



**DENSÍMETRO PARA MONITORAR PRODUÇÃO ARTESANAL DE  
CERVEJA:  
APRIMORAMENTO ENERGÉTICO E TRANSMISSÃO DE DADOS**

**JOÃO VICTOR DE SOUZA POLETTI**

**TRABALHO DE CONCLUSÃO DE CURSO EM ENGENHARIA DE  
COMPUTAÇÃO  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**FACULDADE DE TECNOLOGIA  
UNIVERSIDADE DE BRASÍLIA**

**UNIVERSIDADE DE BRASÍLIA  
FACULDADE DE TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**TRACKING HOMEBREWING WITH A DENSIMETER: ENERGY  
IMPROVEMENT AND DATA TRANSMISSION.**

**DENSÍMETRO PARA MONITORAR PRODUÇÃO ARTESANAL DE  
CERVEJA: APRIMORAMENTO ENERGÉTICO E TRANSMISSÃO  
DE DADOS.**

**JOÃO VICTOR DE SOUZA POLETTI**

**ORIENTADOR: RICARDO ZELENOVSKY**

**TRABALHO DE CONCLUSÃO DE CURSO EM  
ENGENHARIA DE COMPUTAÇÃO**

**PUBLICAÇÃO: PPGEA.TD-001/11**

**BRASÍLIA/DF: MAIO - 2021**

**UNIVERSIDADE DE BRASÍLIA  
FACULDADE DE TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**DENSÍMETRO PARA MONITORAR PRODUÇÃO ARTESANAL DE  
CERVEJA:  
APRIMORAMENTO ENERGÉTICO E TRANSMISSÃO DE DADOS**

**JOÃO VICTOR DE SOUZA POLETTI**

**TRABALHO DE CONCLUSÃO DE CURSO SUBMETIDO AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE GRADUANDO EM ENGENHARIA DE COMPUTAÇÃO.**

**APROVADO POR:**

---

**Prof. Ricardo Zelenovsky – ENE/Universidade de Brasília  
Orientador**

---

**Prof. Edson Mintsu Hung – ENE/Universidade de Brasília  
Examinador interno**

---

**Prof. Eduardo Peixoto Fernandes da Silva – ENE/Universidade de Brasília  
Examinador interno**

**BRASÍLIA, 24 DE MAIO DE 2021.**

## FICHA CATALOGRÁFICA

JOÃO VICTOR DE SOUZA POLETTI

Densímetro para monitorar produção artesanal de cerveja: aprimoramento energético e transmissão de dados [Distrito Federal] 2021.

xii, 63p., 210 x 297 mm (ENE/FT/UnB, Graduando em Engenharia de Computação, Engenharia de Computação, 2021).

Trabalho de Conclusão de Curso – Universidade de Brasília, Faculdade de Tecnologia.

Departamento de Engenharia Elétrica

1. Cerveja

2. Densímetro

3. Rádio

4. Fermentação

I. ENE/FT/UnB

II. Título (série)

## REFERÊNCIA BIBLIOGRÁFICA

POLETTI, J (2021). Densímetro para monitorar produção artesanal de cerveja: aprimoramento energético e transmissão de dados. Trabalho de Conclusão de Curso em Engenharia de Computação, Publicação PPGEA.TD-001/11, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 63p.

## CESSÃO DE DIREITOS

AUTOR: João Victor de Souza Poletti

TÍTULO: Densímetro para monitorar produção artesanal de cerveja: aprimoramento energético e transmissão de dados.

GRAU: Graduando em Engenharia de Computação

ANO: 2021

É concedida à Universidade de Brasília permissão para reproduzir cópias desta trabalho de conclusão de curso e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte dessa trabalho de conclusão de curso pode ser reproduzida sem autorização por escrito do autor.

---

João Victor de Souza Poletti

Departamento de Engenharia Elétrica (ENE) - FT

Universidade de Brasília (UnB)

Campus Darcy Ribeiro

CEP 70919-970 - Brasília - DF - Brasil

*Dedico este trabalho aos meus pais,  
Sandra e João Ricardo.*

## **AGRADECIMENTOS**

*Agradeço os amigos que fiz durante essa dura jornada estudando na Universidade de Brasília. Guardo com carinho os momentos que passamos juntos. Tenho certeza que sem eles essa etapa da minha vida teria sido muito mais difícil.*

*Agradeço à minha família, por terem estado ao meu lado em todos os momentos dessa trajetória e me ajudado de todas as formas possíveis.*

*Agradeço, também, meu orientador e professor Ricardo Zelenovsky pela sua dedicação, conselhos e ensinamentos. Ainda, agradeço pelas suas piadas nas reuniões semanais.*

*Por fim, agradeço a minha cadela Laika, pelo seu imenso companheirismo e pelos passeios diários.*

## **RESUMO**

**Título:** Densímetro para monitorar produção artesanal de cerveja: aprimoramento energético e transmissão de dados.

**Autor:** João Victor de Souza Poletti

**Orientador:** Ricardo Zelenovsky

**Graduação em Engenharia de Computação**

**Brasília, 24 de maio de 2021**

Este projeto dá prosseguimento ao desenvolvimento de um densímetro digital com a habilidade de aferir o estado de fermentação da cerveja de forma contínua.

Tem o objetivo de viabilizar um método de estimação da densidade da cerveja em fermentação de maneira fácil, contínua, eficiente e que atenda às novas demandas do mercado, focando na acessibilidade e baixo custo. Desse modo, os componentes presentes no dispositivo foram escolhidos de acordo com o seu custo benefício para o projeto como um todo.

O densímetro no estado atual utiliza um tubo de formato semi cilíndrico para conter o hardware. Este, por sua vez, emprega um Arduino Nano como controlador, bem como uma placa de circuito impresso de duas camadas. Além disso, contém uma memória Flash para armazenar os dados, como também um módulo de acelerômetro e giroscópio para aferir o posicionamento e angulação do equipamento.

Esta nova versão realizou alterações em diversas partes do projeto, solucionando problemas como a vida útil da bateria por intermédio de um novo sistema de ligar e desligar todo o circuito por um intervalo de tempo programável. Este sistema foi confeccionado com o acréscimo de um módulo de relógio. Ainda, foi adicionado um motor para auxiliar no posicionamento inicial do densímetro dentro do mosto, em razão da dificuldade observada na iteração passada. Também, um vibrador foi afixado com o intento de dispersar eventuais bolhas que possam se alojar no corpo do densímetro.

Por fim, ensaios foram executados para testar parte dessas novas implementações com a finalidade de checar a viabilidade e eficácia dessa nova abordagem.

**Palavras-chave:** Cerveja, Densímetro, Rádio, Fermentação.

## **ABSTRACT**

**Title:** Tracking homebrewing with a densimeter: energy improvement and data transmission.

**Author:** João Victor de Souza Poletti

**Supervisor:** Ricardo Zelenovsky

**Graduate Program in Computer Engineering**

**Brasília, May 24th, 2021**

This project provides a new stage for the development of a digital densimeter to present continuous measurements regarding the status of beer fermentation.

The main goal is to present a method of estimating the density in an easy, continuous, efficient manner that meets the new demands of the market, focusing on accessibility and low cost. Thus, the components were chosen according to their cost-benefit for the project as a whole.

The densimeter in its current state uses a semi-cylindrical-shaped tube to contain the hardware. Employs an Arduino Nano as a microcontroller, as well as a two-layer printed circuit board. In addition, it contains a Flash memory to store the data along with an accelerometer and gyroscope module to measure the positioning and slope of the equipment.

Furthermore, a few modifications all around the project had to take place to solve problems diagnosed in previous versions. Moreover, the battery life issue was solved by adding a new system to turn the entire circuit on and off for a programmable time interval using a clock module. Also, a motor was attached to help position the densimeter inside the wort. Additionally, a vibrator was appended to disperse bubbles that could be lodged in the body of the densimeter.

Finally, tests were conducted to check the feasibility and efficiency of this new approach.

**Keywords:** Beer, Densimeter, Radio, Brew.



## SUMÁRIO

---

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>1</b>
1.1	AMBIENTAÇÃO HISTÓRICA .....	1
1.2	MOTIVAÇÃO .....	2
1.3	TRABALHOS RELACIONADOS .....	4
1.4	PROPOSTA DE SOLUÇÃO .....	5
1.5	METODOLOGIA .....	6
1.6	SÍNTESE DOS CAPÍTULOS .....	7
<b>2</b>	<b>ESTRUTURAÇÃO DO DENSÍMETRO.....</b>	<b>8</b>
2.1	APERFEIÇOAMENTO .....	8
2.2	ARDUINO NANO .....	11
2.3	RÁDIO .....	11
2.4	FUNÇÃO DE LIGAR E DESLIGAR .....	16
2.5	RELÓGIO .....	19
2.6	MOTOR .....	21
2.7	SENSOR HALL .....	23
2.8	OUTROS COMPONENTES E SISTEMAS .....	26
2.8.1	ACELERÔMETRO .....	26
2.8.2	MEMÓRIAS .....	26
2.8.3	TERMÔMETRO .....	26
2.8.4	BATERIA .....	26
2.8.5	VIBRADOR .....	27
2.8.6	PYTHON DASHBOARD .....	27
2.9	MODOS DE TESTE .....	28
2.9.1	MODO FOFOCA .....	28
2.9.2	MODO XERETA .....	29
2.9.3	MODO TRANSMISSÃO CONSTANTE .....	29
2.9.4	MODO RECEPÇÃO CONSTANTE .....	30
2.9.5	MODO LIGA E DESLIGA .....	30
2.9.6	MODO MOTOR .....	30
<b>3</b>	<b>ENSAIOS .....</b>	<b>32</b>
3.1	RÁDIO .....	32
3.2	RELÓGIO E SISTEMA LIGA/DESLIGA .....	35
3.3	ENSAIOS DO LASTRO .....	42

<i>SUMÁRIO</i>	ix
<b>4 CONCLUSÃO.....</b>	<b>48</b>
<b>REFERÊNCIAS.....</b>	<b>48</b>
<b>A APÊNDICE - ENSAIOS PARA AFERIÇÃO DOS LASTROS.....</b>	<b>51</b>

## LISTA DE FIGURAS

---

1.1	Densímetro de vidro. ....	3
1.2	Densímetro com funcionalidade <i>WiFi</i> (PINTARELLI, 2019). ....	4
1.3	Dispositivo comercial TILT™ (EQUIPMENT, 2021). ....	4
1.4	Dispositivo de código aberto iSpindle (ISPINDEL, 2021). ....	5
2.1	Componentes da versão anterior do densímetro (VASCONCELOS, 2019). ....	8
2.2	Placa com circuito elétrico da versão anterior do densímetro (VASCONCELOS, 2019). ....	8
2.3	Densímetro atual. ....	9
2.4	Densímetro atual. ....	10
2.5	Diagrama de blocos dos componentes. ....	10
2.6	Arduino Nano. ....	11
2.7	Rádio digital modelo NRF24L01+. ....	12
2.8	Comunicação entre Mestre e Escravo. ....	12
2.9	Diagrama de múltipla recepção do rádio (SEMICONDUCTOR, 2008). ....	14
2.10	Endereçamento dos pipes (SEMICONDUCTOR, 2008). ....	14
2.11	Rede de rádios (SEMICONDUCTOR, 2008). ....	15
2.12	Circuito para o ensaio do funcionamento da função Liga e Desliga até a exaustão da bateria. ....	17
2.13	Circuito de liga e desliga. ....	18
2.14	<i>Real Time Clock</i> (RTC) DS3231. ....	19
2.15	Estruturação da interface I2C. ....	20
2.16	Gráfico da vida útil da bateria. ....	21
2.17	Motor utilizado (TECHNOLOGY, c2008). ....	22
2.18	Circuito de controle do motor utilizando ponte H. ....	22
2.19	Sensor de efeito Hall. ....	24
2.20	Circuito reduzido do funcionamento dos sensores de efeito Hall. ....	25
2.21	Ilustração do funcionamento do modo Fofoca. ....	28
2.22	Ilustração do funcionamento do modo Xereta. ....	29
3.1	Leitura dos registradores utilizando a biblioteca. ....	32
3.2	Primeira comunicação entre os rádios. ....	33
3.3	Pinagem e conexões do rádio. ....	33
3.4	Curvas de tensão para cada transistor. ....	35
3.5	Primeira tentativa de controle da alimentação. ....	36
3.6	Comportamento do pino SQW e de G1. ....	37

3.7	Controle da alimentação. ....	37
3.8	Diodo D1 em série com D5. ....	38
3.9	Medidas obtidas com o diodo.....	38
3.10	Circuito com dois transistores. ....	39
3.11	Medidas obtidas com G2 em vermelho. ....	39
3.12	Medidas obtidas com G1 em vermelho. ....	40
3.13	Cronograma do sistema Liga/Desliga usado para testar o circuito até a exaustão da bateria.....	40
3.14	Lastro da versão antiga.....	42
3.15	Configurações testados para o lastro. ....	43
3.16	Coordenadas do densímetro.....	43
3.17	Exemplo de uma posição e os ângulos obtidos.....	44
3.18	Ensaio 6 do lastro.....	45
3.19	Ensaio 9 do lastro.....	46
A.1	Configurações testados para o lastro. ....	51
A.2	Ensaio 1 do lastro.....	52
A.3	Ensaio 2 do lastro.....	53
A.4	Ensaio 3 do lastro.....	54
A.5	Ensaio 4 do lastro.....	55
A.6	Ensaio 5 do lastro.....	56
A.7	Ensaio 6 do lastro.....	57
A.8	Ensaio 7 do lastro.....	58
A.9	Ensaio 8 do lastro.....	59
A.10	Ensaio 9 do lastro.....	60
A.11	Ensaio 10 do lastro. ....	61
A.12	Ensaio 11 do lastro. ....	62
A.13	Ensaio 12 do lastro. ....	63

## LISTA DE TABELAS

---

2.1	Medição realizada apenas com o Arduino Nano e o MPU 6050 (VASCON-CELOS, 2019).....	16
2.2	Funcionamento dos sinais. ....	23
2.3	Funcionamento dos sinais. ....	24
2.4	Modos de teste do densímetro. ....	28
3.1	Configuração rádio receptor.....	34
3.2	Configuração rádio transmissor. ....	34
3.3	ENSAIO 6.....	44
3.4	ENSAIO 9.....	46
A.1	ENSAIO 1.....	52
A.2	ENSAIO 2.....	53
A.3	ENSAIO 3.....	54
A.4	ENSAIO 4.....	55
A.5	ENSAIO 5.....	56
A.6	ENSAIO 6.....	57
A.7	ENSAIO 7.....	58
A.8	ENSAIO 8.....	59
A.9	ENSAIO 9.....	60
A.10	ENSAIO 10.....	61
A.11	ENSAIO 11.....	62
A.12	ENSAIO 12.....	63

# 1 INTRODUÇÃO

---

## 1.1 AMBIENTAÇÃO HISTÓRICA

Sabe-se que a cerveja começou a ser produzida pelos sumérios, localizados na Mesopotâmia. Ainda, estima-se que o processo de fabricação dessa bebida teve início há 6 mil anos (PIMENTA et al., 2020).

Antigamente, não existia regulação ou um padrão bem definido de como a cerveja deveria ser produzida. Desse modo, as variações eram abundantes, estando presentes na sua coloração, consistência, teor alcoólico e sabor. Verificou-se, também, que essas variações eram justificadas pela utilização de diferentes insumos para sua produção, como: trigo, sorgo e arroz, bem como a tradicional cevada (PIMENTA et al., 2020).

No Brasil, a popularização e disseminação deste produto teve início no começo do século XIX, quando a família real portuguesa chega ao Brasil. No entanto, essa influência veio dos ingleses, uma vez que, na época, as relações comerciais entre a Inglaterra e Portugal eram muito fortes (ROSALIN, 2021). No entanto, foi após a Independência do Brasil que houve um aumento na importação de produtos de outros países europeus, como a cerveja.

Nesse intervalo de tempo entre a Independência e os anos 1880, muitas cervejarias foram criadas ao redor do país, principalmente nas regiões Sul e Sudeste do Brasil.

Ainda, (ROSALIN, 2021) afirma que "[...] só a partir das fundações da Antarctica Paulista em 1885 e da Brahma em 1888 o mercado cervejeiro atinge maiores patamares de consumo e qualidade. Incentivados pelas medidas econômicas de substituições de importações adotados durante a República Velha (1889-1930), os grandes produtores puderam se destacar através de melhores tecnologias empregadas no processo produtivo, eliminando dessa forma seus competidores locais e regionais."

Essa dominação do mercado por parte da Antarctica e Brahma durou muitos anos, devido, principalmente, a formação de um sindicato e a sua alta capacidade tecnológica (ROSALIN, 2021).

Foi apenas nos últimos anos que o panorama do cenário cervejeiro no Brasil começou a mudar de figura, com o crescimento das micro e nano cervejarias. De acordo com (ROSALIN, 2021) "[...] a ascensão das microcervejarias implica em muitos fatores, como por exemplo a reivindicação do verdadeiro sabor da cerveja pelos consumidores, sabor esse que pode ter se perdido com os produtos standardizados em função das vendas em grande escala."

Outra argumentação plausível para justificar esse aumento seria a de que os consumidores começaram a buscar novos sabores e experiências, requisitando novos estilos de cervejas. Dessa forma, as microcervejarias começam a desenvolver produtos com diferentes características, variando: cor, teor alcoólico, consistência, sabor, cheiro e temperatura (ROSALIN, 2021).

Agora, um dos instrumentos de maior importância, mundialmente, para realizar o controle de qualidade deste produto é o densímetro. Este, no que lhe toca, é uma ferramenta utilizada para realizar aferições da densidade de líquidos, eliminando a necessidade de se utilizar uma balança. O seu funcionamento utiliza como base o princípio da flutuabilidade. Este preceito, por sua vez, corresponde ao estado de equilíbrio que um corpo se encontra quando a ação concomitante da força peso (força da gravidade) e do empuxo resultam em um peso aparente nulo. Logo, o corpo flutua quando a força da gravidade é igual ao empuxo (OLIVEIRA; FILHO; AFONSO, 2013).

Aponta-se que o primeiro conceito de um densímetro surgiu no século IV, graças ao matemático Pappos (Alexandria, c. 290 – c. 350). A bem da verdade, de acordo com algumas cartas datadas do século IV e V, Pappos havia confeccionado um aerômetro. Este, no que lhe diz respeito, é uma ferramenta cujo objetivo é estipular a densidade, ou massa, de gases confinados em recipientes rígidos. Ainda, o aerômetro é capaz de indicar a densidade de líquidos em relação ao ar (OLIVEIRA; FILHO; AFONSO, 2013).

De acordo com (OLIVEIRA; FILHO; AFONSO, 2013), "[...] a principal evolução do instrumento nos últimos 250 anos passou pela diversificação dos tipos de densímetro (peso e volume constante), pela associação com o termômetro e pela mudança de seu elemento de equilíbrio, começando com o mercúrio e passando mais tarde para o emprego de bolas de chumbo unidas com resina polimérica". Ainda, é fato que a densidade exerce uma alta influência no cotidiano das pessoas, como na tipificação de materiais. Ademais, a área médica faz uso desse conceito físico para realizar testes de sangue e urina. Também, é um conceito utilizado para compreender o comportamento da água sob variações de temperatura.

Por fim, nos últimos anos os modelos digitais de densímetro começaram a ser fabricados, facilitando ainda mais esse controle de qualidade do estágio de fermentação da cerveja.

## **1.2 MOTIVAÇÃO**

Como foi citado na seção anterior, o aerômetro pode ser utilizado para realizar aferições de densidade. No entanto, sua precisão é bastante inferior a de um densímetro digital, tanto que não é comumente usado para essa finalidade atualmente.



Figura 1.1 – Densímetro de vidro.

Outro instrumento mais usado nos dias atuais pelos cervejeiros artesanais para monitorar o processo de fermentação é o densímetro de vidro - usualmente chamado de "vareta de vidro"(Figura 1.1). O seu funcionamento requer a remoção de um amostra do mosto, bem como verificações de temperatura, aplicando correções em relação a estas medidas a fim de obter uma estimativa mais precisa. Verifica-se que esse procedimento possui alguns empecilhos, capazes de gerar erros de leitura, prejudicando o processo de fermentação. Ainda, a aferição varia de acordo com o operador e ocorre o desperdício do produto, visto a necessidade de se retirar uma amostra para efetivar a verificação da densidade. É válido lembrar, também, a possibilidade de inconsistência nos intervalos de medição, bem como a possível contaminação do mosto como um todo.

Ainda mais, para realizar esse acompanhamento da fermentação das cervejas artesanais pode-se utilizar um refratômetro. Este, por sua vez, opera baseado no princípio físico da refração da luz. De acordo com (GOMES et al., 2018), "[...] A luz bate no mosto e, com base na quantidade de açúcar dissolvido na solução, a luz refrata numa direção específica, incidindo em uma escala fracionada. A densidade do mosto obtida através do refratômetro representa a razão da densidade da substância analisada (neste caso, da cerveja) pela densidade do ar, sendo assim um valor relativo, sem unidade". No entanto, ainda persiste a dificuldade de ter acesso ao mosto sem violá-lo ou contaminá-lo.

Verifica-se que as soluções disponíveis no mercado não dispõem das características de funcionamento apropriadas para realizar aferições precisas, mantendo o conteúdo do produto imaculado. Paralelamente, necessita-se que este dispositivo tenha um significativo custo benefício.

Adicionalmente, foi possível aprender com a versão anterior do projeto que o densímetro precisava de uma abordagem diferente, sofrendo reestruturações em partes críticas do funcionamento do aparelho. Um dos problemas estava relacionado à fixação dos componentes do densímetro, a qual interferia no comportamento das curvas obtidas. Também, melhorias



em relação ao consumo da bateria foram propostas, bem como uma montagem diferenciada para garantir a flutuabilidade e controle do posicionamento do equipamento.

### 1.3 TRABALHOS RELACIONADOS

Ao realizar pesquisas com o objetivo de encontrar trabalhos similares, foi encontrado o projeto acadêmico descrito em (PINTARELLI, 2019). O equipamento desenvolvido é de instalação imediata e faz uso da rede *WiFi* disponibilizada pelo cliente, sendo automático, possui bateria interna, sem fios e é capaz de fornecer os dados das medidas em servidores *online*.

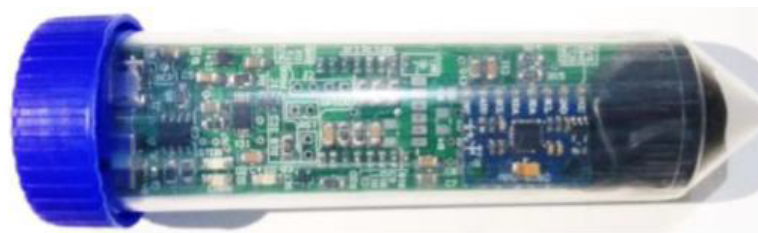


Figura 1.2 – Densímetro com funcionalidade *WiFi* (PINTARELLI, 2019).

Ainda, foi encontrado o Dispositivo comercial TILT™, comercializado pela empresa *Baron Brew Equipment, LLC* (EQUIPMENT, 2021) (Santa Rosa, CA, Estados Unidos), modelos US2014260607-A1 e US9234828-B2. O equipamento produzido pela empresa utiliza a inclinação como método de aferição. Este produto é comercializado no valor de \$135,00 dólares. O presente trabalho faz uso do mesmo conceito físico para obter as medições de densidade.



Figura 1.3 – Dispositivo comercial TILT™ (EQUIPMENT, 2021).

Por fim, foi descoberto um projeto em desenvolvimento denominado *iSpindle* (ISPINDEL, 2021). Assim como o TILT, o *iSpindle* também faz uso da inclinação para realizar medições. A sua comunicação e transmissão de dados é efetuada por intermédio da rede *Wi-Fi* do usuário.



Figura 1.4 – Dispositivo de código aberto iSpindle (ISPINDEL, 2021).

## 1.4 PROPOSTA DE SOLUÇÃO

O presente trabalho tem como foco apresentar um densímetro cujo foco principal é a sua acessibilidade. Assim, seus componentes foram escolhidos tendo em mente os seguintes requisitos: funcionalidade, taxa de consumo de energia, tamanho e, principalmente, custo benefício.

O dispositivo proposto utiliza a inclinação para determinar a densidade do líquido em que está inserido. O seu funcionamento básico consiste em colocá-lo dentro de um recipiente contendo cerveja em fermentação. Em seguida, as medidas coletadas de inclinação e temperatura são armazenadas em memória e capazes de serem acessadas remotamente a qualquer momento, permitindo, assim, acompanhar o estado da fermentação sem correr riscos de violar, contaminar ou desperdiçar parte do produto.

Nessa nova iteração do projeto foi adicionado um motor para controlar o lastro, novos sensores, bem como um relógio com o intuito de aumentar a vida útil da bateria e manter o dispositivo desligado por intervalos de tempos maiores. Ainda, houve uma significativa refatoração do código para acomodar as necessidades desses novos componentes, assim como uma nova implementação do software do rádio, independente de bibliotecas externas.

## 1.5 METODOLOGIA

Neste projeto o foco principal foi o aperfeiçoamento do software do densímetro. Com a inserção de novos componentes, foi necessário reajustar e refatorar o código de acordo com as necessidades de cada hardware adicional. Ainda, conforme novos componentes foram sendo adicionados, novos modos de funcionamento foram, também, sendo criados.

Em um primeiro momento, testes com bibliotecas já criadas foram feitos para obter um maior entendimento a respeito do funcionamento de cada componente. No entanto, neste projeto, foi decidido que a implementação de cada funcionalidade seria independente e autônoma, sem qualquer auxílio de bibliotecas externas (disponibilizadas para os processadores Arduino).

Ainda, ensaios foram realizados com o intuito de esclarecer e nortear a pesquisa. A cada erro encontrado, hipóteses foram levantadas, discutidas e testadas até, finalmente, achar a solução mais adequada, tendo sempre em mente as características fundamentais do projeto: acessibilidade e custo benefício.

## 1.6 SÍNTESE DOS CAPÍTULOS

O presente trabalho foi dividido em 4 capítulos. O presente capítulo tem como objetivo realizar uma ambientação do projeto, situando o leitor em relação a alguns pontos relevantes da história da cerveja e seu processo de fermentação. Também, dispõe as razões que justificaram o desenvolvimento desse densímetro. Ainda, aponta alguns trabalhos cujas características são, a um certo nível, similares ao aparelho confeccionado neste trabalho. Por fim, é explicitado a proposta sugerida, bem como a metodologia empregada para o seu desenvolvimento.

O segundo capítulo é incumbido de explicar os componentes e sistemas mais relevantes deste trabalho. Cada tópico descreve o modelo do dispositivo, suas características principais e razões de uso, sua forma de funcionamento, bem como contém figuras com o intuito de clarificar a discussão feita.

O terceiro capítulo elucida os ensaios mais pertinentes realizados no decorrer do desenvolvimento do densímetro. Dentre eles, foram citados os testes relacionados ao relógio em conjunto com o sistema de liga e desliga, tal como os experimentos referentes ao incremento do *software* para realizar o controle do rádio. Por fim, ocorre a descrição dos testes do lastro, relativos ao seu posicionamento no densímetro.

Finalmente, o quarto e último capítulo aponta algumas melhorias futuras, questionamentos não solucionados, bem como realiza o desfecho do presente trabalho.

# 2 ESTRUTURAÇÃO DO DENSÍMETRO

## 2.1 APERFEIÇOAMENTO

A Figura 2.1 e a Figura 2.2 elucidam a versão anterior do densímetro. É possível observar todos os componentes utilizados, bem como o posicionamento destes na placa. Também, alguns itens não estavam soldados e não era utilizada uma placa de circuito impresso para integrá-los. Ademais, uma bucha foi empregada para conter o lastro na parte inferior do tubo.

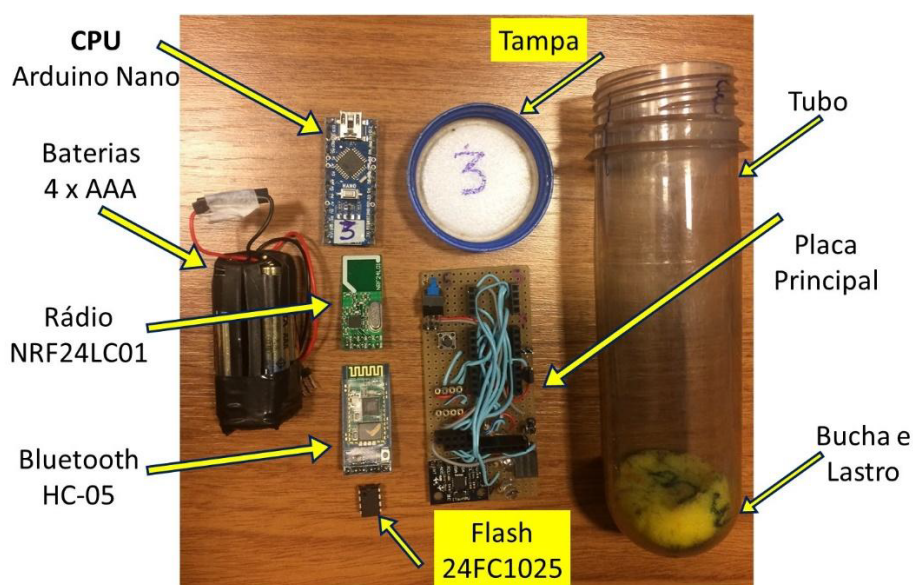


Figura 2.1 – Componentes da versão anterior do densímetro (VASCONCELOS, 2019).

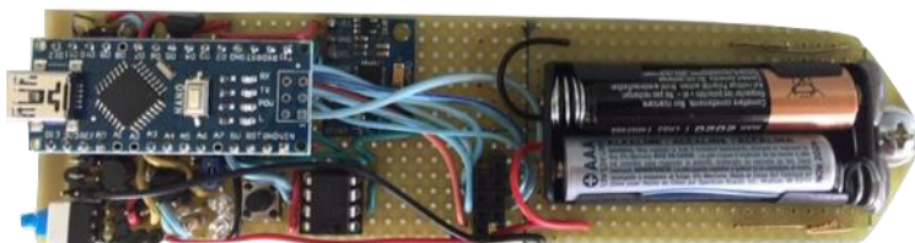


Figura 2.2 – Placa com circuito elétrico da versão anterior do densímetro (VASCONCELOS, 2019).

Em contraste, a Figura 2.3 e a Figura 2.4 apresentam as alterações feitas no equipamento. Nota-se que foi confeccionado uma placa de circuito impresso, desenvolvida por Matheus Rotta Ribeiro (2021), com o objetivo de fixar todos os componentes para dar robustez e aprimorar a distribuição espacial, bem como refinar o comportamento das curvas obtidas.

Também um dos problemas observados na versão passada do aparelho era a duração das baterias usadas - 4 pilhas AAA - cuja durabilidade conjunta não passava de 7 dias em todos os testes realizados. A meta pretendida era de 21 dias. A versão atual do densímetro propõe uma solução para este problema, a qual emprega um relógio, indicado na Figura 2.3, um sistema de ligar e desligar o circuito como um todo, bem como uma nova bateria de maior capacidade.

Ainda, constatou-se que o manuseio do lastro (ajuste do centro de gravidade) era uma tarefa complicada. Este componente era extremamente vulnerável a alterações, não sendo robusto o suficiente, visto a necessidade do densímetro se manter em um ponto de referência crítico. Em outras palavras, a sua fragilidade em relação a alterações na distribuição dos pesos, resultando em mudanças não desejadas de sua posição, indicou ser problemático. A versão atual do equipamento agrega um motor, bem como faz uso de dois sensores de efeito Hall e de um lastro para solucionar essa questão. Em outras palavras, a versão atual permite o ajuste do centro de massa do densímetro. O motor e os sensores Hall podem ser observados na Figura 2.3.

Por fim, foi adicionado um vibrador - vide Figura 2.3 - para libertar as eventuais bolhas que podem se fixar na parte exterior do densímetro. A hipótese gerada é a de que esses corpos estranhos possam alterar a flutuabilidade do equipamento.

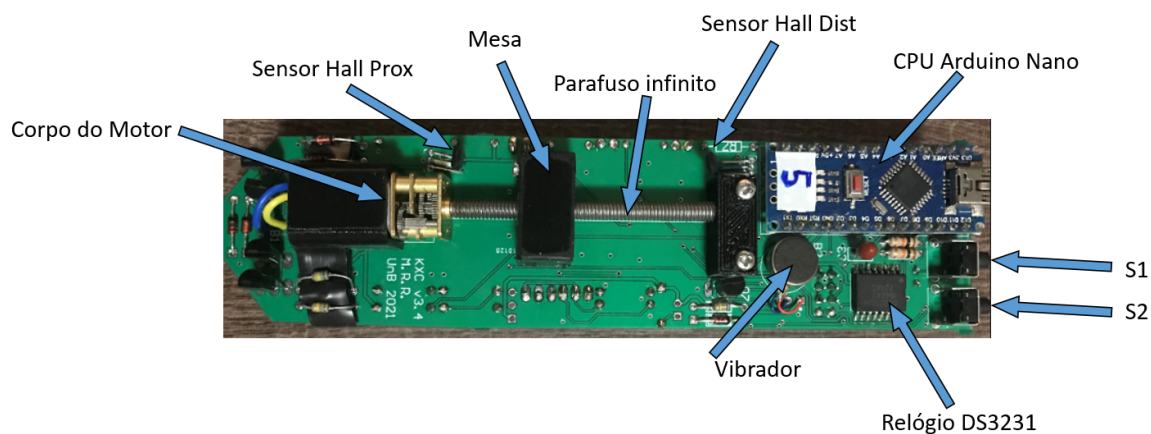


Figura 2.3 – Densímetro atual.

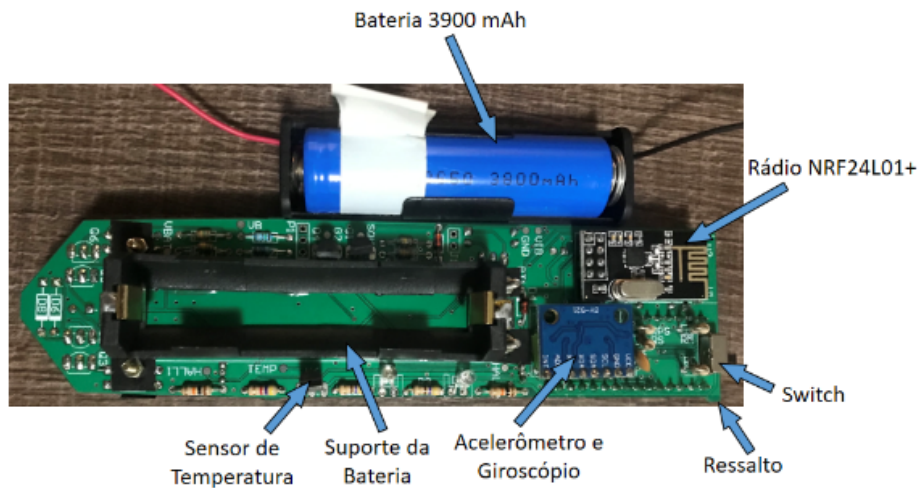


Figura 2.4 – Densímetro atual.

Para facilitar ainda mais o entendimento, o diagrama contido na Figura 2.5 elucida de forma compacta e encapsulada as conexões feitas entre os módulos e o Arduino Nano.

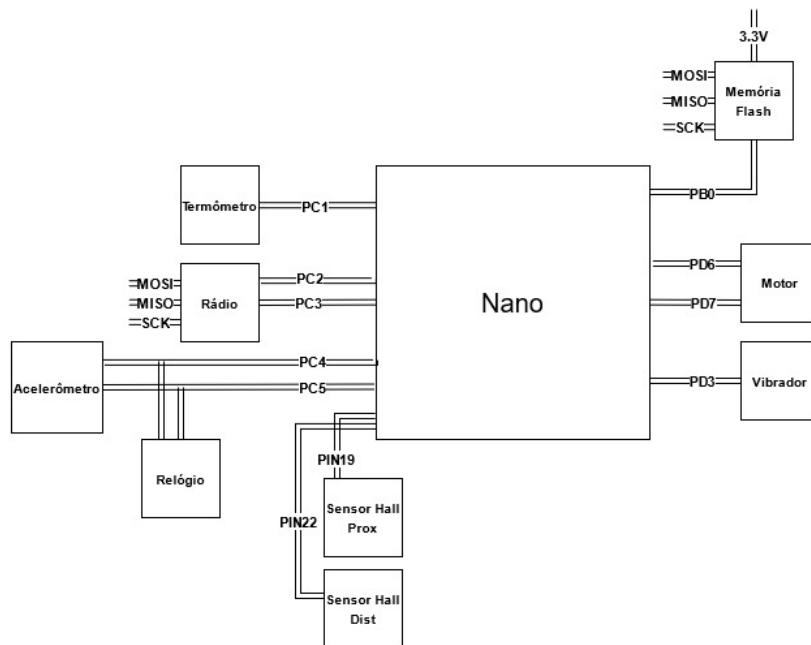


Figura 2.5 – Diagrama de blocos dos componentes.

As seções seguintes realizam explicações relacionadas ao modelo de cada componente pertinente a este trabalho, bem como das características, das razões de uso e o seu funcionamento.

## 2.2 ARDUINO NANO

A placa de prototipagem escolhida para esse projeto foi o Arduino Nano. Essa escolha foi feita levando em consideração as suas dimensões, uma vez que o densímetro consiste em um tubo de plástico de formato aproximadamente cilíndrico. Como o espaço disponível dentro desse tubo é consideravelmente limitado, o Nano foi a escolha mais apropriada.

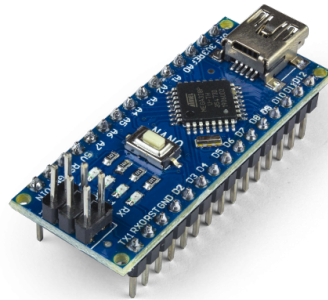


Figura 2.6 – Arduino Nano.

Ainda, o Arduino Nano escolhido usa uma CPU Atmega 328. Esta CPU, por sua vez, oferece performance similar aos arduinos de maior porte. A voltagem de operação está na faixa de 1,8V até 5,5V, possui uma memória flash com capacidade para 32 KB, uma *Electrically-Erasable Programmable Read-Only Memory* (EEPROM) de 1 KB e uma *Static Random Access Memory* (SRAM) de 2 KB. Ademais, possui 14 pinos digitais de entrada e saída (I/O), dos quais 6 permitem a utilização de *Pulse-Width Modulation* (PWM) (NANO, 2021).

Por fim, foi utilizado a *Integrated Development Environment* (IDE) padrão do Arduino para esse projeto para auxiliar o desenvolvimento do código.

## 2.3 RÁDIO

Neste projeto, o rádio tem a função de efetivar o acompanhamento da fermentação. Para tal, fornece a comunicação necessária para acessar os dados gravados na memória do densímetro inserido no barril.

O modelo do rádio escolhido foi o NRF24L01+. Ele é desenvolvido pela *Nordic Semiconductor* (SEMICONDUCTOR, 2008). Faz uso a banda de 2,4 GHz e pode operar com taxas de transmissão de 250 kbps a 2 Mbps. Se usado em espaço aberto e com a taxa de



transmissão mais baixa, seu alcance pode chegar a até 100 metros.

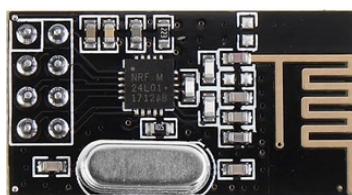


Figura 2.7 – Rádio digital modelo NRF24L01+.

Ainda, o rádio emprega a comunicação *Serial Peripheral Interface* (SPI). A interface SPI possui no mínimo 4 linhas, velocidade de transmissão de até 2 Mbps e opera com a comunicação do tipo *Full Duplex*. De acordo com (ZELENOVSKY RICARDO E MENDONÇA, 2019), "[...] a comunicação SPI é feita com a conexão de dois registradores de deslocamento, um do Mestre e o outro do Escravo [...]. A transferência é sempre full-duplex, ou seja, sempre nos dois sentidos. Enquanto o Escravo envia bits para registrador de deslocamento do Mestre (linha MISO), o Mestre envia bits para o registrador de deslocamento do Escravo (linha MOSI). Note que se forma um anel. Para cada bit que o Mestre envia, ele recebe um bit do Escravo. Ao final de 8 batidas do relógio SCK os bits estarão permutados.". É válido lembrar que a comunicação é sempre iniciada pelo Mestre, colocando em nível baixo a linha SS (*Slave Select*) do Escravo com o qual ele deseja se comunicar. A Figura 2.8 ilustra o funcionamento dessa comunicação. Por fim, é válido citar que esta comunicação é feita de forma síncrona.

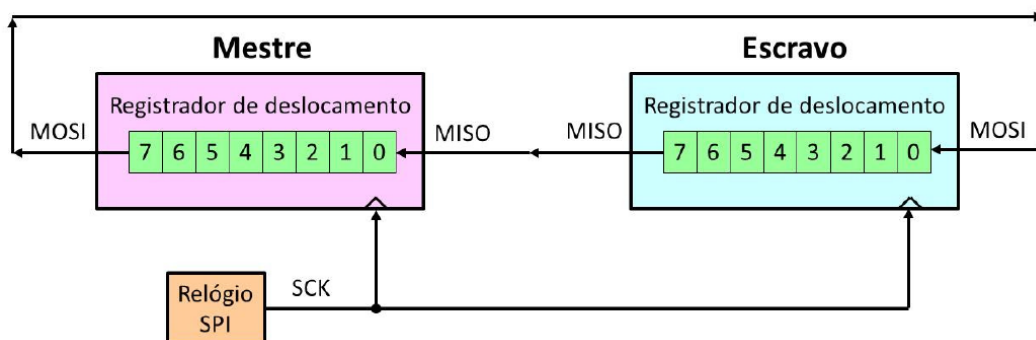


Figura 2.8 – Comunicação entre Mestre e Escravo.

O módulo rádio pode usar 125 canais diferentes, permitindo ter uma rede de 125 rádios trabalhando independentemente em um só lugar. Cada canal pode ter até 6 endereços, em outras palavras, cada rádio pode se comunicar com até 6 outras unidades simultaneamente.

O consumo de corrente deste módulo é por volta de 11,3 mA durante a transmissão e 13,5 mA durante uma recepção a uma taxa de 2 Mbps. A tensão de operação do módulo é de 1,9 a 3,6 V. Porém, os pinos toleram valores lógicos de até 5V. Logo, é possível conectá-lo a

um Arduino sem usar nenhum conversor de nível lógico, visto que internamente possui um regulador de tensão permitindo ser alimentado por uma fonte de 5V.

Este componente possui 8 pinos, um VCC e outro GND e os outros estão descritos a seguir:

O pino **CE** é sempre uma entrada em relação ao rádio. É usado para controlar a transmissão e recepção de dados nos modos TX e RX, respectivamente.

O pino **CSN** habilita o barramento SPI, e é ativado quando estiver em nível baixo. Deve-se mantê-lo sempre em nível alto, exceto quando estiver enviando ao dispositivo um comando SPI ou obtendo dados no barramento SPI do chip. Quando esse pino passa para o nível baixo, o dispositivo fica inativo para comunicação SPI.

O pino **SCK** Fornece o pulso de *clock*, regendo o funcionamento da comunicação SPI.

O pino **MOSI** é conectado ao pino MOSI do *Microcontroller Unit* (MCU), para o módulo receber dados do MCU.

O pino **MISO** é conectado ao pino MISO do MCU, para o módulo enviar dados do MCU.

O pino **IRQ** não foi empregado nesse projeto, mas é utilizado para gerar interrupções por recepção ou por transmissão de dados.

Este rádio, também, implementa a funcionalidade *Enhanced ShockBurst™*. Esta, por sua vez, é uma camada de enlace de dados baseada em pacotes que apresenta montagem automática de pacotes e temporização, reconhecimento automático e retransmissões de pacotes. *Enhanced ShockBurst™* permite a implementação de comunicação de baixo consumo de energia e alto desempenho. Os recursos fornecidos permitem melhorias significativas de eficiência de energia para bidirecionamento e sistemas unidirecionais, sem adicionar novas camadas de complexidade.

Agora, na iteração anterior deste projeto foi utilizado a biblioteca RF24.h, a qual contém funções já implementadas, facilitando o fluxo de aprendizagem e acelerando o trabalho com o rádio, tornando-o mais amigável. Encapsula detalhes de registradores e realiza configurações como transmissor ou receptor. Porém, todas as funcionalidades que foram necessárias para este trabalho foram escritas sem o auxílio dessa biblioteca, fazendo uso, apenas, do *datasheet* do componente como guia.

Como citado anteriormente, cada rádio, em modo de recepção, é capaz de se comunicar e receber pacotes de até outros 6 rádios que estejam em modo de transmissão. Cada conexão é *full-duplex*, como verificado na Figura 2.9.

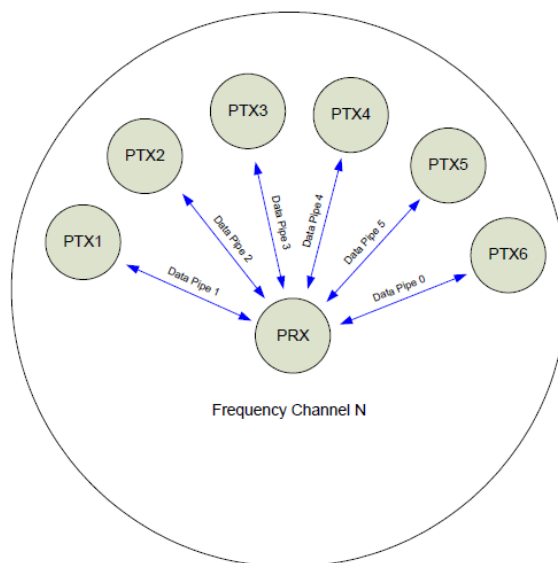


Figura 2.9 – Diagrama de múltipla recepção do rádio (SEMICONDUCTOR, 2008).

Essa capacidade é devida ao fato de que o rádio é equipado com 6 *pipes* paralelos, cada um com um endereço único - configurável -, evitando conflitos e sobreposições de pacotes. A busca pelos endereços dos pipes é realizada simultaneamente, no entanto, apenas um pipe pode receber um pacote em um dado momento. Ainda, cada pipe pode fazer uso do *Enhanced ShockBurst™*.

	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0
Data pipe 0 (RX_ADDR_P0)	0xE7	0xD3	0xF0	0x35	0x77
Data pipe 1 (RX_ADDR_P1)	0xC2	0xC2	0xC2	0xC2	0xC2
Data pipe 2 (RX_ADDR_P2)	0xC2	0xC2	0xC2	0xC2	0xC3
Data pipe 3 (RX_ADDR_P3)	0xC2	0xC2	0xC2	0xC2	0xC4
Data pipe 4 (RX_ADDR_P4)	0xC2	0xC2	0xC2	0xC2	0xC5
Data pipe 5 (RX_ADDR_P5)	0xC2	0xC2	0xC2	0xC2	0xC6

Figura 2.10 – Endereçamento dos pipes (SEMICONDUCTOR, 2008).

A Figura 2.10 ilustra um exemplo de configuração dos endereços dos pipes. Observa-se que os pipes de 1 a 5 possuem os mesmos 4 bytes mais significativos no seu endereço. No entanto, o byte menos significativo é particular para cada um, assim, individualiza os pipes. Caso seja do interesse do usuário, é possível colocar um endereço de menor tamanho, mas nunca excedendo 5 bytes. A Figura 2.11 mostra um exemplo de configuração de uma rede de rádios.

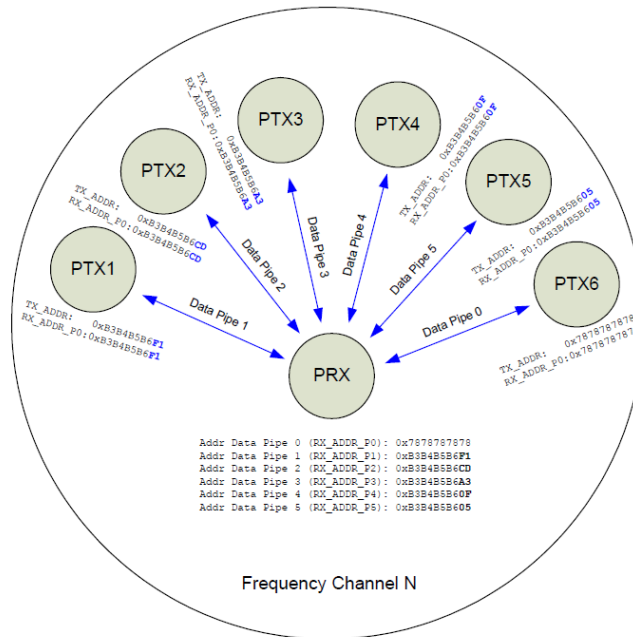


Figura 2.11 – Rede de rádios (SEMICONDUCTOR, 2008).

A comunicação *full-duplex* é realizada por intermédio do pacote de autoreconhecimento que o receptor envia para o transmissor ao receber um pacote deste. Claro, com o objetivo de realizar esse envio, assim que o receptor adquire um novo pacote, o rádio altera seu modo para o de transmissão de forma automática - se a tecnologia de *Enhanced ShockBurst™* e a flag de envio de pacotes de autoreconhecimento com *payload* estiverem ativas - e é capaz de inserir nesse pacote de resposta um *payload* contendo alguma informação extra (*payload*). Caso a flag de adicionar um *payload* a esse pacote de autoreconhecimento estiver desativada, apenas a informação de autoreconhecimento é enviada. Assim que o envio for finalizado, o rádio retorna ao modo de recepção.

Por outro lado, assim que o transmissor enviar um pacote (considerando o uso de *Enhanced ShockBurst™*), o rádio muda para o modo de recepção e espera por um pacote de autoreconhecimento. Caso o transmissor primário não receba o pacote de autoreconhecimento imediatamente, *Enhanced ShockBurst™* automaticamente retransmite o pacote de dados original após um atraso programável. A quantidade de retransmissões é configurável. Caso o pacote não seja recebido neste intervalo de tempo, o rádio para de realizar envios e entra em modo de *standby*.

O tamanho do pacote que é enviado e recebido pelo rádio é editável, podendo chegar a 32 bytes. Na implementação atual do projeto os pacotes são sempre de 32 bytes. No entanto, para lidar com pacotes cuja informação útil é menor que 32 bytes, completa-se a mensagem com zeros, de forma que o receptor sempre receba pacotes do tamanho previamente configurado, mesmo que a informação útil seja menor. O *Enhanced ShockBurst™*

possui a funcionalidade de lidar, de forma automática, com pacotes com tamanhos dinâmicos, eliminando essa necessidade de enviar mensagens vazias a fim de completar o tamanho configurado. No entanto, essa funcionalidade ainda não está sendo utilizada no projeto.

Por fim, é perceptível que o software desenvolvido para o rádio NRF24L01+ foi expandido de forma a facilitar sua integração com outros componentes do densímetro, viabilizando a capacidade deles conversarem entre si via uma interface desenvolvida cujo cerne é a comunicação SPI.

## 2.4 FUNÇÃO DE LIGAR E DESLIGAR

Na versão anterior deste trabalho, desenvolvido por (VASCONCELOS, 2019), foi observado que a fonte de alimentação utilizada - 4 pilhas AAA de 1,5V cada - não durava o tempo necessário para um ciclo completo de uma fermentação (por volta de 2 a 4 semanas). A justificava encontrada diz respeito ao tempo em que o arduino ficava dormindo. O processo para tal iniciava quando o arduino entrava no modo de sono profundo (*Power-Down*), onde a maior parte dos recursos internos da CPU são desligados, sendo acordado pelo *Watchdog Timer* (WDT) 8 segundos após entrar nesse estado. Ainda, houve uma redução do clock da CPU, de 16 MHz para 1 MHz, com o intuito de reduzir o consumo de energia (vide tabela 2.1).

Tabela 2.1 – Medição realizada apenas com o Arduino Nano e o MPU 6050 (VASCONCELOS, 2019).

Clock	Ativo	Power-Down	Período do WDT
16 MHz	13,4 mA	2,4 mA	8,683 seg / Vcc = 4,7 V
1 MHz	5,7 mA	2,4 mA	8,694 seg / Vcc = 4,7 V

O funcionamento do densímetro ainda é igual ao do projeto anterior. Ocorrem medições periódicas a cada hora e após cada medição ele verifica se existe algum receptor presente (seu funcionamento é explicitado em mais detalhes na seção **2.9.2** deste capítulo). Se não obtiver resposta do receptor dentro de um intervalo programável, o densímetro retorna ao estado de sono profundo. Caso contrário, ele transmite todos os dados das memórias Flash e EEPROM até aquele momento e volta a dormir.

No entanto, o restante do hardware permanecia energizado durante esse intervalo. Assim, foi inevitável verificar uma alta drenagem da bateria. Desse modo, mudanças em relação ao hardware foram feitas, bem como ajustes no código do projeto.

A fim de diminuir a taxa de consumo da bateria e aumentar a vida útil desta, o circuito da Figura 2.12 foi desenvolvido com a intenção de testar a solução proposta para esse problema.

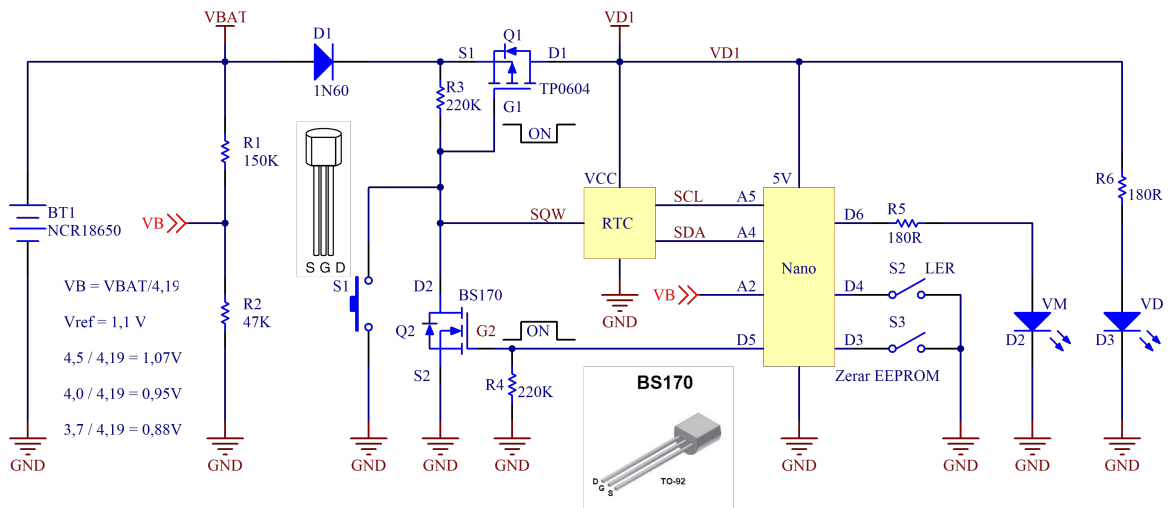


Figura 2.12 – Circuito para o ensaio do funcionamento da função Liga e Desliga até a exaustão da bateria.

Os dispositivos mais relevantes da Figura 2.12 são:

- Bateria (BT1)
- Diodo (D1)
- Transistor Q1 (canal P)
- Transistor Q2 (canal N)
- RTC
- Arduino Nano
- Resistores R3 e R4

O diodo, por sua vez, tem a finalidade de impedir que a tensão injetada pela porta USB, quando ligada ao arduino nano, trafegue para a bateria BT1. Ainda, o diodo escolhido é feito de germânio, em oposição ao de silício - o qual possui uma queda de tensão em condução de 0,7 - possuindo uma queda de tensão em condução por volta de 0,2 a 0,3 V.

O transistor Q1 de canal P passou por vários testes de tensão para definir qual modelo utilizar. Dentre os examinados estão o TP0604, TP0606 e IRF9530. Esses transistores foram ensaiados porque foram os únicos encontrados no fornecedor local com encapsulamento TO-92. O TP0604 mostrou-se mais adequado para as finalidades do projeto. Pela Figura 2.12 é possível observar que ao colocar o seu GATE em nível baixo, o transistor Q1 liga. Por outro lado, se for colocado em nível alto, o transistor Q1 ficará desligado.

Inicialmente, a intenção era usar apenas um transistor. Desse modo, conectou-se o GATE de Q1 ao pino D5 do microcontrolador. Entretanto, para evitar o fluxo de corrente pelo

transistor de canal P é necessário manter seu GATE em nível alto. Assim, para desligar o Arduino basta colocar o GATE de Q1 em nível alto. Quando isto é feito, a tensão da CPU começa a cair e também a tensão no pino que estava em nível alto. Isto começa a levar o transistor de volta à condução. Em suma, com apenas Q1 não era possível cortar completamente a alimentação de todo o circuito. O transistor Q1 ainda permanecia dentro da faixa de condução, permitindo a passagem de uma pequena corrente.

Era necessário, então, fazer ao contrário. Encontrar uma forma de cortar a alimentação com uma saída em nível baixo. A solução encontrada para esse problema foi adicionar outro transistor, agora de canal N - conectando-o ao Nano pelo pino D5 - o qual permite fluxo de corrente quando mantido em nível alto. Assim que a alimentação é ligada, o programa tem como primeira atividade configurar e garantir que o Gate de Q2 esteja em nível alto. Para garantir a partida, é preciso de alguma forma colocar o GATE de Q1 em nível baixo. Isso é obtido com a ação do usuário ao acionar a chave S1 que curto-circuita o GATE de Q1 para terra. Durante a operação normal, com o uso do relógio, por ocasião do momento de ligar o circuito, a saída SQW (Figura 2.13) deste relógio coloca em nível baixo o GATE de Q1. Paralelamente, os resistores R3 (*pull-up*) e R4 (*pull-down*) foram adicionados para que, na eventualidade de mau contato ou uma casualidade não prevista, seja possível definir um nível de tensão no GATE. É válido ressaltar que a adição desses resistores é típica neste tipo de circuito.

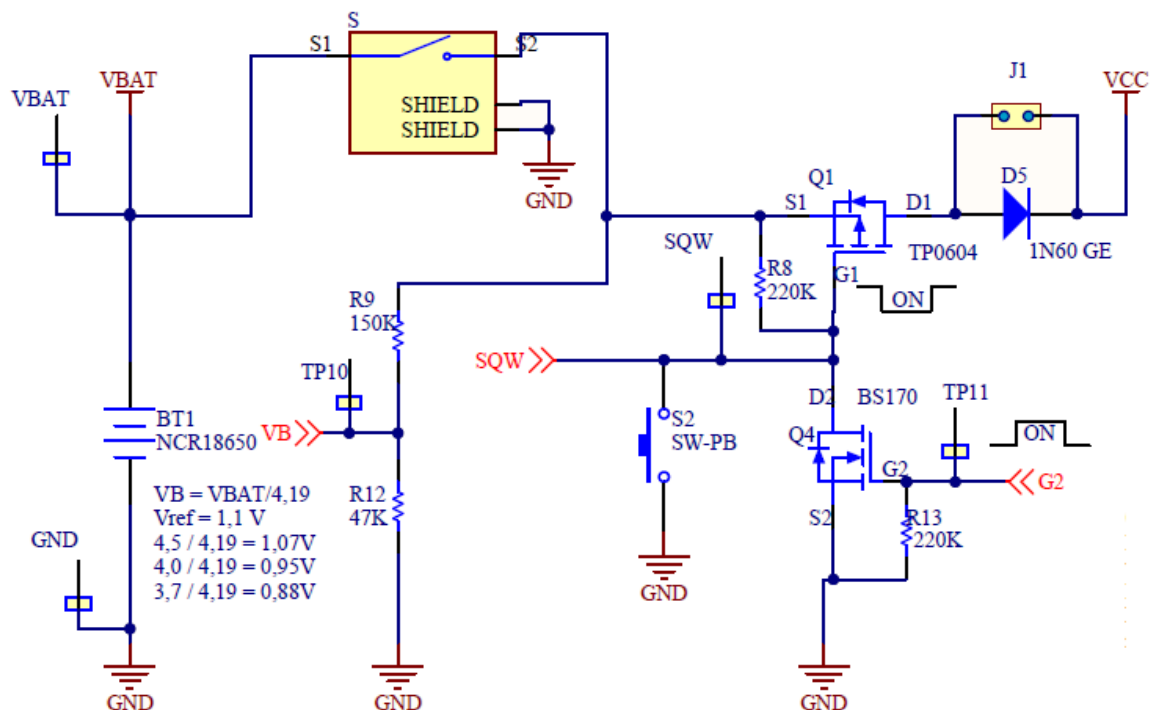


Figura 2.13 – Circuito de liga e desliga.

Tendo em mente a eficiência do ensaio realizado utilizando o circuito da Figura 2.12, o

circuito final foi desenvolvido e é ilustrado pela Figura 2.13. Observa-se que foi adicionado uma chave em série com a bateria para que o usuário possa cortar toda a alimentação do circuito. O único circuito que permanece energizado (não está mostrado na figura) é o suporte para manter o relógio de tempo permanente (RTC). Vale ressaltar que ao fechar a chave o circuito não está ligado ainda, restando apertar o botão S2 para efetivar o fluxo de corrente. Ao apertá-lo, coloca-se o GATE de Q1 em contato com o terra do circuito, ligando-o e fechando o circuito. Adicionalmente, outra maneira de ligar é por intermédio do relógio (SQW), o qual também é capaz de puxar o GATE G1 para nível baixo. Por fim, ao ligar o Arduino o GATE G2 fica em nível alto, ligando o transistor Q4 e realizando o curto para a terra, fechando o circuito e efetivando o devido comportamento deste como um todo.

## 2.5 RELÓGIO

O relógio, modelo RTC DS3231 (vide Figura 2.14), produzido pela *Dallas Semiconductor Maxim* (MAXIM, 2010), está sendo utilizado neste projeto.

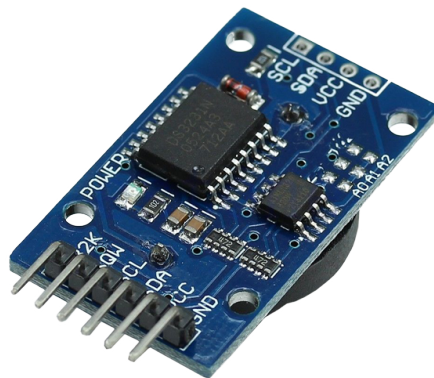


Figura 2.14 – RTC DS3231.

Este modelo é caracterizado por possuir baixo consumo e utilizar a interface I2C. Este tipo de comunicação faz uso de um barramento, é realizado de forma síncrona sendo bidirecional, mas não full-duplex. Para seu funcionamento são necessários 2 condutores, utilizando a metodologia de Mestre-Escravo. Possui endereçamento de 7 bits (capacitando 128 endereços) e velocidade de transmissão, no modo *fast-mode*, de 400 KBits por segundo.



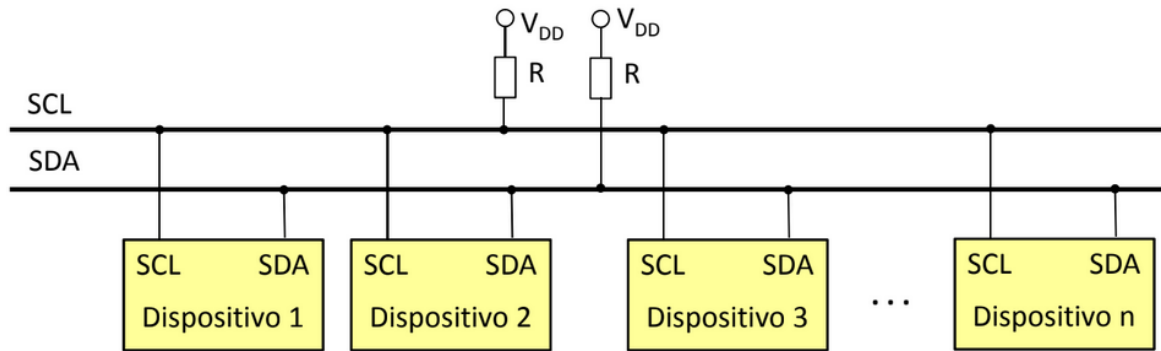


Figura 2.15 – Estruturação da interface I2C.

A linha (SCL ou SDA) vai para nível baixo quando um dos dispositivos a ela conectado faz sua saída igual a zero. Paralelamente, o nível alto é atingido quando um dos dispositivos está com sua saída em um. Finalmente, o Mestre é o dispositivo que inicia e finaliza a transmissão e está incumbido, também, de gerar o relógio SCL. O Escravo, por sua vez, só entra em ação quando endereçado pelo Mestre. O transmissor é constituído pelo dispositivo que dispõe dados no barramento, ao passo que o Receptor é aquele que realiza leituras a partir deste barramento.

Adicionalmente, o relógio opera com tensão de 3,3V a 5V. É capaz de trabalhar segundos, minutos, horas, dias da semana, dias do mês, meses e anos em tempo real. Vem equipado, também, com um sensor de temperatura com  $\pm 3$  °C de exatidão.

Ademais, o relógio dispõe de um sistema para automaticamente detectar eventos de falha de energia, alterando para a energia reserva. Isso ocorre graças ao circuito comparador presente, trabalhando em conjunto com uma voltagem de referência baseada em temperatura para verificar, constantemente, os níveis de *Voltage Common Collector* (V<sub>cc</sub>).

No presente trabalho, o relógio tem como finalidade ajudar no controle do consumo de energia do dispositivo, dado que o densímetro possui apenas a bateria como fonte de energia e está isolado dentro de um recipiente sem ter acesso às fontes externas. Desse modo, é possível programar intervalos de tempo em que o densímetro ficará desligado e o relógio terá a responsabilidade de acordá-lo por intermédio de um alarme configurável. Para testes, o intervalo selecionado foi de 1 minuto. Idealmente, em um ambiente de produção, o intervalo deverá ser de 1 hora.

Com o intuito de demonstrar a importância do relógio no circuito e como ele influencia a vida útil da bateria, a Figura 2.16 ilustra seu comportamento em ação.

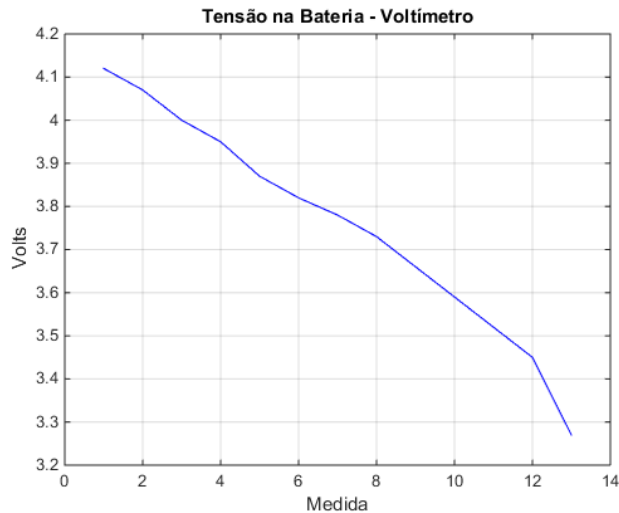


Figura 2.16 – Gráfico da vida útil da bateria.

Para a realização deste teste, o circuito foi reduzido ao máximo, estando equipado apenas com RTC e um LED - além de outros componentes, como transistores, diodos e resistores cujo consumo é insignificante para o presente teste -, o qual fica acesso em todos os momentos. Isso foi feito para acelerar um pouco o consumo da bateria. Ainda, o relógio foi programado para despertar a cada 60 segundos. Verificou-se que o consumo total de corrente, quando o dispositivo estava ligado, era de 16,5 mA. Por outro lado, quando encontrava-se desligado, o consumo diminuiu para 2,2 mA, o qual corresponde ao do LED do RTC, pois neste ensaio foi usada uma pequena placa com esse CI e um LED. As 13 medidas foram obtidas ao longo de 6 dias.

É válido citar que neste ciclo de 60 segundos, o dispositivo ficava ligado por 45 segundos (75% do tempo) e desligado nos restantes 15 segundos. Extrapolando para uma aplicação real, onde ficaria ligado 1 minuto a cada hora, tem-se alimentação para mais de 200 dias. Excedendo em muito a proposta do projeto.

## 2.6 MOTOR

Constatou-se uma certa dificuldade, na versão anterior deste projeto, de conseguir manusear o lastro do densímetro, de forma a mantê-lo na posição desejada. O dispositivo revelou grande sensibilidade na posição do seu centro de gravidade. A cada manuseio do equipamento, por exemplo uma troca de bateria, o centro de gravidade era alterado, modificando completamente o comportamento do densímetro. Se mostrou necessário desenvolver uma forma de ajustar esse centro de gravidade.

A presente versão utiliza um motor, modelo GA12-N20 (TECHNOLOGY, c2008), para

realizar esse controle e efetivar o posicionamento adequado do densímetro ao ser inserido no barril de fermentação. O motor escolhido deve ter uma tensão de alimentação de 3 a 5 V. Ainda, foram realizadas verificações para determinar o melhor intervalo de rotações por minuto em que o motor irá trabalhar. Nos densímetros de teste confeccionados, esse valor varia de 60 a 150 RPM.



Figura 2.17 – Motor utilizado (TECHNOLOGY, c2008).

A Figura 2.17 ilustra esse componente. Ainda, foi adicionado à sua extremidade um parafuso sem fim, com o intuito de estender a distância percorrida pela mesa (lastro), a qual ficará fixada nesse parafuso. Foi obtido um maior intervalo de ângulos de inclinação com esse aumento da distância útil a ser transitada pela mesa.

O seu funcionamento é realizado de forma conjunta com dois sensores de efeito Hall (descritos na seção seguinte). Uma ponte H foi construída para fazer o motor girar nos dois sentidos e é representada na Figura 2.18.

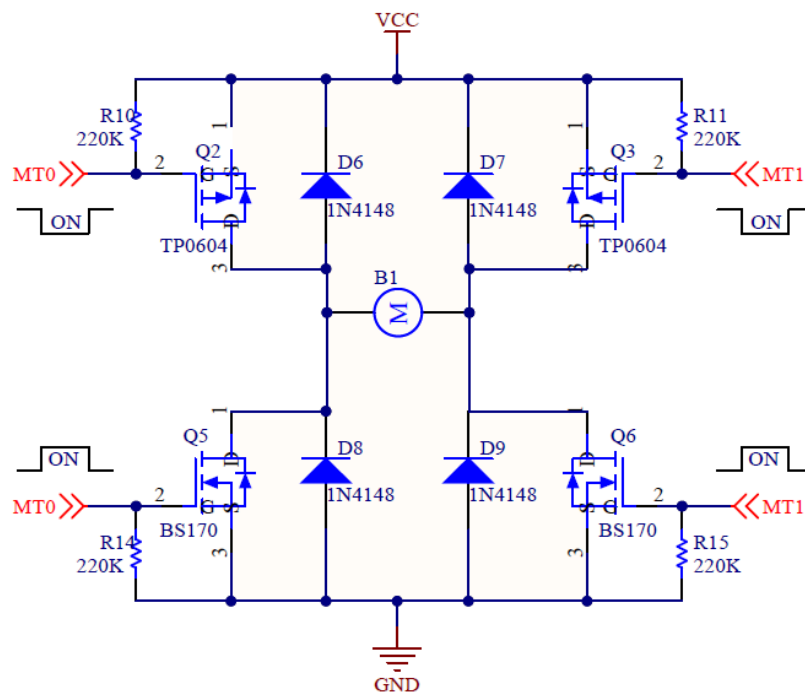


Figura 2.18 – Circuito de controle do motor utilizando ponte H.

Tabela 2.2 – Funcionamento dos sinais.

MT0	MT1	Q2	Q3	Q5	Q6
NÍVEL BAIXO	NÍVEL ALTO	LIGADO	DESLIGADO	DESLIGADO	LIGADO
NÍVEL ALTO	NÍVEL BAIXO	DESLIGADO	LIGADO	LIGADO	DESLIGADO

A tabela 2.2 demonstra o funcionamento da ponte H. O primeiro estado (MT0 em nível baixo e MT1 em nível alto) resulta em uma corrente fluindo entre o transistor Q2 e o transistor Q6. Alternativamente, no segundo estado, a corrente flui entre o transistor Q5 e o transistor Q3.

Desse modo, é possível controlar a direção em que a rotação do motor ocorre. Os diodos D6, D7, D8 e D9 tem a finalidade de proteção da força contraeletromotriz quando ocorre a supressão da corrente do motor. Esse fenômeno ocorre devido ao fato do motor ser construído com indutores, logo é passível de gerar altas tensões quando desligado. Se não tivessem esses diodos para consumirem essa energia, os transistores ficariam submetidos a tensões elevadas e poderiam queimar.

Por fim, o propósito principal do motor é efetivar o posicionamento do densímetro na inclinação desejada, geralmente 20 a 30 graus quando imerso em água pura. Esta faixa de angulação foi obtida por meio de teste realizados no trabalho anterior ((VASCONCELOS, 2019)) com diversos recipientes contendo, em cada um, água pura misturada com sal para variar a densidade do líquido. Desse modo, foi observado que se a inclinação do densímetro em água comum ficar entre 20 e 30 graus, a curva da inclinação se aproxima do caso linear desejado. Assim que a angulação programada for alcançada, o motor finaliza sua movimentação e, idealmente, não deverá mais operar até o fim da fermentação.

## 2.7 SENSOR HALL

Este componente, bem como o motor, é de suma importância para o presente trabalho. O modelo utilizado neste projeto, SS49E produzido pela *SEC ELECTRONICS INC.* (INC., 2008), pode trabalhar com uma voltagem dentro do intervalo de 4,5V até 6V. Além disso, possui uma significativa sensibilidade para campos magnéticos em volta dele. Ainda, possui um baixo consumo de corrente. Devido à sua ampla faixa de temperatura para trabalhar (de -40°C até 85°C), é passível de ser utilizado em diversas áreas, como: comercial e industrial. Ademais, as suas dimensões são mínimas, sendo um atrativo e uma das razões pelas quais foi decidido utilizá-lo neste trabalho.

Adicionalmente, foi necessário utilizar o conversor AD (analógico-digital), contido no AVR e tendo uma tensão de referência de 1,1V, para orquestrar o funcionamento adequado dos sensores de efeito Hall. Estes trabalham com sinais analógicos, logo, conectou-se os

pinos A6 e A7 do Nano para realizar essas leituras de forma adequada.

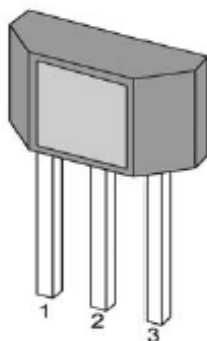


Figura 2.19 – Sensor de efeito Hall.

Tabela 2.3 – Funcionamento dos sinais.

Nome	Número	Descrição
Vdd	1	Fonte de Alimentação
Gnd	2	Terra
Output	3	Saída

A Tabela 2.3 explicita a pinagem do sensor ilustrado na Figura 2.19.

Sua principal finalidade para o circuito do densímetro é observar e capturar a proximidade da mesa ao corpo do motor, bem como a sua proximidade ao fim de curso - extremidade oposta. Em cada extremo do parafuso sem fim, é posicionado um sensor de efeito Hall. O controle desses sensores é relativamente simples. Ocorre o monitoramento constante de uma variável que acusa a proximidade da mesa em relação a eles. Em outras palavras, quando a variável que observa a proximidade acusar 1 (sim, a mesa está próximo deste sensor), inverte-se a rotação do motor.

O seu funcionamento pode ser exemplificado da seguinte forma. O sensor, quando não está sobre a influência de um campo magnético, indica o valor médio da tensão de alimentação - nesta explicação será de 4V. Ou seja, a tensão de referência é de 2V. Ao aproximar um dos polos do ímã, a tensão indicada pelo sensor começa a subir em direção aos 4V. Por outro lado, ao aproximá-lo do outro polo do ímã, a tensão de referência começa a cair para 0V. Em outras palavras, em um polo ocorre o incremento da tensão, ao passo que no outro a tensão é decrementada, partindo do nível médio. É calculado sempre o valor absoluto da diferença entre a medida obtida do Hall e os 2 V (valor médio da tensão). Houve a tentativa de acertar a polaridade dos ímãs, mas pela reduzida dimensão deles, esta tarefa acabou se mostrando muito difícil.

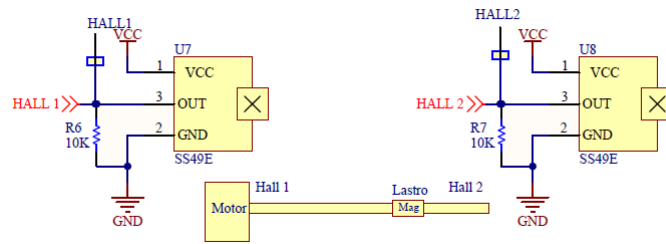


Figura 2.20 – Circuito reduzido do funcionamento dos sensores de efeito Hall.

É possível observar pela Figura 2.20 o circuito confeccionado para o seu funcionamento. Um ímã foi fixado no lastro, com o intuito de fornecer um campo magnético perceptível para os sensores de efeito Hall. Estes, por sua vez, são posicionados nos extremos da haste - constituída por um parafuso sem fim.

## 2.8 OUTROS COMPONENTES E SISTEMAS

Em um trabalho complementar a este, descrito em (BEAL, 2021), é possível verificar mudanças e acréscimos pertinentes ao projeto como um todo, as quais são citadas brevemente nas seguintes subseções.

### 2.8.1 Acelerômetro

O acelerômetro modelo MPU6050, desenvolvido pela *InvenSense*, está sendo usado no presente projeto. Este dispositivo também possui um giroscópio inserido no mesmo chip. Ainda, possui 3 eixos dedicados para ambas as funcionalidades e opera utilizando a interface I2C para realizar sua comunicação.

A sua principal função no trabalho é estimar valores de aceleração da gravidade nos eixos, bem como aferir a taxa de inclinação do densímetro. Desse modo, é possível determinar a densidade do líquido em que o equipamento está inserido.

### 2.8.2 Memórias

Paralelamente, o densímetro faz uso da memória EEPROM interna na CPU (Atmega 328). Esta, por sua vez, possui 1 KB de espaço de armazenamento e tem o objetivo de guardar informações únicas do densímetro.

Além disso, foi necessário adicionar uma memória *Flash*. O modelo escolhido foi o W25Q32BV, o qual utiliza a comunicação SPI e possui 4 MB de capacidade de armazenamento. A sua incorporação ao projeto diz respeito a necessidade de persistir os dados coletados ao decorrer do ciclo da fermentação.

### 2.8.3 Termômetro

Adicionalmente, um termômetro de alta precisão - modelo DS18B20 - foi fixado no densímetro. Utilizando o protocolo 1-Wire, este dispositivo está incumbido de realizar aferições de temperatura durante o processo de fermentação, sendo determinante para o êxito da observação do ciclo.

### 2.8.4 Bateria

Ainda, conforme explanado na seção 2.4, o densímetro fica isolado dentro de um barril de fermentação, sem contato com qualquer fonte de energia externa. Portanto, foi adicionado uma pilha recarregável, modelo 18650, com capacidade de 3800 mAh e aproximadamente

4,5 V de tensão quando totalmente carregada.

### **2.8.5 Vibrador**

Em contrapartida à versão anterior do densímetro, a presente iteração está equipada de um vibrador. Sua função parte da hipótese de que, durante a fermentação é possível que bolhas se fixem no corpo do equipamento, alterando a flutuabilidade deste. Logo, é de responsabilidade do vibrador eliminá-las.

### **2.8.6 Python Dashboard**

Ademais, revelou-se necessário o desenvolvimento de uma ferramenta cujo objetivo principal é auxiliar na leitura e interpretação dos dados decorrentes das fermentações. Assim, um programa em linguagem Python foi concebido, utilizando a biblioteca PyQt.

No presente estágio do projeto, os dados coletados incluem:

- Variações angulares.
- Coordenadas do acelerômetro (X, Y e Z).
- Tensão da bateria.
- Temperatura.

Finalmente, as informações listadas acima são lidas via porta serial e dispostas em gráficos.



## 2.9 MODOS DE TESTE

O densímetro deste trabalho possui, atualmente, 21 modos de operação. A maior parte deles servem o propósito de efetuar ensaios. A tabela 2.4 abaixo lista todos os modos usados atualmente.

Tabela 2.4 – Modos de teste do densímetro.

Modo	Descrição
1	Leds
2	Chave SW
3	DS18B20
4	RTC
5	Liga e Desliga
6	MPU
7	W25Q32 (Flash)
8	EEPROM
9	Vibrador
10	Motor
11	Carga da Bateria
12	Transmissão Contínua
13	Recepção Contínua
14	Fofoca
15	Xereta
16	Ensaio de Uso
17	Configuração
18	Vertical
19	25 Graus
20	Lastro
21	Acompanhar Fermentação

Abaixo são listados e descritos os principais modos que fazem uso do Rádio NRF24L01+.

### 2.9.1 Modo Fofoca

Com o objetivo de realizar uma comunicação mais direta com o computador, o modo Fofoca foi concebido. Basicamente, um densímetro, equipado com um módulo de rádio, fica conectado ao computador por intermédio da porta serial. Determinou-se que o endereço do rádio, neste modo, será FFK0. Isto significa que os densímetros remotos sempre transmitem para o rádio com endereço FFK0. Em outras palavras, no modo Fofoca (FFK0), o densímetro é um *transceiver*. Envia para sua porta serial tudo que chegar pelo rádio e, por outro lado, envia pelo rádio tudo que chegar pela sua porta serial. Seu funcionamento foi esboçado na Figura 2.21.

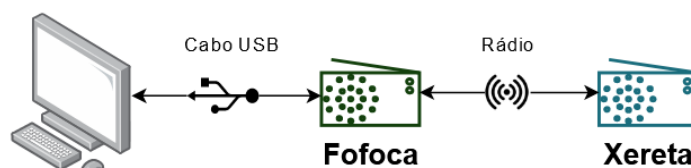


Figura 2.21 – Ilustração do funcionamento do modo Fofoca.

### 2.9.2 Modo Xereta

Por outro lado, o modo Xereta foi projetado para coletar todas as informações que o densímetro remoto - inserido dentro do barril de fermentação - possui até aquele instante. Na prática, o densímetro que se encontra neste modo é posicionado em cima do barril e seu endereço é alterado para XRT0. Quando o densímetro imerso no líquido, acompanhando a fermentação, acorda (intervalo de 1 hora), este sempre busca, via rádio, por um densímetro cujo endereço é XRT0. Ao encontrar, começa a transmitir todos os seus dados, armazenados em memória, que possui até aquele momento.

Assim, o densímetro que recebeu todos esses dados passa a ter uma cópia espelhada de todas as informações em memória do dispositivo acompanhando a fermentação. Com esses dados, o Xereta é encaminhado para a estação do computador, a qual inclui um densímetro no modo Fofoca, e a comunicação entre eles é estabelecida. Ao final dessa nova transferência, todos os dados que, em um primeiro momento, estavam isolados dentro do barril de fermentação, agora estão disponíveis no computador. Desse modo, é possível acompanhar o estado da fermentação sem correr riscos de violar, contaminar ou desperdiçar parte do produto.

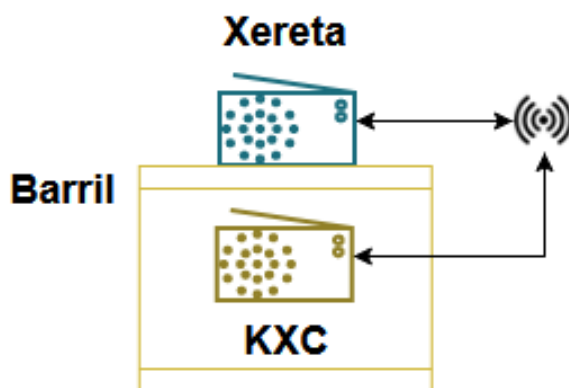


Figura 2.22 – Ilustração do funcionamento do modo Xereta.

A Figura 2.22 demonstra o funcionamento deste modo. Para fins ilustrativos, o densímetro foi reduzido a imagem de um rádio. Verifica-se a possibilidade do Xereta enviar, também, pacotes para o KXC, por intermédio dos *payloads* do seu pacote de autorreconhecimento. Idealmente, ao terminar o envio de seus dados contidos em memória, o KXC reassume sua operação normal.

### 2.9.3 Modo Transmissão Constante

Este modo foi criado para a realização de testes. Sua concepção tem como utilidade testar a função de transmissão do rádio, verificando se todos os registradores foram atribuídos da

forma correta, enviando mensagens de teste para outro rádio, o qual esteja em modo de recepção constante.

#### **2.9.4 Modo Recepção Constante**

Este modo foi criado para a realização de testes. Sua concepção tem como utilidade testar a função de recepção do rádio, verificando se todos os registradores foram atribuídos da forma correta, enviando mensagens de autorreconhecimento para outro rádio, o qual esteja em modo de transmissão constante.

#### **2.9.5 Modo Liga e Desliga**

Com o interesse de realizar ensaios relativos ao funcionamento adequado do sistema de liga e desliga, este modo foi concebido. Inicialmente, programa-se o alarme para ser ativado a cada minuto, bem como prepara-se a EEPROM com um código especial - o qual identifica este modo - para que, assim que o Nano acordar, este entrar diretamente neste modo.

Para melhorar a visualização do comportamento do Liga e Desliga, foram atribuídas as seguintes responsabilidades:

- No segundo 40, acende-se o LED vermelho.
- No segundo 45, desliga-se a alimentação (polling).
- No segundo 05, o relógio dispara o alarme (saída SQW) que irá religar a alimentação do Arduino.

Por fim, este modo vem equipado com a constante observação do input do usuário. Caso este digite no monitor serial o comando "X", termina-se o processo e o densímetro sai do modo Liga e Desliga.

#### **2.9.6 Modo Motor**

Com a finalidade de testar o funcionamento do motor, este modo foi criado.

Primeiramente, o motor deve estar parado para dar início a operação do modo. A posição da mesa (lastro) é controlada por comandos, os quais são inseridos via monitor serial e estão listados abaixo:

- P ou p: O movimento do motor é interrompido até ficar estacionário. Adicionalmente, os LEDs indicativos são desligados.

- S ou s: Este é o modo de subida. Aqui, o LED azul é acionado e o motor se move nesta direção enquanto SW (botão) estiver sendo pressionado. Em outras palavras, a mesa se afasta do corpo do motor.
- D ou d: Modo de descida. Ocorre a ativação do LED vermelho e o motor inicia a sua atividade nesta direção enquanto SW (botão) estiver acionada. Ou seja, a mesa se aproxima do corpo do motor.
- Z ou z: Modo de "zigzague" entre os dois sensores Hall. Vale lembrar que o LED azul fixado no densímetro tem o propósito de indicar o movimento de subida da mesa, ao passo que o LED vermelho indica o movimento contrário. Ainda, ocorre a impressão do contador de segundos no monitor serial.

Vale ressaltar que durante este ensaio ocorre a impressão das leituras do sensores de efeito Hall com o intuito de verificar se o modo está funcionando da forma esperada.

# 3 ENSAIOS

## 3.1 RÁDIO

Inicialmente, para realizar a configuração dos rádios, foi utilizada a biblioteca RF24.h, bem como a biblioteca SPI.h. Claro, esse passo inicial foi dado para ajudar na compreensão das necessidades dos rádios, assim como para auxiliar na estruturação de estratégias para implementar tal arquitetura.

A primeira etapa foi realizar a leitura de algum registrador do rádio, modelo NRF24L01+. Para cumprir esta etapa verificou-se no *Datasheet* do dispositivo a existência de um comando, enviado via SPI, chamado *read\_register*. É válido lembrar que toda a configuração da interface SPI já estava preenchida pela biblioteca sendo usada. Esta tarefa não apresentou muitas dificuldades e foi possível ler os principais registradores do rádio, como pode ser observado na Figura 3.1.

```
17:39:07.654 -> SPI Speedz      = 10 Mhz
17:39:07.654 -> STATUS          = 0x0e RX_DR=0 TX_DS=0 MAX_RT=0 RX_P_NO=7 TX_FULL=0
17:39:07.721 -> RX_ADDR_P0-1     = 0xb00b1e5000 0xc2c2c2c2c2
17:39:07.721 -> RX_ADDR_P2-5     = 0xc3 0xc4 0xc5 0xc6
17:39:07.789 -> TX_ADDR          = 0xb00b1e5000
17:39:07.789 -> RX_PW_P0-6          = 0x20 0x00 0x00 0x00 0x00 0x00
17:39:07.858 -> EN_AA            = 0x3f
17:39:07.858 -> EN_RXADDR        = 0x03
17:39:07.891 -> RF_CH            = 0x4c
17:39:07.891 -> RF_SETUP        = 0x07
17:39:07.925 -> CONFIG          = 0x0e
17:39:07.959 -> DYNPD/FEATURE     = 0x03 0x06
17:39:07.959 -> Data Rate        = 1MBPS
17:39:07.994 -> Model            = nRF24L01+
17:39:08.028 -> CRC Length      = 16 bits
17:39:08.028 -> PA Power       = PA_MAX
```

Figura 3.1 – Leitura dos registradores utilizando a biblioteca.

A figura acima demonstra uma configuração padrão de um rádio receptor. O registrador CONFIG possui 8 bits, um deles - bit 0 (PRIM\_RX) - é dedicado para controlar o perfil do dispositivo, configurando-o como receptor quando estiver atribuído para 0 e configurando-o como transmissor quando PRIM\_RX for igual a 1.

Seguindo essa linha de raciocínio e mantendo a configuração do outro rádio quase idêntica, apenas alterando o perfil e alguns registradores auxiliares foi possível completar a comunicação entre eles, como visto na Figura 3.2.

```

transmitter_code
1 #include <SPI.h>
2 #include <NRF24L01.h>
3 #include <RF24.h>
4
5 RF24 radio(5, 4); // CE, CSN
6
7 const byte address[6] = "00001";
8
9 void setup() {
10   radio.begin();
11   radio.openWritingPipe(address);
12   radio.setPALevel(RF24_PA_MIN);
13   radio.stopListening();
14 }
15 void loop() {
16   const char text[] = "Hello World";
17   radio.write(text, sizeof(text));
18   delay(1000);
19 }

```

**Programa Transmissor**

```

receiver_code
1 #include <SPI.h>
2 #include <NRF24L01.h>
3 #include <RF24.h>
4
5 RF24 radio(9, 10); // CE, CSN
6
7 const byte address[6] = "00001";
8
9 void setup() {
10   Serial.begin(9600);
11   radio.begin();
12   radio.openReadingPipe(0, address);
13   radio.setPALevel(RF24_PA_MIN);
14   radio.startListening();
15 }
16
17 void loop() {
18   if (radio.available()) {
19     const char text[] = radio.read();
20     Serial.println(text);
21   }
22 }

```

**Programa Receptor**

**Mensagem comprovando a conexão dos rádios.**

Figura 3.2 – Primeira comunicação entre os rádios.

Observa-se que o transmissor entrega um pacote contendo a mensagem "Hello World" em um intervalo de um segundo.

Concluído a parte inicial de coletar informações a respeito do funcionamento do rádio e como ele se comporta, iniciou-se o desenvolvimento do código sem qualquer auxílio de bibliotecas externas. Logo, o passo inaugural é efetivar a configuração da interface SPI. Esta etapa foi feita tendo em mente as instruções e os ensinamentos de (ZELENOVSKY RICARDO E MENDONÇA, 2019), bem como as conexões utilizadas entre o Nano, o rádio e a memória *Flash* - vide Figura 3.3.

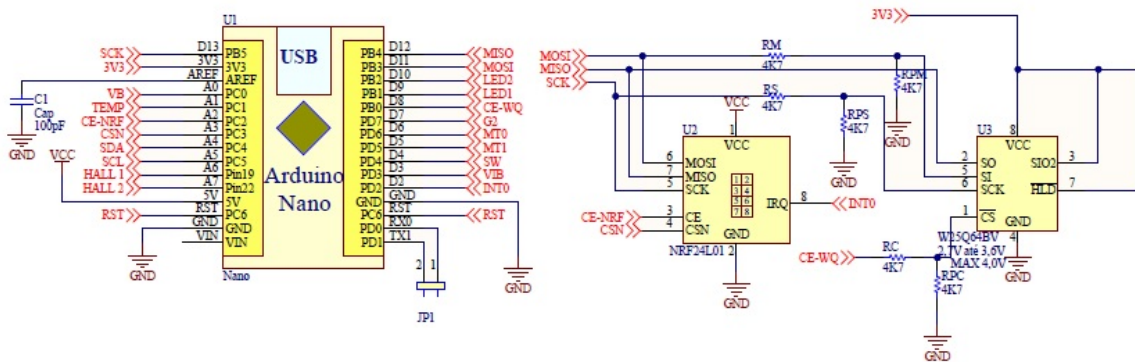


Figura 3.3 – Pinagem e conexões do rádio.

Uma vez finalizada a configuração da interface SPI, a etapa seguinte foi configurar os registradores do rádio. As tabelas abaixo elucidam seus valores.

Tabela 3.1 – Configuração rádio receptor.

Registadores	Valores
STATUS	0x0E (RX_DR=0 TX_DS=0 MAX_RT=0 RX_P_NO=7 TX_FULL=0)
CONFIG	0x0F
EN_AA	0x3F
EN_RXADDR	0x01
SETUP_AW	0x03
SETUP_RETR	0x5F
RF_CH	0x4C
RF_SETUP	0x01
OBSERVE_TX	0x00
RPD	0x00
RX_ADDR_P0	0x31 0x32 0x33 0x34 0x35
RX_ADDR_P1-5	0xC2 0xC3 0xC4 0xC5 0xC6
TX_ADDR	0xE7
RX_PW_P0-6	0x20 0x00 0x00 0x00 0x00 0x00
FIFO_STATUS	0x11
DYNPD	0x00
FEATURE	0x00

A tabela 3.1 apresenta a configuração utilizada por um rádio receptor, o qual utiliza o pipe 0 para receber os pacotes. Estes, por sua vez, devem ter o tamanho de 32 bytes - conforme configurado pelo registrador RX\_PW\_P0). A velocidade de recepção é de 1Mbps (configurado pelo registrador RF\_SETUP) e, tendo em mente que este módulo é habilitado para operar na faixa de frequência de 2400 GHz até 2525 GHz, a frequência para o canal escolhido é de 2476 GHz, considerando que a seguinte fórmula, onde F corresponde à frequência desejada:

$$F = 2400 + RF\_CH[MHz].$$

Ainda, foi configurado autorreconhecimento em todos os pipes, por intermédio da atribuição no registrador EN\_AA. Também, o endereço configurado no pipe 0 foi "12345", visualizado no valor do registrador RX\_ADDR\_P0.

Tabela 3.2 – Configuração rádio transmissor.

Registadores	Valores
STATUS	0x0E (RX_DR=0 TX_DS=0 MAX_RT=0 RX_P_NO=7 TX_FULL=0)
CONFIG	0x0E
EN_AA	0x3F
EN_RXADDR	0x01
SETUP_AW	0x03
SETUP_RETR	0x5F
RF_CH	0x4C
RF_SETUP	0x01
OBSERVE_TX	0x0F
RPD	0x00
RX_ADDR_P0	0x31 0x32 0x33 0x34 0x35
RX_ADDR_P1-5	0xC2 0xC3 0xC4 0xC5 0xC6
TX_ADDR	0x31 0x32 0x33 0x34 0x35
RX_PW_P0-6	0x20 0x00 0x00 0x00 0x00 0x00
FIFO_STATUS	0x11
DYNPD	0x00
FEATURE	0x00

A tabela 3.2 apresenta as alterações necessárias dos valores dos registradores para o perfil

de transmissão de pacotes. Observe que ambas as configurações possuem o mesmo valor de RF\_CH. Isso é necessário para efetivar a comunicação entre os rádios. Ademais, verifica-se que no transmissor os valores de RX\_ADDR\_P0 e TX\_ADDR, em virtude da habilitação do autorreconhecimento fornecido pela funcionalidade de *Enhanced Shockburst*. Claro, não é coincidência que TX\_ADDR e RX\_ADDR\_P0 do transmissor é igual ao RX\_ADDR\_P0 do receptor, uma vez que esses endereços devem ser iguais para consumir a comunicação entre eles. Também, foi configurado para que houvesse até cinco retransmissões na eventualidade de falha da recepção do pacote de autorreconhecimento advindo do receptor primário. Adicionalmente, o intervalo entre o fim de uma transmissão e o início da próxima foi de 4000 microssegundos. Ambas configurações são feitas mediante o registrador SETUP\_RETR.

### 3.2 RELÓGIO E SISTEMA LIGA/DESLIGA

Com o intuito de solucionar os problemas observados na versão anterior do densímetro, os quais dizem respeito à duração da bateria, foram realizados ensaios para verificar o comportamento e eficiência da solução proposta.

Primeiramente, mostrou-se necessário escolher qual transistor seria utilizado no circuito para cortar a alimentação (MOSFET canal P). Para realizar o teste do comportamento do nível de tensão foram selecionados os modelos:

- TP0604.
- TP0606.
- IRF9530.

Os três modelos listados acima foram selecionados dado a disponibilidade no fornecedor local e o preço de cada um. Foram levantadas as curvas para os três transistores, com três diferentes cargas: 3K3, 300 e 100 Ohms. Essas curvas estão apresentadas na Figura 3.4.

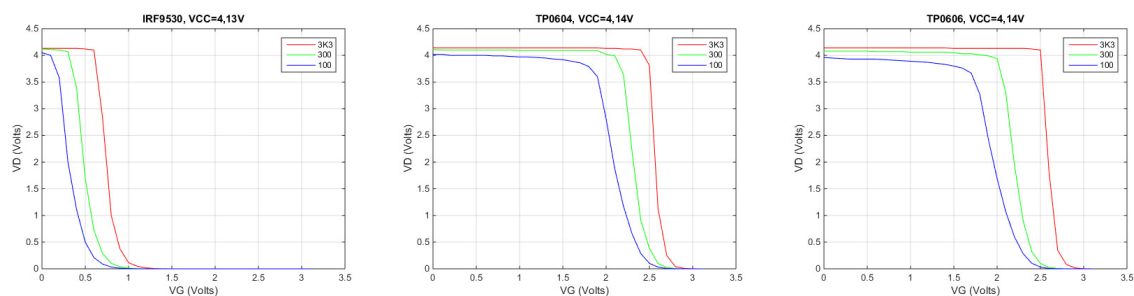


Figura 3.4 – Curvas de tensão para cada transistor.



O transistor IRF9530 seria o mais adequado para o projeto. Porém, só foi possível comprá-lo com encapsulamento TO-220, o que era muito grande para o densímetro. Assim, a decisão ficou entre os transistores TP0604 e TP0606. A curva azul da Figura 3.4 do TP0604 demonstra que uma tensão (VG) acima de 1,7 V apresenta uma diminuição da condução, ao passo que acima de 2,5 V ocorre o corte. Paralelamente, a curva azul do TP0606 ilustra uma boa condução quando a tensão VG está abaixo de 1,7 V e acima de 2,5 V verifica-se o corte. Por fim, a curva azul do IRF9530 certifica que até 0,2 V tem-se uma boa condução, enquanto que uma tensão acima de 1 V resulta no corte. Este transistor foi descartado devido as suas dimensões. Apesar das curvas do TP0604 e TP0606 serem similares, nota-se que a condução no último ocorre ligeiramente mais cedo que o do primeiro (tensão um pouco mais baixa). Desse modo, o transistor modelo TP0604 foi selecionado para os propósitos deste trabalho.

Tendo definido qual transistor utilizar no trabalho, o circuito da Figura 3.5 foi projetado para realizar os testes iniciais.

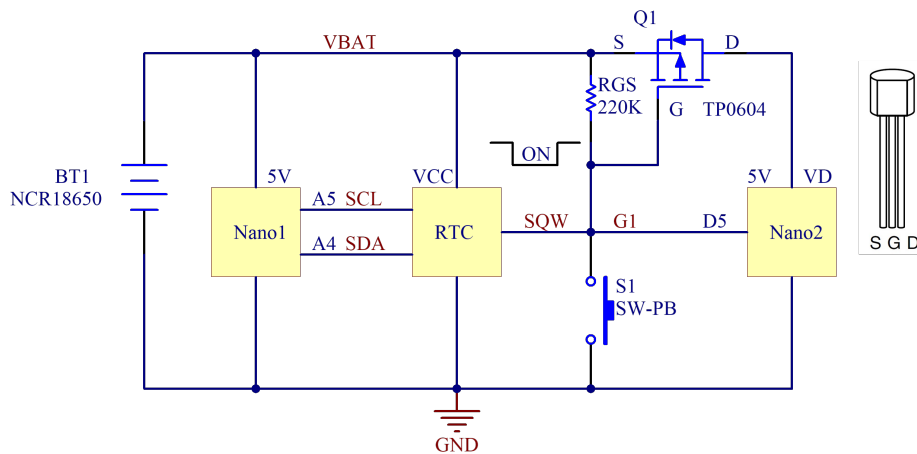


Figura 3.5 – Primeira tentativa de controle da alimentação.

A imagem acima contém a bateria (BT1), dois Arduinos Nano, um relógio RTC, bem como o transistor eleito para o trabalho. Este foi o primeiro ensaio, por isso, se fez uso de dois arduinos. Um só para controlar o RTC e outro para fazer o papel do arduino do densímetro. O Nano1 realiza o controle do RTC com o objetivo de colocar o seu pino SQW para nível baixo, resultando na ativação da alimentação do Nano2. Este, por sua vez, quando acordar conduz o seu pino D5 para baixa tensão e mantem o seu funcionamento por alguns segundos até voltar a desligar. Observe que ao fazer isso o GATE do TP0604 vai para 0, ligando-o e permitindo o fluxo de corrente.

No Nano1, que controla o RTC, ativa de forma artificial o pino de alarme (SQW) do RTC, ou seja, permitindo a simulação do alarme a qualquer momento. Quando isso acontece, uma correspondência entre os registradores que realizam a cronometragem e qualquer um dos registros de alarme ativa o pino SQW - apenas se o alarme estiver habilitado. Logo,

programa-se um alarme para disparar a cada segundo. Após o primeiro seg, o bit A1F vai para 1 e assim permanece se não for apagado. Com o bit A1IE, o Nano 1 consegue controlar o estado da saída SQW = LOW/HIGH.

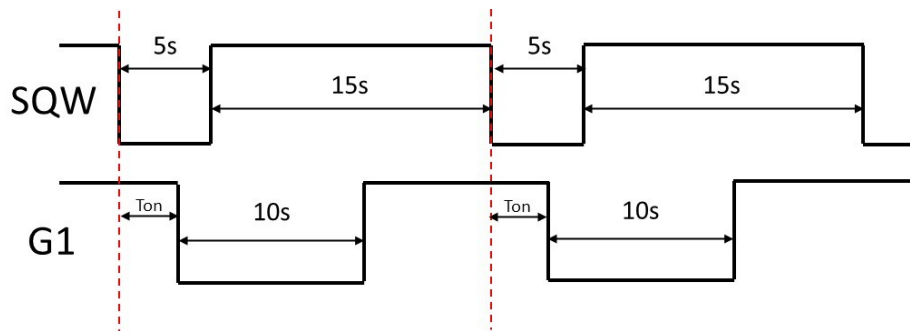


Figura 3.6 – Comportamento do pino SQW e de G1.

Pela Figura 3.6 é possível observar o funcionamento do circuito, onde o Nano 1 realiza o controle sobre o pino SQw, atribuindo-o o nível lógico 0 por 5 segundos. Este tempo é suficiente para que o Nano 2 consiga ligar e fazer o pino D5 - consequentemente o G1 - ir para 0. Nano 1 mantém G1 em 0 por 10 segundos e, em seguida, atribui G1 para 1, o que resulta no corte da energia. Em seguida, reinicia-se o ciclo.

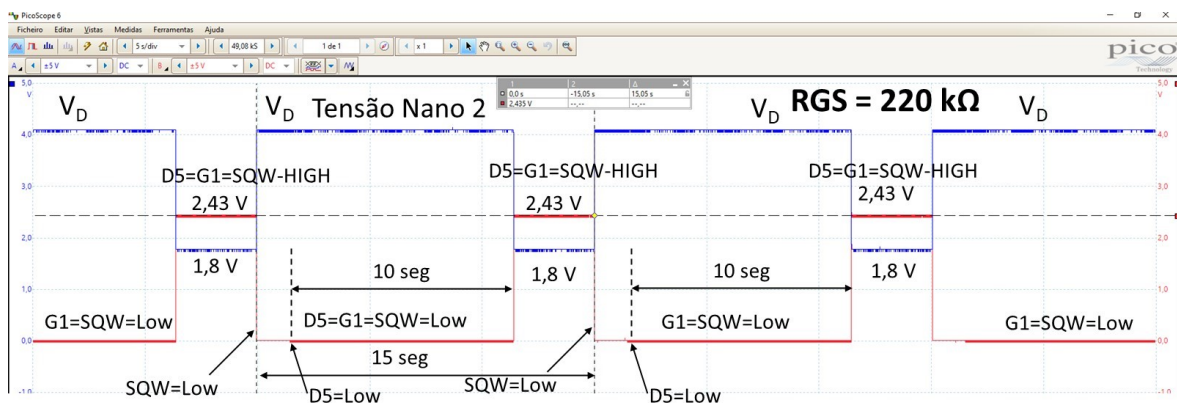


Figura 3.7 – Controle da alimentação.

A Figura 3.7 apresenta a imagem de um osciloscópio com os detalhes importante do circuito da Figura 3.5 em funcionamento. A curva azul representa o pino SQW, ao passo que a curva vermelha ilustra o comportamento de G1. Quando desligado, a tensão em G1 é de 2,43 V, permitindo a condução por Q1 (transistor). Dessa forma, ainda é possível observar 1,8 V em SQW, mesmo estando desligado. Ainda, observa-se que o Nano necessita de um intervalo equivalente a 1,5 segundos para que o programa do usuário assuma controle da CPU.

A hipótese levantada para justificar a tensão elevada em G1 quando D5 = SQW = HIGH sugere que quando a alimentação está cortada, a impedância interna do pino do Arduino deve

estar alta, resultando em um divisor resistivo com RGS (220 kΩ), ocasionando uma tensão de 2,43 V em G1. É válido lembrar que, pela Figura 3.4, o transistor TP0604 apresenta condução quando VGS é igual a 2,43 V.

No entanto, essa suposição de um divisor resistivo foi descartada assim que o resistor RGS foi trocado por um equivalente a esse divisor (15 kΩ). As medidas obtidas não resultaram no comportamento esperado.

Assim, houve a adição de um diodo - modelo 1N4148 - em série com D5, com o objetivo de elevar a tensão em G1. Logo, 0,7V somado aos 2,43 V deveria resultar em 3,1 V. Assim, de acordo com a Figura 3.4, haveria um corte.

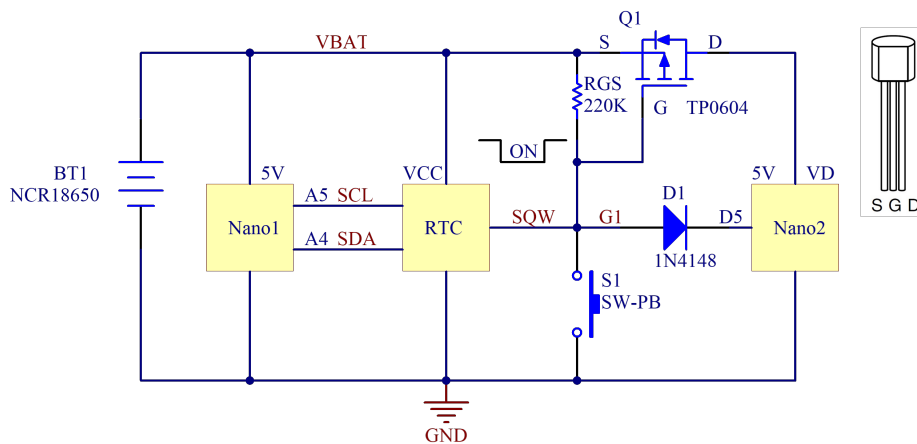


Figura 3.8 – Diodo D1 em série com D5.

A Figura 3.8 apresenta o resultado do circuito com o diodo adicionado.

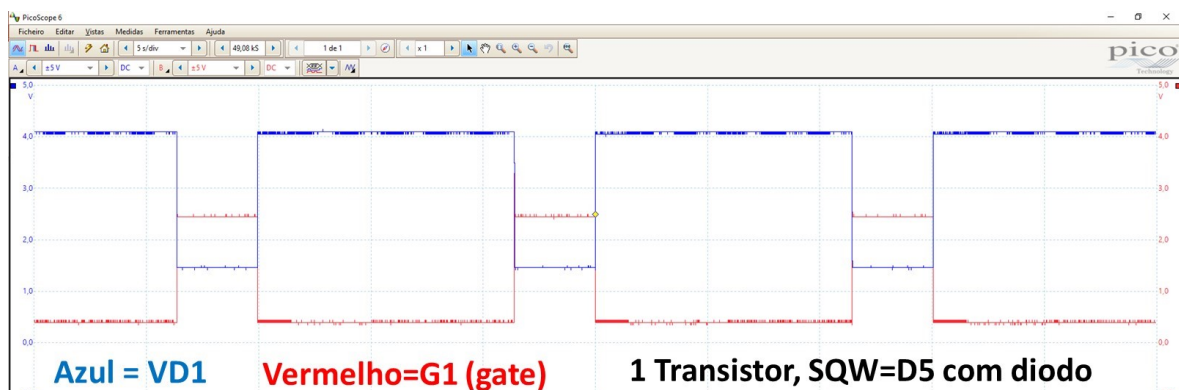


Figura 3.9 – Medidas obtidas com o diodo.

Como pode ser visto na Figura 3.9, ocorreu uma pequena melhora, mas ainda não foi possível cortar completamente a alimentação.

Outra hipótese levantada sugere que os resultados parecem indicar que o transistor de

saída do RTC poderia manter essa tensão no gate do transistor Q1. Adicionalmente, foram feitas as seguintes tentativas:

- Colocar D5 e SQW em curto e ligá-los ao GATE (G1) por um diodo.
- Ligar D5 ao GATE (G1) com dois diodos em série.

Ambas alternativas propostas, listadas acima, mostraram-se infrutíferas.

Para solucionar esse problema, fez necessário o uso de um segundo transistor. A utilização de dois transistores, um de canal P (TP0604) e outro de canal N (BS170). Este componente foi integrado ao projeto devido a sua disponibilidade. O projeto do circuito está na Figura 3.10.

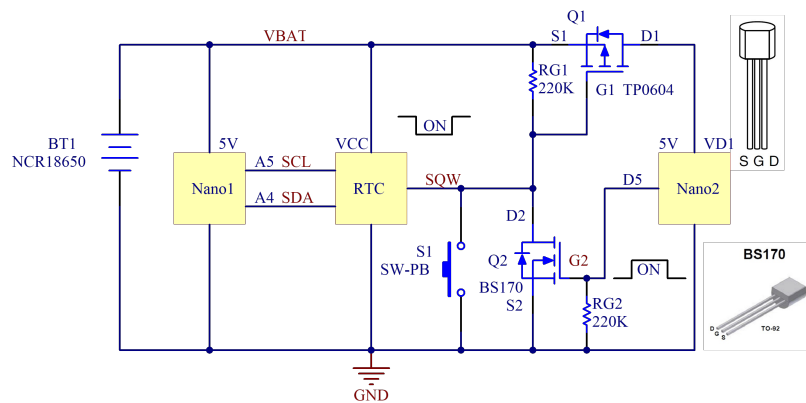


Figura 3.10 – Circuito com dois transistores.

O circuito acima mostrou melhores resultados, ocasionando o desejado  $VD = 0V$  quando a alimentação é interrompida.

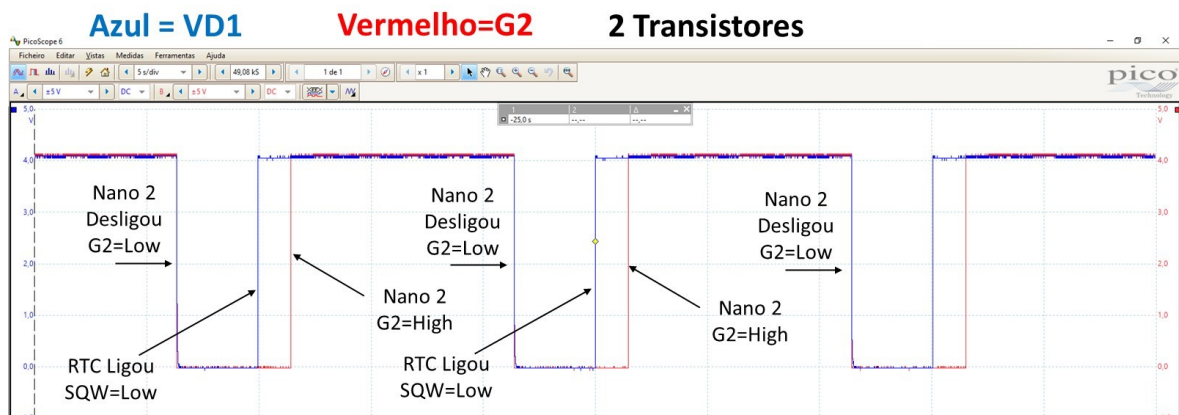


Figura 3.11 – Medidas obtidas com G2 em vermelho.

Na Figura 3.11, a linha azul corresponde a alimentação do Nano 2, ou seja, é a tensão em VD1. A subida desta linha indica o instante em que o RTC, comandado pelo Nano 1 aterrou

o GATE G1 (SQW = LOW) e ligou a alimentação, pois Q1 entrou em condução. Neste ponto, o transistor Q2 está cortado. Depois de 1,5 segundos, o Nano 2 começa a executar o programa e atribui D5 = G2 = HIGH e, assim, liga o transistor Q2, o qual passa a garantir nível baixo no GATE G1. A partir deste ponto o RTC pode colocar SQW = HIGH pois Q2 garante que Q1 fique ligado. Para desligar a alimentação, o Nano 2 faz D5 = G2 = LOW, resultando no desligamento de Q1.

Ainda, observa-se pela Figura 3.11 que quando a alimentação é desligada, a tensão VD1 realmente transita para zero. Logo, a alimentação é realmente desligada.

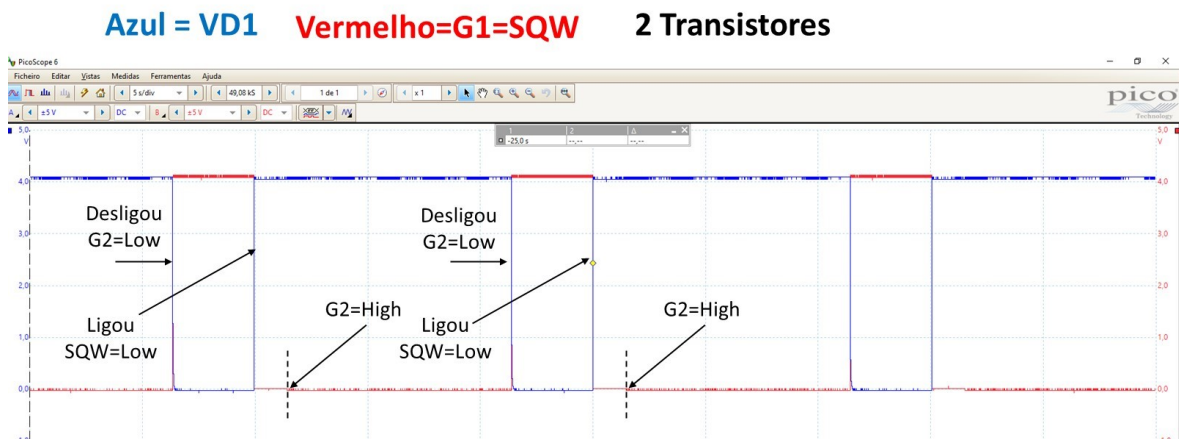


Figura 3.12 – Medidas obtidas com G1 em vermelho.

Paralelamente, na Figura 3.12 a linha azul indica a alimentação e a linha vermelha a tensão no GATE de Q1, ou seja, G1 = SQW. Constatou-se que se G1 = SQW = LOW, a alimentação é ligada. Ainda, é possível ver o instante em que o Nano 2 faz G2 = HIGH, garantindo a alimentação.

É válido citar que o botão (S1) para ligar também é capaz de puxar G1 para zero. Ainda, assim que é energizado, o Nano 2 deve fazer D5 = HIGH para assim garantir a continuidade da alimentação. Se D5 = HIGH, o transistor Q2 conduz e aterra G1.

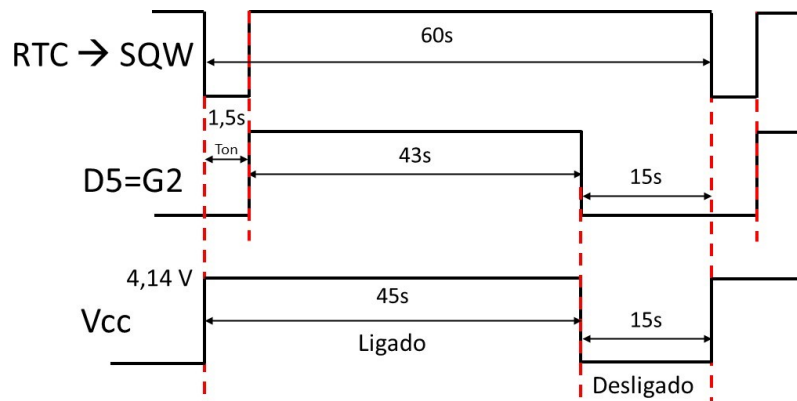


Figura 3.13 – Cronograma do sistema Liga/Desliga usado para testar o circuito até a exaustão da bateria.

Para fazer o teste definitivo, o RTC foi programado para despertar a cada 60 segundos. A Figura 3.13 ilustra a temporização. Ao despertar, depois de aproximadamente 1,5 segundos, o programa do Nano faz  $D5 = G2 = HIGH$  e passa a garantir a alimentação. Em seguida, comanda o RTC para apagar o alarme. Ainda, Permanece ligado por mais 43 segundos e se desliga fazendo  $D5 = G2 = LOW$ . Depois de aproximadamente mais 15 segundos, o alarme do RTC dispara e tudo volta a repetir. Assim, é possível obter 45 segundos ligado e 15 desligados, aproximadamente.

Finalmente, pela Figura 2.16 observou-se 9313 ciclos de liga e desliga do sistema. Onde, 75% do tempo ficava ativo. Assim, considerando que o densímetro ficará ligado por 1 minuto a cada hora dentro do barril de fermentação, é possível obter, aproximadamente, 291 dias de vida útil da bateria utilizando este circuito limitado contendo o Arduino e o relógio.

### 3.3 ENSAIOS DO LASTRO

Na versão anterior do projeto foi notado uma certa dificuldade em realizar o controle do posicionamento do densímetro em uma inclinação desejada na qual este ficasse em equilíbrio. Foram utilizadas bolinhas de chumbo para este propósito. O peso variou de 20 gramas a 25 gramas e as conclusões retiradas dos ensaios apontavam para a necessidade de um sistema mais robusto e seguro de efetivar esse equilíbrio, uma vez que o manuseio dessas bolinhas revelou-se ser uma tarefa complicada.

A Figura 3.14 elucida as bolinhas de chumbo, acompanhadas de uma esponja para não deixá-las se movimentar ao longo do tubo, o que poderia tornar ainda mais imprevisível o seu comportamento.



Figura 3.14 – Lastro da versão antiga.

Houve, também, a tentativa de amarrar cada bolinha e distribuir este conjunto pela placa do circuito. Porém, ao testar essa configuração em tubos diferentes foi possível verificar inconsistências nos resultados do posicionamento obtidos. Qualquer pequena intervenção podia mudar o centro de gravidade e alterar completamente essa inclinação inicial.

Assim, tendo em mente a necessidade de se adicionar um lastro no densímetro com o objetivo de equilibrá-lo e mantê-lo em uma determinada posição, bem como para alterar o centro de massa do densímetro, foram executados testes para esclarecer questionamentos a respeito do posicionamento e do peso deste item. Desse modo, pesos utilizados para balanceamento de pneus de carro foram selecionados para este propósito. Foram escolhidos os pesos de 5 e 10 gramas devido ao tamanho reduzido deles. É válido lembrar que o lastro tem a finalidade de ser o peso principal, ao passo que a mesa e o motor tem o propósito de realizarem o ajuste fino do posicionamento do corpo do densímetro.

A Figura A.1 ilustra os posicionamentos escolhidos para dar início aos ensaios. Os testes foram conduzidos mantendo o lastro em alguma das 4 posições e movimentando a mesa de uma extremidade a outra do parafuso sem fim, por uma quantidade definida de vezes, e gravando as informações do acelerômetro e giroscópio, bem como os valores dos sensores de efeito Hall com o propósito de plotar gráficos para efetuar o estudo de seu comportamento.

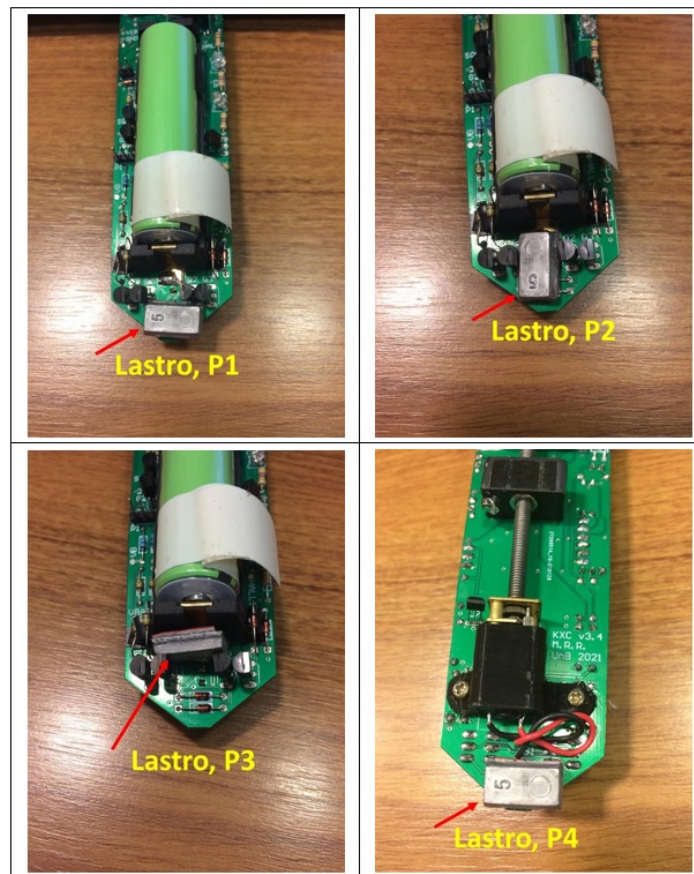


Figura 3.15 – Configurações testados para o lastro.

Para facilitar a explicação e a visualização dos valores obtidos das coordenadas pelo acelerômetro e giroscópio, a Figura 3.16 foi confeccionada.

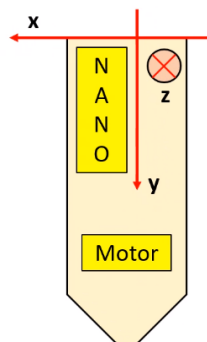


Figura 3.16 – Coordenadas do densímetro.



A Figura 3.17, retirada do trabalho (VASCONCELOS, 2019) elucidada, de forma visual, como esses ângulos podem ser medidos. Na imagem, verifica-se que o densímetro está em um inclinação de 23 graus em relação à perpendicular à linha d'água.

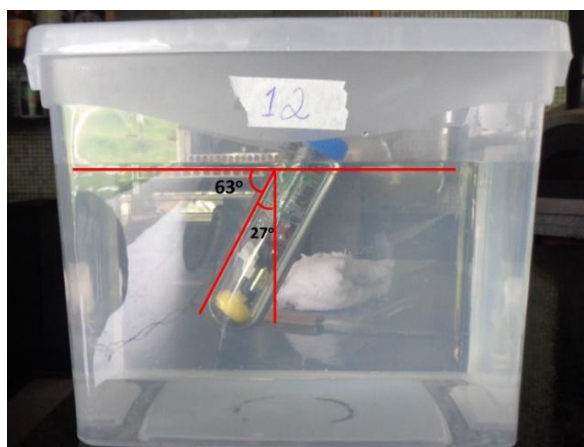


Figura 3.17 – Exemplo de uma posição e os ângulos obtidos.

Foram feitos 12 ensaios, variando o posicionamento do lastro principal e do peso adicionado a mesa. A tabela 3.3 mostra uma dessas configurações com o densímetro na água.

Tabela 3.3 – ENSAIO 6

<b>P1</b>	<b>P2</b>	<b>P3</b>	<b>P4</b>	<b>Mesa</b>
0	5g	0	0	10g

A angulação desejada para a inclinação do densímetro quando mergulhado na água é de 25 a 30 graus. Como é possível verificar no gráfico superior do lado esquerdo, ao movimentar a mesa de uma extremidade a outra, esse ângulo varia dentro desse intervalo a maior parte do tempo. Mais especificamente, quando a mesa move-se em direção ao corpo do motor, localizado na parte inferior da placa, o equipamento apresenta uma diminuição na sua angulação, indicando uma maior verticalidade, ao passo que ao fazer com que a mesa translade na posição oposta, observa-se um aumento deste valor, apresentando uma proximidade com a orientação horizontal. Ademais, verifica-se que essa movimentação de subida e descida foi repetida várias vezes para observar a consistência do comportamento.

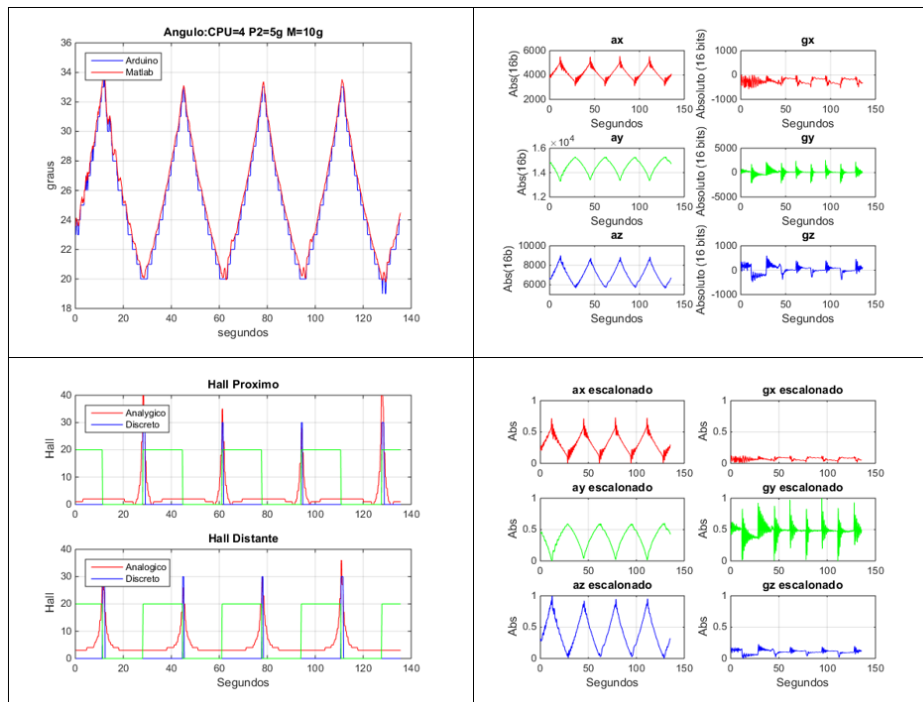


Figura 3.18 – Ensaio 6 do lastro.

Ademais, o gráfico inferior do lado esquerdo fornece informações das medidas dos sensores Hall. O hall próximo corresponde ao sensor mais próximo do corpo do motor, ao passo que o hall distante diz respeito a extremidade oposta. Adicionalmente, a curva verde indica o sentido do motor, quando em nível alto indica que a mesa está subindo, enquanto que em nível baixo a mesa está descendo. Levando em consideração o intervalo de 20 a 40 segundos do gráfico do hall, verifica-se que entre 20 e 30 segundos a mesa está descendo, aproximando-se deste sensor. Na marca de 30 e poucos segundos é possível ver um pico das curvas analógica e discreta, o que indica que o sensor captou o campo magnético, resultando na inversão da rotação do motor. Agora, a mesa se movimenta para cima, em direção ao hall distante. No momento dos 40 segundos, o sensor hall distante captura o campo magnético, assinalando a proximidade da mesa a ele, ocasionando a inversão do motor.

É necessário citar que a curva vermelha demonstra as medidas lidas pelo sensor, enquanto que a azul corresponde aos valores discretos destas. Ainda, foram escolhidos limiares máximos e mínimos para obter as medidas discretas, determinando a quantização dos valores. Com o objetivo de melhorar a visibilidade, os valores discretos foram multiplicados por 30.

Paralelamente, os gráficos posicionados na direita superior indicam as medidas obtidas do acelerômetro (ax, ay e az) e do giroscópio (gx, gy e gz). O primeiro tem a utilidade de expressar a inclinação, ao passo que o segundo mostra se o densímetro realiza movimentos de rotação enquanto a mesa se mexe. Porém, é notório que os gráficos de ambos encontram-se em escalas diferentes, dificultando a apreciação dos dados. Assim, foram plotados os gráficos da direita inferior com as medidas devidamente escalonadas. Para isso, foi subtraído

de todos os eixos o valor mínimo e dos três eixos foi pego o valor máximo, dividindo todos por esta medida máxima. Logo, a máxima aceleração é 1 e a mínima aceleração é 0. É válido citar que os valores dispostos são absolutos, uma vez que o sentido do eixo não é muito relevante nestes resultados. O mesmo processo foi executado para as medidas do giroscópio.

Observando os gráficos escalonados é possível verificar movimento em todos os eixos. O desejado era que o acelerômetro nos eixos x e z não acusasse muita movimentação, uma vez que a mesa desloca-se pelo eixo y. No entanto, quando o densímetro for inserido no mosto, o motor irá situar a mesa em uma posição capaz de manter o aparelho em uma angulação próxima a 25 graus, realizando eventuais ajustes quando essa inclinação estiver errada. Porém, deve ser feita uma verificação para avaliar o quanto isso influencia nas medidas. É fato que, devido à distribuição irregular da sua massa, o densímetro gire à medida que muda sua inclinação. Também, nota-se que ocorre uma agitação, principalmente no eixo y quando muda o sentido de rotação. Isto é esperado pela inércia da mesa, a qual tem um lastro à ela grudado.

Em contraste, o ensaio 9 apresentou resultados consideravelmente piores em relação ao teste citado acima.

Tabela 3.4 – ENSAIO 9

P1	P2	P3	P4	Mesa
0	0	5g	0	5g

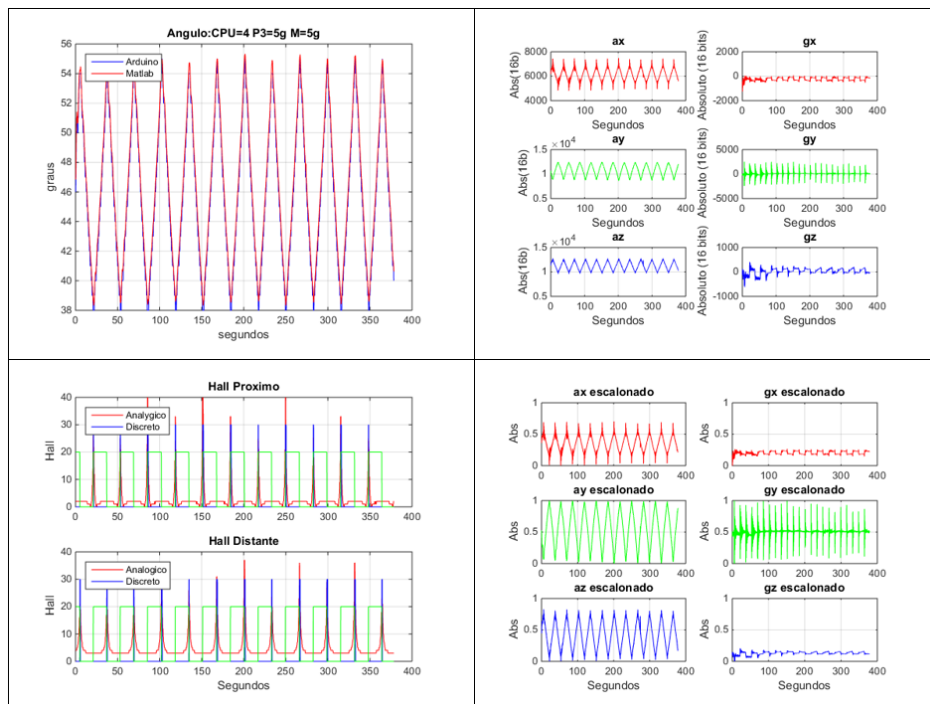


Figura 3.19 – Ensaio 9 do lastro.

Verifica-se que o intervalo de inclinações obtido não foi o esperado, variando de 55 a 38 graus. Também, apesar da agitação não ser o fator mais relevante nestes ensaios, é possível notar uma alta oscilação no eixo y (gráfico gy escalonado) nessa distribuição de pesos.

Finalmente, a intenção final era conseguir uma composição onde o lastro tivesse mais peso que a mesa, dado o espaço limitado entre o topo da mesa e o tubo. No entanto, tal arranjo não apresentou resultados satisfatórios. Adicionalmente, os resultados obtidos desses ensaios demonstra a necessidade de algumas mudanças. Nos próximos teste o motor terá uma rotação mais lenta e irá parar de operar quando a mesa chegar em uma das extremidades - apontado pelo sensor de efeito Hall. Em seguida, o motor deverá inverter a direção de sua rotação e ligar. Porém, Não será possível eliminar por completo a oscilação, visto que ao ligar, o motor irá inevitavelmente gerar essa agitação. Todavia, pode-se criar uma rotina especial, a qual irá aumentando suavemente o PWM até o motor ganhar velocidade plena e seguir adiante.

# 4 CONCLUSÃO

---

O densímetro digital desempenha um papel fundamental no processo de produção de cervejas artesanais. Com ele é possível realizar aferições de densidade informando o estágio da fermentação. Porém, as opções deste tipo de aparelho no mercado, atualmente, não são abundantes. Também, essas alternativas não possuem um bom custo benefício, sendo muito caras ou não contêm as características necessárias para que a fermentação da cerveja seja feita de forma a assegurar a integridade de qualidade do mosto.

Na versão anterior deste projeto, descrita por (VASCONCELOS, 2019), foram realizados numerosos ensaios com o densímetro. Estes auxiliaram no evidenciamento de problemas relacionados ao equipamento. Dentre as dificuldades encontradas estão a duração da bateria, a incapacidade de desligar o sistema inteiro por um intervalo de tempo programável, o posicionamento inconsistente do densímetro quando inserido em um líquido, bem como a dificuldade de manipulação do lastro para efetivar o seu equilíbrio.

À vista disso, uma nova abordagem foi tomada. Na versão atual, os problemas citados anteriormente foram solucionados em parte, visto que mais ensaios são necessários para determinar a qualidade da conduta tomada até então. O sistema de liga e desliga mostra-se promissor e seu funcionamento já foi colocado a prova, assim como o lastro apresenta um comportamento previsível e estável. No entanto, é essencial realizar testes em diferentes líquidos com densidades diferentes e observar o desempenho do densímetro, como também implementar verificações relacionados aos modos "Xereta" e "Fofoca".

Em suma, a iteração desenvolvida apresenta avanços significativos, sempre tendo como cerne, para a tomada de decisões e solução de adversidades, a característica fundamental do projeto desde a sua concepção: custo-benefício.

## REFERÊNCIAS BIBLIOGRÁFICAS

---

- BEAL, P. V. R. Densímetro para monitorar produção artesanal de cerveja: coleta e análise de dados. 2021.
- EQUIPMENT, B. B. *TILT Wireless Hydrometer and Thermometer*. [S.l.]: Disponível em: <<https://tilthydrometer.com/>>. Acesso em: maio, 2021.
- GOMES, M. J. d. S. et al. Acompanhamento da densidade do mosto durante a fermentação de cervejas produzidas em uma cervejaria artesanal. Universidade Federal Rural do Semi-Árido, 2018.
- INC., S. E. *Linear Hall Effect Sensor V 2.00*. [S.l.]: SEC ELECTRONICS INC. Disponível em: <<http://www.secsemi.com/products/Linear-Hall-ICs/SS49E.html>>, 2008.
- ISPINDEL. [S.l.]: Disponível em: <<https://github.com/universam1/iSpindel>>. Acesso em: maio, 2021.
- MAXIM, D. S. *Extremely Accurate I2C-Integrated RTC/TCXO/Crystal*. [S.l.]: Dallas Semiconductor Maxim. Disponível em: <<https://www.maximintegrated.com/en/products/analog/real-time-clocks/DS3231.html>>, 2010.
- NANO, A. *Arduino Nano*. [S.l.]: Obtenido de Arduino: <<https://store.arduino.cc/usa/arduino-nano>>. Acesso em: maio, 2021.
- OLIVEIRA, B. d. M.; FILHO, J. M. M.; AFONSO, J. C. A densidade e a evolução do densímetro. *Revista Brasileira de Ensino de Física*, SciELO Brasil, v. 35, n. 1, p. 1–10, 2013.
- PIMENTA, L. B. et al. A história e o processo da produção da cerveja: uma revisão. *Cadernos de Ciência & Tecnologia*, v. 37, n. 3, p. 26715, 2020.
- PINTARELLI, G. B. Desenvolvimento de um densímetro eletrônico para produção de cerveja. 2019.
- ROSALIN, J. P. A trajetória da cerveja no brasil: uma proposta de aproximação com a teoria da sucessão dos meios geográficos. *GEOGRAFIA (Londrina)*, v. 30, n. 1, p. 149–165, 2021.
- SEMICONDUCTOR, N. *nRF24L01+ Single Chip 2.4GHz Transceiver Product Specification v1.0*. [S.l.]: Nordic Semiconductor. Disponível em: <<https://www.nordicsemi.com/Products/Low-power-short-range-wireless/nRF24-series>>, 2008.
- TECHNOLOGY, H. *G12-N20 Geared Mini DC Motor*. [S.l.]: HANDSON TECHNOLOGY. Disponível em: <[shorturl.at/bmoDW](http://shorturl.at/bmoDW)>, c2008.
- VASCONCELOS, B. B. de A. F. Projeto de densímetro digital para a produção artesanal de cerveja. 2019.
- ZELENOVSKY RICARDO E MENDONÇA, A. *Arduino - Guia Avançado Para Projetos*. [S.l.]: Interciência; NOVA EDIÇÃO<sup>a</sup> (11 julho 2019), 2019.

# APÊNDICE

# A APÊNDICE - ENSAIOS PARA AFERIÇÃO DOS LASTROS.

Os ensaios seguintes foram realizados para determinar o posicionamento, bem como o peso do lastro a ser utilizado no densímetro. Testes feitos no trabalho anterior (VASCONCELOS, 2019) indicaram que uma boa disposição dos lastros é a que deixa o densímetro, quando imerso em água pura, com uma inclinação entre 20 e 30 graus.

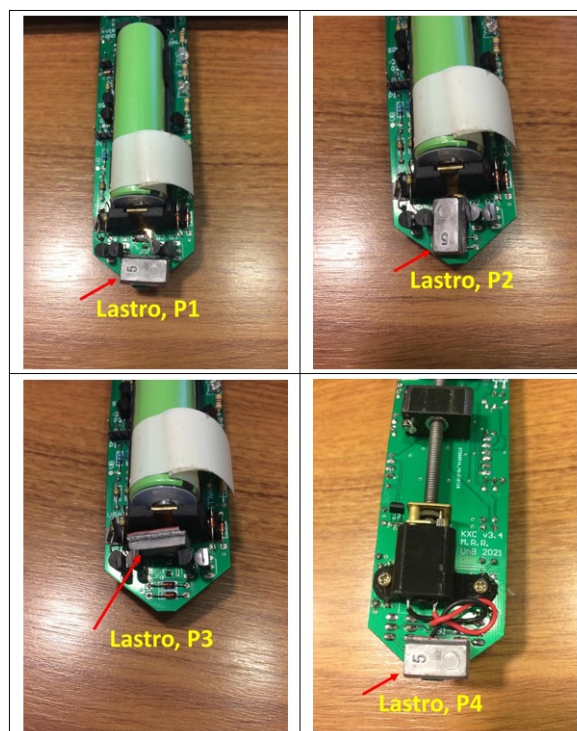


Figura A.1 – Configurações testados para o lastro.

São dois os lastros usados.

- Lastro 1: Colado na placa para dar uma certa inclinação ao densímetro, temos 4 posições mostradas na Figura A.1.
- Lastro 2: Colado sobre a mesa, a qual ao movimentar vai fazer a sintonia fina da inclinação.

Foram feitos vários ensaios com diferentes valores de Lastro 1 e Lastro 2 para verificar a combinação que melhor favorece-se o posicionamento no ângulo desejado (entre 20 e 30 graus).



Tabela A.1 – ENSAIO 1

P1	P2	P3	P4	Mesa
0	0	0	0	0

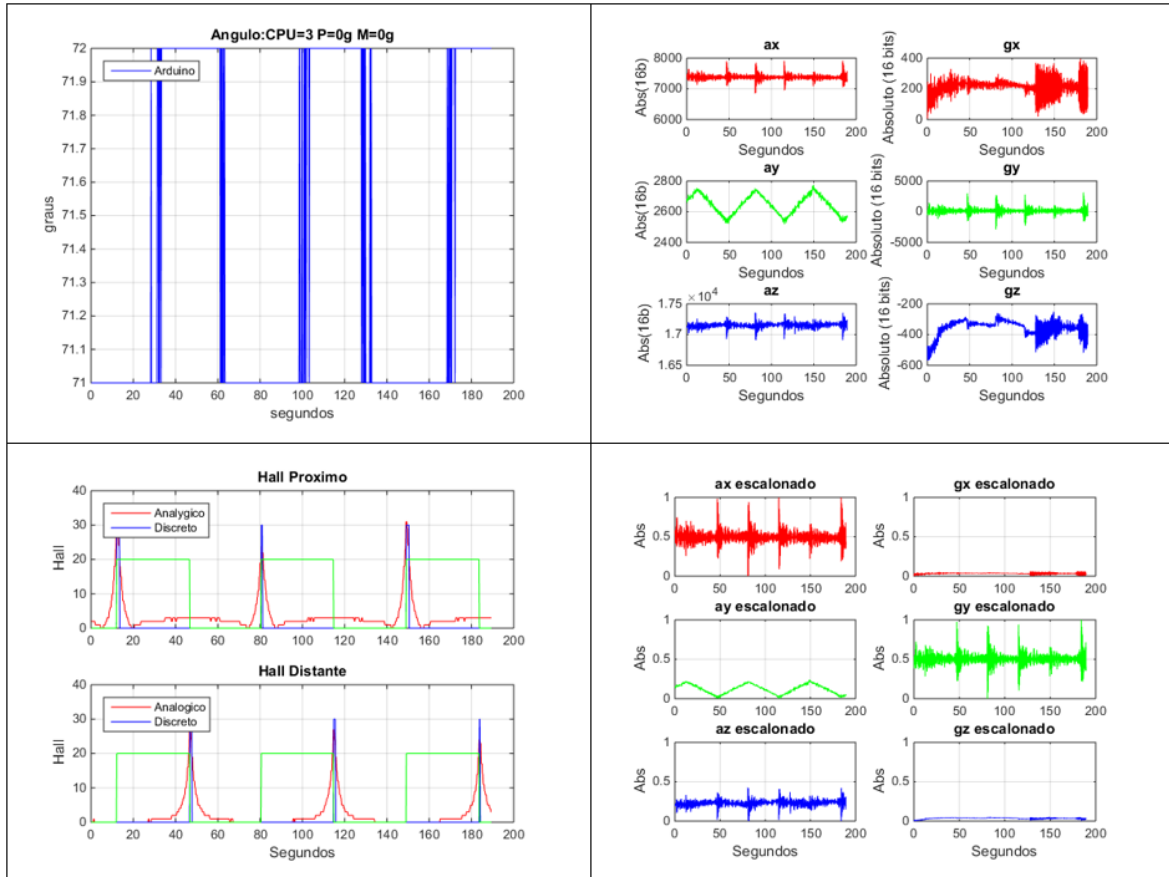


Figura A.2 – Ensaio 1 do lastro.

Tabela A.2 – ENSAIO 2

P1	P2	P3	P4	Mesa
0	0	0	0	5g

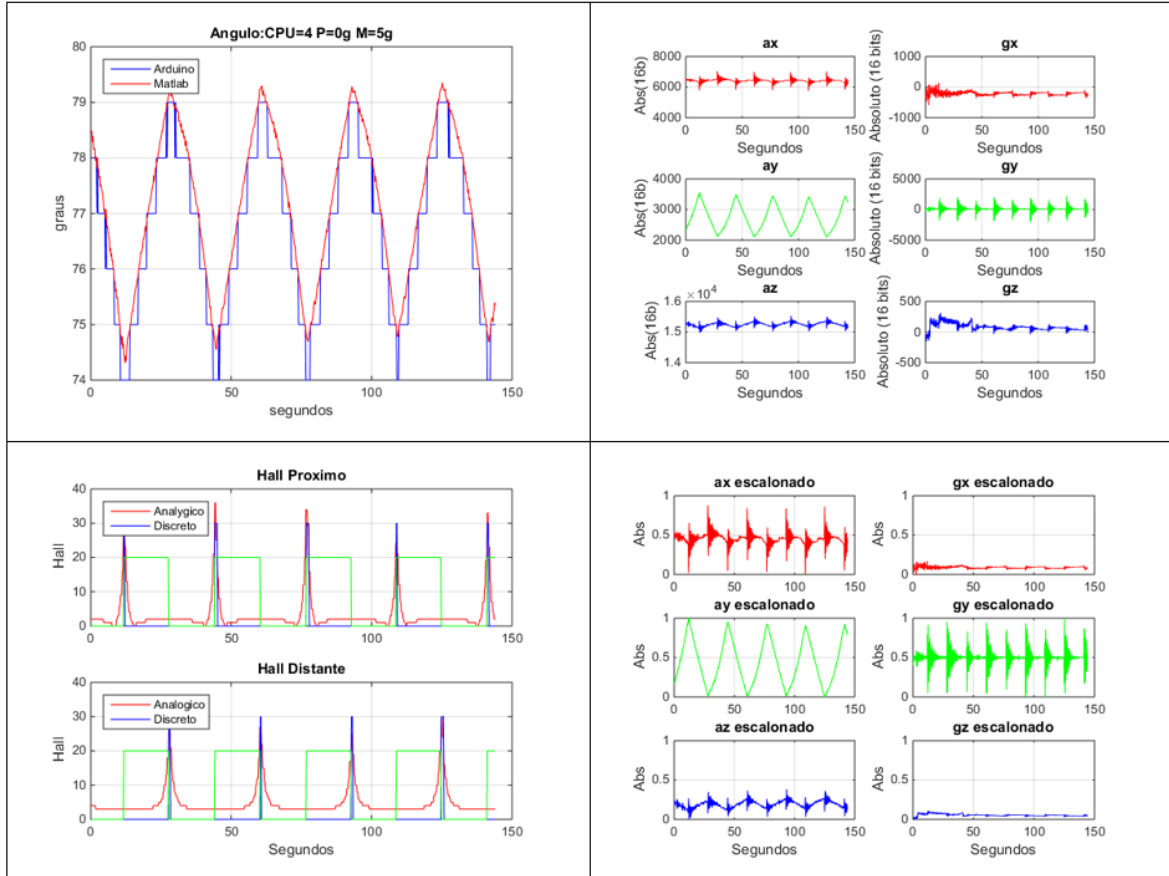


Figura A.3 – Ensaio 2 do lastro.

Tabela A.3 – ENSAIO 3

P1	P2	P3	P4	Mesa
0	0	0	0	10g

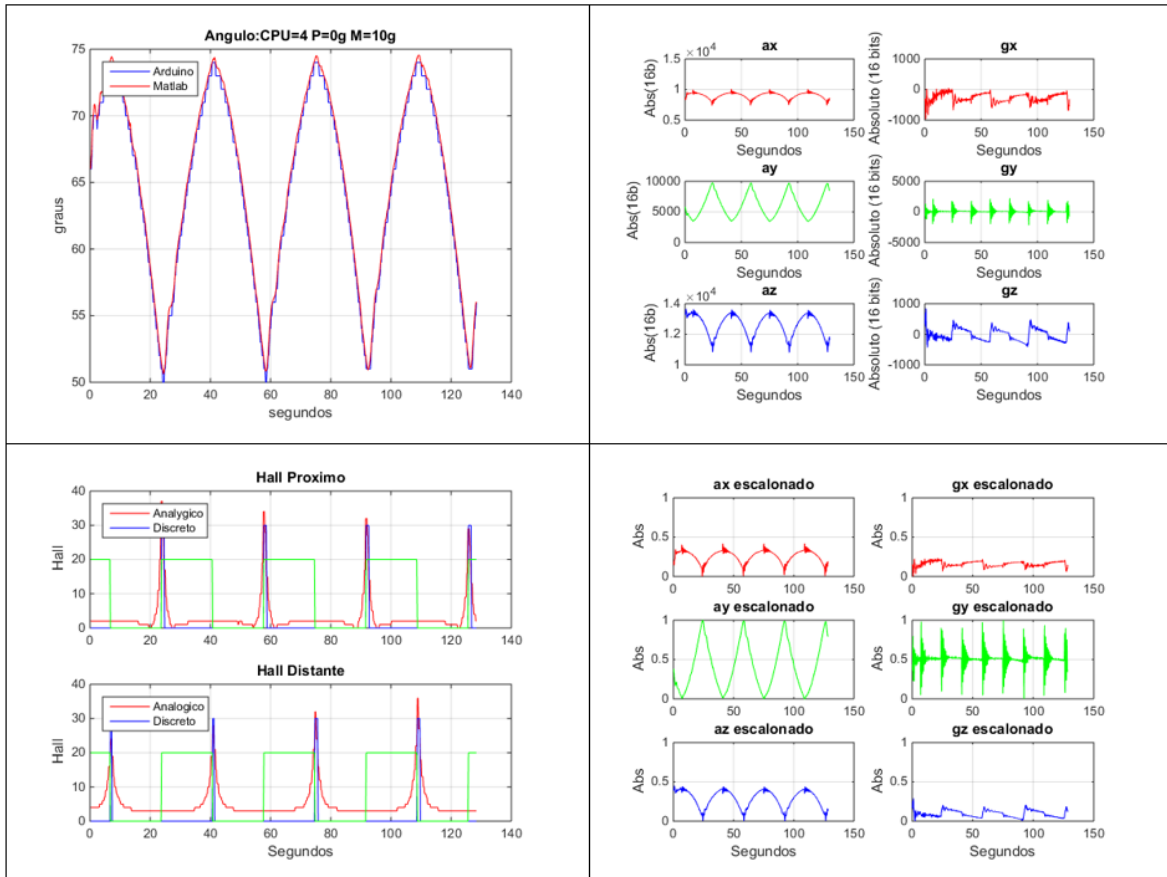


Figura A.4 – Ensaio 3 do lastro.

Tabela A.4 – ENSAIO 4

P1	P2	P3	P4	Mesa
5g	0	0	0	5g

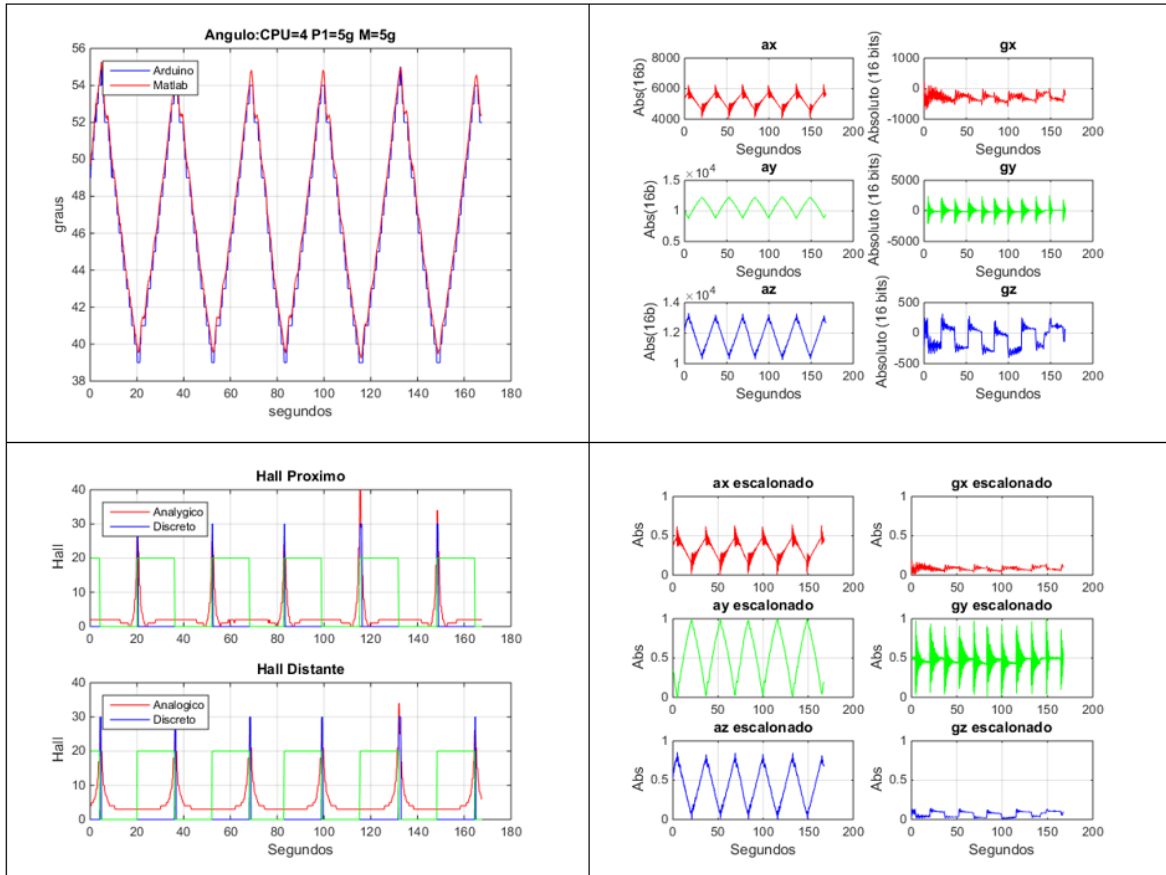


Figura A.5 – Ensaio 4 do lastro.

Tabela A.5 – ENSAIO 5

P1	P2	P3	P4	Mesa
0	5g	0	0	5g

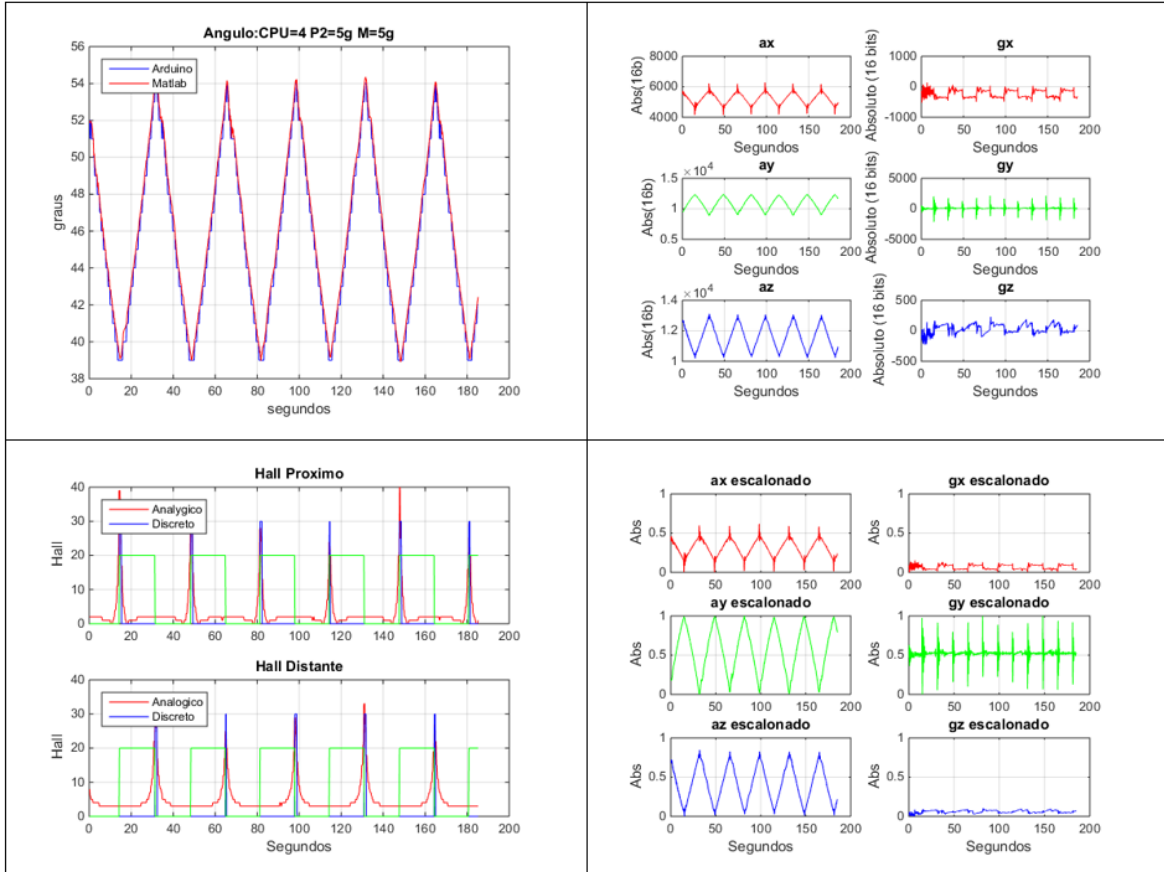


Figura A.6 – Ensaio 5 do lastro.

Tabela A.6 – ENSAIO 6

P1	P2	P3	P4	Mesa
0	5g	0	0	10g

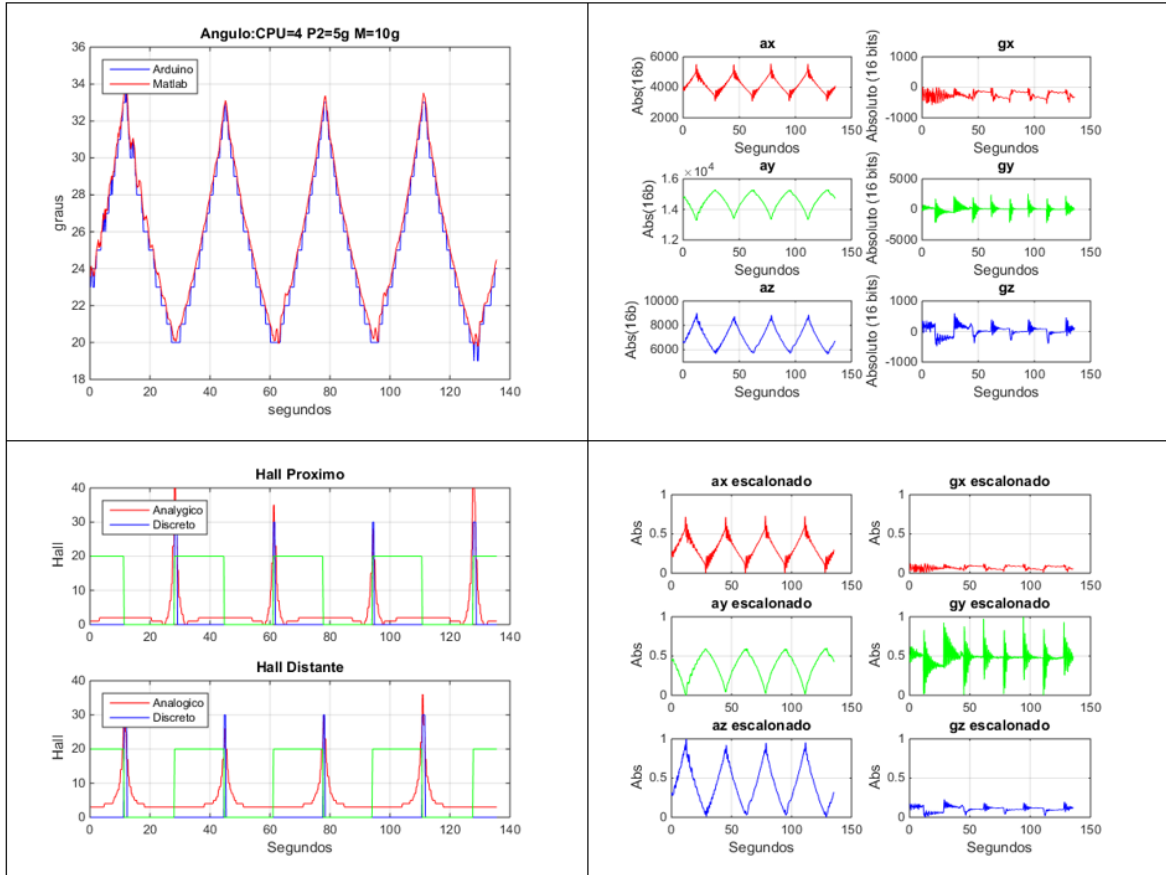


Figura A.7 – Ensaio 6 do lastro.

Tabela A.7 – ENSAIO 7

P1	P2	P3	P4	Mesa
0	0	2,5g	0	2,5g

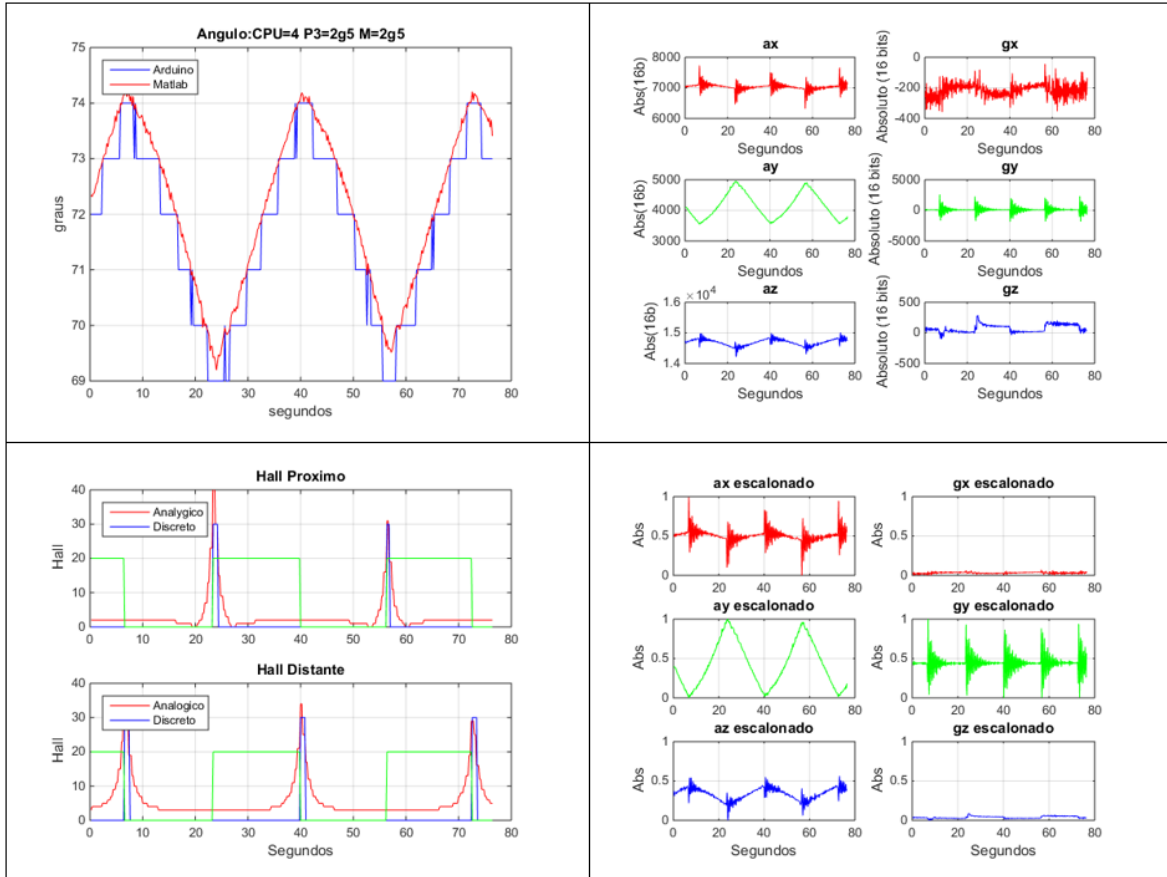


Figura A.8 – Ensaio 7 do lastro.

Tabela A.8 – ENSAIO 8

P1	P2	P3	P4	Mesa
0	0	5g	0	2,5g

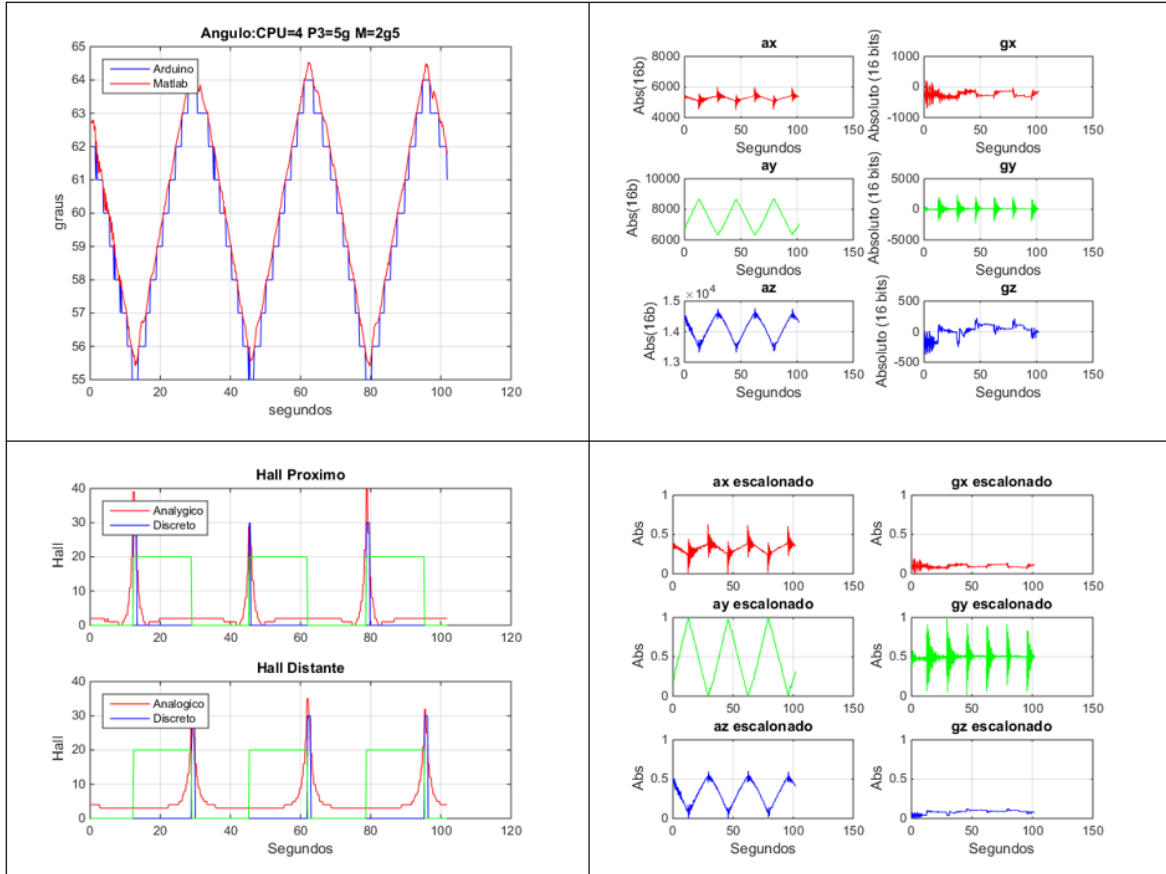


Figura A.9 – Ensaio 8 do lastro.



Tabela A.9 – ENSAIO 9

P1	P2	P3	P4	Mesa
0	0	5g	0	5g

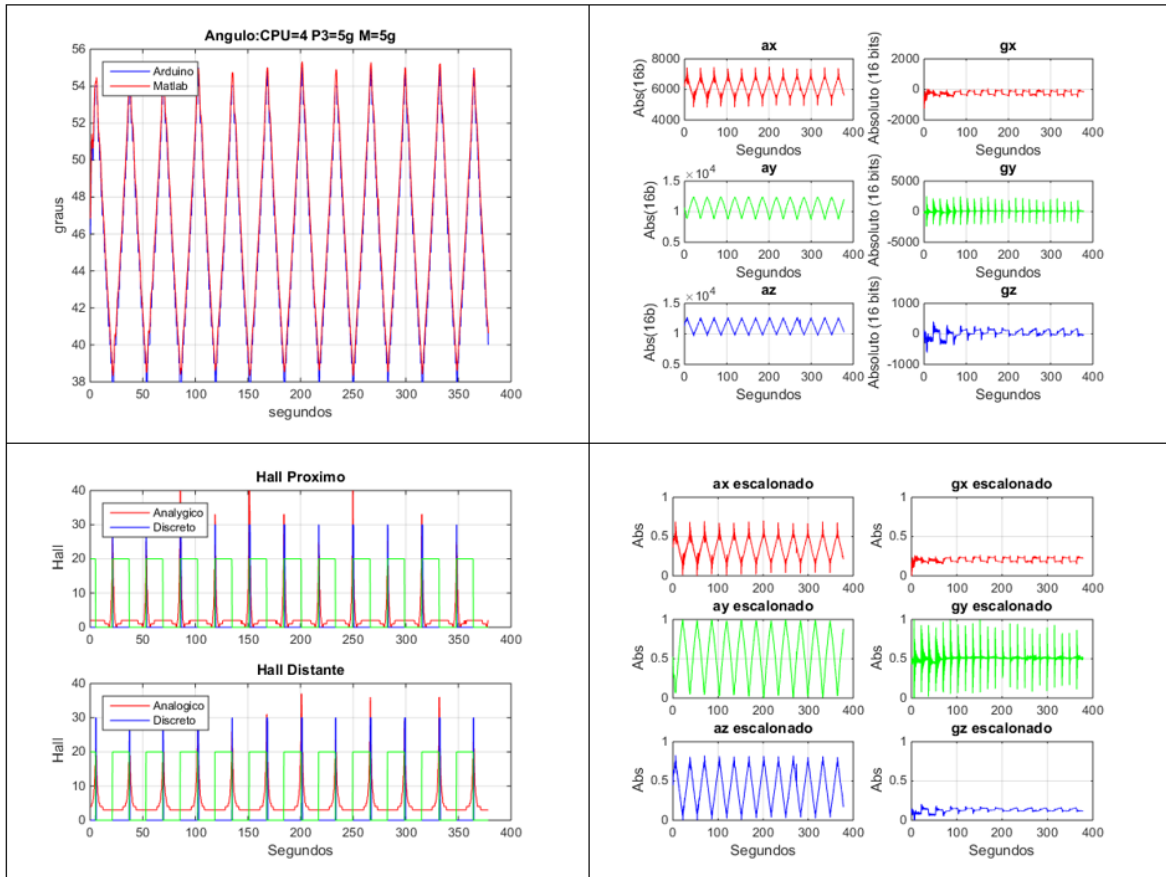


Figura A.10 – Ensaio 9 do lastro.

Tabela A.10 – ENSAIO 10

P1	P2	P3	P4	Mesa
0	0	7,5g	0	2,5g

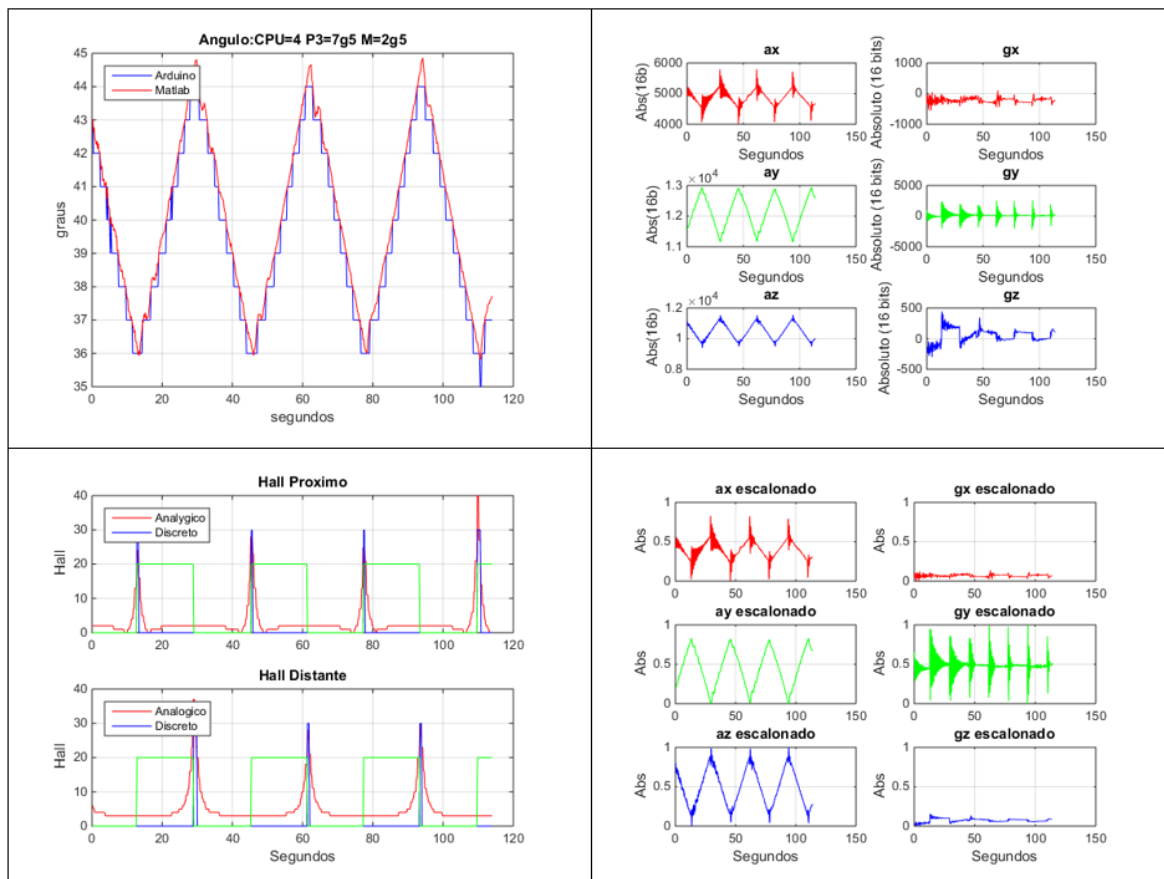


Figura A.11 – Ensaio 10 do lastro.

Tabela A.11 – ENSAIO 11

P1	P2	P3	P4	Mesa
0	0	0	5g	10g

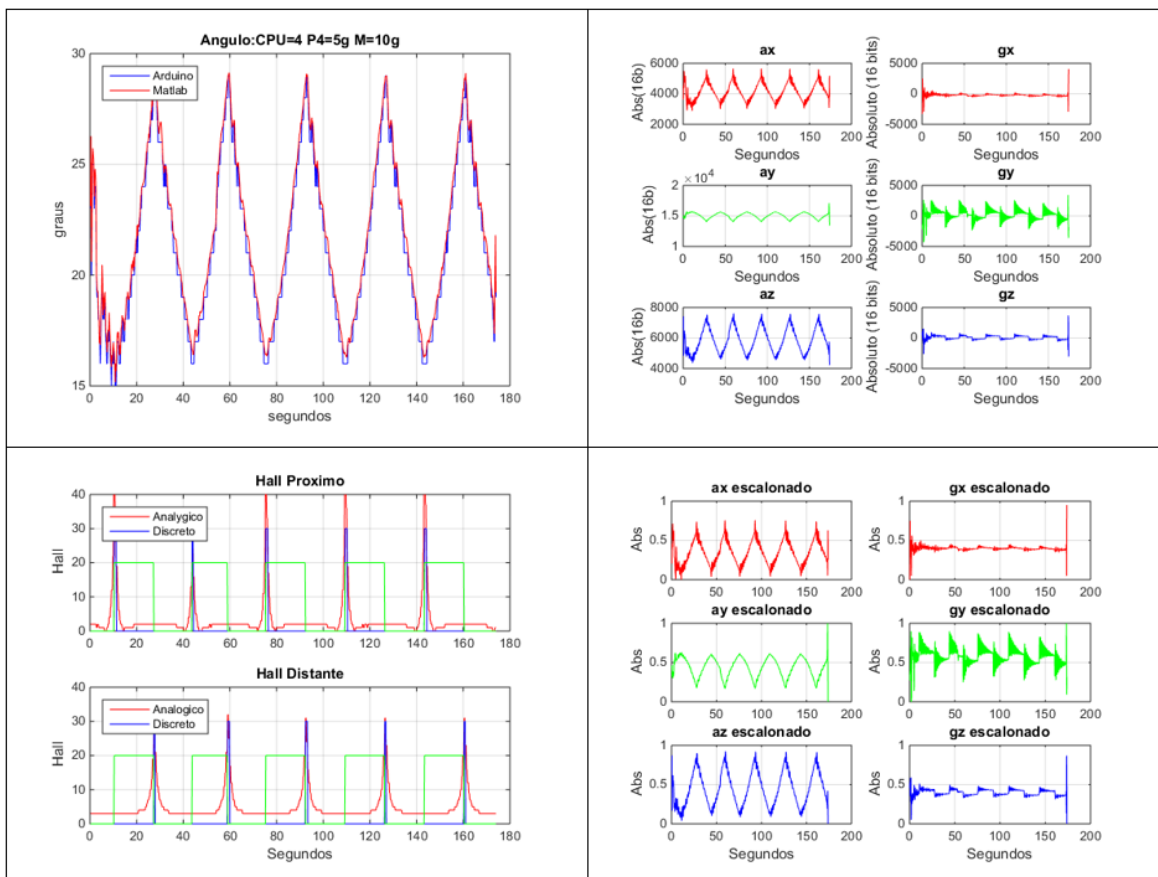


Figura A.12 – Ensaio 11 do lastro.

Tabela A.12 – ENSAIO 12

P1	P2	P3	P4	Mesa
0	0	0	10g	5g

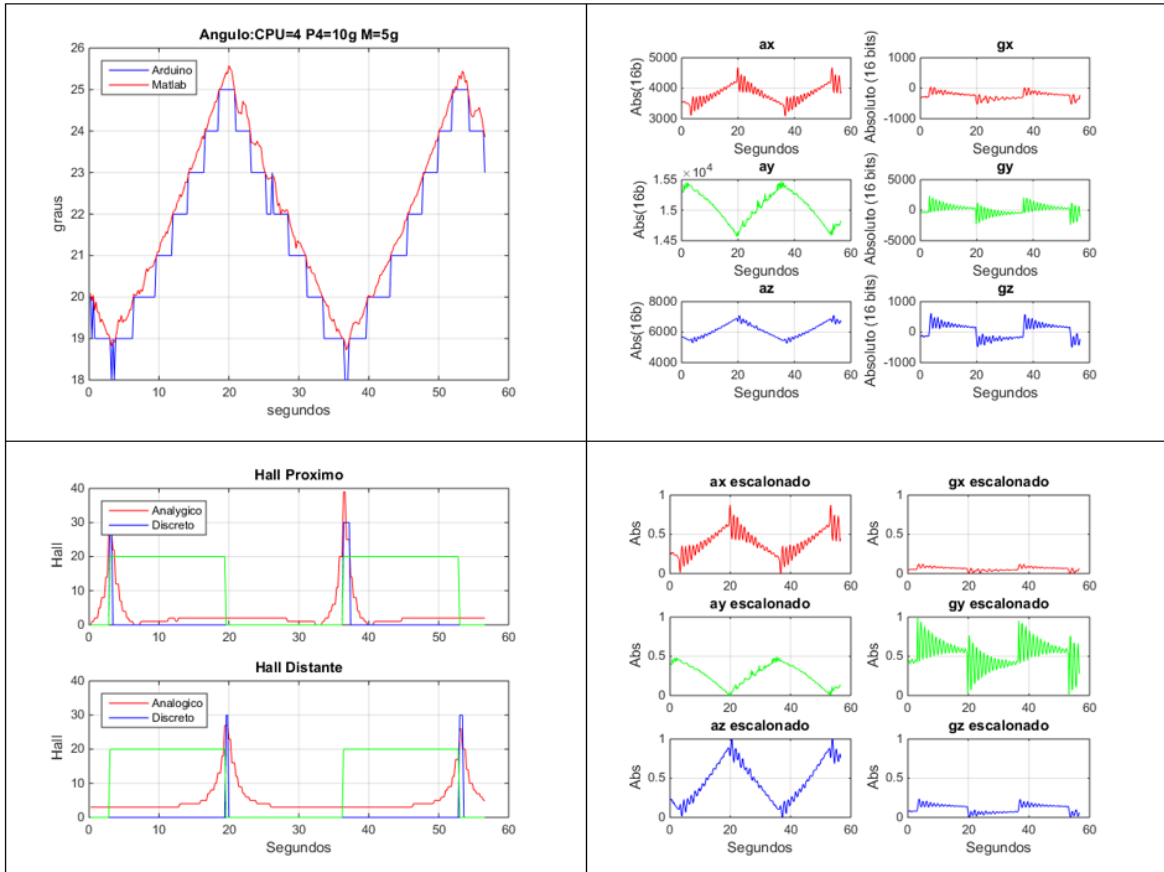


Figura A.13 – Ensaio 12 do lastro.