa é a quantidade de memória necessária para armazenar 10 ensaios e mais os dados de configuração.
4,5	Ler sinais de tensão na faixa de +10V a -10V com 12 bits de precisão	A interface analógica do motor produz sinais nesta faixa de tensão que são proporcionais ao torque e à velocidade do motor
4,5	Gerar sinais de tensão na faixa de +10V a -10V com 12 bits de precisão	O motor é controlador por sinais desta magnitude
4	Possuir suporte à comunicação serial	A configuração do motor é feita através de comunicação serial
6	Ler os sinais a uma taxa mínima de 50 amostras/seg	Essa é a taxa que reduz os erros de medida a praticamente zero segundo testes realizados.
8	Possuir fonte de alimentação independente da fonte de alimentação do reômetro capaz de prover +12V, -12V e +5V e capacidade de corrente de até 1A.	Uma fonte de tensão independente evita interferência entre a operação dos motores e o controlador. A interface analógica utiliza os sinais de +12V e -12V para produzir os sinais para o motor e precisa de +5V para alimentar o circuito digital. O consumo de corrente é decorrente das simulações dos circuitos que compõem o sistema

2	Possuir sistema operacional que permita execução de firmware multithreading, simultâneas	Para oferecer conexões simultâneas o dispositivo deve executar várias operações simultâneas, o que só é possível com o uso de threads.
7	Possuir servidor web embarcado com API para integração	A API web provê integração com qualquer linguagem de programação.

Tabela 5 – Requisitos de engenharia e suas respectivas justificativas

## 2.3 Restrições

### Tecnologia

#### Microcontrolador

O microcontrolador utilizado deve ser o KL25Z128, com processador ARM Cortex M0, da Freescale, operando a 48 MHz, pois este microcontrolador está disponível em larga escala no Departamento de Sistemas Eletrônicos.

#### Reprodutibilidade

A documentação de projeto e a montagem do sistema devem ser facilmente executáveis pelo Laboratório de Microestruturas.

#### Compatibilidade

O sistema deve ser compatível com a montagem atual do reômetro, podendo ser facilmente revertido para a montagem original em caso de falha.

#### Sustentabilidade

O sistema deve ser tolerante a falhas, permitindo a execução dos ensaios com o mínimo ou nenhum suporte técnico.

## 2.4 Padrões técnicos

O controlador embarcado de reômetro deve obedecer uma série de padrões técnicos relativos, principalmente, às diversas interface de comunicação.

### HTTP

O protocolo HTTP/1.1, definido pela norma RFC2616 é um protocolo de comunicação entre os computadores clientes e os gateways e proxies, e entre estes e outros sistemas conectados à rede.

É um protocolo de requisição e resposta. O cliente envia uma requisição ao servidor requisitando um método e informando o endereço do dado requisitado (URI) e o tipo de dado requisitado. O servidor responde aplicando o método requisitado e informando o status da operação. Além disso, uma vez que o cliente está conectado ao servidor, a conexão deve manter-se ativa até que um dos dois agentes encerre a comunicação.

A figura 3 exemplifica este processo para o método GET.

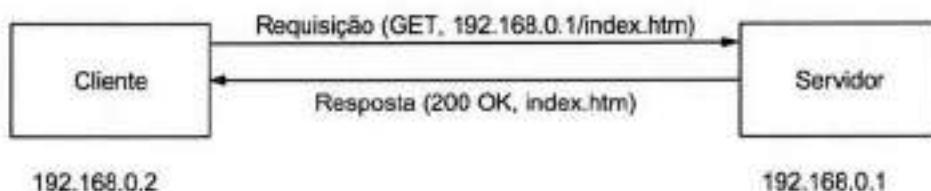


Figura 3 – Exemplo de comunicação via HTTP

As figuras 4 e 5 mostram o formato das mensagens HTTP de requisição e resposta.

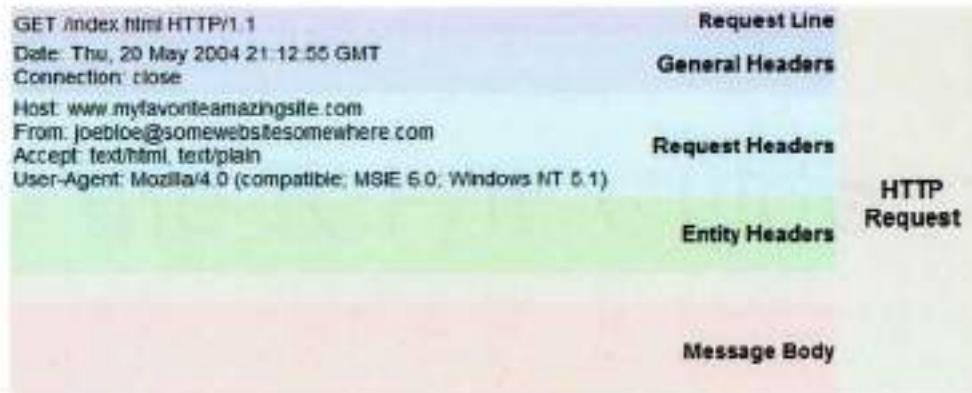


Figura 4 – Mensagem de requisição HTTP (GUIDE, 2005a)

HTTP/1.1 200 OK	Status Line
Date: Thu, 20 May 2004 21:12:58 GMT	General Headers
Connection: close	
Server: Apache/1.3.27	Response Headers
Accept-Ranges: bytes	
Content-Type: text/html	Entity Headers
Content-Length: 170	
Last-Modified: Tue, 18 May 2004 10:14:49 GMT	
<html>	HTTP Response
<head>	
<title>Welcome to the Amazing Site!</title>	
</head>	
<body>	Message Body
<p>This site is under construction. Please come back later. Sorry!</p>	
</body>	
</html>	

Figura 5 – Mensagem de resposta HTTP (GUIDE, 2005b)

O servidor deve implementar dois métodos principais: GET e POST. O método GET consiste em enviar para o cliente o dado requisitado na URI. O método POST consiste em armazenar dados no servidor, enviados pelo cliente ou executar ações de acordo com os dados passado pelo cliente.(IETF, 2014)

## SPI

SPI é uma interface utilizada na comunicação entre chips de um sistema. Funciona como uma comunicação mestre-escravo, em que um dispositivo é definido como mestre enviando requisições e os outros dispositivos atuam como escravos, respondendo às requisições do mestre.

A interface é definida por quatro sinais:

- SCLK: sinal de clock que estabelece uma referência de tempo para toda a comunicação. Todos os sinais são referenciados pelo SCLK.
- SS: o sinal slave select seleciona com qual dispositivo escravo o dispositivo mestre irá se comunicar.
- MOSI: é a via de dados estabelecida do mestre para o escravo.
- MISO: é a via de dados estabelecida do escravo para o mestre.

(PARADIGM, 2015)

As figuras 6 e 7 apresenta a interconexão entre os dispositivos e o diagrama de tempos típico de uma comunicação SPI.

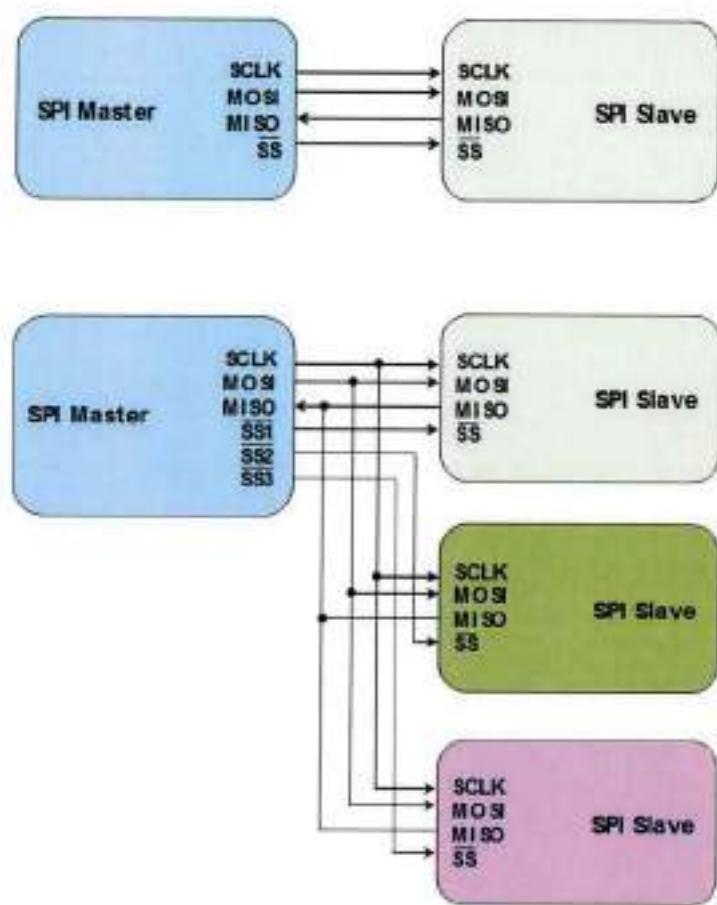


Figura 6 – Interconexão de dispositivos na interface SPI (PARADIGM, 2015)

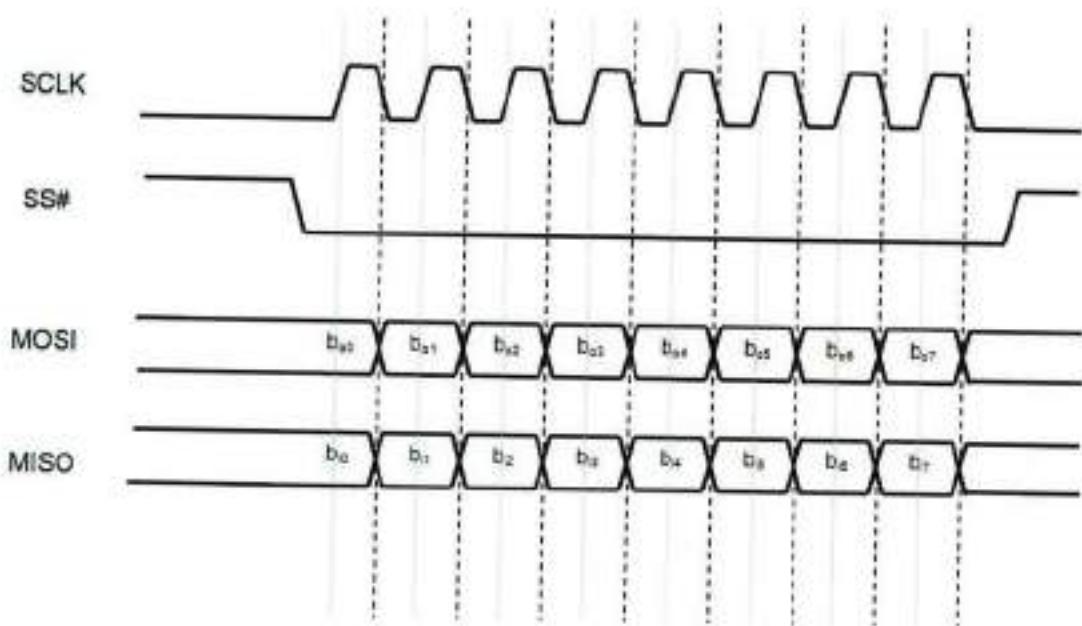


Figura 7 – Interconexão de dispositivos na interface SPI (PARADIGM, 2015)

## 2.5 Benchmark competitivo

Como o controlador embarcado sem fio de reômetro é um produto que não é comercializado de forma direta no mercado, não é possível estabelecer um benchmark competitivo com produtos de mesma natureza. O que ocorre no mercado é a comercialização de reômetros já prontos, que possuem requisitos de engenharia diferentes dos que devem ter um controlador. Sendo assim, estabeleceu-se um benchmark com placas de aquisição de dados, os quais não são utilizados especificamente para o controle de reômetros, mas são os produtos que mais se assemelham ao desenvolvido nesse projeto. Essa comparação, entretanto, deve ser feita com ressalvas, uma vez que tais produtos, terão necessidades diferentes. Os produtos utilizados para comparação foram o USB-201-OEM da Measurement computing (MCCDAQ, 2015) e o NI 9481 da National Instruments (NI, 2015) . A tabela 6 mostra a comparação entre os três produtos.

	USB-201-OEM	NI 9481	Nosso produto
USB	Sim	Sim	Sim
RS232	Não	Não	Sim
Wi-fi	Não	Sim	Sim
Faixa de leitura e geração de sinais	-10V a +10V	0 a 60V	-10V a +10V
Resolução	12 bits	12 bits	12 bits

Tabela 6 – Benchmark competitivo

### 3 Geração de conceito

Através do levantamento de requisitos de engenharia do que compõem o sistema, foram elaboradas as tabelas de conceitos que elencam opções para cada requisito de engenharia. A tabela 7 abrange os requisitos de hardware, a tabela 8 abrange os requisitos de comunicação e a tabela 9 abrange os requisitos de software.

	Memória de 128MB	Leitura de sinais	Geração de sinais	Fonte de alimentação
Opção 1	Nand Flash	ADC interno	DAC interno	Fonte chaveada
Opção 2	SDCard	ADC externo	DAC externo	Regulador linear
Opção 3	Pendrive			Divisor resistivo

Tabela 7 – Opções para requisitos de engenharia de hardware

	Comunicação sem fio	RS232	USB
Opção 1	Wifi	Chip MAX3232	USB interno
Opção 2	Bluetooth	Chip MAX3222	Chip FT232
Opção 3	nRF24L01		

Tabela 8 – Opções para requisitos de engenharia de comunicação

	Servidor Web	Sistema operacional
Opção 1	Servidor concorrente	Linux
Opção 2	Servidor não concorrente	freeRTOS
Opção 3		SO próprio

Tabela 9 – Opções para requisitos de engenharia de software

Através da tabela de conceitos procedemos à análise das opções disponíveis a fim de determinar as melhores opções para garantir a implementação de cada requisito. A metodologia utilizada para a decisão de soluções muito similares é a análise de forças e fraquezas.

### Comunicação sem fio

Para a comunicação sem fio é importante que a solução escolhida não exija hardware adicional para a recepção, a fim de maximizar a compatibilidade com os dispositivos que o usuário pode utilizar como interface para o controlador.

O Wi-fi consiste numa rede local sem fio que permite comunicação entre diversos computadores, celulares, tablets e dispositivos embarcados capazes de conectar-se à rede. Utiliza o protocolo TCP/IP como meio de comunicação entre os dispositivos. Atualmente existe uma grande disponibilidade de módulos Wi-fi de baixo custo, que implementam o *stack* do protocolo TCP/IP, tais como o ESP8266 que tornam a comunicação Wi-fi uma tecnologia de fácil implementação.

O Bluetooth é uma tecnologia de comunicação recente, presente na maioria dos dispositivos atuais. Permite a comunicação entre os dispositivos através de um protocolo próprio. Também possui ampla disponibilidade de módulos que permitem facilidade na implementação.

Módulos RF como o NRF24L01 são módulos transceptores que permitem a comunicação entre dois módulos, um atuando como transmissor e outro como receptor. Não possui protocolo de comunicação definido, ficando a cargo do desenvolvedor implementá-lo. Sua utilização, porém é fácil, se comparado às outras tecnologias de comunicação.

A tabela 10 apresenta a análise das três opções. Pode-se perceber que a melhor opção a ser escolhida para o projeto é a comunicação via Wi-fi.

	Vantagens	Desvantagens	Total
Wi-fi	Não necessita de hardware adicional +++ Permite usar estrutura Web ++ Baixo custo +	Pouca documentação para os módulos existentes -	3
Bluetooth	Ampla disponibilidade de módulos prontos + Ampla disponibilidade de documentação + Não necessita de hardware adicional +++	Custo mediano - Não permite usar estrutura Web -	2
Módulo RF	Ampla disponibilidade de módulos prontos + Ampla disponibilidade de documentação + Fácil implementação ++ Baixo custo +	Exige hardware adicional — Não permite usar estrutura Web -	0

Tabela 10 – Análise de forças/fraquezas para a comunicação sem fio

## Memória

Para implementar a memória do dispositivo deve-se considerar que os dados armazenados devem estar disponíveis para análise posterior mesmo sem alimentação do dispositivo. Logo deve-se implementar uma memória não volátil. Todas as opções apresentadas: memória nand flash, sdcard e pendrive, na verdade são variações da mesma tecnologia: a memória nand flash. Memória Nand Flash é um tipo de memória não volátil programável eletricamente que pode ser soldada e programada numa placa de circuito impresso.

Tanto o SDCard quanto o pendrive consistem de dispositivos de armazenagem que possuem memórias nand flash internas e que provêm interfaces para comunicação com o microcontrolador.

Além de serem muito mais caros, estes dispositivos apresentam problemas de confiabilidade com relação à manutenção dos dados armazenados sendo muito afetados por ruído eletromagnético, além de serem dispositivos removíveis que podem acabar desprendendo-se do controlador devido à vibração do próprio reômetro, assim, estas opções

não são viáveis para o projeto.

A opção escolhida, então, é soldar um chip de memória nand flash na própria placa do controlador.

### Leitura de sinais

O microcontrolador KL25Z128 utilizado para este projeto possui um ADC integrado, porém ele só é capaz de ler sinais de 0 a 3.3V. Para ler sinais na faixa de -10V a +10V será necessário implementar um circuito conversor como o da figura 8.

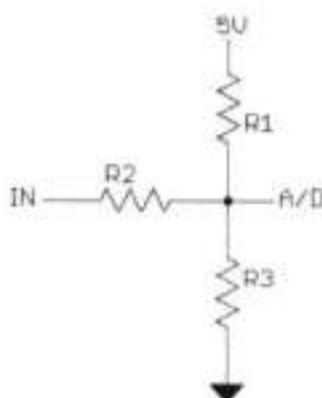


Figura 8 – Circuito de conversão de amplitude para o ADC interno

Esta implementação, embora simples, exige resistores de alta precisão, o que compromete a reprodutibilidade do projeto, esta opção, portanto, não é viável para o projeto. Uma outra opção seria utilizar um ADC externo, capaz de ler sinais de alta amplitude. Esta opção, embora de maior custo e mais complexa, uma vez que exige a comunicação entre o ADC e o microcontrolador, garante robustez e atende os requisitos do projeto. Portanto, esta será a opção escolhida.

### Geração de sinais

O microcontrolador KL25Z128 utilizado para este projeto possui um DAC integrado, porém ele só é capaz de gerar sinais de 0 a 3.3V. Para gerar sinais na faixa de +10V a -10V seria necessário utilizar duas saídas analógicas ligadas cada uma a um amplificador. Uma ligada a um amplificador inversor para gerar sinais de 0 a -10V e outra ligada a um amplificador não inversor, para gerar sinais de 0 a +10V. A figura 9 mostra o circuito equivalente.

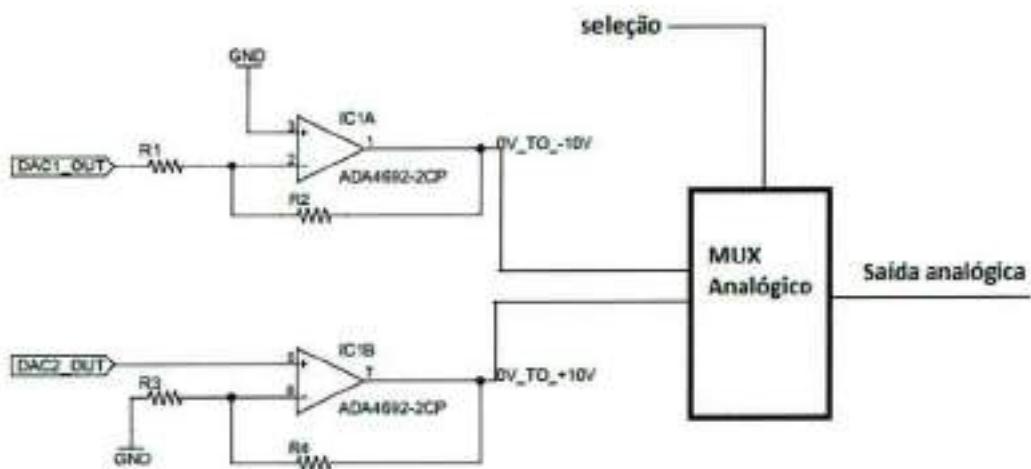


Figura 9 – Circuito de conversão de amplitude para o DAC interno

Outra opção, semelhante à utilizada para a leitura de sinais, é utilizar um DAC externo. A tabela 11 mostra a análise de força/fraqueza para as duas opções.

	Vantagens	Desvantagens	Total
DAC interno	Maior controle sobre a tensão de saída +	Exige um circuito grande - Possibilidade de inserção de ruído devido aos amplificadores -	-1
DAC externo	Exige um circuito pequeno + Maior imunidade a ruído +	Sem controle sobre a tensão de saída -	1

Tabela 11 – Análise de forças/fraquezas para a geração de sinais

Podemos notar que a opção a ser escolhida é utilizar um DAC externo ao microcontrolador, com comunicação SPI, como feito com o ADC.

### Comunicação serial

Para implementar a comunicação serial existem duas possibilidades de rede: RS232 e RS485. RS485 é um tipo de comunicação atual baseada na arquitetura mestre-escravo,

onde todas as transações são controladas pelo dispositivo determinado como mestre, sendo que o dispositivo escravo só realiza transmissão quando determinado pelo mestre. (CATSOULIS, 2005)

RS232 é um tipo de comunicação serial antigo utilizado para comunicações cabeadas de até 25 m de distância e com taxa de transmissão de até 38.4kbps (CATSOULIS, 2005)

O sistema atual utiliza comunicação serial RS232. Como o controlador deve manter a compatibilidade com o sistema atual, não é possível utilizar comunicação serial RS485. Logo a opção escolhida será RS232.

## Comunicação USB

O microcontrolador KL25Z128 possui suporte interno à comunicação USB permitindo a ligação direta com o computador por meio de um cabo USB. Outra opção de implementação é utilizar o chip FT232 da FTDI que converte a comunicação serial UART do microcontrolador para USB. Esta última opção oferece maior facilidade de implementação do ponto de vista do software, entretanto aumenta em muito a quantidade de componentes necessários, consequentemente aumentando os custos do projeto e o tamanho necessário para a placa. Logo é mais plausível utilizar a interface integrada ao microcontrolador.

## Fonte de alimentação

Embora implementada como uma única fonte, a fonte de alimentação do circuito consiste de duas fontes distintas: uma fonte simétrica com saída +12V e -12V com capacidade de corrente de 100mA para a interface analógica, e uma fonte fixa de 5V com capacidade de corrente de 900mA para os circuitos digitais. Analisaremos então as opções disponíveis para cada fonte. A tensão de entrada disponível para as duas fontes é de +18V e -18V.

### Fonte simétrica

Para implementar a fonte simétrica temos duas opções: utilizar um regulador linear ou uma fonte chaveada. A tabela 12 mostra a análise de força/fraqueza para os dois casos.

	Vantagens	Desvantagens	Total
Regulador Linear	Simplicidade de implementação ++ Filtragem de ruídos de alta frequência +++	Baixa eficiência - Alta geração de calor -	2
Fonte chaveada	Alta eficiência + Baixa geração de calor ++	Circuito complexo - Produção de ruído de alta frequência —	-2

Tabela 12 – Análise de forças/fraquezas para a fonte simétrica

### Fonte fixa

Considerando a diferença entre a tensão de entrada e saída como sendo de  $V_d = 13V$ , temos que a potência dissipada por um regulador linear, dada por  $P_d = V_d \cdot I$ , temos  $P_d = 13 \cdot 0.9 = 12W$  aproximadamente. Tal dissipação de potência geraria muito calor no sistema como um todo e exigiria um grande investimento em dissipação de calor. Isto torna o regulador linear inviável neste caso. Logo, para a fonte fixa a única opção disponível é utilizar uma topologia DC-DC buck abaixadora de tensão, visto que a potência dissipada neste caso é muito menor.

### Servidor Web

A implementação de um servidor web, no modelo cliente-servidor, pode ser feita de duas maneiras: com um servidor concorrente ou com um servidor não concorrente. O servidor concorrente atende as requisições dos clientes simultaneamente, utilizando-se de uma estrutura multithreading provida pelo sistema operacional. Sua implementação exige sincronização entre as diversas threads o que a torna mais complexa.

O servidor não concorrente, ao contrário, atende uma requisição por vez, somente atendendo a requisição do próximo cliente quando a primeira requisição por finalizada. Como não se utiliza de um sistema de threads possui implementação mais simples e não necessita de um sistema operacional. (PAI, 1999)

Como um dos requisitos de marketing é garantir acesso simultâneo de diversos usuários, não é viável utilizar um servidor não concorrente. A opção escolhida, então, é o servido concorrente.

## Sistema operacional

Para o sistema operacional tem-se três opções, das quais duas são amplamente conhecidas no mercado. O sistema operacional Linux é um sistema de código aberto, que permite customização de todo o kernel e aplicações. Devido à alta gama de aplicações disponíveis o uso deste sistema operacional tornaria o processo de desenvolvimento menos atrelado ao microcontrolador, tornando o sistema escalável. Infelizmente, o uso de Linux não é suportado pelo KL25Z128, pois esse não dispõe de unidade de gerenciamento de memória (MMU).

O sistema operacional freeRTOS não possui kernel customizável, nem dispõe de aplicações como o Linux, mas é um sistema gratuito, o qual permite um ambiente multithreading adequado às necessidades do projeto.

Outra opção seria criar um sistema operacional próprio que escalonasse tarefas, a fim de implementar o ambiente multithreading necessário. Tal opção, no entanto, demandaria muito esforço, não estando no escopo do projeto.

Optou-se então pelo freeRTOS como sistema operacional embarcado.

## 4 Decomposição funcional

### 4.1 Nível 0

A figura 10 mostra o diagrama de blocos do sistema como um todo.

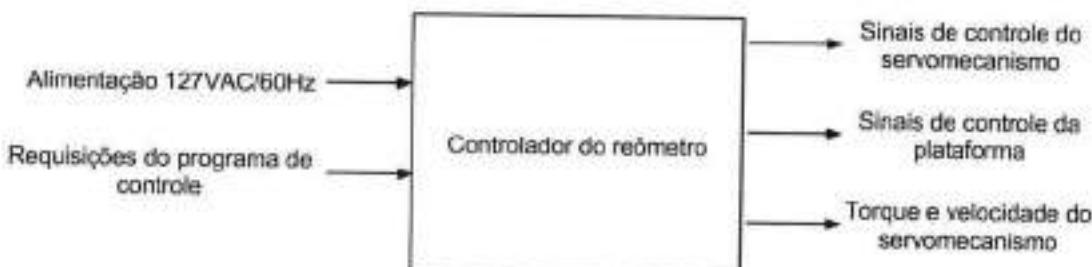


Figura 10 – Diagrama de blocos de nível 0

O controlador é alimentado pela rede elétrica de 127VAC/60Hz e recebe requisições do usuário e atua controlando a máquina ou enviando ao usuário informações sobre o torque e a velocidade do servomecanismo, dependendo da requisição efetuada.

### 4.2 Nível 1

A figura 11 mostra o diagrama de nível 1 do sistema. A figura 12 mostra o diagrama de estados do software embarcado do sistema.

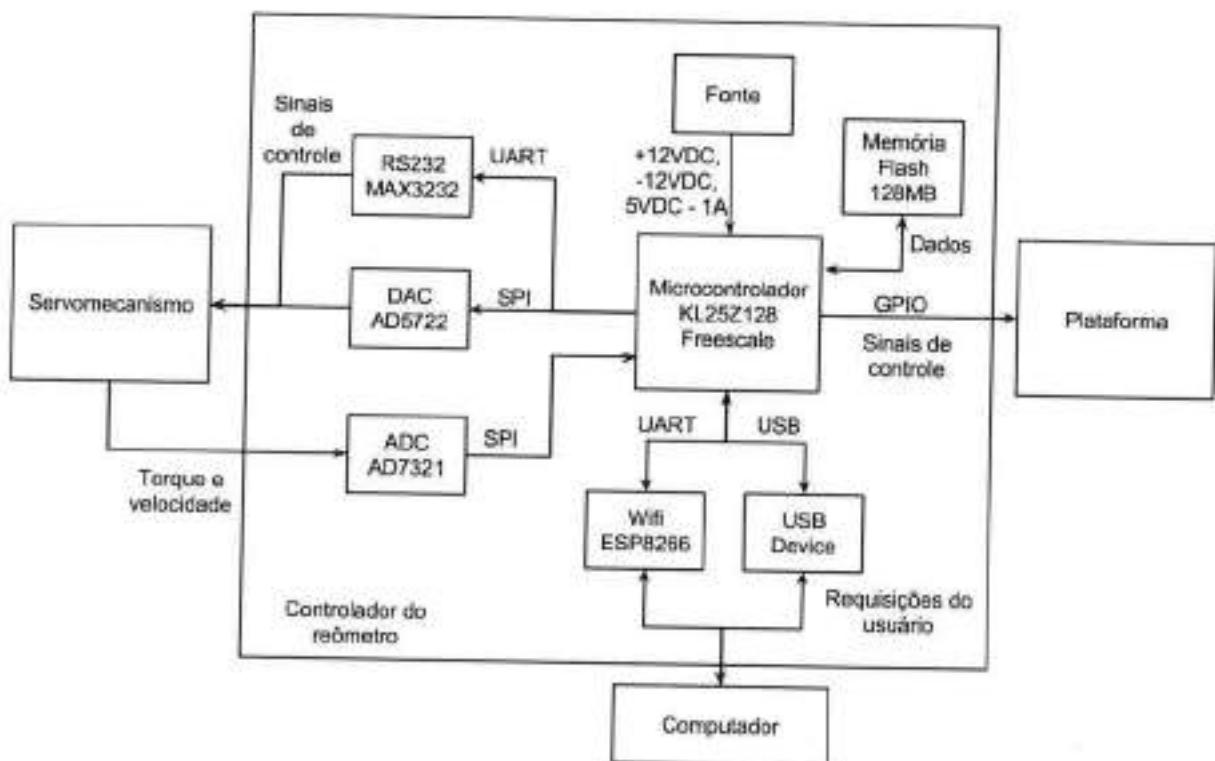


Figura 11 – Diagrama de blocos de nível 1

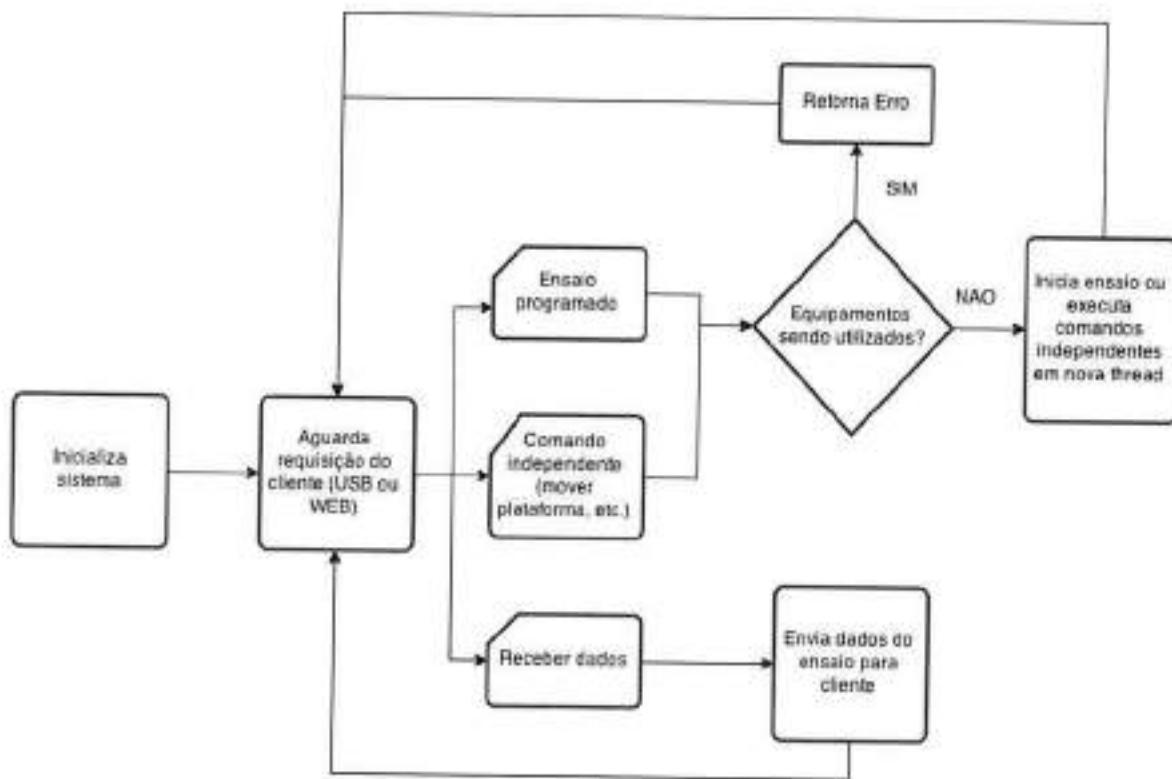


Figura 12 – Diagrama de fluxo de nível 1

Os diagramas mostram como os elementos de hardware e software funcionam dentro do sistema. Do ponto de vista do hardware o microcontrolador é o núcleo do sistema. É ele quem controla os chips de ADC e DAC, realiza a comunicação serial com o servomecanismo, controla a plataforma através dos sinais de controle, recebe e processa as requisições dos usuários e envia as respostas correspondentes.

O ADC recebe, através da interface analógica do servomecanismo, sinais de tensão proporcionais ao torque e à corrente do motor e os envia por SPI ao microcontrolador.

O DAC recebe do microcontrolador os sinais digitais de tensão que devem ser enviados ao servomecanismo, também através da interface analógica.

O RS232 recebe os dados de configuração do servomecanismo, através do microcontrolador e os envia através de comunicação serial.

O Wi-fi e o USB recebem as requisições do cliente e as enviam para o microcontrolador. Quando este responde, estes dois componentes enviam a resposta ao usuário.

Por fim, a memória flash armazena os dados obtidos no experimento e também dados de configuração. Além disso ela armazena as páginas HTML do servidor web programado dentro do microcontrolador.

Do ponto de vista do software o sistema segue o seguinte procedimento:

- O sistema realiza todas as etapas de inicialização do sistema, tais como configurações do ADC e DAC e configurações do motor.
- Iniciado o sistema ele aguarda requisições dos clientes.
- Quando uma requisição é recebida o servidor analisa qual o tipo de requisição.
- Se a requisição apenas solicitar dados, isto é, não solicitar acesso ao servomecanismo ou à plataforma, o servidor solicita os dados ao ADC e responde ao usuário.
- Caso a solicitação seja de controle, então o servidor verifica se os dispositivos não estão em uso.
- Se estiverem, o servidor envia uma mensagem de erro ao usuário.
- Caso contrário, ele executa a ação de controle solicitada.

O projeto fica então totalmente descrito por estes dois diagramas.

### 4.3 Nível 2

A descrição de Nível 2 é feita de modo a detalhar o funcionamento do software e a sua interação com os elementos de hardware. O software embarcado utiliza-se de um sistema operacional de tempo real, o freeRTOS. Por isso, toda a estrutura do software é executada por meio de tarefas que são executadas simultaneamente.

#### Inicialização do sistema

O programa é iniciado na função main(), cuja única função é inicializar os GPIOs do microcontrolador, a fim de garantir que todos estejam em nível de tensão conhecido. Em seguida, o pino relativo ao botão de emergência é avaliado: caso ele esteja em nível alto o sistema vai entrar em modo de testes, caso contrário, entrará em operação no modo normal. Por fim, a função main() inicializa o sistema operacional e cria a tarefa principal do sistema. Quando, o sistema operacional é iniciado, ele cria uma tarefa 'idle' que é a responsável por executar o escalonador. O escalonador é uma rotina que é responsável por executar cada tarefa criada por um determinado tempo, criando, assim, a impressão de simultaneidade. A figura 13 mostra o fluxo correspondente.

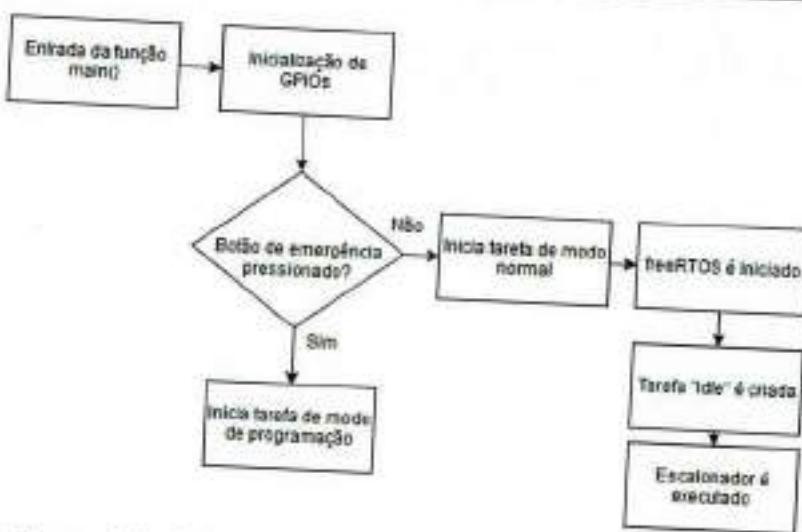


Figura 13 – Diagrama de fluxo da inicialização do sistema

### Modo de testes

A função de um modo de testes é fornecer um ambiente simples onde os elementos de hardware possam ser testados individualmente, sem que haja interação entre eles. No modo de testes apenas a interface USB recebe e envia dados. A comunicação entre a placa e o computador é feita através de um terminal de texto. A rotina executa um loop que aguarda comandos de textos simples através do cabo USB. Esses comandos são decodificados e executam a função correspondente. No Anexo C podemos ver uma lista com os comandos e suas funções. A figura 14 ilustra a execução no modo de testes.

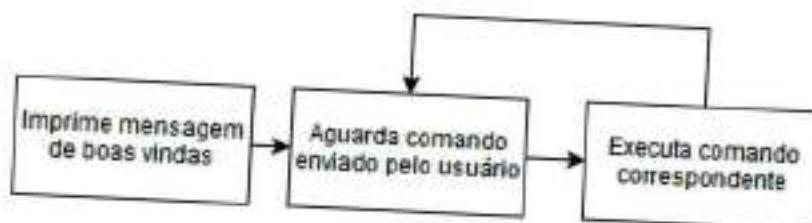


Figura 14 – Diagrama de fluxo do modo de testes

### Modo normal

No modo normal os elementos de hardware são primeiramente inicializados conforme se segue:

- Memória flash: a memória flash lê todos os 1024 blocos e identifica se algum deles está com defeito (bad block). Os blocos com defeito são armazenados numa tabela, pois não se pode escrever neles, sob o risco de perda de dados.

- Servomotor: a inicialização do servomotor ocorre com o envio dos parâmetros de configuração para o DAC e o ADC através da interface SPI;
- Plataforma: a plataforma executa um movimento de calibração, com a finalidade de se posicionar na posição zero;
- Módulo Wifi: o módulo wifi é reiniciado, configurado como host e cliente e configura a rede wifi com o nome e a senha padrões. Em seguida ele é novamente reiniciado.

Após a inicialização dos módulos de hardware o servidor web é iniciado e entra em loop para aguardar requisições.

### Requisições HTTP

As requisições recebidas pelo servidor web obedecem à especificação HTTP, conforme já mencionado anteriormente. O servidor web implementa dois métodos do protocolo HTTP: GET e POST. As requisições do tipo GET possuem o seguinte formato:

GET /comando?parametro1=valor1&parametro2=valor2 HTTP1.1

este método é utilizado quando o servidor precisa retornar informações para o cliente que está fazendo a requisição.

Já o método POST possui o seguinte formato:

POST /comando HTTP1.1 parametro1=valor1&parametro2=valor2

O método POST é utilizado quando o servidor terá seu estado mudado pela requisição e quando ele não precisa retornar valores ao cliente. Quando uma requisição é recebida, o servidor retorna códigos de sucesso ou erro no seguinte formato:

GET [código de erro] [mensagem de erro] HTTP1.1 POST [código de erro] [mensagem de erro] HTTP1.1

Há dois valores possíveis para o código de erro: 200 e 500. O código 200 indica que tudo ocorreu normalmente. Já o código 500 indica erro interno no servidor.

Os valores possíveis para as mensagens são:

- 200 OK: indica que a requisição foi recebida corretamente e será executada;
- 500 Command not found: indica que a requisição foi recebida, mas o comando é inválido;
- 500 Invalid params: indica que a requisição foi recebida, mas os parâmetros do comando estão incorretos;
- 500 Hardware in use: indica que o comando não pode ser executado no momento;

### Execução de comandos

Quando o servidor web recebe uma requisição, ele realiza a decodificação dela, extraindo o nome do comando e os parâmetros recebidos. Se o comando for de rápida execução, ele é executado na tarefa principal. Caso contrário, uma nova tarefa é criada para executá-lo. Em geral, tarefas como mover a plataforma e iniciar um experimento são executadas em tarefas separadas, pois possuem tempo de execução superior a 10 segundos. O Anexo D apresenta uma lista dos comandos disponibilizados pelo servidor web. Existem três tipos básicos de comandos:

- Comandos de recepção de dados: são comandos que retornam dados ao cliente (ex: retornar uma página html, retornar dados do ensaio sendo executado). Normalmente acessam a memória flash. Em geral, são do tipo GET;
- Comandos de hardware: são comandos para executar alguma ação de hardware (ex: acender LED, mover plataforma, mover servomotor). Estes comandos são todos do tipo POST e não retornam dados;
- Comandos de programação de ensaio: são comandos que permitem ao usuário iniciar um ensaio, definindo a posição da plataforma e o comportamento do servomotor (ex: girar o servomotor a 3000rpm por 2 horas com a plataforma a 10 cm).

Antes de executar um comando de hardware ou de ensaio programado, o sistema verifica se já há algum comando do tipo sendo executado. Se houver, o sistema não executa o comando e envia um erro do tipo 500 Hardware in use para o cliente.

## 5 Gerenciamento de projeto

### 5.1 Cronograma

O cronograma completo do projeto, na forma de diagrama de Gantt, encontra-se no Anexo E.

### 5.2 Principais riscos

Podemos notar que as atividades críticas são as seguintes:

- Fabricação da placa: a fabricação da placa é um processo demorado e que depende de agentes externos, que são os fabricantes de placas. Como a fabricação se dará no exterior, devido ao menor custo, o início da fabricação não pode atrasar mais do que um mês. Para evitar ao máximo o atraso no recebimento da placa, todos os membros da equipe estão de comum acordo em pagar um frete expresso, mais caro.
- Configuração e compilação do freeRTOS: o sistema operacional é o que comandará o sistema, há o risco baixo de o microcontrolador não suportar o sistema operacional. Deve-se prever uma folga caso haja a necessidade de nova pesquisa.
- Desenvolvimento da comunicação wifi: há a possibilidade de o ruído do equipamento afetar a comunicação de maneira fatal. Deve-se elaborar testes in loco tão logo a comunicação wifi esteja programada. Além disso, deve-se prever no projeto a utilização de caixa metálica para blindar o circuito contra interferência, caso necessário.

### 5.3 Custos

A tabela 13 mostra a lista de materiais necessária para a execução bem como a projeção de custos de todos os componentes. Constatou-se que o preço de desenvolvimento do protótipo é de, aproximadamente, R\$600,00.

Item	Qtde	Custo unitário (R\$)	Custo total (R\$)
Bateria CR2032	1	1,00	1,00
Capacitor Cerâmico 0,1uF/16V SMD 0603	21	0,10	2,10
Capacitor Cerâmico 10uF/25V SMD 0603	6	0,50	3,00

Capacitor Cerâmico 56pF/16V SMD 0603	2	0,10	0,20
Capacitor Cerâmico 680nF/16V SMD 0603	1	0,10	0,10
Capacitor Tântalo 10uF/25V SMD 1206	6	1,00	6,00
CI AD5722R - ADC	1	15,00	15,00
CI AD5722R - DAC	1	15,00	15,00
CI MAX3232 - RS232	1	3,00	3,00
Conector BNC	3	6,00	18,00
Conector Header 10 pinos macho	1	3,00	3,00
Conector header 8 pinos fêmea	1	3,00	3,00
Conector KK 2,54 de 2 terminais	2	0,50	1,00
Conector KK 2,54 de 3 terminais	1	0,50	0,50
Conector Micro B	1	4,00	4,00
Conector RJ11	1	2,00	2,00
Cristal 8MHz/30pF	1	0,50	0,50
Diodo Schottky	4	1,00	4,00
Diodo TVS	1	1,00	1,00
Ferrite bead 300nH SMD 0603	2	0,50	1,00
LED verde SMD 0603	1	1,00	1,00
Memória Nand 128MB	1	15,00	15,00
Microcontrolador KL25Z128	1	20,00	20,00
Regulador de tensão AP1117E33	2	3,00	6,00
Resistor 100k SMD 0603	3	0,10	0,30
Resistor 10k SMD 0603	6	0,10	0,60
Resistor 1k SMD 0603	7	0,10	0,70
Resistor 330R SMD 0603	1	0,10	0,10
Resistor 33R SMD 0402	2	0,10	0,20
Switch PTH	1	0,10	0,10
PCB 4 camadas	1	300,00	300,00
Stencil	1	100,00	100,00
Caixa	1	5,00	5,00
<b>Custo total</b>			<b>532,40</b>

Tabela 13 – Lista de materiais

## 6 Prova de conceito

Com a finalidade de comprovar a viabilidade do ponto principal e mais crítico do projeto, que é a comunicação wifi, foi desenvolvido um primeiro protótipo funcional, utilizando uma placa de desenvolvimento da Freescale, FRDM-KL25Z, a qual contém o microcontrolador a ser utilizado no projeto e terminais de acesso aos pinos, conforme mostra a figura 15.



Figura 15 – FRDM-KL25Z: placa de desenvolvimento do KL25Z128

Como o módulo de Wifi possui um consumo muito grande de corrente de pico (aproximadamente 300 mA), foi necessária a fabricação e montagem de uma placa de alimentação e comunicação do módulo, conforme mostram as figuras 16 e 17. A placa possui um regulador linear próprio, além de uma entrada micro USB para que fosse possível a conexão de uma fonte de 5V. O esquema elétrico e o layout da placa encontram-se no Anexo A.



Figura 16 – Placa de alimentação do módulo wifi: vista inferior

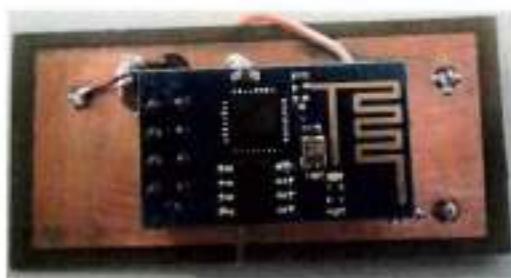


Figura 17 – Placa de alimentação do módulo wifi: vista superior

A figura 18 mostra a montagem final do conjunto.

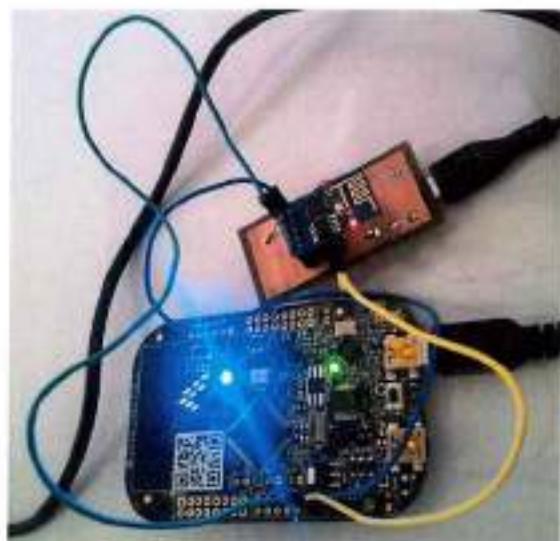


Figura 18 – Vista do conjunto montado e operando

O módulo Wifi é configurado pelo microcontrolador através da interface serial responsável pela comunicação entre as duas placas. O processo de configuração do módulo WiFi é feito da seguinte forma:

- O microcontrolador envia um comando de teste e aguarda um 'OK' do módulo, com a finalidade de testar o funcionamento do canal de comunicação.
- O microcontrolador envia comandos para que o módulo opere como servidor e configura uma rede wifi, de nome REOMETRO, que ficará disponível para conexão do computador cliente.
- O módulo é reiniciado para aplicar as configurações. Ao final do boot a rede wifi aparecerá disponível para qualquer dispositivo com capacidade de conexão.
- No computador conecta-se à rede recém criada e digita-se num navegador o endereço IP do módulo wifi (192.168.4.1) para acessar o servidor web que está rodando no microcontrolador.
- Os comandos inseridos pelo usuário na página web são enviados através do módulo para o microcontrolador.

A figura 19 mostra a tela da página web criada. Nela é possível controlar a cor do LED RGB presente na placa de desenvolvimento.



Figura 19 – Tela do WebServer aberta no navegador Web Firefox

Este protótipo valida a possibilidade de controle do hardware por meio de uma interface wifi. O procedimento adotado para controle da cor do LED também se aplica ao controle dos motores do reômetro sem a necessidade de alteração do protocolo de comunicação entre o módulo e o microcontrolador.

## 7 Protótipo

Validada a ideia central do projeto, através da prova de conceito, deu-se inicio à criação de um protótipo para validação e testes. O protótipo contempla todas as funcionalidades previstas no sistema e atende a todos os requisitos de engenharia propostos no inicio do projeto. A construção do protótipo é feita em duas etapas: o projeto e fabricação do hardware e o projeto e codificação do software embarcado.

### 7.1 Hardware

#### Placa principal

O esquema elétrico e o layout da placa de circuito impresso que foram enviados para fabricação encontram-se no Anexo B. Deu-se prioridade à utilização de componentes com encapsulamento SMD para maior imunidade à ruídos elétricos e para manter um design compacto do controlador. A placa foi projetada tendo 4 camadas elétricas, com núcleo de FR4, com a disposição das camadas nesta ordem: sinais, plano de terra, plano de alimentação, sinais. Para facilitar o processo de soldagem os componentes SMD e os componentes PTH foram colocados em lados opostos da placa de circuito impresso. Visando compatibilidade eletromagnética o plano de terra da parte analógica foi segmentado do plano de terra da parte digital tendo apenas dois pontos de união localizados logo abaixo dos chips que fazem a interface analógico-digital. Para facilitar a montagem e posterior manipulação todos os componentes possuem silkscreen indicando suas funções e posições. As figuras 20 e 21 mostram a placa após sua fabricação e posterior soldagem em linha de montagem SMD. O LED foi colocado para servir de indicador de funcionamento da placa e das fontes de alimentação.



Figura 20 – Foto da placa, lado TOP



Figura 21 – Foto da placa, lado BOTTOM

### Alimentação

Para alimentar a placa controladora foram utilizadas duas fontes adaptadas para encaixarem-se nos conectores utilizados na placa. A primeira fonte é uma fonte chaveada de 5V/600mA que garante alimentação suficiente para os circuitos digitais da placa, incluindo aí o módulo wifi acoplado. A outra fonte é uma fonte chaveada de +12V/100mA e -12V/100mA que é utilizada como tensão de referência para que os chips de ADC e DAC possam produzir e receber os sinais de +10V a -10V.

## Programação

Para realizar a programação do microcontrolador a placa de desenvolvimento utilizada na prova de conceito foi adaptada para funcionar como programadora. A programação é feita através de interface SWD. Tal interface possui 5 pinos de conexão os quais permitem resetar o microcontrolador para um estado conhecido e transmitir o firmware através de comunicação serial com a memória Flash do microcontrolador. A figura 22 mostra a montagem realizada para fazer a programação do firmware.

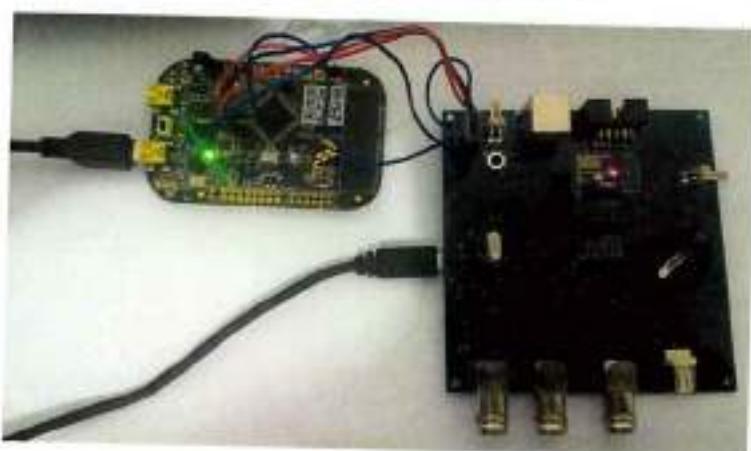


Figura 22 – Conexão da placa com a programadora

## Motor de plataforma

A conexão com o motor de plataforma é feita através de uma placa de interface entre a lógica digital de controle e a parte de potência necessária para exercer a atuação sobre o motor. Tal placa foi desenvolvida pelo Prof. Dr. Gustavo Pamplona Rehder anteriormente à existência deste trabalho. A interface digital possui 3 sinais de saída que devem ser gerados pela placa controladora, a fim de controlar o motor de plataforma, e dois sinais de entrada, os quais devem ser lidos para permitir também o controle manual da plataforma. Além disso, também há um sensor hall que indica o quanto a plataforma subiu ou desceu para fechar a malha de controle do motor. A figura 23 mostram a conexão entre a placa de plataforma e a placa controladora principal.



Figura 23 – Conexão com a placa controladora do motor de plataforma

### Servomotor

O servomotor utilizado no reômetro consiste num motor SCA05 da WEG que possui uma interface analógico-digital, a qual permite o controle de sua velocidade ou torque. As conexões com o motor são realizadas através do controlador de servomotor utilizando uma conexão serial RS232 para enviar comandos e receber dados do motor e também através de três conexões analógicas para enviar sinais de referência de velocidade e para receber sinais de tensão proporcional ao torque e à velocidade. A figura 24 mostra o controlador conectado à placa controladora.



Figura 24 – Servomotor conectado à placa controladora

A conexão serial é feita por um cabo com conector RJ12 de 6 vias que transporta sinais digitais de 5V, conforme mostra a figura 26.

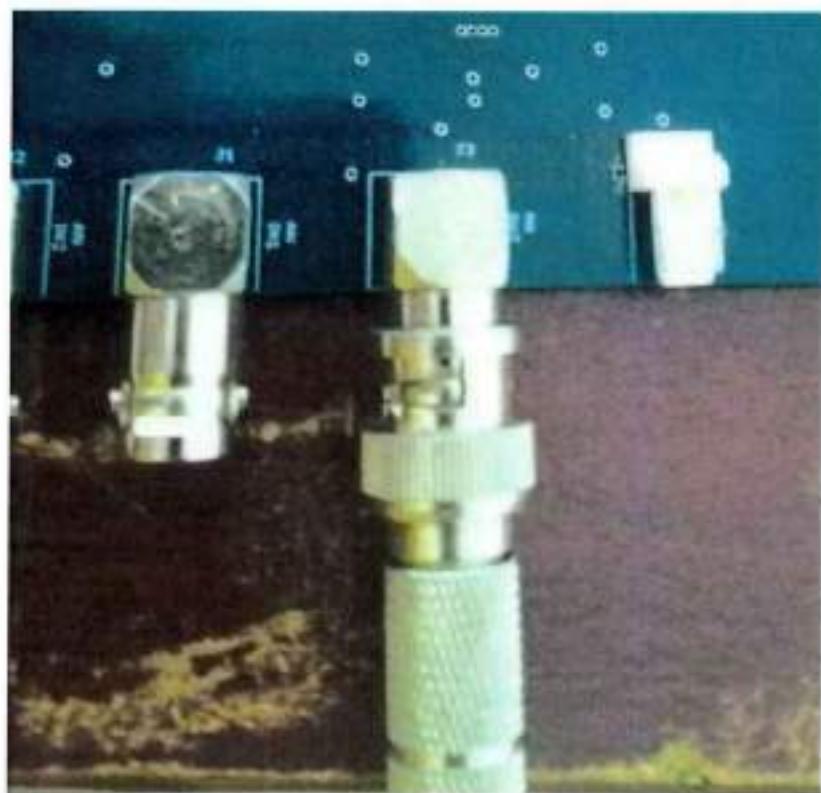


Figura 25 – Conexão analógica

As conexões analógicas são feitas utilizando cabo coaxial blindado com conectores BNC conforme mostra a figura 25. O cabo blindado garante imunidade a ruídos externos, notadamente o ruído oriunda da rede elétrica. Os sinais trafegados pela interface analógica são sinais de tensão na faixa de +10V a -10V.

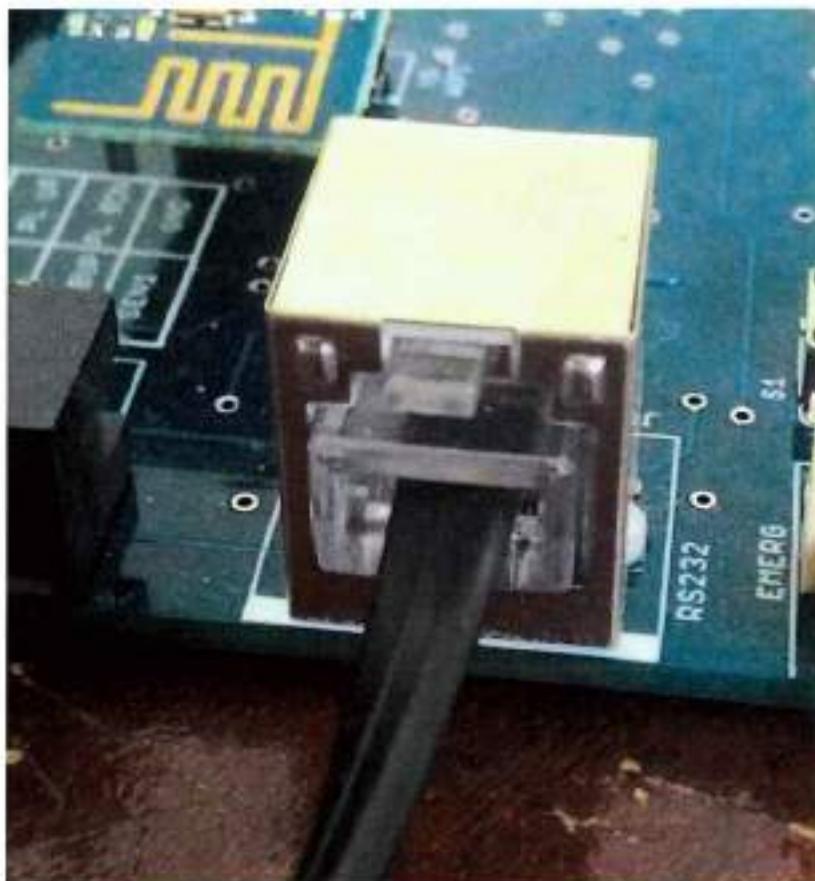


Figura 26 – Conexão serial RS232

## 7.2 Software

O software embarcado foi programado em C++ e utiliza os conceitos de programação orientada a objetos onde o programa como um todo é partido em pequenos pedaços de software, ditos objetos, que são responsáveis por partes específicas e que conversam entre si numa função central. Assim há um objeto responsável por abstrair o Sistema de arquivos, o servidor embarcado, entre outros, estes objetos são ditos componentes do sistema. Ainda há objetos responsáveis por fornecer interface com os elementos de hardware, estes são chamados drivers. Assim, de forma análoga aos componentes há um objeto que abstrai a comunicação com o módulo wifi, com a memória nand flash, com a placa de plataforma, entre outros. As figuras 27 e 28 mostram um exemplo de um driver de hardware escrito como objeto em C++.

```
* Serial.h]

#ifndef SERIAL_H_
#define SERIAL_H_

#include <string>
extern "C" {
#include "USB.h"
}

namespace Hardware {

    class USBSerial {
    public:
        USBSerial();
        virtual ~USBSerial();
        static bool write(const std::string& text);
        static bool read(std::string& text, int data_size);
        static bool readline(std::string &line);
        static uint16_t total_received_data(void);
    };
}

/* namespace Hardware */

#endif /* SERIAL_H_ */
```

Figura 27 – Exemplo de objeto em C++ (header)

```
* Serial.cpp]

#include "USBSerial.h"
#include "util.h"

using namespace std;

namespace Hardware {

    USBSerial::USBSerial() {
    }

    USBSerial::~USBSerial() {
    }

    bool USBSerial::write(const string& text) {
        uint16_t result = USB_SendString((unsigned char*)text.c_str());
        return result == ERR_OK;
    }

    bool USBSerial::read(string &text, int data_size) {
        text.clear();
        uint8_t result;
        unsigned char chr;
        int i = 0;
        int timeout = 5000;
        do {
            if (USB_GetCharInRxBuf()) {
                result = USB_RecvChar(&chr);
                if(result != ERR_OK)
                    break;
                text += chr;
                i++;
                if (i == data_size)
                    break;
            }
        } while (true);
    }
}
```

Figura 28 – Exemplo de objeto em C++ (source)

A fim de facilitar o processo de desenvolvimento, o software, bem como o hardware foram desenvolvidos utilizando um sistema de versionamento denominado Git, o qual permite que se armazene o código fonte e esquemáticos em nuvem, além de realizar o controle de versões e modificações do projeto, permitindo o retorno a versões anteriores caso haja qualquer problema nas versões mais atuais.

### Ambiente de compilação e debug

O processador contido dentro do microcontrolador KL25Z possui arquitetura ARM. Como a arquitetura do processador do ambiente de desenvolvimento e do processador do ambiente de execução são diferentes, é necessário realizar um procedimento conhecido como compilação cruzada, no qual o código desenvolvido numa arquitetura x86 é compilado para arquitetura ARM através do compilador arm-gcc, usando a toolchain arm-none-eabi, a qual, contém não só o compilador de código como bibliotecas necessárias para realizar a execução do código na arquitetura ARM. O procedimento de debug é realizado através da interface SWD, utilizando o programa arm-gdb, o qual permite que acompanhemos a execução do programa instrução por instrução, facilitando enormemente o processo de depuração.

### Arquitetura central

A arquitetura central do sistema consiste no fluxo descrito na Descrição funcional de nível 1. O programa inicia colocando todos os pinos de entrada e saída do microcontrolador em estado conhecido. Em seguida desabilita as interrupções e inicializa os componentes de hardware do microcontrolador. Quando este processo é finalizado o sistema operacional FreeRTOS é inicializado e assume o comando do sistema. A partir deste momento o sistema operacional cria a tarefa principal do sistema. Há duas possibilidades neste caso, ou o sistema entrará em modo de programação ou em modo normal. O que decide qual dos modos será executado é se o botão de emergência está ou não acionado no momento do boot, conforme mostra a figura 29. Se o botão estiver acionado, entramos no modo de programação, caso contrário, entramos no modo normal.



Figura 29 – Seleção de modo de programação

O modo de programação é utilizado para programar a página html do servidor na memória flash e para depurar os componentes de hardware. Ele é composto de uma rotina em laço infinito que aguarda comandos de texto através da porta USB. Os comandos e arquivos são enviados através de um programa emulador de terminal denominado gtkterm. A figura 30 ilustra este processo

```

Atividades  GtkTerm
Ter. 21:59
GtkTerm - /dev/ttyACM0 9600-B-N-1

File Edit Log Configuration Control signals View
PPR0x_MODE
SHELL - PEOMETRO
Inicializando NAND...
NAND inicializado com sucesso
CMD>send_html[]

```

Figura 30 – Modo de programação

O modo normal consiste na execução normal do programa. Neste modo, os drivers de hardware e componentes de sistema são inicializados para colocar o sistema num estado conhecido. Após a inicialização o servidor é iniciado e fica aguardando requisições que podem vir através da interface Wifi ou da interface USB. Quando o servidor recebe uma requisição ele verifica se esta requisição é válida e executa o comando correspondente. Se o hardware estiver sendo utilizado no momento ele retorna ao cliente uma mensagem de erro.

## 8 Testes e validação

O procedimento de validação e de testes do hardware consiste na construção do software embarcado e na sua interação com o hardware. O primeiro teste realizado foi a programação de uma página html na memória nand flash através do modo de programação. Este teste garante que a comunicação USB está operando corretamente e que a comunicação com a memória flash também está operando corretamente. Com isso dois requisitos de engenharia foram validados: o sistema possui memória com capacidade de 128 MB e comunicação USB. O segundo teste foi construir um web server que recebesse uma requisição do navegador de internet, através de rede wifi e devolvesse a página html programada na memória. A figura 31 ilustra a página lida através da interface wifi.

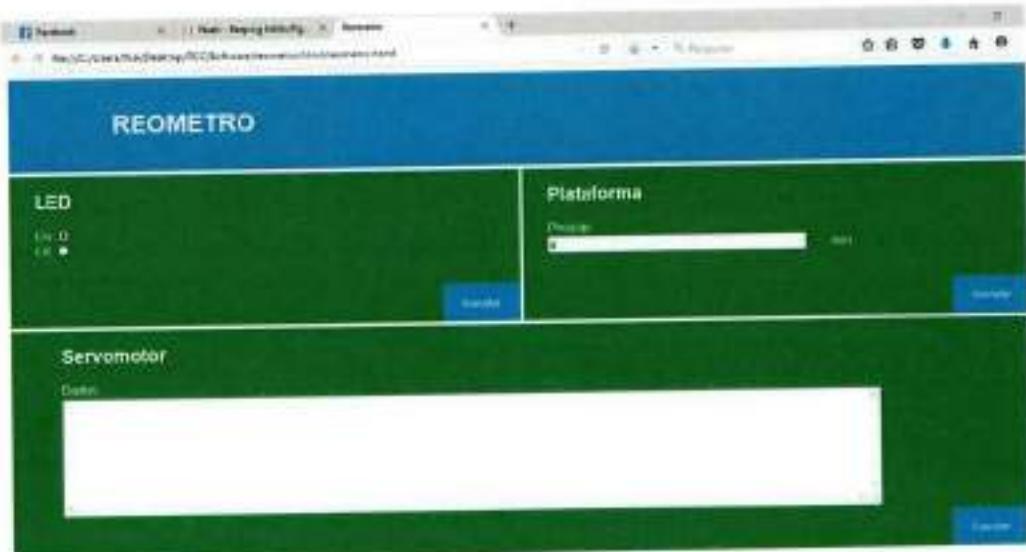


Figura 31 – Página html sendo servida pela placa controladora

Este procedimento garante que a comunicação wifi está operando corretamente. A página foi acessada de dispositivos diferentes (um notebook e dois smartphones) ao mesmo tempo, a fim de garantir também que o sistema operacional está operando corretamente. Assim, os requisitos que estipulavam que o sistema deveria possuir comunicação sem fio e permitir comunicação simultânea para até 3 pessoas foram validados. Dentro da página html há a possibilidade de se controlar o motor de plataforma e o servomotor, garantindo que o sistema é capaz de receber comandos através do wifi, que o controle do motor de plataforma está operando corretamente e que o controlador consegue comunicar-se tanto na interface serial, quanto na analógica, com o servomotor. Os testes com o motor de plataforma foram feitos ligando a placa controladora à placa de interface do motor e deslocando o motor de 10 em 10 cm, para cima e para baixo. Com isso validou-se os

acionamentos do motor, os sinais de parada e a conexão com o sensor hall, que permite saber o posicionamento da plataforma. A cada deslocamento foi medida a posição deslocada com uma régua, obtendo-se um erro máximo de 1 cm, o qual não compromete a operação do equipamento. Os testes com o servomotor foram feitos ligando-se o servomotor a uma rede trifásica 220 V, tendo o conector acoplado a ele, conforme ligações já descritas acima. Foi imposta uma tensão de 3 V, oriunda do DAC, como referência de velocidade, o que equivale, para o motor, a uma rotação de 3000 rpm no sentido horário. Em seguida, enviou-se comandos através da interface serial para iniciar a operação do motor. Quando o motor, começou a se mover, os sinais recebidos pelo ADC foram impressos na tela e também enviados via wifi para o browser, obtendo-se a leitura média de 3 V e -90 mV que equivalem à velocidade programada e ao torque correspondente. Em seguida realizou-se o mesmo procedimento para um sinal de -3 V, equivalente a 3000 rpm no sentido anti-horário, obtendo-se as leituras de -3 V e 90 mV. Sendo assim, todos os elementos de software e hardware foram testados e validados comprovando que foram corretamente projetados.

## 9 Conclusão

Os testes mostraram que o projeto atendeu aos requisitos de engenharia propostos obtendo um desempenho satisfatório. As limitações de velocidade e memória RAM inerentes ao microcontrolador implicam em quedas não comprometedoras da velocidade do controlador como um todo e da capacidade de sustentação de muitas comunicações simultâneas. Os testes mostram que o sistema deixa de ser confiável quando mais de 3 pessoas conectam-se ao mesmo tempo ao sistema. Uma sugestão de melhoria do ponto de vista de hardware é a troca do microcontrolador por um com, pelo menos, 32kB de memória RAM. O projeto também não previu uma estrutura que permitisse a expansão de conexões conforme a necessidade do usuário. Assim, para que seja possível acoplar novos sensores, é necessário desenvolver um novo hardware. Sugere-se prever uma interface de expansão com conexões de SPI, UART e GPIOs para acoplar sensores conforme a necessidade. Do ponto de vista da arquitetura de software, sugere-se a criação de um bootloader, para permitir o update de software em campo, sem a necessidade de uma programadora externa. Além disso, melhorias na transmissão e recepção de dados devem ser feitas. Por fim, este projeto permitiu obter uma visão completa do processo de desenvolvimento de hardware e software que incorporam um sistema embarcado complexo.

## Referências

- CATSOULIS, J. *Designing Embedded Hardware*. [S.l.]: O' Reilly, 2005. Citado na página 36.
- ELETROONICS, F. *What is a Microcontroller?* 2015. Disponível em: <<https://www.futureelectronics.com/en/Microcontrollers/microcontrollers.aspx>>. Citado na página 19.
- ESC. *Uma pessoa interage com 20 sistemas embarcados por dia.* 2015. Disponível em: <<http://www.escbrazil.com.br/pt/multimedia/indice-de-noticias/382-uma-pessoa-interage-com-20-sistemas-embarcados-por-dia>>. Citado na página 18.
- GUIDE, T. T. *HTTP Request Message Format*. 2005. Disponível em: <[http://www.tcpipguide.com/free/t\\_HTTPRequestMessageFormat.htm](http://www.tcpipguide.com/free/t_HTTPRequestMessageFormat.htm)>. Citado 2 vezes nas páginas 9 e 26.
- GUIDE, T. T. *HTTP Response Message Format*. 2005. Disponível em: <[http://www.tcpipguide.com/free/t\\_HTTPResponseMessageFormat.htm](http://www.tcpipguide.com/free/t_HTTPResponseMessageFormat.htm)>. Citado 2 vezes nas páginas 9 e 27.
- IETF. *Hypertext Transfer Protocol – HTTP/1.1*. 2014. Disponível em: <<http://www.w3.org/Protocols/rfc2616/rfc2616.html>>. Citado na página 27.
- JOHN, F. A. C. R. G. P. V. M. Caracterização reológica de argamassas pelo método de squeeze-flow. *VI Simpósio Brasileiro de Tecnologia de Argamassas*, Maio 2006. Citado na página 17.
- MCCDAQ. *USB-201-OEM*. 2015. Disponível em: <<http://www.mccdaq.com/usb-data-acquisition/usb-201-oem.aspx>>. Citado na página 29.
- NI. *Sistema de medição NI 9481*. 2015. Disponível em: <<http://sine.ni.com/nips/cds/view/p/lang/pt/nid/209923>>. Citado na página 29.
- PAAR, A. *Reômetro*. 2015. Disponível em: <<http://www.anton-paar.com/br-pt/produtos/grupo/reometro/>>. Citado na página 16.
- PAI, V. S. Flash: An efficient and portable web server. *Proceedings of the 1999 USENIX - Annual Technical Conference*, Junho 1999. Disponível em: <[https://www.usenix.org/legacy/event/usenix99/full\\_papers/pai/pai.pdf](https://www.usenix.org/legacy/event/usenix99/full_papers/pai/pai.pdf)>. Citado na página 37.
- PARADIGM, B. *Introduction to I2C and SPI protocols*. 2015. Disponível em: <<http://www.bytesparadigm.com/applications/introduction-to-i2c-and-spi-protocols/?/article/AA-00255/22/Introduction-to-SPI-and-IC-protocols.html>>. Citado 4 vezes nas páginas 9, 27, 28 e 29.
- WIKIPEDIA. *Analog-to-digital converter*. 2015. Disponível em: <[http://en.wikipedia.org/wiki/Analog-to-digital\\_converter](http://en.wikipedia.org/wiki/Analog-to-digital_converter)>. Citado na página 19.

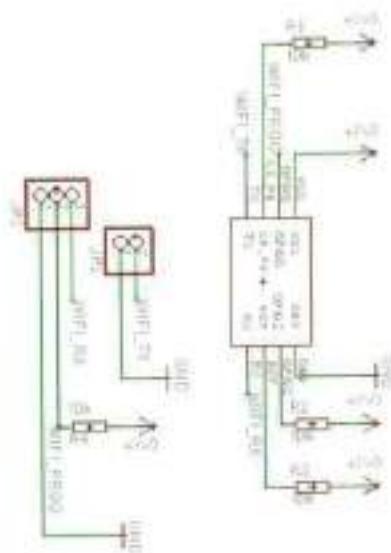
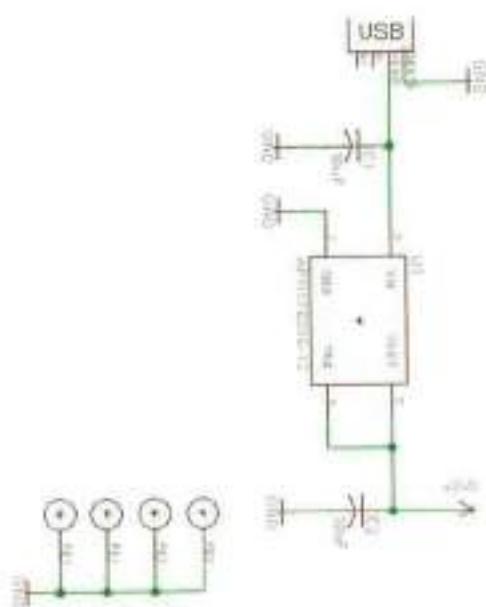
WIKIPEDIA. *Memória flash*. 2015. Disponível em: <[http://pt.wikipedia.org/wiki/Mem%C3%B3ria\\_flash](http://pt.wikipedia.org/wiki/Mem%C3%B3ria_flash)>. Citado na página 19.

WIKIPEDIA. *Wi-Fi*. 2015. Disponível em: <<http://en.wikipedia.org/wiki/Wi-Fi>>. Citado na página 20.

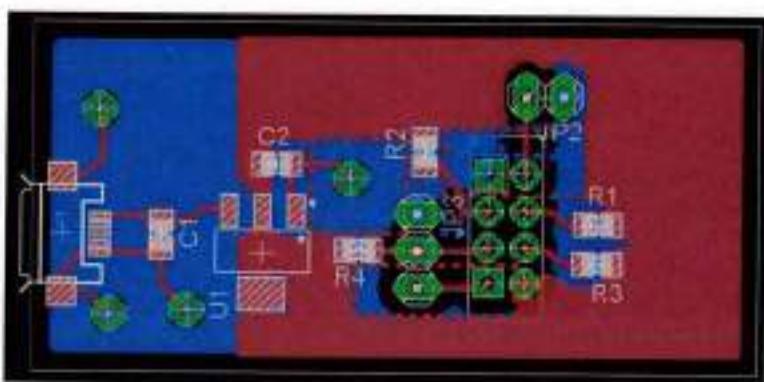
## Anexos

## ANEXO A - Arquivos de projeto elétrico da placa de alimentação do módulo Wifi

Esquema elétrico da placa de alimentação do módulo Wifi



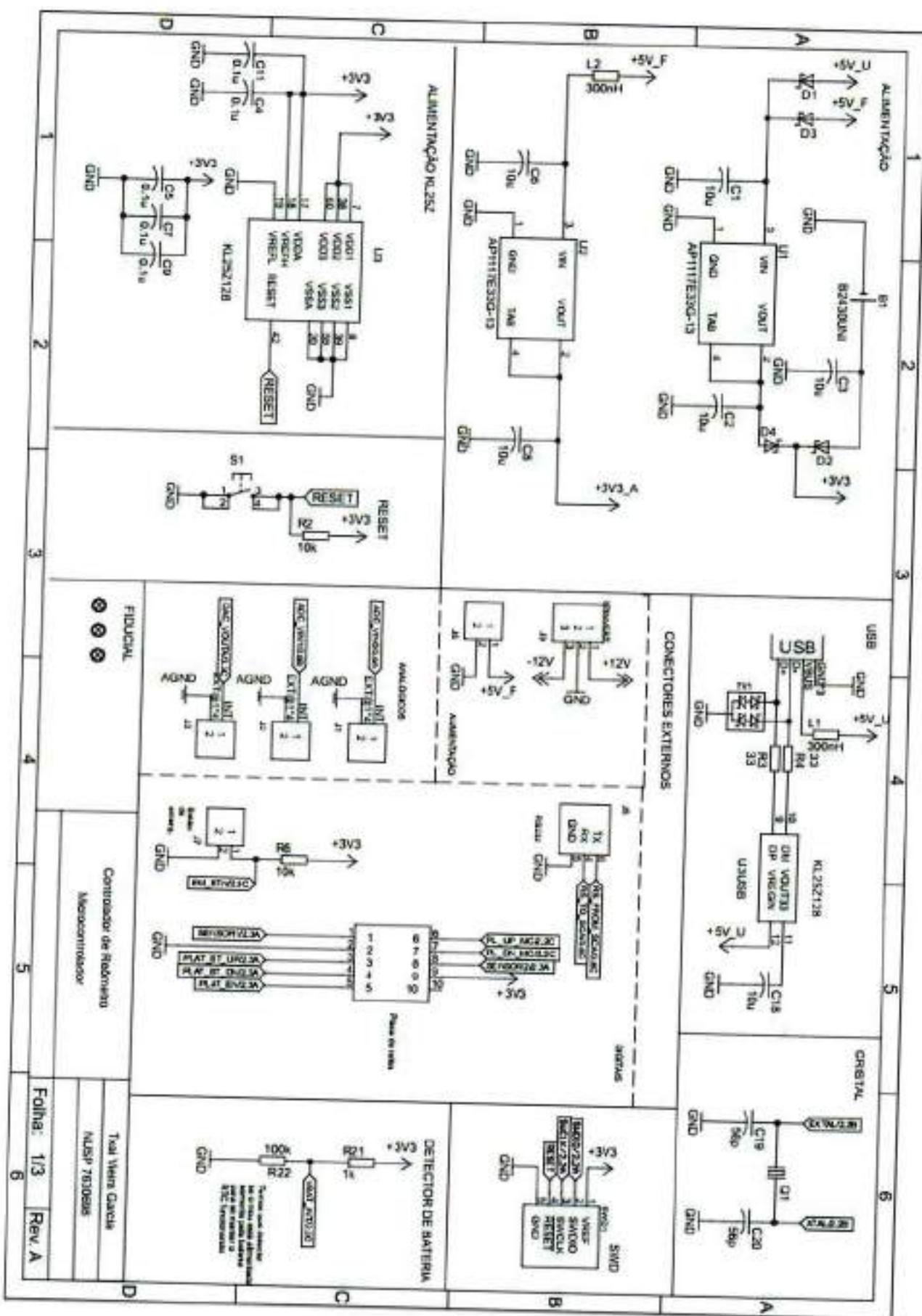
## Layout da placa de alimentação do módulo Wifi

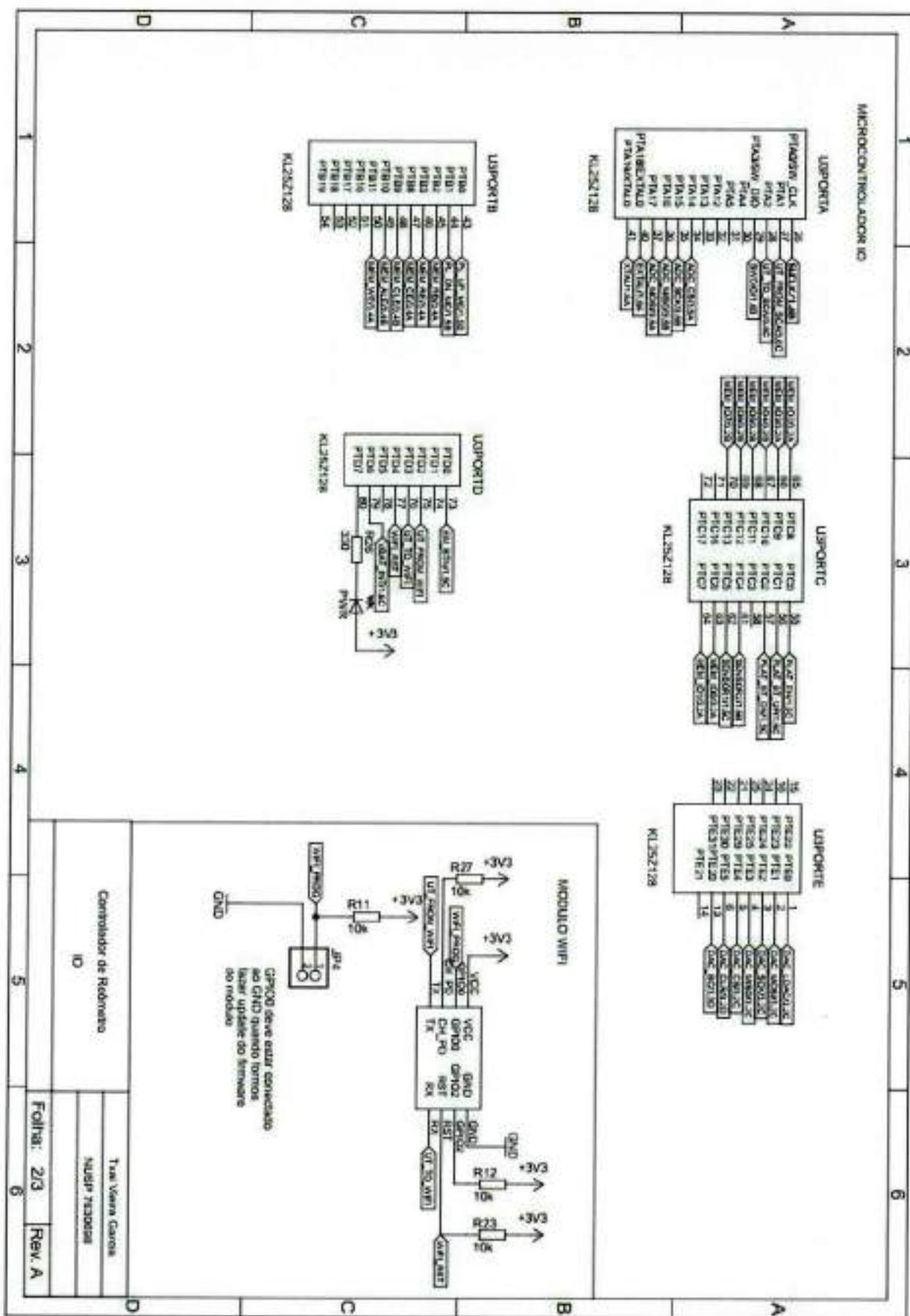


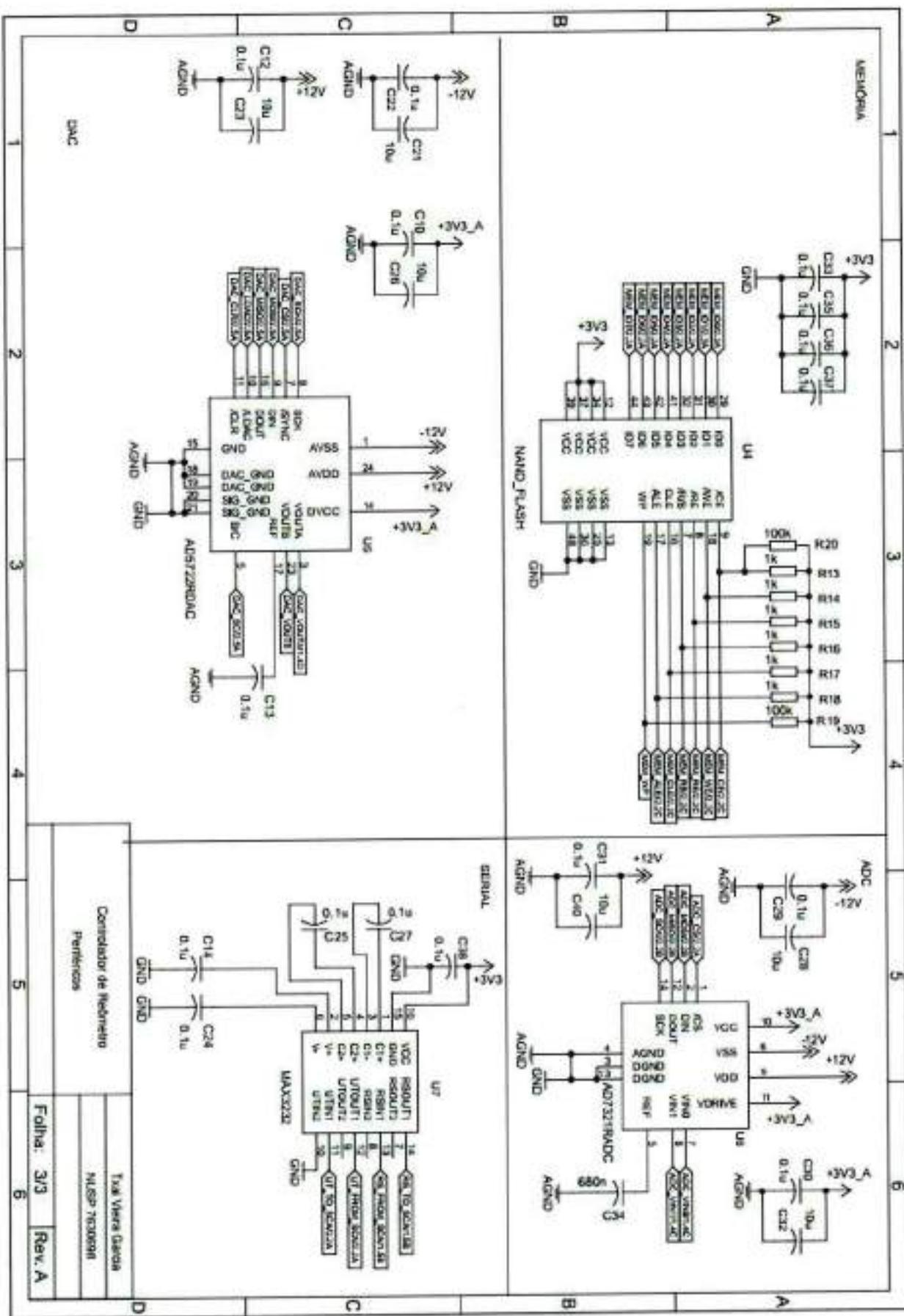
## ANEXO B - Arquivos de projeto elétrico da placa de controladora de reômetro

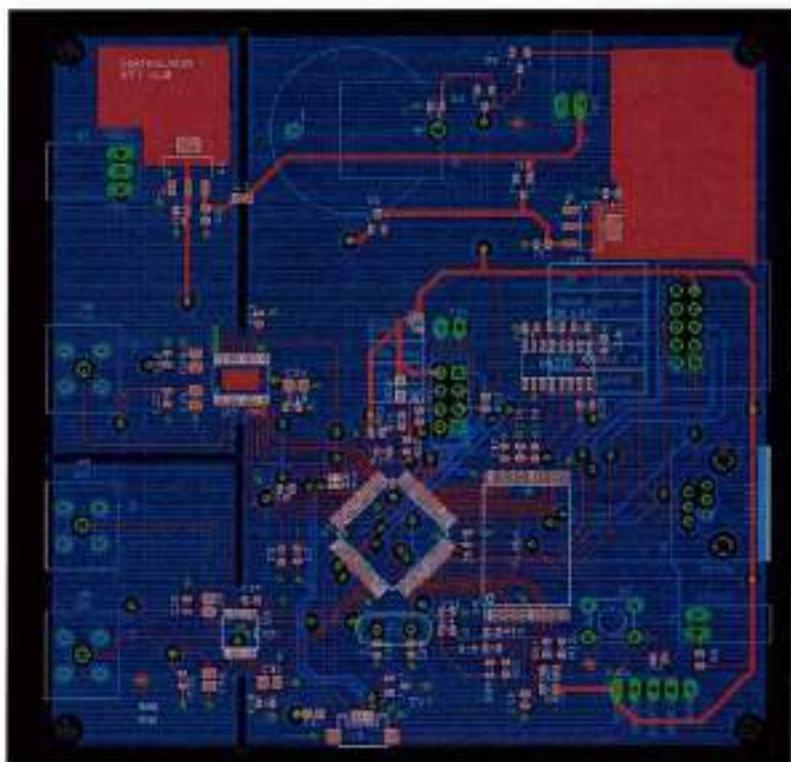
Esquema elétrico da placa controladora de reômetro

Layout da placa controladora de reômetro









## ANEXO C - Comandos do modo de testes

- `test`: imprime "teste ok" no terminal para comprovar que a interface USB está funcionando;
- `send_html`: recebe uma página html vinda do terminal de texto e grava a página na memória flash;
- `read_html`: lê a página html da memória flash e a imprime no terminal de texto;
- `plat_up`: faz a plataforma mover-se para cima indefinidamente;
- `plat_down`: faz a plataforma mover-se para baixo indefinidamente;
- `plat_stop`: faz a plataforma parar;
- `plat_pos`: aguarda uma posição e faz a plataforma ir para esta posição;
- `erase_mem`: apaga todos os dados da memória flash;
- `wifi_pass`: permite enviar comandos diretamente para o módulo wifi;

## ANEXO D - Requisições HTTP da interface wifi

- Requisição: change\_led  
tipo: POST;  
parâmetros: led\_status('on'/'off');  
retorna valores: não  
Descrição: liga ou desliga o LED.
- Requisição: move\_platform  
tipo: POST;  
parâmetros: position(valores numéricos);  
retorna valores: não  
Descrição: move a plataforma para uma dada posição.
- Requisição: get\_servo\_data  
tipo: GET;  
parâmetros: nsr(valores numéricos);  
retorna valores: sim  
Descrição: o cliente recebe os dados do experimento de acordo com o número de nsr que é enviado na requisição. O número é sequencial e corresponde à uma página na memória flash.
- Requisição: move\_servo  
tipo: POST;  
parâmetros: servo\_speed(valores numéricos);  
retorna valores: não  
Descrição: move o servomotor a uma dada velocidade. Se o valor for positivo o servo se moverá no sentido horário, caso contrário, ele se moverá em sentido anti-horário.
- Requisição: stop\_servo  
tipo: POST;  
parâmetros: não há;

retorna valores: não

Descrição: para o servomotor.

- Requisição: send\_servo\_command

tipo: POST;

parâmetros: register(P + valor numérico) e data(valor numérico);

retorna valores: não

Descrição: envia um comando para o servomotor através da interface serial RS232.

- Requisição: start\_experiment

tipo: POST;

parâmetros: platform\_position(valor numérico), points(valor numérico), p1(valor numérico), t1(valor numérico), m1('c', 'r'ou 'd'), p2, t2, m2, ...;

retorna valores: não

Descrição: Este comando programa e inicia um experimento. A plataforma é ajustada para a posição definida em platform\_position. O número de parâmetros p, t e m é definido pelo parâmetro points. Os parâmetros p, t e m definem, num dado momento t, a velocidade p do servomotor e qual o modo que será utilizado para se chegar à velocidade (manter a velocidade anterior (c), subir ou descer num movimento contínuo de rampa (r) ou subir ou descer bruscamente, num movimento de degrau (d)).

## ANEXO E - Cronograma

