

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

LUKAS GABRIEL DIAS GOMES

DISPOSITIVO PARA MONITORAÇÃO DO PROCESSO DE FERMENTAÇÃO DE
BEBIDAS ALCOÓLICAS

Uberlândia

2022

Lukas Gabriel Dias Gomes

**DISPOSITIVO PARA MONITORAÇÃO DO PROCESSO DE FERMENTAÇÃO DE
BEBIDAS ALCOÓLICAS**

Trabalho de Conclusão de Curso de Engenharia de Controle e Automação da Universidade Federal de Uberlândia – UFU – Campus Santa Mônica, como requisito para a obtenção do título de Graduação em Engenharia de Controle e Automação.

Área de concentração: Engenharia de Controle e Automação

Orientador: Prof. Dr. Marcelo Barros de Almeida

Uberlândia

2022

Gabriel Dias Gomes, Lukas

Dispositivo para monitoração do processo de fermentação de bebidas alcoólicas/ **Lukas Gabriel Dias Gomes.** – **UBERLÂNDIA, 2022-** 87 p, 30 cm.

Orientador: Prof. Dr. Marcelo Barros de Almeida

Trabalho de Conclusão de Curso – Universidade Federal de Uberlândia – UFU
Faculdade de Engenharia Elétrica. **2022.**

Inclui bibliografia.

1. Sistemas embarcados. 2. Comunicação sem fio. 3. Automatização do processo.
Orientador Prof. Dr. Marcelo Barros de Almeida. II. Universidade Federal de Uberlândia. III. Faculdade de Engenharia Elétrica. IV. Engenharia de Controle e Automação.

LUKAS GABRIEL DIAS GOMES

Dispositivo para monitoração do processo de fermentação de bebidas alcoólicas

Trabalho de Conclusão de Curso de Engenharia de Controle e Automação da Universidade Federal de Uberlândia – UFU – Campus Santa Mônica, como requisito para a obtenção do título de Graduação em Engenharia de Controle e Automação.

Área de concentração: Engenharia de Controle e Automação

Uberlândia (MG), 29 de março de 2022.

Banca Examinadora:

Prof. Dr. Marcelo Barros de Almeida – Orientador

Prof. Dr. Josué Silva de Moraes – Membro

Prof. Dr. Marcio José da Cunha – Membro

AGRADECIMENTOS

Agradeço primeiramente a Deus, por sempre me abençoar na minha trajetória de vida e me dar motivação para ir atrás dos meus objetivos.

Agradeço aos meus pais Dilson e Sonia e ao meu irmão, que estão sempre me apoiando e motivando nas minhas escolhas, e trajetórias pessoais e profissionais. Meus exemplos de perseverança e dedicação, que os seus árduos esforços para sempre conseguir me proporcionar uma boa educação, sempre me motivam para sempre me esforçar em fazer o meu melhor.

Agradeço ao meu professor e orientador Marcelo Barros de Almeida por possibilitar junto comigo a concluir esse trabalho com êxito. Sendo uma pessoa referência para mim como exemplo profissional, me orientando desde minha primeira iniciação científica em 2017, e sou muito grato pelas oportunidades de experiências profissionais que ele me possibilitou ao longo da minha carreira acadêmica e acreditando no meu potencial.

“Por vezes sentimos que aquilo que fazemos não é senão uma gota de água no mar. Mas o mar seria menor se faltasse uma gota”

(Madre Teresa de Calcuta)

RESUMO

O processo de fermentação alcoólica, consiste em uma das etapas mais importantes para o processo de fabricação de bebidas alcoólicas, sendo essencial para formação de aromas característicos e da concentração de álcool, entre outros. O densímetro de vidro é um dos equipamentos utilizados tradicionalmente nesse processo, para medição da densidade. Porém esse equipamento apresenta limitações, pois ele consiste em medições de forma manual, sendo necessário um operador para realização das medições durante o processo de fermentação, e isso pode ocasionar risco de contaminação do produto e o seu desperdício, no momento de coletar amostras da bebida alcoólica para leitura da densidade. Então, este trabalho apresenta o desenvolvimento de um dispositivo para aprimorar a monitoração no processo de fermentação de bebidas alcoólicas, com objetivo de automatizar essa monitoração da densidade e reduzir os riscos de contaminação e desperdício do produto a ser fermentado.

No projeto foi levado em consideração o desenvolvimento de um hardware, capaz de fazer a aquisição de densidade a partir do sensor de acelerômetro e sensor de temperatura. E que seja capaz de enviar essas informações de aquisição para o cliente por meio de uma comunicação sem fio utilizando Bluetooth de baixo consumo (BLE). É um dispositivo que cumpra os requisitos de segurança alimentícia, que possa ser inserido dentro do fermentador, sem correr riscos de contaminação da bebida alcoólica durante o processo de fermentação. Os resultados dos dispositivos em funcionamento foram apresentados.

Palavras-chave: Sistemas embarcados; Comunicação sem fio; Automatização do processo.

ABSTRACT

The process of alcoholic fermentation, consists of one of the most important steps for the process of manufacture of alcoholic beverages, being essential for the formation of characteristic aromas and alcohol concentration, among others. The glass densimeter is one of the equipment traditionally used in this process, for density measurement. However, this equipment has limitations, because it consists of manual measurements, and an operator is required to perform the measurements during the fermentation process, and this can cause a risk of contamination of the product and its waste, when collecting samples of the alcoholic beverage to read the density. So, this work presents the development of a device to improve monitoring in the fermentation process of alcoholic beverages, to automate this density monitoring and reduce the risks of contamination and waste of the product to be fermented.

In the project was taken into consideration the development of a hardware, able to make the acquisition of density from the accelerometer sensor and temperature sensor. And that you can send this acquisition information to the customer through wireless communication using low-power Bluetooth (BLE). And a device that meets the requirements of food safety, which can be inserted into the fermenter, without running risks of contamination of the alcoholic beverage during the fermentation process. The results of the devices in operation were presented.

Keywords: Embedded systems; Wireless communication; Process automation.

LISTA DE ILUSTRAÇÕES

Figura 1 - Dispositivo comercial TILT™.....	15
Figura 2 - Dispositivo do projeto de código aberto iSpindel.....	15
Figura 3 - Forças que atuam no princípio do empuxo.....	16
Figura 4 - Fonte de alimentação.....	21
Figura 5 - Alimentação/Capacitores de desacoplamento e filtro.....	21
Figura 6 - Circuito USB.....	22
Figura 7 - Circuito do Microcontrolador e periféricos.....	23
Figura 8 - Circuito antena.....	24
Figura 9 - Circuito sensor inercial.....	24
Figura 10 - Circuito sensor de temperatura.....	25
Figura 11 - AS duas camadas juntas (Top + Bottom).....	25
Figura 12 - Primeira camada (Top).....	26
Figura 13 - Segunda camada (Bottom).....	26
Figura 14 - Recomendações de padrão de trilha de PCI para a linha de transmissão de RF, ao utilizar o componente MLPF-WB55-01E3.....	28
Figura 15 - Dimensões da antena de PCI (em mm).....	29
Figura 16 - PCI fabricada na JLCPCB.....	30
Figura 17 - Stencil utilizado no processo de soldagem dos componentes eletrônicos na PCI.....	31
Figura 18 - Aplicação da pasta de solda nas pedadas dos componentes eletrônicos na PCI.....	31
Figura 19 - Componentes eletrônicos posicionados na PCI.....	31
Figura 20 - Forno de soldagem de componentes SMD utilizado.....	32
Figura 21 - Componentes eletrônicos já soldados na PCI.....	32
Figura 22 - Recipiente plástico (tubete).....	33
Figura 23 - Estrutura 3D.....	33
Figura 24 - Estrutura 3D (lateral).....	34
Figura 25 - Criação do projeto de firmware.....	35
Figura 26 - Seleção do microcontrolador utilizado.....	35
Figura 27 - Pinos disponíveis do microcontrolador.....	36
Figura 28 - Configuração do HSEM.....	36
Figura 29 - Configuração do IPCC.....	37
Figura 30 - Configuração do RCC.....	37
Figura 31 - Configuração do debug.....	38

Figura 32 - Configuração do ADC para o sensor de temperatura.	38
Figura 33 - Configuração do RTC.....	39
Figura 34 - Configuração da I2C, para o sensor inercial.....	39
Figura 35 - Configuração do RF.....	40
Figura 36 - Configuração do RNG.	40
Figura 37 - Configuração do middleware.....	41
Figura 38 -Configuração das aplicações e serviços BLE.	41
Figura 39 - Configuração do GATT BLE.	42
Figura 40 - Configuração do serviço referente ao sensor inercial.....	42
Figura 41 - Configuração geral da característica referente ao evento do acelerômetro (dados de SG).....	43
Figura 42 - Configuração das propriedades da característica referente ao evento do acelerômetro (dados de SG).	44
Figura 43 - Configuração das permissões da característica referente ao evento do acelerômetro (dados de SG).	44
Figura 44 - Configuração dos eventos de GATT da característica referente ao evento do acelerômetro (dados de SG).	44
Figura 45 - Configuração geral da característica referente a inclinação.	45
Figura 46 - Configuração das propriedades da característica referente a inclinação.	45
Figura 47 - Configuração das permissões da característica referente a inclinação.	46
Figura 48 - Configuração dos eventos de GATT da característica referente a inclinação.	46
Figura 49 - Configuração do serviço referente ao sensor de temperatura.	46
Figura 50 - Configuração geral da característica referente a temperatura.....	47
Figura 51 - Configuração das propriedades da característica referente a temperatura.....	48
Figura 52 - Configuração das permissões da característica referente a temperatura.....	48
Figura 53 - Configuração dos eventos de GATT da característica referente a temperatura. ...	48
Figura 54 - Configurações gerais do middleware.....	49
Figura 55 - Configurações de emparelhamento do BLE.	50
Figura 56 - Configurações dos periféricos digitais de saída referentes aos LEDs e chave de gerenciamento de carga para os sensores.	51
Figura 57 - Configuração do relógio.	51
Figura 58 - Configuração do LSE e HSE.	52
Figura 59 - Configuração do relógio do RNG.....	53
Figura 60 - Configuração do sistema de wakeup do RF.....	53

Figura 61 - Configurações avançadas das funções geradas.....	53
Figura 62 - Pontos de configuração.....	54
Figura 63 - Árvore de projeto gerada no STM32CubeIde.....	55
Figura 64 - Implementação de firmware para leitura de acelerômetro.....	56
Figura 65 - Implementação de firmware para leitura de inclinação.....	56
Figura 66 - Implementação de firmware para leitura de temperatura.	57
Figura 67 - Leitura do ADC referente ao sensor de temperatura LMT86.....	57
Figura 68 - Leitura do ADC referente ao sensor de temperatura LMT86.....	57
Figura 69 - Parâmetros da máscara dos bits das características.	59
Figura 70 – Alterações na função de solicitação de publicidade.....	60
Figura 71 – Inclusão de biblioteca e declaração de variáveis.....	60
Figura 72 – Declaração de variáveis para armazenar os dados obtidos dos sensores.	61
Figura 73 – Definições dos tempos de envio da notificação e definições de macros para ordenação dos bytes da notificação.	61
Figura 74 - Tarefas para atualização das características.....	62
Figura 75 - Callback para os temporizadores de software, referentes as tarefas.....	62
Figura 76 – Inicializações da aplicação BLE.	63
Figura 77 - Inicialização do sensor inercial e das variáveis de status de notificação.....	63
Figura 78 – Implementação na aplicação da notificação referente ao evento de acelerômetro.	64
Figura 79 - Implementação na aplicação da notificação referente a inclinação.....	64
Figura 80 - Implementação na aplicação da notificação referente ao evento de acelerômetro.....	65
Figura 81 – Atualização dos dados que serão adicionados na notificação do evento de acelerômetro.	65
Figura 82 - Função de leitura do SG (gravidade específica).	66
Figura 83 - Atualização dos dados que serão adicionados na notificação de inclinação.	66
Figura 84 - Atualização dos dados que serão adicionados na notificação de temperatura.....	67
Figura 85 – Envio de notificação referente aos dados de evento de acelerômetro (gravidade específica).	67
Figura 86 - Envio de notificação referente aos dados de inclinação.	67
Figura 87 - Envio de notificação referente aos dados de temperatura.....	68
Figura 88 - Balança digital.	68
Figura 89 - Refratômetro.	69
Figura 90 - Recipiente para mistura.	69

Figura 91 - Calculadora para obtenção da curva de calibração do dispositivo.	71
Figura 92 - Hardware fixado na estrutura mecânica.....	72
Figura 93 - Montagem completa do dispositivo de monitoração do processo de fermentação de bebidas alcoólicas.	73
Figura 94 - Dispositivo imerso na água.....	74
Figura 95 - Dados de temperatura.	74
Figura 96 - Dados de inclinação.....	75
Figura 97 - Dados de SG.	76
Figura 98 - Dados de RSSI.....	76
Figura 99 - Dados comparativos entres dois dispositivos na medição da densidade de soluções de água e açúcar.....	77

LISTA DE TABELAS

Tabela 1 - Formato de dados de evento do acelerômetro (comprimento de 5 bytes).....	43
Tabela 2 – Formato de dados de inclinação (comprimento de 4 bytes).....	45
Tabela 3 – Formato de dados de temperatura (comprimento de 4 bytes).....	47
Tabela 4 - Formatação de campo específica do fornecedor do BlueST-SDK.....	58
Tabela 5 - Bits referente as características utilizadas.....	59
Tabela 6 - Dados de inclinação e SG obtidos na calibração do dispositivo.....	70

LISTA DE ABREVIATURAS E SIGLAS

ADC	<i>Analog-to-Digital Converter</i> (Conversor analógico-digital)
ATT	<i>Attribute Protocol</i>
BLE	<i>Bluetooth Low Energy</i>
EMC	<i>Electromagnetic Compatibility</i>
ESD	<i>Electrostatic Discharge</i>
GAP	<i>Generic Access Profile</i>
GATT	<i>Generic Attribute Profile</i>
GND	<i>Ground</i>
HSE	<i>High Speed Clock</i>
HSEM	<i>Hardware Semaphores</i>
I2C	<i>Inter Integrated Circuit</i>
IMU	<i>Inertial measurement unit</i> (Unidade de medida inercial)
IPCC	<i>Inter-Process Communication Controller</i>
LED	<i>Light Emitting Diode</i> (Diodo emissor de luz)
LSE	<i>Low Speed Clock</i>
MAC	<i>Media Access Control</i>
mAh	Miliampere-hora
MHz	Megahertz
NVIC	<i>Nested Vectored Interrupt Controller</i>
PCI	Placa de circuito impresso
PET	Politereftalato de etileno
PIN	<i>Personal Identification Number</i> (Número de identificação pessoal)
RAM	<i>Random Access Memory</i>
RCC	<i>Reset and Clock Controller</i>
RF	<i>Radio Frequency</i>
RNG	<i>Randow Number Generator</i>
RSSI	<i>Received Signal Strength Indicator</i>
RTC	<i>Real-Time Clock</i>
SCL	<i>Serial Clock</i>
SDA	<i>Serial Data</i>
SG	<i>Specific Gravity</i> (Gravidade específica)

SMD *Surface-Mount Device*
ST *STMicroelectronics*
SWD *Serial Wire Debug*
SYS *System*
USB *Universal Serial Bus*
WIFI *Wireless Fidelity*

SUMÁRIO

1	INTRODUÇÃO	12
1.1	Justificativa	13
1.2	Objetivo	13
2	REFERENCIAL TEÓRICO	14
2.1	Tecnologias correlatas	14
2.2	Cálculo da Densidade de Líquidos	15
2.3	Invólucro do densímetro compatível com alimentos	17
2.4	Tecnologias de comunicação sem fio para envio de dados	17
2.5	Termos e conceitos do protocolo de comunicação Bluetooth de baixa energia	18
2.5.1	GAP (<i>Generic Access Profile</i>)	18
2.5.2	GATT (<i>Generic Attribute Profile</i>)	18
2.5.3	ATT (<i>Attribute Protocol</i>)	18
3	DESENVOLVIMENTO	19
3.1	Levantamento de requisitos	19
3.2	Hardware	20
3.2.1	Esquema elétrico	20
3.2.1.1	Fonte de alimentação	20
3.2.1.2	Alimentação/capacitores de desacoplamento e filtro	21
3.2.1.3	Circuito USB	21
3.2.1.4	Circuito do Microcontrolador e outros periféricos	22
3.2.1.5	Circuito antena	23
3.2.1.6	Circuito do sensor inercial	24
3.2.1.7	Circuito sensor de temperatura	25
3.2.2	Layout	25
3.2.2.1	Camadas da PCI	26
3.2.2.2	Planos e vias de Terra	26
3.2.2.3	Material dielétrico	27
3.2.2.4	Geometria das trilhas	27
3.2.2.5	Espessura da PCI	28
3.2.2.6	Antena feita na própria PCI	29
3.2.2.7	Fabricação da PCI	29
3.2.2.8	Soldagem dos componentes eletrônicos na PCI	30

3.3	Estrutura mecânica	33
3.4	Firmware	34
3.4.1	Configuração de inicialização dos periféricos	34
3.4.2	Implementação de firmware para os sensores	55
3.4.2.1	<i>Firmware sensor inercial</i>	<i>56</i>
3.4.2.2	<i>Firmware sensor de temperatura do sensor ICM-20689</i>	<i>56</i>
3.4.2.3	<i>Firmware sensor de temperatura do sensor LMT86DCK.....</i>	<i>57</i>
3.4.3	Implementação de firmware para o BLE.....	58
3.5	Processo de Calibração.....	68
4	RESULTADOS E DISCUSSÕES.....	72
5	CONCLUSÃO.....	78
6	REFERÊNCIAS BIBLIOGRÁFICAS	79

1 INTRODUÇÃO

No processo de fabricação de bebidas alcoólicas como por exemplo da cerveja que é mais conhecida, é utilizado como matéria prima ingredientes básicos como o lúpulo, leveduras (micro-organismos utilizados no processo de fermentação), malte (grãos de cereal) e água. No processo de fermentação dessa bebida alcoólica é gerado subprodutos como álcool, dióxido de carbono e aromas característicos da bebida, que foram realizados a partir de um processo químico pelas leveduras no processamento dos açúcares do malte (PINTARELLI; MARCONDES, 2019).

Em relação as características finais das bebidas alcoólicas são necessárias levar em consideração fatores que podem impactar na sua qualidade, dentre esses fatores estão as características intrínsecas as leveduras: genética, tipo e crescimento. Além das leveduras outros fatores que podem impactar são qualidade dos equipamentos, possíveis contaminações, agitação da mistura, composição e aspectos químicos do mosto (PINTARELLI; MARCONDES, 2019).

No monitoramento de variáveis importantes no processo de fermentação como a densidade é bastante utilizada o densímetro de vidro. Esse equipamento é simples de ser utilizado e possui baixo custo para a realização de experiências físicas e químicas que envolvem o conceito de densidade de líquidos, sem a necessidade de utilizar instrumentos de medidas como balança (OLIVEIRA; MELO FILHO; AFONSO, 2013).

Os métodos de monitoramento no processo de fermentação de bebidas alcoólicas podem variar amplamente em custo e complexidade. Na produção de cerveja pode-se conseguir medições simples a partir da observação empírica e através de testes manuais básicos utilizando equipamentos como termômetros e densímetros. Há também medições mais sofisticadas, que geralmente são usadas em cervejarias comerciais maiores, que utilizam sistemas de teste de computador. Na monitoração no processo de fermentação é importante analisar variáveis como temperatura, gravidade específica, pH, oxigênio e dióxido de carbono. Havendo a necessidade de monitorar essas variáveis regulamentos para análise do progresso da fermentação. Pois é de extrema importância essa monitoração, principalmente no início do processo, para detecção de possíveis problemas, que podem ocasionar desperdício do produto (WHITE; ZAINASHEFF, 2010).

1.1 Justificativa

A necessidade do desenvolvimento de um dispositivo que realize o monitoramento em tempo real do processo de fermentação de bebidas alcoólicas, é devido a automação do processo e evitar erros durante a fermentação. Problemas como fermentações travadas ou lentas, o atraso na execução das medidas corretiva que o podem afetar as características da bebida alcoólica de forma indesejada e de perdas na produção. O monitoramento remoto visa evitar contaminações externas, que possam ocorrer em um método manual em que é feito coletas de forma direta de dentro do fermentador.

1.2 Objetivo

O projeto a ser desenvolvido é um densímetro eletrônico para auxílio na produção de bebidas alcoólicas, para medição de densidade e temperatura desse líquido. Esse dispositivo tem como objetivo o monitoramento do processo de fermentação na fabricação de bebidas alcoólicas de forma remota a partir de tecnologia de comunicação sem fio como Bluetooth de baixo consumo (BLE). Assim, o intuito deste trabalho é apresentar um dispositivo compacto e impermeável que seja imerso a um tanque de fermentação de bebidas alcoólicas, e que seja capaz de automatizar a supervisão do processo de fermentação, evitando que haja o contato com o meio externo com a bebidas alcoólica na aferição de densidade do fluido. Essa abordagem permitirá um processo mais seguro e rápido de medição de densidade, pois não haverá a necessidade de abrir o fermentador ao longo do processo de fermentação para a aferição das variáveis do processo, evitando que ocorra riscos de contaminação. Possibilitando que se tenha um processo automatizado que executa medições em tempo real, diferente do processo manual que é mais demorado e caro.

2 REFERENCIAL TEÓRICO

O referencial teórico é constituído por aspectos como: “Tecnologias correlatas”, “Cálculo da densidade em líquidos”, “Invólucro do densímetro compatível com alimentos”, Tecnologias de comunicação sem fio para envio de dados” e “Termos e conceitos do protocolo de comunicação Bluetooth de baixa energia”.

2.1 Tecnologias correlatas

Antes de dar início ao desenvolvimento do dispositivo, foi feito um levantamento sobre as tecnologias correlatas ao dispositivo desenvolvido. Levando em consideração nessa análise, os princípios de funcionamento de cada um, tipos de sensores e protocolos de comunicação sem fio utilizados.

O primeiro dispositivo encontrando foi o TILT™ da Figura 1, que é vendido pela *Barob Brew Equipment*. Esse dispositivo faz a medição de gravida específica, utilizando o princípio da inclinação. Além da gravidade específica, ele também faz a medição de temperatura. A tecnologia de comunicação sem fio utilizada é o Bluetooth 4.0+. Em relação a esse dispositivo foi notado melhorias que podem ser empregadas para o projeto do dispositivo desenvolvido, com por exemplo, a utilização de um rádio Bluetooth com tecnologia mais atual. Como no caso do Bluetooth 5.2, que apresenta diferenças significativas em relação ao Bluetooth 4.0+, como maiores taxas de transmissão (Duas vezes maior – 2 Mbps), maior faixa de alcance de distância, menor consumo energético, confiabilidade e entre fatores que evidenciam as vantagens de se utilizar o Bluetooth 5.2 em relação ao Bluetooth 4.0+.

Outro dispositivo encontrado foi um densímetro de documentação de código aberto, do projeto *iSpindle*, Figura 2, que se utiliza do mesmo princípio de medição de gravidade específica por inclinação. Esse projeto serve de base para diversos outros dispositivos que são comercializados no mercado, utilizando os mesmos componentes de hardware. A tecnologia de comunicação sem fio empregada nesse dispositivo é um ESP32 WIFI. Porém o WIFI possui desvantagens em relação ao consumo energético, devido apresentar um grande consumo de corrente em relação ao BLE, sendo limitado para casos de processo de fermentação de bebidas alcoólicas que demandam uma longa duração. Outro detalhe analisado desse dispositivo, foi a utilização de diversos módulos de hardware, que descaracteriza a ideia de um produto profissional, e sim se tratando de um protótipo.

Figura 1 - Dispositivo comercial TILT™



Fonte: BARON BREW EQUIPMENT.

Figura 2 - Dispositivo do projeto de código aberto iSpindel.



Fonte: https://www.ispindel.de/docs/README_pt.html.

2.2 Cálculo da Densidade de Líquidos

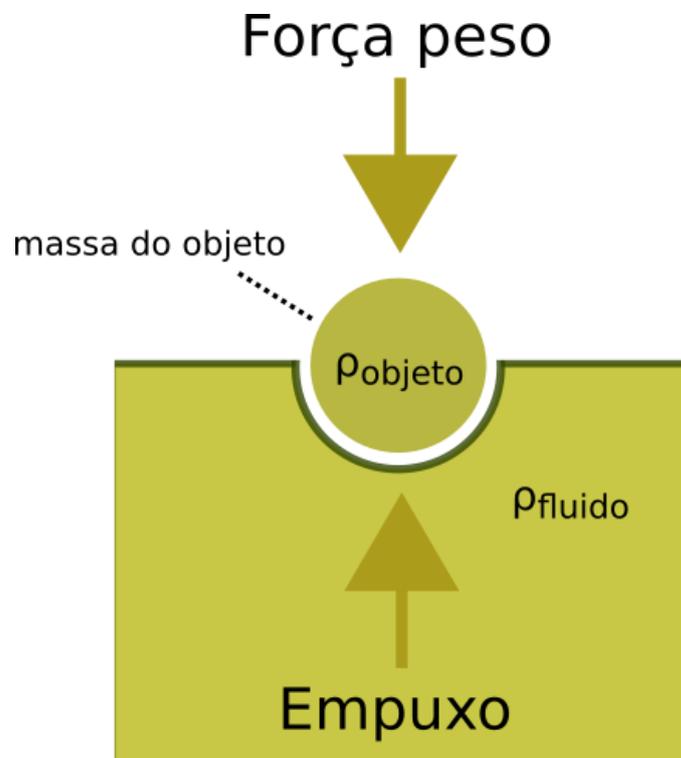
No processo de fermentação da bebida alcóolica, a variável a ser estudada e analisada durante esse processo é a densidade relativa. O conceito de densidade relativa (δ) de acordo com a equação (1), é a razão entre a densidade (massa (m)) de um material e o seu volume (V) de uma substância (ρ) e a densidade de um dado material de referência como a água (ρ_{H_2O}).

$$\delta = \frac{\rho}{\rho_{H2O}} \quad (1)$$

As obtenções de medidas de densidade de líquidos podem ser feitas através de métodos como densímetro (um método de flutuação para líquidos), balança hidrostática, método do corpo imerso, picnômetro e densímetro oscilante. Dentre esses métodos, o método do densímetro de flutuação é o mais recomendando para medir a densidade de bebidas alcólicas, pois sua viscosidade é na ordem de $\text{mPa}\cdot\text{s}^{-1}$ e esse método consegue medir viscosidades de líquidos de até $5 \text{ Pa}\cdot\text{s}^{-1}$ (OECD, 2012).

O princípio de funcionamento do densímetro para obtenção da densidade está relacionado com o fenômeno da flutuabilidade. O densímetro flutua a partir do momento que ele se encontra em equilíbrio, quando a força peso é igual a de empuxo ou a massa do densímetro é igual à do fluido, como representando na Figura 3. Então quando ocorre a alteração da densidade do fluido, conseqüentemente muda a flutuabilidade do densímetro. (OLIVEIRA; MELO FILHO; AFONSO, 2013).

Figura 3 - Forças que atuam no princípio do empuxo.



Outro princípio utilizado por trás do funcionamento do densímetro é o conceito de metacentro. Esse conceito se dá quando um corpo imerso a um líquido muda o seu ângulo de inclinação, quando a densidade do líquido mudar em relação ao seu centro de massa e o centro de empuxo (OECD, 2012).

2.3 Invólucro do densímetro compatível com alimentos

Para a escolha do invólucro do densímetro desenvolvido, foi feita uma pesquisa sobre a normas de fiscalização de produtos alimentícios, que abordam sobre as características do material do involucro que é permitido para entrar em contato com os alimentos. Dentre as normas analisadas, estão as descritas abaixo:

No ramo alimentício como de produção de bebidas alcólicas devem ser respeitadas normas de fiscalização como a “INSTRUÇÃO NORMATIVA Nº 5, DE 31 DE MARÇO DE 2000”, estabelecida pelo Ministério da Agricultura, Pecuária e Abastecimento. Que rege que todos os materiais de todos equipamentos e utensílios que entram em contato com bebidas devem apresentar características não tóxicas, sem odores, nem sabores, e sejam impermeabilizados, bem como resistentes à corrosão e a repetidas operações de limpeza e desinfecção (MINISTÉRIO DA AGRICULTURA PECUÁRIA E ABASTECIMENTO, 2000).

2.4 Tecnologias de comunicação sem fio para envio de dados

No dispositivo a ser desenvolvido haverá a necessidade de envio de dados através de um protocolo de comunicação sem fio. Sendo necessário um protocolo que permita taxas de comunicação suficientes para envio de dados de temperatura e densidade, baixo consumo de energia, que tenha alcance suficiente para se comunicar com o usuário de fora de um fermentador. A facilidade de conexão com dispositivos pessoais como smartphones, tablets e notebooks foram determinantes para a escolha de tecnologias como por exemplo o WIFI e Bluetooth, dentre as tecnologias de comunicação presentes no mercado atualmente.

Levando em comparação ao consumo energético, os dispositivos Bluetooth apresentaram menores gastos energéticos do que dispositivos WIFI (Thomas; Wilkie; Irvine, 2016). Como o densímetro eletrônico é um dispositivo alimentado a bateria, então o uso de um protocolo de baixo consumo como Bluetooth (BLE) é a melhor opção.

2.5 Termos e conceitos do protocolo de comunicação Bluetooth de baixa energia

Foram importantes entender alguns termos e conceitos do Bluetooth, que foram utilizados como base para o desenvolvimento do firmware. Os termos e conceitos utilizados como referencial teórico serão descritos a seguir:

2.5.1 GAP (*Generic Access Profile*)

O GAP é responsável pelo controle das conexões e publicidades no protocolo de comunicação Bluetooth. Permitindo que os dispositivos Bluetooth sejam visíveis e como devem interagir entre si. Esses dispositivos podem ser classificados em dois tipos: os dispositivos periféricos que são os de baixa potência como por exemplo o dispositivo desenvolvido de medição de densidade e temperatura, e como dispositivos centrais que são como por exemplo celulares, que contém maior quantidade de memória e poder de processamento.

2.5.2 GATT (*Generic Attribute Profile*)

O GATT define a maneira como dois dispositivos BLE transferem dados, utilizando os conceitos chamados de serviços e características. Fazendo a utilização do ATT que é um protocolo de dados genérico, que possibilita armazenar serviços, características e dados relacionados em uma simples tabela de pesquisa, contendo IDs de 16 bits para cada entrada.

Nesta etapa o dispositivo central (smartphone) pergunta ao dispositivo periférico (dispositivo de monitoração de bebidas alcoólicas), referente os serviços e características que ele tem a oferecer. Obtendo depois a resposta do dispositivo periférico sobre qual é o serviço e suas características (leitura, escrita e notificação).

2.5.3 ATT (*Attribute Protocol*)

Este protocolo define como um servidor expõe seus dados a um cliente. Esses dados expostos pelo servidor são estruturados como atributos. Os atributos são armazenados no servidor e listados como tabelas. Os atributos são tipos de dados expostos pelo servidor, como por exemplo serviços e características.

Um serviço corresponde a um agrupamento de um ou mais atributos, que contém características. Essas características são itens de dados presentes dentro de um serviço, esses itens consistem em um tipo, valor, propriedades, descritores e permissões.

3 DESENVOLVIMENTO

O desenvolvimento do dispositivo de monitoração do processo de fermentação de bebidas alcoólicas ou dispositivo de medição de densidade, foi dividido em quatro etapas: levantamento de requisitos, hardware, firmware e processo de calibração. Abaixo será detalhado sobre cada uma dessas etapas.

3.1 Levantamento de requisitos

Nessa primeira etapa, foram levantados os requisitos necessários para o desenvolvimento do protótipo, em função do que deveria ou não conter no projeto. A intenção inicial no desenvolvimento desse protótipo, foi projetar um dispositivo com tamanho reduzido em comparação aos dispositivos disponíveis atualmente no mercado (descritos na seção 2.1), de baixo consumo alimentado por bateria, baixo custo, com tudo integrado sem a utilização de módulos, e que permita ser imerso em um recipiente com líquido.

Evidenciando um tamanho de hardware reduzido e de baixo consumo, foi encontrado uma opção de microcontrolador com tudo integrado, tanto com o circuito de microprocessamento e periférias, quanto o rádio, em apenas um único chip, reduzindo o número de componentes para o hardware. O microcontrolador da série STM32WB55, além de minimizar o projeto do circuito de hardware, esse microcontrolador possui baixo consumo, pois conta com configurações de modo de consumo, e o rádio de baixa potência que conta com opções de tecnologia de comunicação como o BLE e ZigBee.

Outro requisito importante foi o da escolha da bateria/pilha. A escolha do uso do padrão bateria/pilha AAA, foi pela facilidade de encontrar disponível no mercado, tamanho reduzido e possui grande capacidade de armazenamento de carga de até 1200 mAh.

Com a necessidade do dispositivo ser imerso em líquidos por um tempo limitado durante o processo de fermentação de bebidas alcoólicas, foi levantando o requisito de grau de proteção para o invólucro do dispositivo, sendo necessário um invólucro que seja capaz de proteger a eletrônica contra imersão de água contínua, com características de proteção IP68 conforme a norma IEC 60529. A opção de invólucro encontrada, foi de um tubo plástico PET Figura 22,

com fechamento de tampa por rosca, sendo ainda adicionado um anel de vedação oring para melhor vedação. Esse involucro foi testado dentro de um recipiente líquido, garantindo que não seja possível a entrada de água dentro do recipiente e cause danos na eletrônica.

3.2 Hardware

3.2.1 Esquema elétrico

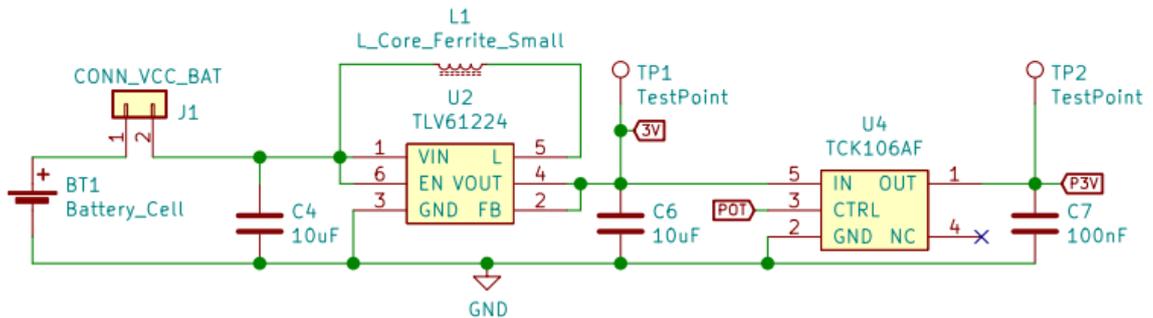
O desenvolvimento do esquema elétrico foi feito utilizando a ferramenta de projeto eletrônico KiCad. Essa ferramenta permite representar todos os componentes eletrônicos necessários no circuito e os interligá-los entre si, formando uma representação do sistema completo. Os datasheets e manuais de referência dos componentes eletrônicos foram utilizados como referência para o projeto do circuito, estabelecendo de forma correta os requisitos de alimentação, desacoplamento e periferia adicional, conforme descritos nos datasheets e manuais.

3.2.1.1 Fonte de alimentação

Contém um circuito de regulagem de tensão da bateria/pilha para uma faixa de tensão de entrada operacional entre 0,7 V e 3 V, regulando para uma saída de tensão fixa de 3 V, que é a tensão operacional de todos os componentes presentes no circuito, como sensores e microcontrolador. O uso de um regulador de tensão chaveado (U2) permite uma maior eficiência de alimentação, pois se tratando de um sistema alimentado por bateria/pilha, em que há a descarga ao longo do tempo da capacidade da alimentação, há a necessidade de diminuir o número de perdas na regulagem da tensão. E a escolha por um conversor *Step-Up*, para elevação da tensão de entrada é devido permitir o uso de baterias/pilhas de tamanho reduzido como as de padrão AAA, que geralmente possuem tensão entre 1,2 e 1,5 V.

Os capacitores de desacoplamento na entrada (C4) e na saída (C6), possuem a função de estabilizar a alimentação de entrada e saída em relação aos ruídos de alta frequência. Foi utilizado uma chave de carga (U4) para gerenciamento de energia, em relação a alimentação dos sensores, uma estratégia para diminuir o consumo, permitindo alimentar o circuito dos sensores, apenas no momento da aquisição, evitando consumir indevidamente corrente nos momentos em que não está sendo feito a aquisição.

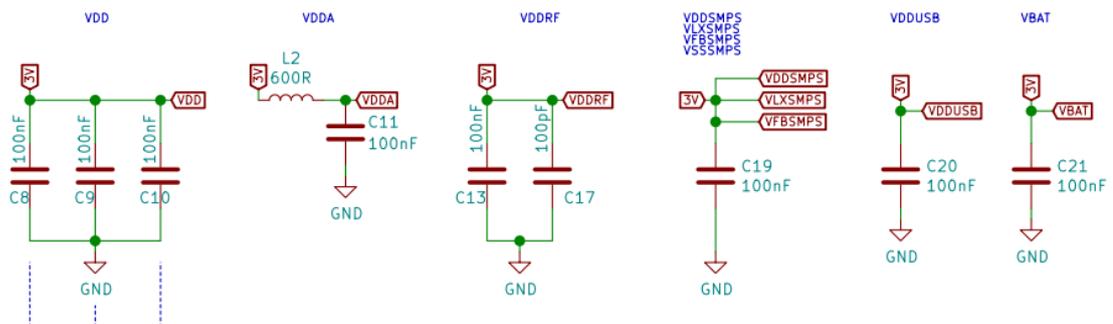
Figura 4 - Fonte de alimentação.



3.2.1.2 Alimentação/capacitores de desacoplamento e filtro

São requeridos capacitores de desacoplamento antes das entradas de alimentação do microcontrolador para prover uma alimentação estável durante os ruídos de alta frequência. Os capacitores de desacoplamento (C8, C9, C10) são para a alimentação digital (VDD), (C11) para alimentação analógica (VDDA) em conjunto com um indutor (L2) para impedir ruídos provenientes da alimentação digital, (C13 e C17) para alimentação digital do circuito de rádio frequência (VDDRF), (C20) para alimentação digital da USB (VDDUSB), e (C21) para alimentação do VBAT que é destinado por alimentar o RTC interno do microcontrolador.

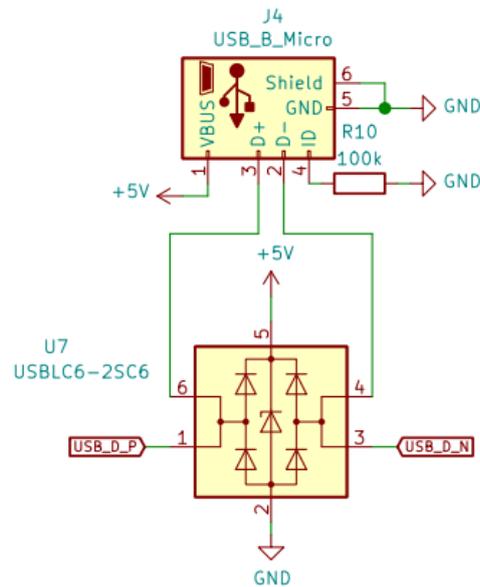
Figura 5 - Alimentação/Capacitores de desacoplamento e filtro.



3.2.1.3 Circuito USB

O uso da USB serve como opção para transmitir e armazenar os dados das leituras dos sensores, e até mesmo para depuração. Nesse circuito contém o componente U7 (USBLIC6-2SC6) que é utilizado para proteção ESD, comum em circuitos com USB. E como conector USB está sendo usado o tipo micro USB (J4).

Figura 6 - Circuito USB.



3.2.1.4 Circuito do Microcontrolador e outros periféricos

O componente U3 é o microcontrolador STM32WB55CEU6 utilizado no projeto. A sua escolha foi devido ao fato de já conter em um único chip, tanto a parte de rádio, quanto o microcontrolador e por suportar padrões de comunicação sem fio de baixo consumo como o Bluetooth LE 5.2 e IEEE 802.15.4.

Circuitos com LEDs (D1 e D2, e resistores de limitação de corrente R3 e R4) servem para indicar alguma sinalização pré-estabelecida para o usuário, como por exemplo, se o dispositivo está em modo de pareamento.

O circuito de BOOT é usado para selecionar a opção de inicialização do microcontrolador, que pode ser a partir da memória do sistema, memória Flash ou da RAM. Esse é composto por uma chave (SW3) e resistores (R5 e R6) que permitem que o nível lógico do pino de entrada BOOT0 seja colocado em alto ou baixo. O microcontrolador faz a leitura desse sinal para entender de onde deve buscar o seu código inicial.

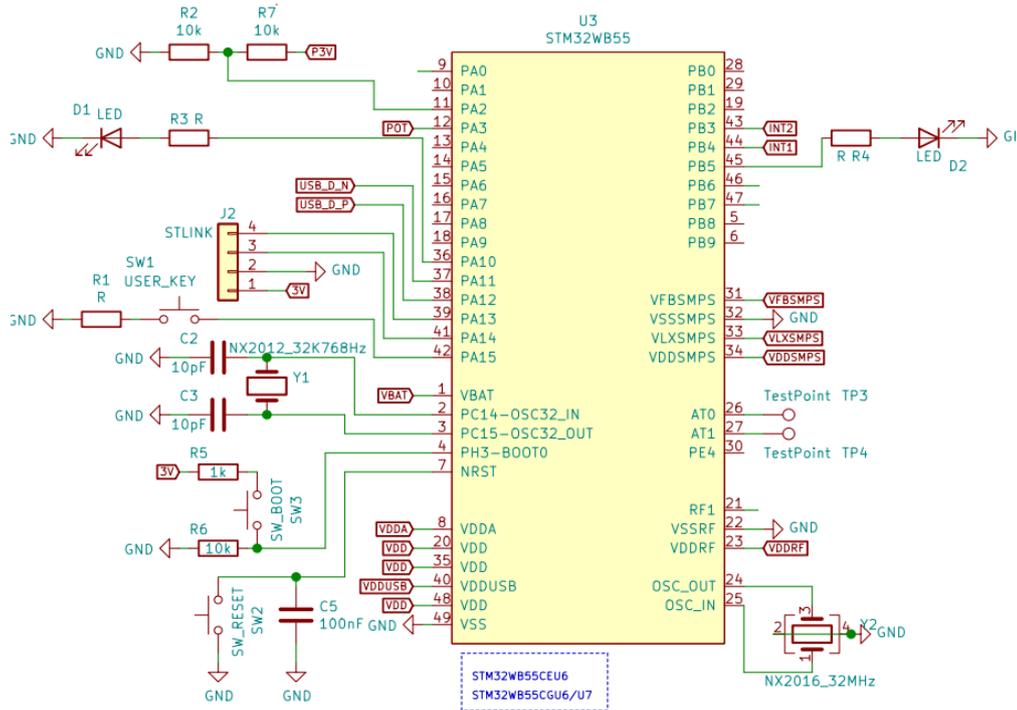
O header J2 é uma interface de depuração e programação do microcontrolador.

Cristal oscilador externo (Y2) do rádio, sua sintonização é feita por firmware, que é possível selecionar os valores dos capacitores corretos, gerando a frequência de 32 MHz requerida para funcionamento do rádio.

Cristal oscilador externo (Y1) destinado ao RTC do microcontrolador, sintonizado com capacitores (C2 e C3), gerando a frequência de 32,768 KHz.

O circuito contém também duas chaves, uma destinada ao usuário que é a (SW1), que ele pode predefinir uma ação a ela. E a chave (SW2) que é usada para resetar o microcontrolador.

Figura 7 - Circuito do Microcontrolador e periféricos.



3.2.1.5 Circuito antena

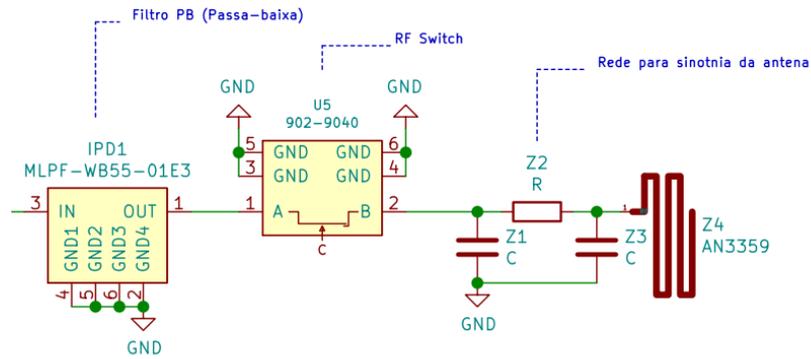
O circuito integrado MLPF-WB55-01E3 (IPD1) integra uma rede de correspondência de impedância e filtro de harmônicos. A rede de impedância correspondente foi adaptada para maximizar o desempenho de RF do STM32WB. Este dispositivo usa a tecnologia STMicroelectronics IPD em substrato de vidro não condutor que otimiza o desempenho de RF.

A antena (Z4) que é do tipo PCI, uma antena feita na própria placa.

Capacitores para sintonizar o rádio, através de um casamento de impedâncias (Z1, Z2 e Z3) da linha de transmissão. Os valores finais são definidos após testes com equipamentos específicos para RF.

Comutador RF (U5), de impedância 50 ohms, para a conexão de um analisador vetorial de sinal, permitindo a sintonia da antena (seleção dos valores dos componentes passivos Z1, Z2 e Z3)

Figura 8 - Circuito antenna.



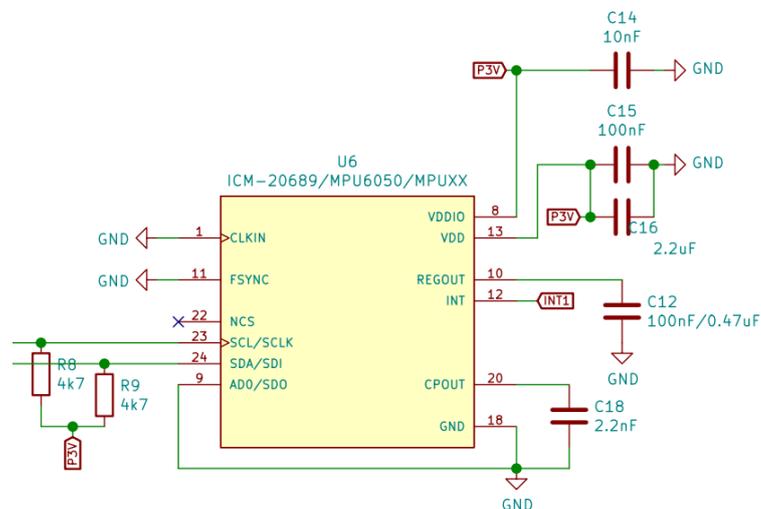
3.2.1.6 Circuito do sensor inercial

Contém uma unidade de medição inercial (U6) de 6 eixos, que combina um giroscópio de 3 eixos e um acelerômetro de 3 eixos.

E foi necessário capacitores de desacoplamento nas entradas de alimentação da IMU, prover uma alimentação estável durante os ruídos de alta frequência. Os capacitores de desacoplamento (C15 e C16), são para alimentação digital (VDD), o capacitor (C14) para alimentação logica digital (VDDIO), e o capacitor (C12) no pino de saída do regulador (REGOUT) que é destinado para o filtro do regulador.

Para a comunicação entre o microcontrolador e o ICM-20689, foi utilizado a interface de comunicação I2C, sendo necessário a adição de resistores de *pull-up* (R8 e R9) nas linhas de comunicação SDA e SCL.

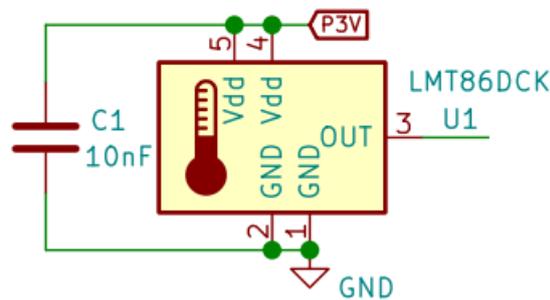
Figura 9 - Circuito sensor inercial.



3.2.1.7 Circuito sensor de temperatura

No circuito do sensor de temperatura, contém apenas um sensor de temperatura LMT86DCK (U1), que possui uma saída analógica, que varia conforme a temperatura do ambiente. E para desacoplamento na alimentação digital (VDD), possui um capacitor (C1) para prover uma alimentação estável durante os ruídos de alta frequência.

Figura 10 - Circuito sensor de temperatura.



3.2.2 Layout

O projeto de *layout* da Placa de Circuito Impresso (PCI) do dispositivo de monitoramento do processo de fabricação de bebidas levou em consideração critérios e técnicas para atender as condições necessárias ao funcionamento satisfatório do dispositivo. E como referência de projeto de *layout*, foi utilizado o *Application note* (AN5165), que aborda regras de *layout* para o desenvolvimento de hardware de RF para microcontroladores STM32WB, possibilitando um projeto de RF que tenha melhor desempenho e performance possível. Abaixo seguem as descrições de cada um dos critérios e técnicas utilizados para o desenvolvimento do *layout*.

Figura 11 - AS duas camadas juntas (*Top + Bottom*)

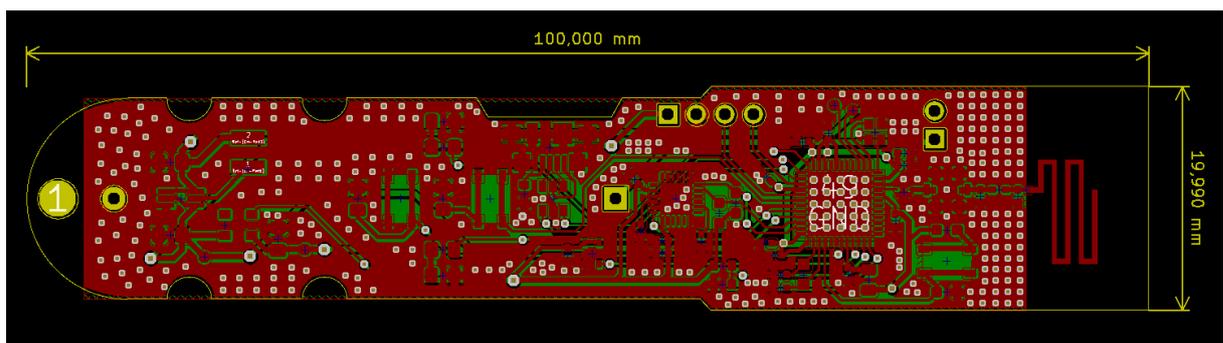
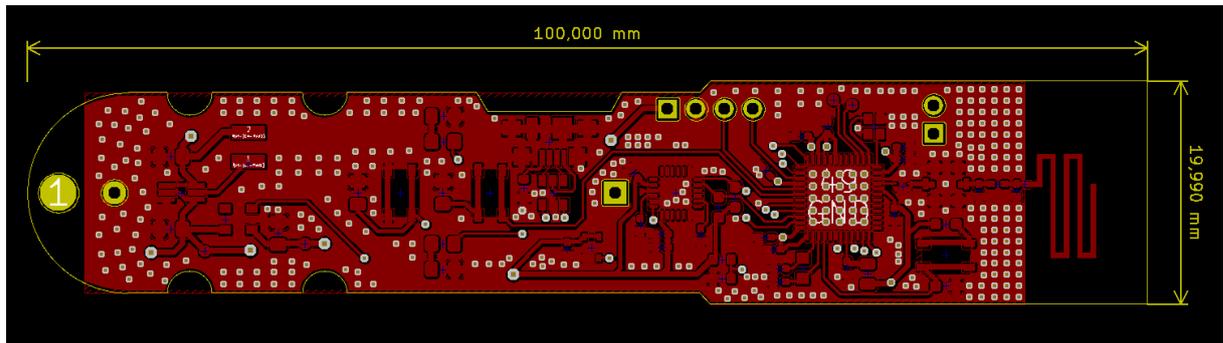
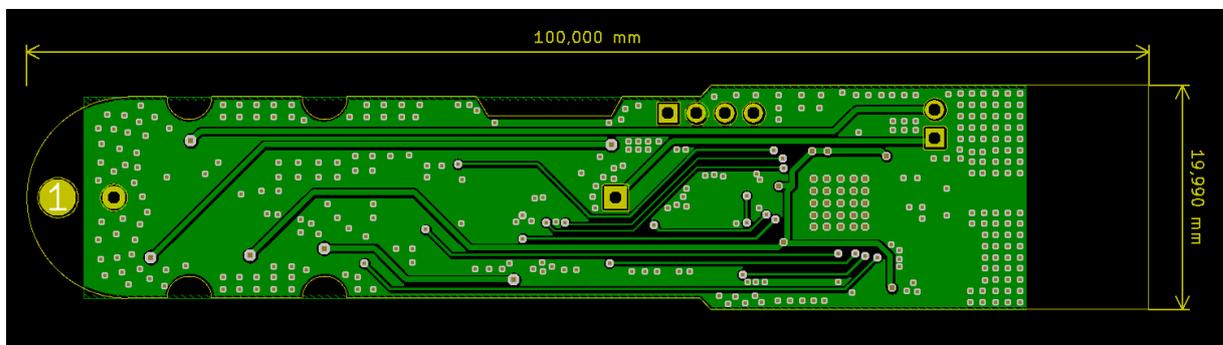


Figura 12 - Primeira camada (*Top*)Figura 13 - Segunda camada (*Bottom*)

3.2.2.1 Camadas da PCI

As trilhas e planos da placa de circuito impresso foram dispostas em duas camadas empilhadas. Essa escolha por apenas duas camadas foi devido ao menor custo para fabricação, e também pela STM32 ter um *Application note* (AN5165) de um layout de circuito de rádio frequência com o microcontrolador STM32WB para duas camadas, que serviu como base para projetar de forma correta levando em considerações critérios de projeto para minimizar interferências no circuito de Rádio Frequência.

A camada 1 (*Top*) contém as ilhas para soldagem dos componentes eletrônicos, as trilhas de sinais, as de alimentação de energia e planos GND. A camada 2 (*Bottom*) contém trilhas de sinais, as de alimentação de energia, planos de terra e o conector da bateria.

3.2.2.2 Planos e vias de Terra

No *layout* da PCI utilizou-se planos de terra em todas as duas camadas para garantir o melhor funcionamento do circuito e minimizar os problemas de Compatibilidade Eletromagnética (EMC). As áreas de terra da PCI foram maximizadas e unidas por furos de passagem entre planos (vias), reduzindo a indutância do sistema, o caminho de retorno das correntes, as emissões eletromagnéticas e interferências cruzadas entre trilhas.

3.2.2.3 Material dielétrico

A placa de circuito impresso é composta por um componente construtivo denominado de FR4-Standard Tg 140C, que garante um circuito impresso seguro, eficiente e de qualidade. O FR4 é um material composto por fibra de vidro e resina de epóxi, que permitem a placa se tornar altamente resistente. Além dos aspectos de uso construtivo, esse material é dielétrico – garante isolamento e controle de impedância entre as trilhas de alimentação e sinais de altas frequências. De acordo com a certificação do material FR4, conforme o utilizado pelo fabricante das placas de circuito impresso JLCPCB (*PCB Prototype & PCB Fabrication Manufacturer*), em testes com frequência de 1 MHz, a constante dielétrica do material equivale à $\epsilon_r = 4.58$.

3.2.2.4 Geometria das trilhas

No dimensionamento das larguras e espessuras das trilhas, é levado em considerações fatores como espessura do cobre, resistividade do material, corrente elétrica e temperatura de funcionamento. Baseado nessas variáveis para dimensionar a espessura de uma trilha, existe a Norma IPC-2221 que define as equações necessárias para o cálculo de dimensionamento. Dentre essas equações foi utilizado para o projeto ao mensurar a larguras das trilhas a equação abaixo.

$$I = K * dt^{0.44} * (W * H)^{0.725} \quad (2)$$

Onde:

I = máxima corrente em amperes.

dt = aumento da temperatura acima da temperatura ambiente em °C.

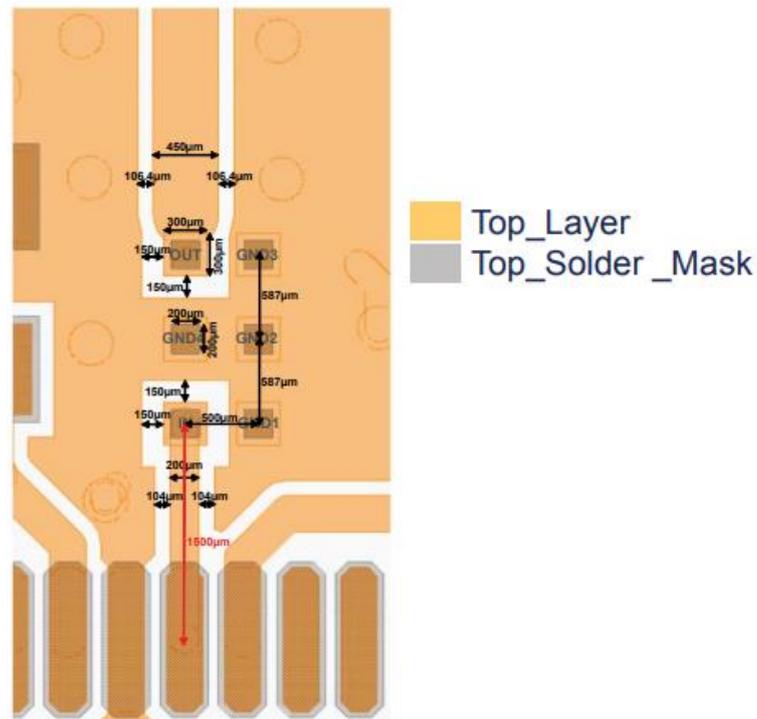
W, H = largura e espessura em mils.

K = 0.024 para trilhas internas ou 0.048 para trilhas externas.

Além do dimensionamento da trilha quanto a capacidade de corrente elétrica também houve precauções em função da geometria das trilhas referente ao controle da impedância intrínseca desejada para o correto funcionamento dos sinais de alta frequência do rádio. Levando em consideração a constante dielétrica da placa, a impedância de 50 desejada, restringiu a espessura das trilhas, bem como o espaçamento entre elas e os planos de GND. A espessura da trilha destinada a linha de transmissão de RF de 50 Ω , foi definida a partir do datasheet do componente MLPF-WB55-01E3, que recomenda utilizar as dimensões de trilhas

representadas na Figura 14 para uma PCI que contém duas camadas. Pois esse componente eletrônico apresenta correspondência de impedância (50Ω) integrada para o microcontrolador STM32WB55Cx utilizado.

Figura 14 - Recomendações de padrão de trilha de PCI para a linha de transmissão de RF, ao utilizar o componente MLPF-WB55-01E3.



Fonte: Datasheet do MLPF-WB55-01E3.

Por fim, no projeto da geometria das trilhas adotou-se ângulos mais suaves, evitando trilhas com ângulos de 90° , para evitar que ocorra reflexão no funcionamento.

3.2.2.5 Espessura da PCI

Dentre os parâmetros fornecidos pelo fabricante da confecção da placa de circuito impresso, temos a espessura como sendo um fator que deve ser considerado. O fabricante escolhido JLCPCB (*PCB Prototype & PCB Fabrication Manufacturer*), estabelece que a espessura deve ser definida conforme o número de camadas e espessura dos dielétricos entre camadas. Sendo assim, levando em consideração que o projeto de circuito impresso é composto por duas camadas, e que se deseja uma placa rígida, a espessura da PCI selecionada foi de 1,6 mm.

3.2.2.6 Antena feita na própria PCI

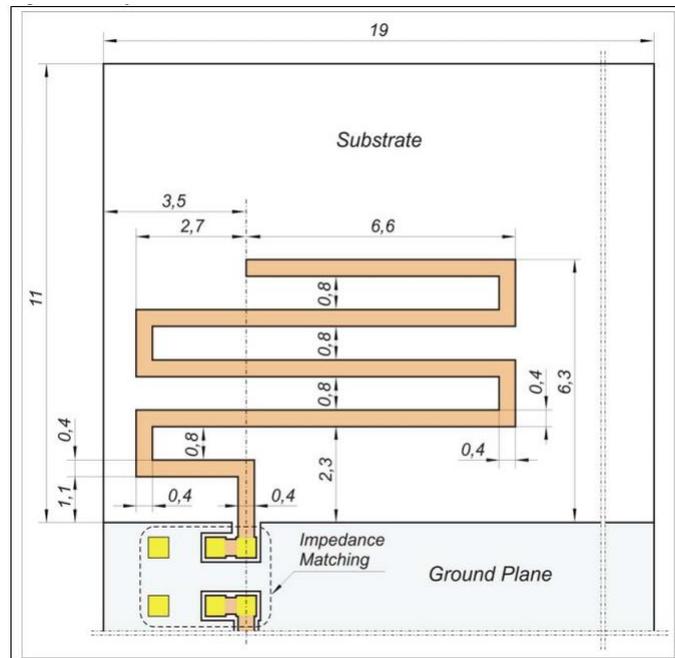
A antena presente no projeto de circuito impresso é uma antena de PCI, conforme Figura 15. Essa antena possui um ganho máximo de + 1,95dBi e os requisitos gerais de tamanho para ela são de 19×11 mm (não incluindo o plano de aterramento).

A escolha por usar essa antena de PCI é devido ao custo ser reduzido, também em função do seu desempenho ser semelhante à de uma antena cerâmica SMD, e por possuir uma área de cobertura maior que a antena cerâmica SMD. Além dessa antena, ainda ser dedicada aos microcontroladores da série STM32WB, conforme o *application note* AN5129.

A impedância da antena está ajustada para 50Ω na placa. No entanto, a impedância pode sofrer variações. Essas variações são provocadas por alguns fatores, como por exemplo: diferentes tamanhos e formas do plano de terra no campo próximo da antena, montagens mecânicas (caixa, baterias etc.), espessura e material a placa de circuito impresso, dentre outros.

As medidas de cada uma das arestas da antena estão na Figura 15.

Figura 15 - Dimensões da antena de PCI (em mm).



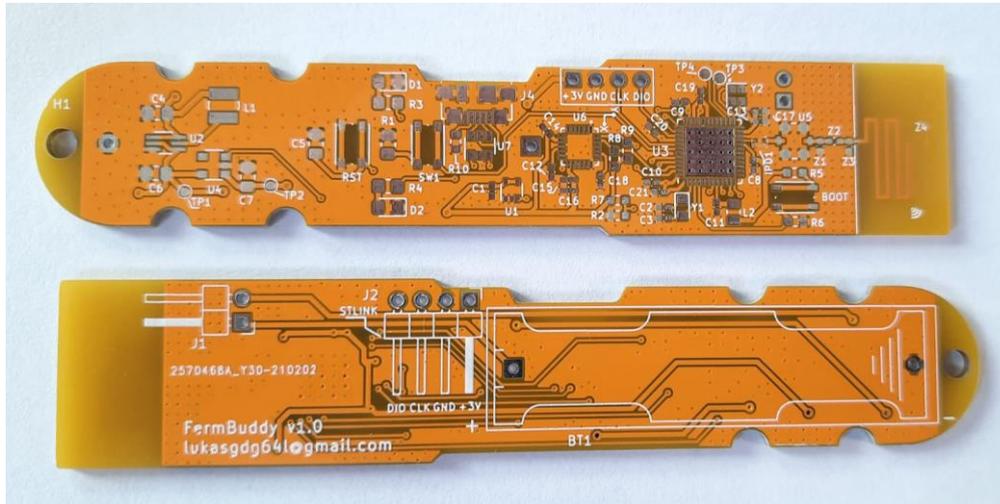
Fonte: *Application note* AN5129.

3.2.2.7 Fabricação da PCI

Na fabricação das placas de circuito impresso, referente ao densímetro eletrônico, escolheu-se o fabricante JLCPCB Factory. Esse fabricante é um dos mais antigos em fabricação

de protótipos e produção em pequenos lotes de PCI na China. Abaixo está representado na Figura 16, a PCI fabricada pela JLCPCB Factory.

Figura 16 - PCI fabricada na JLCPCB.



3.2.2.8 Soldagem dos componentes eletrônicos na PCI

A soldagem dos componentes eletrônicos na PCI, foi feito utilizando o método do uso do stencil. Nesse método é utilizado o stencil que é uma ferramenta utilizada para aplicar a pasta de solda em placas de circuito impresso, para depois ser realizada a montagem e soldagem dos componentes. O stencil é uma chapa metálica contendo o formato das pegadas dos componentes eletrônicos SMD, como apresentado na Figura 17.

O processo para fixação dos componentes na placa de circuito impresso, foi realizado em quatro etapas: adição da pasta de solda, colocação dos componentes eletrônicos na PCI, soldagem e verificação de qualidade. Na primeira etapa foi feita a aplicação da pasta de solda, com o auxílio da ferramenta stencil, que foi apoiada em cima da placa e em seguida com o uso de um rolo, foi a distribuição da pasta de solda em toda placa, formando uma espécie de almofada de pasta de solda nas pegadas dos componentes eletrônicos, como apresentada na Figura 18. Na próxima etapa foi feito o posicionamento dos componentes eletrônicos nas suas respectivas posições na PCI Figura 19, com o auxílio de uma pinça eletrostática. Na terceira etapa foi realizado a soldagem dos componentes eletrônicos na PCI, com o auxílio de um forno caseiro Figura 20, que possui um sistema de controle que permite gerar uma curva de soldagem (variação da temperatura interna no forno ao longo do tempo) com base nas características da pasta de solda utilizada. E por último foi feito a verificação de qualidade da solda pós processo de soldagem dos componentes eletrônicos, utilizando um microscópio para inspeção visual de

todos os pontos de solda e um multímetro para medição de continuidade de todos os pontos de ligação e verificação de possíveis curtos-circuitos que possam ter na PCI, após o processo de soldagem.

Figura 17 - Stencil utilizado no processo de soldagem dos componentes eletrônicos na PCI.

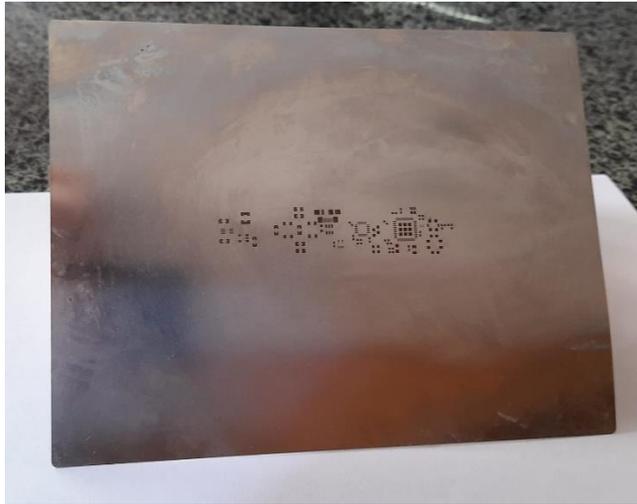


Figura 18 - Aplicação da pasta de solda nas pedadas dos componentes eletrônicos na PCI.

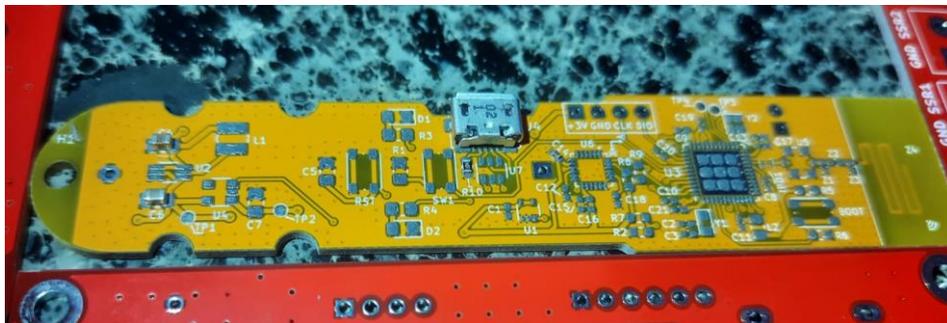


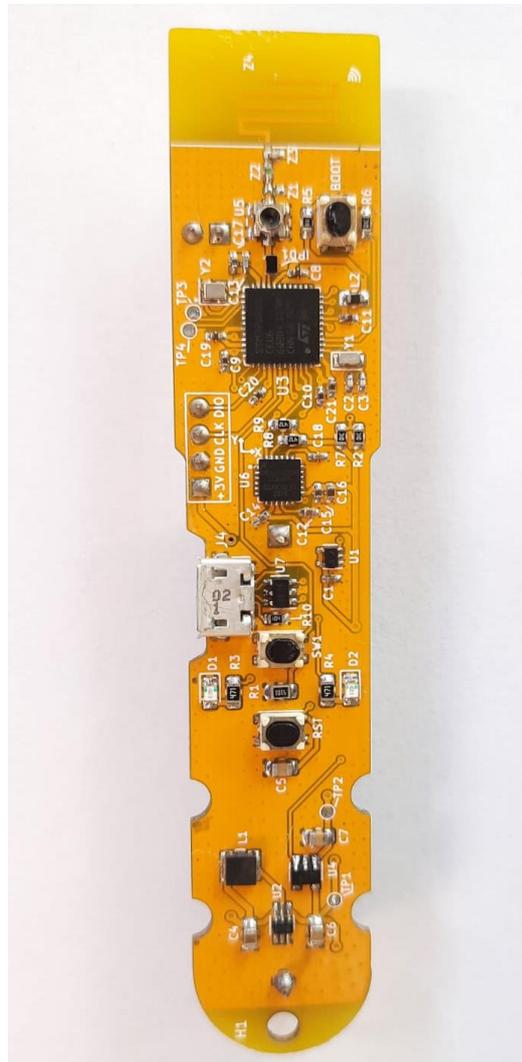
Figura 19 - Componentes eletrônicos posicionados na PCI.



Figura 20 - Forno de soldagem de componentes SMD utilizado.



Figura 21 - Componentes eletrônicos já soldados na PCI.



3.3 Estrutura mecânica

A estrutura é essencial para que sirva de base para o hardware dentro do recipiente plástico tubete Figura 22, possibilitando que o hardware se mantenha imóvel dentro do recipiente durante o processo de monitoramento da bebida alcóolica, e não tenha interferência no sensor acelerômetro, que possa ser ocasionado caso o hardware esteja solto dentro do recipiente. E essa estrutura conta com um suporte para pesos (ruelas metálicas), marcado em vermelho na Figura 24, para que seja possível alterar o centro de massa do dispositivo, que por consequência altera a inclinação do dispositivo ao ser imerso em água (o líquido água é a referência no processo de calibração que vai ser descrito no tópico 3.5). Para o desenvolvimento do modelo 3D da estrutura Figuras 23 e 24, foi utilizado o software de modelagem Fusion 360.

Figura 22 - Recipiente plástico (tubete).



Figura 23 - Estrutura 3D.

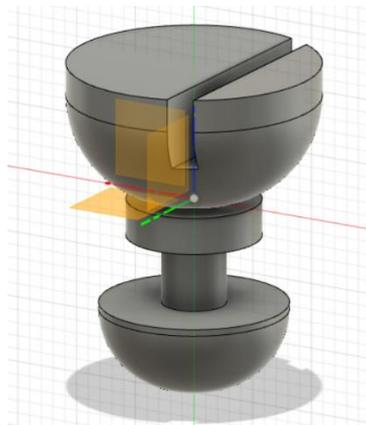
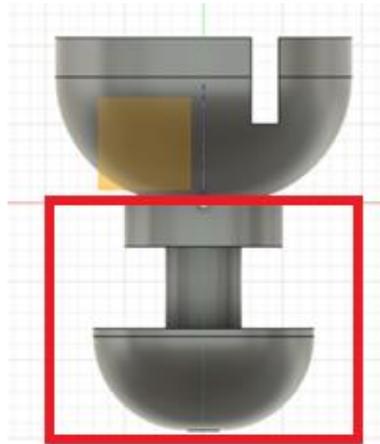


Figura 24 - Estrutura 3D (lateral).



3.4 Firmware

Para comunicação com a parte de baixo nível como o hardware foi necessário o desenvolvimento de um firmware. Que contém uma implementação para se comunicar com os sensores e outra para comunicação por protocolo BLE. Como ferramenta de ambiente de desenvolvimento C/C++ foi utilizado a plataforma STM32CubeIde, que possibilita a geração e compilação de código, e recursos de depuração para microcontroladores e microprocessadores STM32.

As implementações realizadas no firmware serão explicadas a seguir de forma detalhada, como a geração da inicialização dos periféricos, a aplicação que envolve a aquisição dos sensores e as configurações da comunicação do BLE.

3.4.1 Configuração de inicialização dos periféricos

No primeiro momento no desenvolvimento do firmware, é feito a configuração dos periféricos que serão utilizados. Em que é feito um projeto inicial na ferramenta STM32CubeIde como na Figura 25 e selecionado o microcontrolador que está sendo utilizado como na Figura 26, a partir disso será apresentado o encapsulamento do microcontrolador Figura 27 com todos os pinos disponíveis. Então foi feita a configuração dos periféricos com base nas conexões de hardware, como os cristais, RF, sensores, interrupções, RTC, depuração, GPIOs, essas configurações serão detalhadas a seguir.

Figura 25 - Criação do projeto de firmware.

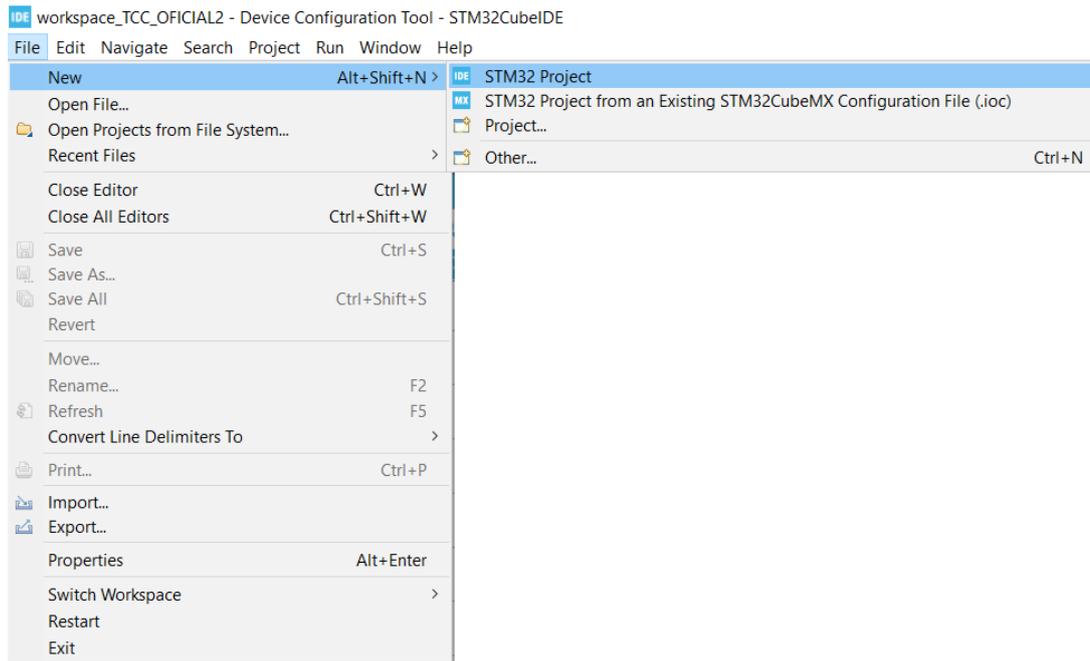


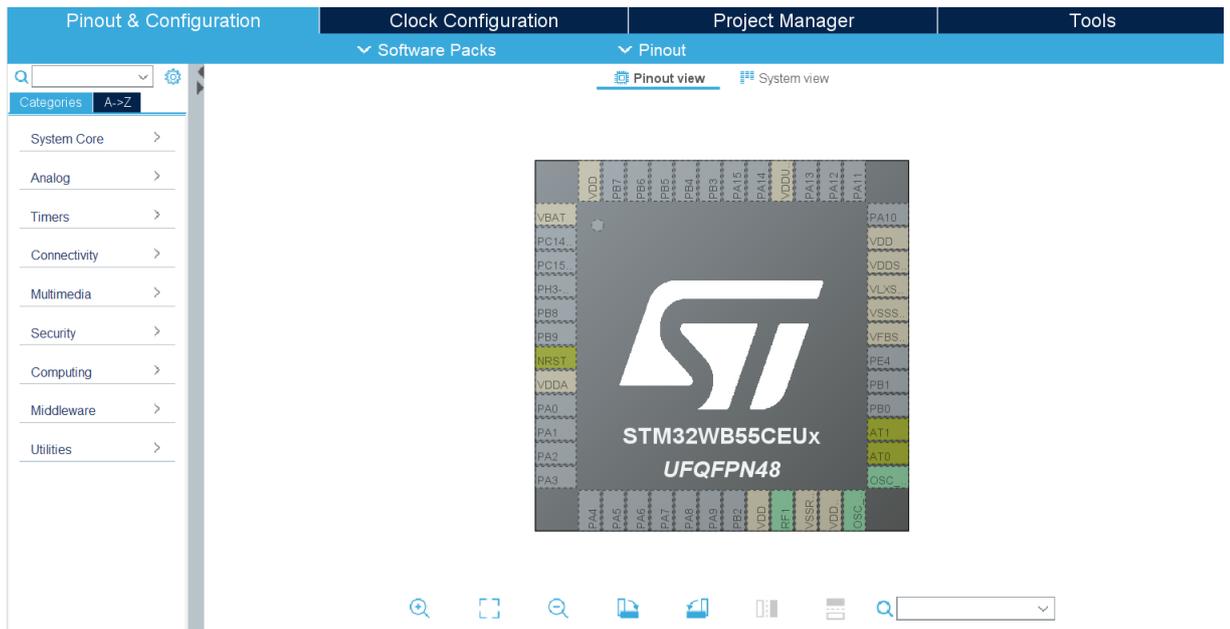
Figura 26 - Seleção do microcontrolador utilizado.

The screenshot displays the STM32Cube Configurator interface. The 'MCU/MPU Selector' tab is active, showing filters for the selected part number: STM32WB55CE. The interface includes tabs for Features, Block Diagram, Docs & Resources, Datasheet, and Buy. A large image shows the STM32U5 ultra-low-power MCU series with the STM32Cube ecosystem logo. Below the image, a table lists the selected MCU/MPU.

MCUs/MPUs List: 1 item

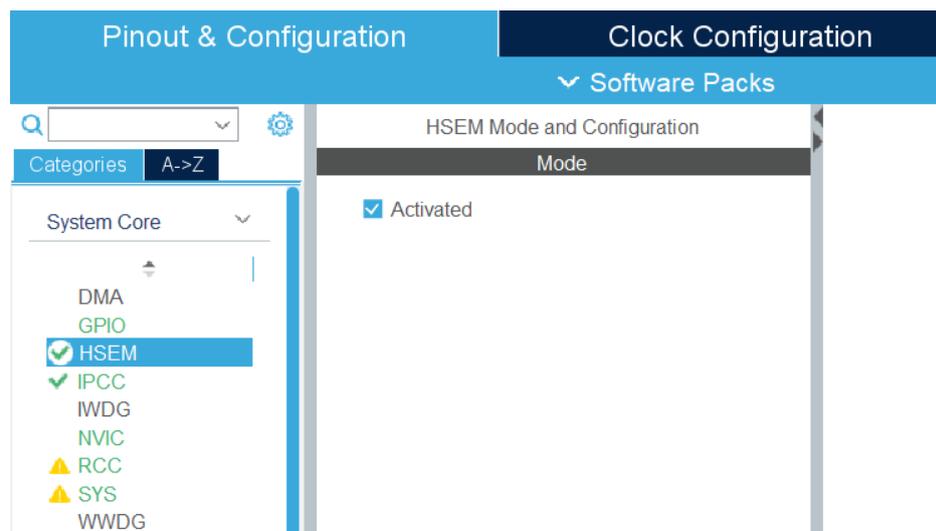
*	Part No.	Reference	Market...	Unit Pri...	Board	Package	Flash	RAM	IO	Freq.
☆	STM32...	STM32...	Active	2.43		UFQFP...	512 kByt...	256 kByt...	30	64 MHz

Figura 27 - Pinos disponíveis do microcontrolador.



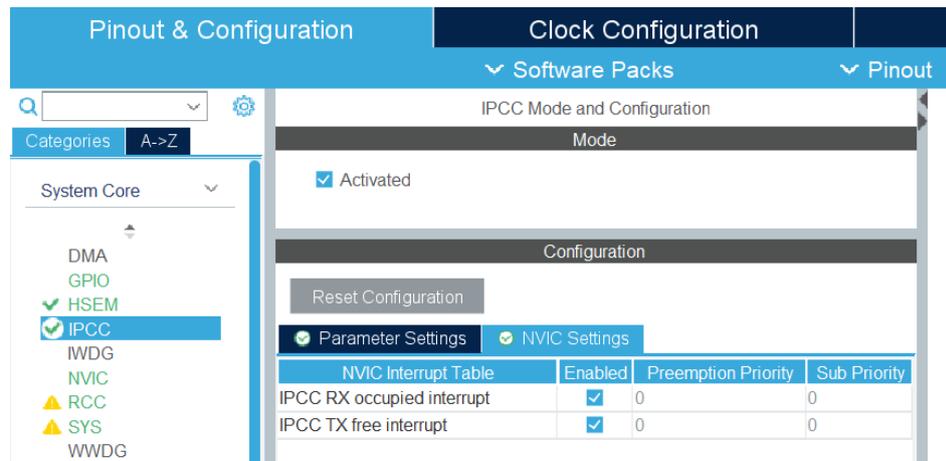
1 - A primeira configuração que deve ser feita é habilitar o HSEM. Que fica localizado em System Core -> HSEM e marcar o quadrado escrito *Activated*. O HSEM (*Hardware Semaphore*) é responsável por fazer a sincronização de processos e gerenciamento do acesso a periféricos compartilhados (como registros, RTC, etc...).

Figura 28 - Configuração do HSEM.



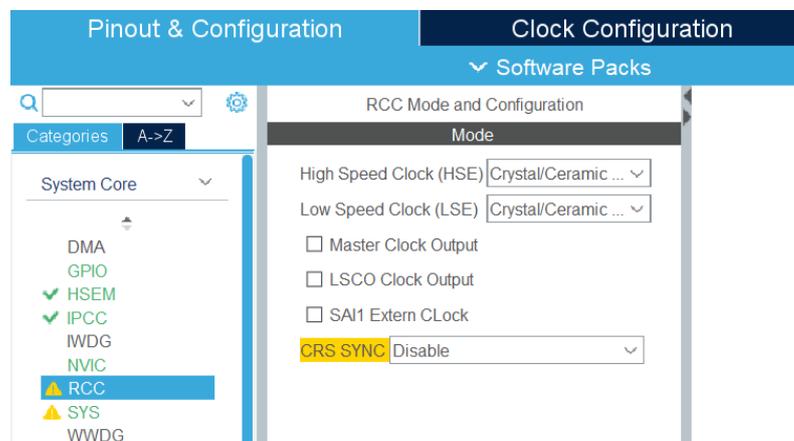
2 - A segunda configuração é habilitar o serviço de interrupção IPCC, que permite a comunicação entre os processadores Cortex-M4 com Cortex-M0+. Essa configuração é feita em System Core -> IPCC -> NVIC Settings e habilitar as duas interrupções.

Figura 29 - Configuração do IPCC.



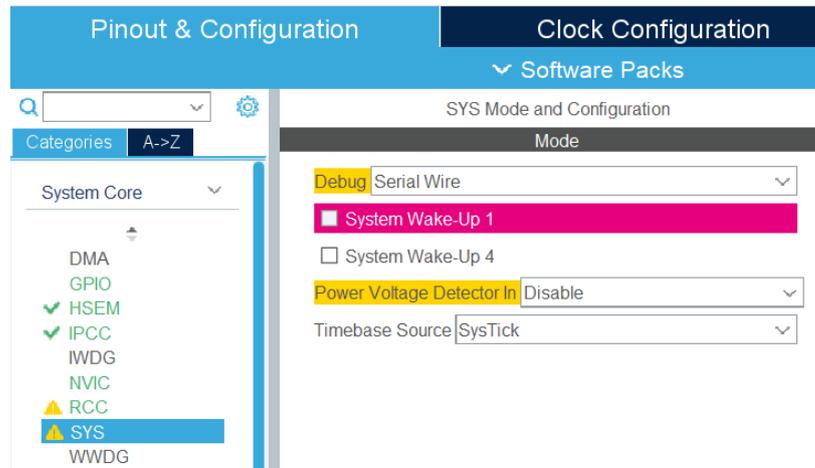
3 – Em seguida, é necessário determinar o RCC que vai ser responsável por estabelecer a base de tempo do BLE e do RTC. Para os relógios de alta e baixa velocidade é configurado como *Cristal/Ceramic Resonator*.

Figura 30 - Configuração do RCC.



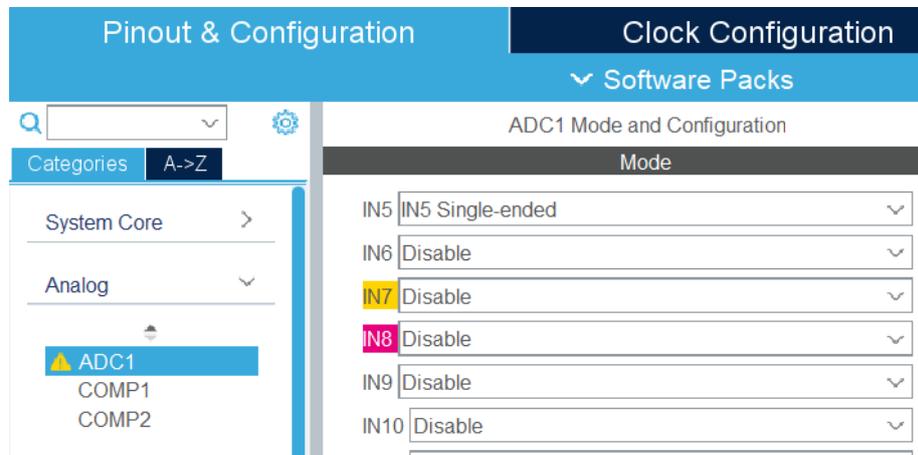
4 – Depois é preciso habilitar o modo de depuração em System Core -> SYS. Que foi configurado como Serial Wire. Essa configuração é feita para permitir que o usuário possa comunicar com o microcontrolador através de alguma ferramenta de depuração da ST em seu computador.

Figura 31 - Configuração do debug.



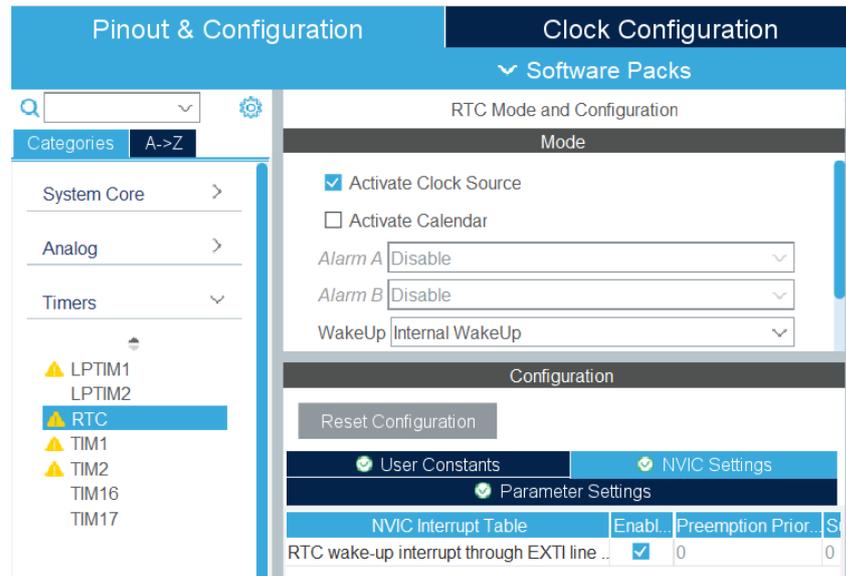
5 – O sensor de temperatura tem saída analógica, então é necessário configurar o ADC do microcontrolador no canal 5, que é onde está conectado fisicamente no hardware o pino de saída do sensor com o pino PA0 do microcontrolador.

Figura 32 - Configuração do ADC para o sensor de temperatura.



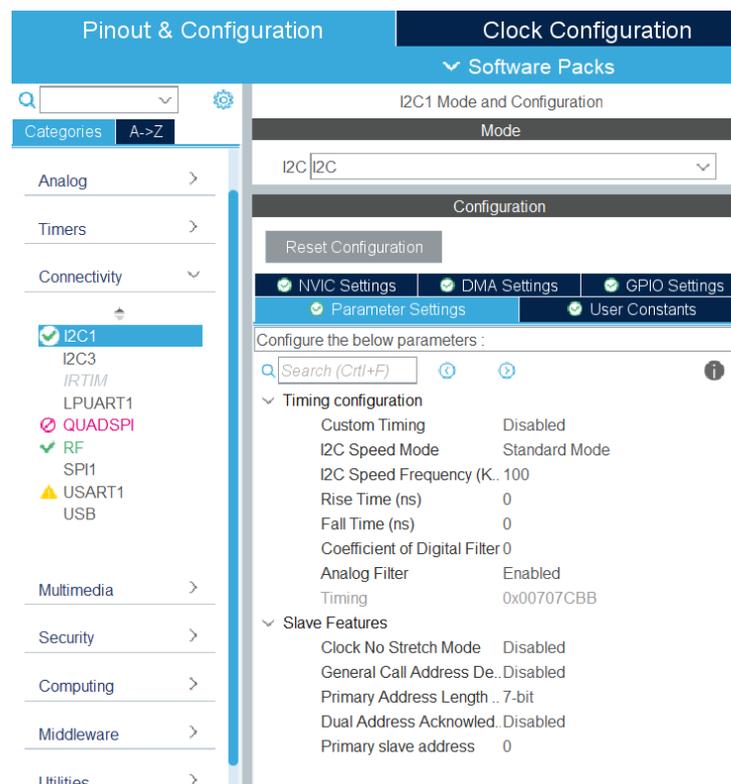
6 – Depois disso, é preciso ativar a fonte de relógio em *Timers* -> RTC e habilitar a interrupção do RTC em configurações de NVIC. O RTC é utilizado pelo Middleware para o gerenciamento do sistema de ativação do RF.

Figura 33 - Configuração do RTC.



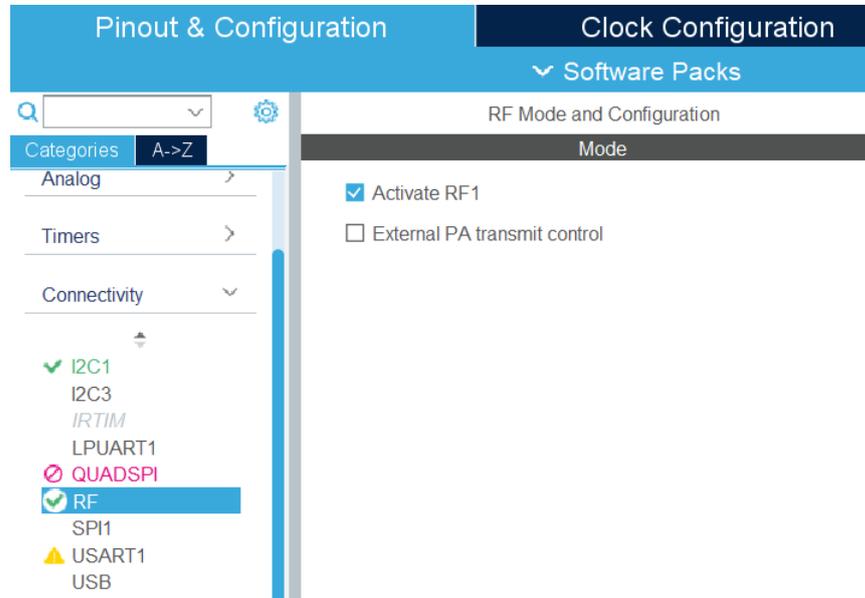
7 – O próximo passo é fazer a configuração dos periféricos responsáveis pela comunicação entre o microcontrolador e o sensor inercial ICM-20689. Como o sensor inercial utiliza o protocolo de comunicação I2C, então é necessário habilitar os pinos de I2C que são o PB7 (SDA) e PB6 (SCL) e configurar o modo dele como *Standard Mode*.

Figura 34 - Configuração da I2C, para o sensor inercial.



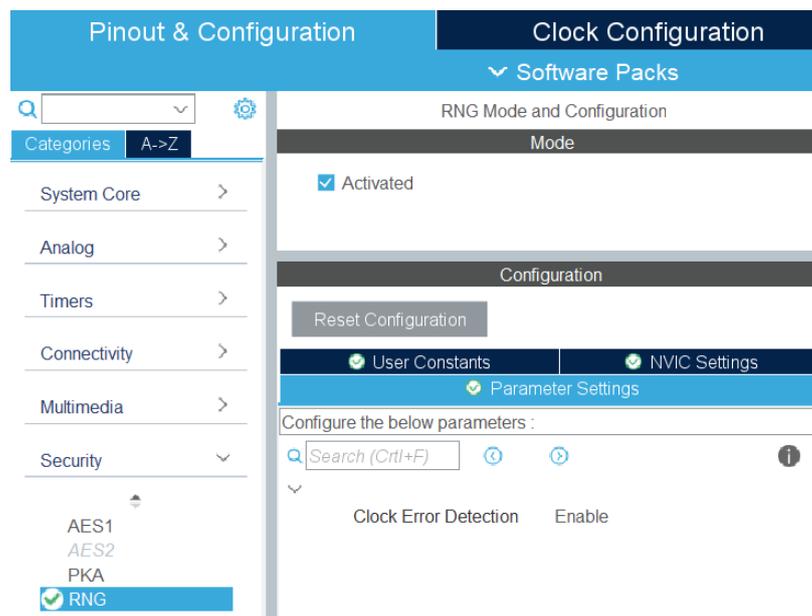
8 – Em seguida, é necessário ativar o RF1 em Connectivity -> RF, que é o pino responsável pela comunicação sem fio do rádio.

Figura 35 - Configuração do RF.



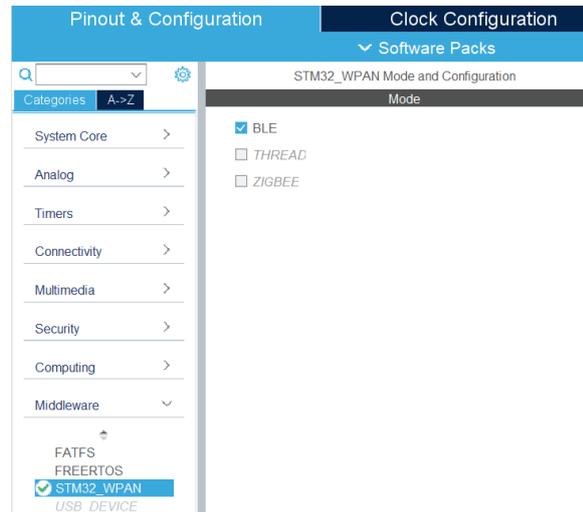
9 – Depois foi ativado o RNG em Security -> RNG. O RNG permite a criptografia MAC de camada inferior e camada superior. Um recurso de armazenamento de chaves do cliente que pode ser usado para manter as chaves escondidas.

Figura 36 - Configuração do RNG.



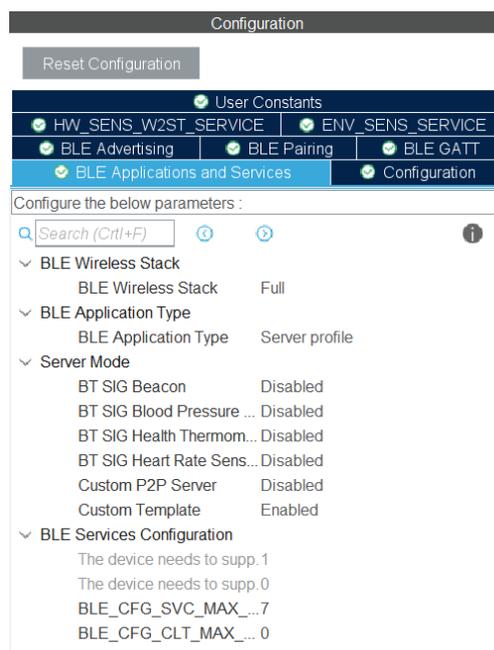
10 - Na configuração final, é preciso ativar a função BLE, que está localizado em Middleware-> STM32_WPAN -> BLE. Nessa configuração determina o modo do protocolo de comunicação a ser utilizado.

Figura 37 - Configuração do middleware.



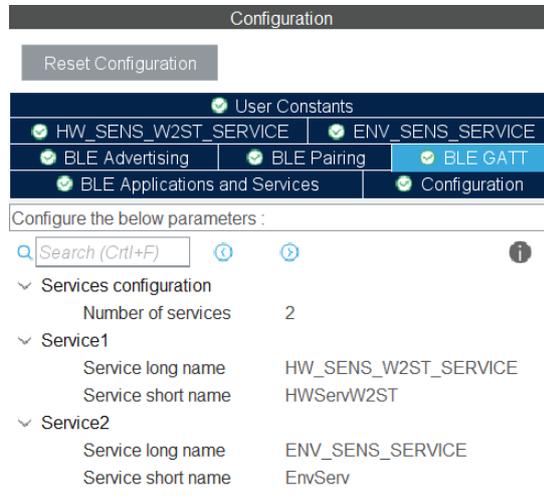
11 - Na guia de aplicações e serviços do BLE, é configurado o modo de servidor que será utilizado, que no caso foi habilitado o *Custom Template*, que é um servidor customizado para aplicação em específico, sendo possível determinar os serviços e suas características.

Figura 38 -Configuração das aplicações e serviços BLE.



12 - Na guia BLE GATT foi configurado o número de serviços e seus respectivos nomes.

Figura 39 - Configuração do GATT BLE.



13 - A seguir foi configurado o primeiro serviço, sendo o primeiro serviço relacionado aos dados de inclinação provenientes do sensor acelerômetro (densidade SG e inclinação). Foi configurado no serviço o número de características sendo 2, e o respectivo número do UUID do serviço.

Figura 40 - Configuração do serviço referente ao sensor inercial.



14 - Em seguida Figura 41 foi configurado a primeira característica, foi definido o número do UUID e o tamanho do valor da característica com base nos parâmetros do aplicativo Android da ST utilizado que é o app ST BLE Sensor. Foi usado como referência o manual de usuário UM2496 que descreve o protocolo BlueST, que é o protocolo utilizado no aplicativo ST BLE Sensor. Isso permite que tanto o dispositivo periférico (dispositivo medidor de densidade), quanto o dispositivo central (Celular + aplicativo Android) estejam falando sobre os mesmos atributos que são as características, que são identificadas através do UUID.

No manual do protocolo do aplicativo descreve que para característica referente ao evento de acelerômetro (usado para enviar os dados de densidade SG), O UUID padrão da característica é 0x00000400-0001-11e1-ac36-0002a5d5c51b e o formato do dado para o evento de acelerômetro deve ser igual da Tabela 1, contendo 2 bytes para o *Timestamp*, 1 byte para o *Event* e 2 bytes para os *Steps* (o valor referente a densidade), totalizando 5 bytes o tamanho do valor.

Tabela 1 - Formato de dados de evento do acelerômetro (comprimento de 5 bytes)

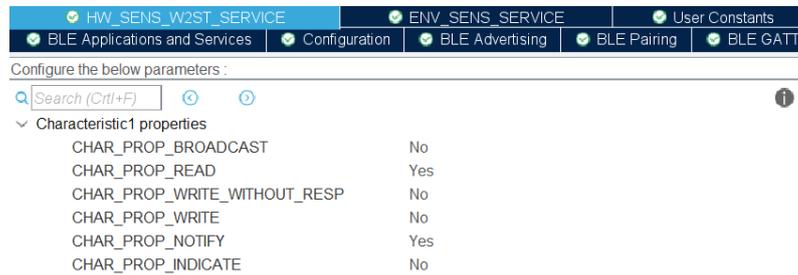
Bytes	Descrição
0	Timestamp
1	
2	Event
3	# Steps (Uint16)
4	

Figura 41 - Configuração geral da característica referente ao evento do acelerômetro (dados de SG).



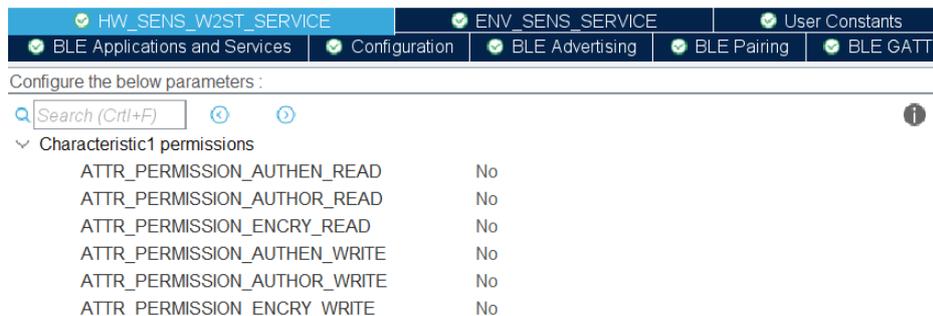
15 - Na configuração das propriedades da característica 1, deve ser habilitado para conter propriedades de leitura e notificação.

Figura 42 - Configuração das propriedades da característica referente ao evento do acelerômetro (dados de SG).



16 - Não foi configurado nenhuma permissão referente a característica 1.

Figura 43 - Configuração das permissões da característica referente ao evento do acelerômetro (dados de SG).



17 - Foi habilitado o evento em que o servidor GATT notifica a solicitação de leitura e aguarda a resposta da aplicação, em determinados pedidos de ATT.

Figura 44 - Configuração dos eventos de GATT da característica referente ao evento do acelerômetro (dados de SG).



18 - A seguir foi configurado a segunda característica, que foi utilizada para representar os dados da inclinação do dispositivo. Foi utilizado conforme o manual do protocolo do aplicativo a característica representada pelo nome *Direction of arrival*, o UUID padrão da característica é 0x10000000-0001-11e1-ac36-0002a5d5c51b e o formato do dado para a

direção de chegada é definida no manual como na Tabela 2, contendo 2 bytes para o *Timestamp*, 2 bytes para o valor do ângulo, totalizando 4 bytes o tamanho do valor.

Tabela 2 – Formato de dados de inclinação (comprimento de 4 bytes).

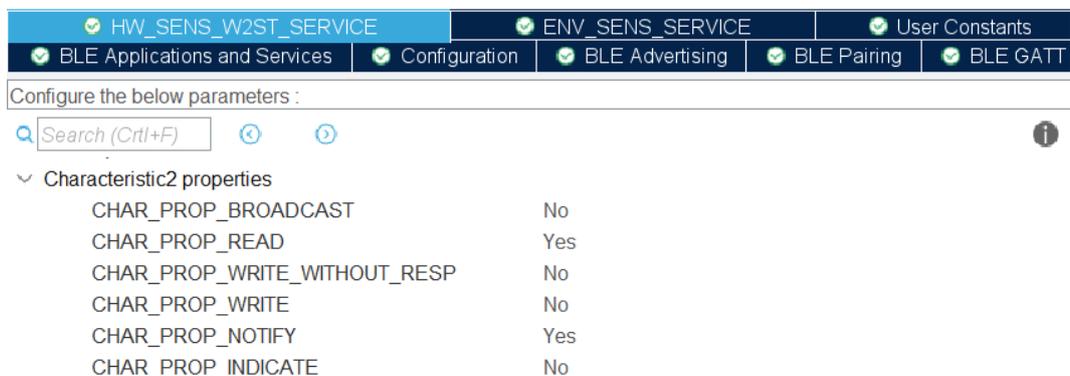
Bytes	Descrição
0	Timestamp
1	
2	Angle (int16)
3	

Figura 45 - Configuração geral da característica referente a inclinação.



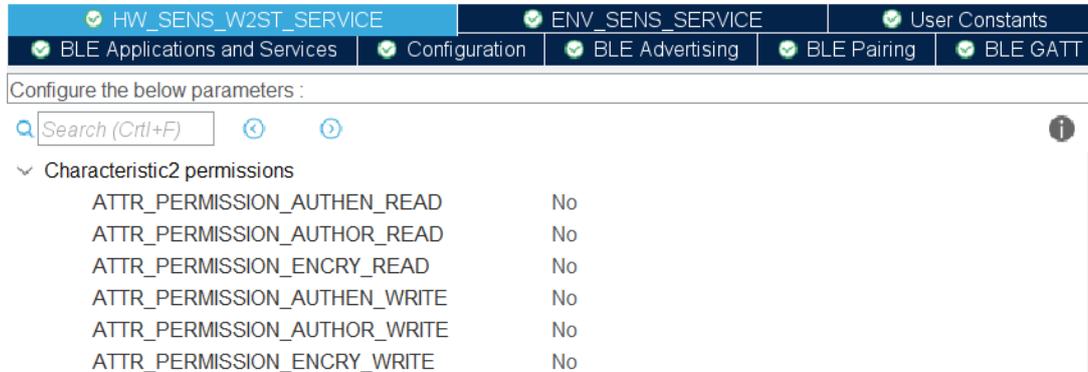
19 - Na configuração das propriedades da característica 2, deve ser habilitado para conter propriedades de leitura e notificação.

Figura 46 - Configuração das propriedades da característica referente a inclinação.



20 - Não foi configurado nenhuma permissão referente a característica 2.

Figura 47 - Configuração das permissões da característica referente a inclinação.



21 - Foi habilitado o evento em que o servidor GATT notifica a solicitação de leitura e aguarda a resposta da aplicação, em determinados pedidos de ATT.

Figura 48 - Configuração dos eventos de GATT da característica referente a inclinação.



22 - A seguir foi configurado o segundo serviço, sendo esse serviço relacionado aos dados do sensor de ambiente que monitora a temperatura. Foi configurado no serviço o número de características sendo 1, e o respectivo número do UUID do serviço.

Figura 49 - Configuração do serviço referente ao sensor de temperatura.



23 - A seguir foi configurado a característica do serviço de sensor de ambiente, que foi utilizada para representar os dados de temperatura do dispositivo. Foi utilizado conforme o manual do protocolo do aplicativo a característica representada pelo nome *Temperature*, o UUID padrão da característica é 0x00040000-0001-11e1-ac36-0002a5d5c51b e o formato do dado para a temperatura é definida no manual como na Tabela 3, contendo 2 bytes para o *Timestamp*, 2 bytes para o valor da temperatura, totalizando 4 bytes o tamanho do valor.

Tabela 3 – Formato de dados de temperatura (comprimento de 4 bytes).

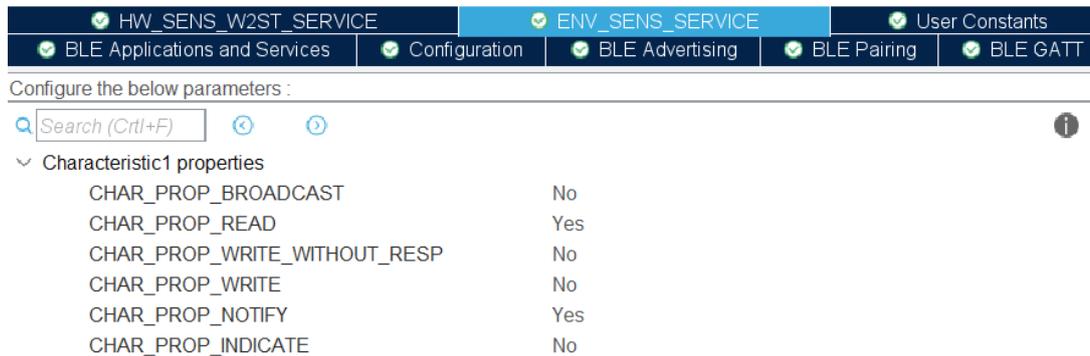
Bytes	Descrição
0	Timestamp
1	
2	Temperature*10
3	

Figura 50 - Configuração geral da característica referente a temperatura.



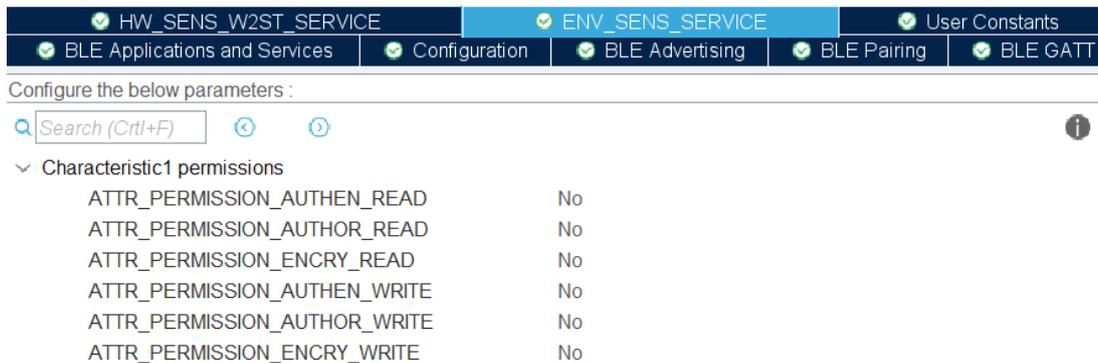
24 - Na configuração das propriedades da característica 1 do serviço de sensor de ambiente, deve ser habilitado para conter propriedades de leitura e notificação.

Figura 51 - Configuração das propriedades da característica referente a temperatura.



25 - Não foi configurado nenhuma permissão referente a característica 1.

Figura 52 - Configuração das permissões da característica referente a temperatura.



26 - Foi habilitado o evento em que o servidor GATT notifica a solicitação de leitura e aguarda a resposta da aplicação, em determinados pedidos de ATT.

Figura 53 - Configuração dos eventos de GATT da característica referente a temperatura.



27 - Abaixo estão as configurações gerais do middleware, referente aos temporizadores do servidor, depuração, parâmetros genéricos e parâmetros de aplicação.

Figura 54 - Configurações gerais do middleware.

Configuration

Reset Configuration

BLE GATT HW_SENS_W2ST_SERVICE ENV_SENS_SERVICE User Constants

BLE Applications and Services Configuration BLE Advertising BLE Pairing

Configure the below parameters :

Search (Ctrl+F)

HW Timer Server

CFG_HW_TS_MAX_NBR_CONCURRENT_TIMER	6
CFG_HW_TS_NVIC_RTC_WAKEUP_IT_PREEMPTPRIO	3
CFG_HW_TS_NVIC_RTC_WAKEUP_IT_SUBPRIO	0
CFG_HW_TS_USE_PRIMASK_AS_CRITICAL_SECTION	1
CFG_HW_TS_RTC_HANDLER_MAX_DELAY	(10 * (LSI_VALUE/1000))
CFG_HW_TS_RTC_WAKEUP_HANDLER_ID	RTC_WKUP_IRQn

HW UART

CFG_HW_LPUART1_ENABLED	Disabled
CFG_HW_LPUART1_DMA_TX_SUPPORTED	Disabled
CFG_HW_USART1_ENABLED	Disabled
CFG_HW_USART1_DMA_TX_SUPPORTED	Disabled

Generic parameters

CFG_HW_RESET_BY_FW	Enabled
CFG_USE_SMPs	Disabled
CFG_LPM_SUPPORTED	Enabled
CFG_DEBUGGER_SUPPORTED	Enabled
CFG_DEBUG_BLE_TRACE	Disabled
CFG_DEBUG_APP_TRACE	Disabled
CFG_DEBUG_TRACE_LIGHT	Disabled
CFG_DEBUG_TRACE_FULL	Disabled
DBG_TRACE_USE_CIRCULAR_QUEUE	Enabled
DBG_TRACE_MSG_QUEUE_SIZE	4096
MAX_DBG_TRACE_MSG_SIZE	1024

Application parameters

CFG_TX_POWER	0dBm (0x19)
CFG_DEBUG_TRACE_UART	You need to activate either CFG_HW_UART1 or CFG_HW_LPUART1(wh..
CFG_CONSOLE_MENU	You need to activate either CFG_HW_UART1 or CFG_HW_LPUART1(wh..
CFG_ADV_BD_ADDRESS	0x7257acd87a6c
CFG_FAST_CONN_ADV_INTERVAL_MIN	80
CFG_FAST_CONN_ADV_INTERVAL_MAX	100
CFG_LP_CONN_ADV_INTERVAL_MIN	1000
CFG_LP_CONN_ADV_INTERVAL_MAX	2500
CFG_IO_CAPABILITY	Display Yes No (0x01)
CFG_MITM_PROTECTION	MITM protection required (0x01)
CFG_RTCCLK_DIVIDER_CONF	0
CFG_RTCCLK_DIV	16
CFG_RTC_WUCKSEL_DIVIDER	0
CFG_RTC_ASYNC_PRESCALER	0x0F
CFG_RTC_SYNC_PRESCALER	0x7FFF
CFG_BLE_NUM_LINK	2
CFG_BLE_NUM_GATT_SERVICES	8
CFG_BLE_NUM_GATT_ATTRIBUTES	68
CFG_BLE_MAX_ATT_MTU	156
CFG_BLE_ATT_VALUE_ARRAY_SIZE	1344
CFG_BLE_DATA_LENGTH_EXTENSION	Enabled
CFG_BLE_SLAVE_SCA	500
CFG_BLE_MASTER_SCA	0
CFG_BLE_LSE_SOURCE	Disabled
CFG_BLE_HSE_STARTUP_TIME	0x148
CFG_BLE_MAX_CONN_EVENT_LENGTH	0xFFFFFFFF
CFG_BLE_VITERBI_MODE	Enabled
CFG_BLE_OPTIONS	BLE stack Options flags:
- CFG_BLE_OPTIONS_LL	SHCI_C2_BLE_INIT_OPTIONS_LL_HOST
- CFG BLE OPTIONS SVC	SHCI C2 BLE INIT OPTIONS WITH SVC CHANGE DESC
- CFG_BLE_OPTIONS_DEVICE_NAME	SHCI_C2_BLE_INIT_OPTIONS_DEVICE_NAME_RW
- CFG_BLE_OPTIONS_POWER_CLASS	SHCI_C2_BLE_INIT_OPTIONS_POWER_CLASS_2_3
CFG_BLE_MAX_COC_INITIATOR_NBR	32
CFG_BLE_MIN_TX_POWER	0
CFG_BLE_MAX_TX_POWER	0
CFG_TLBLE_EVT_QUEUE_LENGTH	5
CFG_TLBLE_MOST_EVENT_PAYLOAD_SIZE	255

Debug options

BLE_DBG_APP_EN	Disabled
----------------	----------

29 - Na guia de emparelhamento do BLE, deve ser habilitado os parâmetros de emparelhamento, essa configuração é importante para estabelecer automaticamente uma conexão entre dispositivos, de forma segura, exigindo PINs e códigos de segurança. Os parâmetros de emparelhamento, foram os configurados abaixo.

Figura 55 - Configurações de emparelhamento do BLE.

HW_SENS_W2ST_SERVICE	ENV_SENS_SERVICE	User Constants
BLE Applications and Services	Configuration	BLE Advertising
		BLE Pairing
		BLE GATT

Configure the below parameters :

Search (Ctrl+F) [] [] []

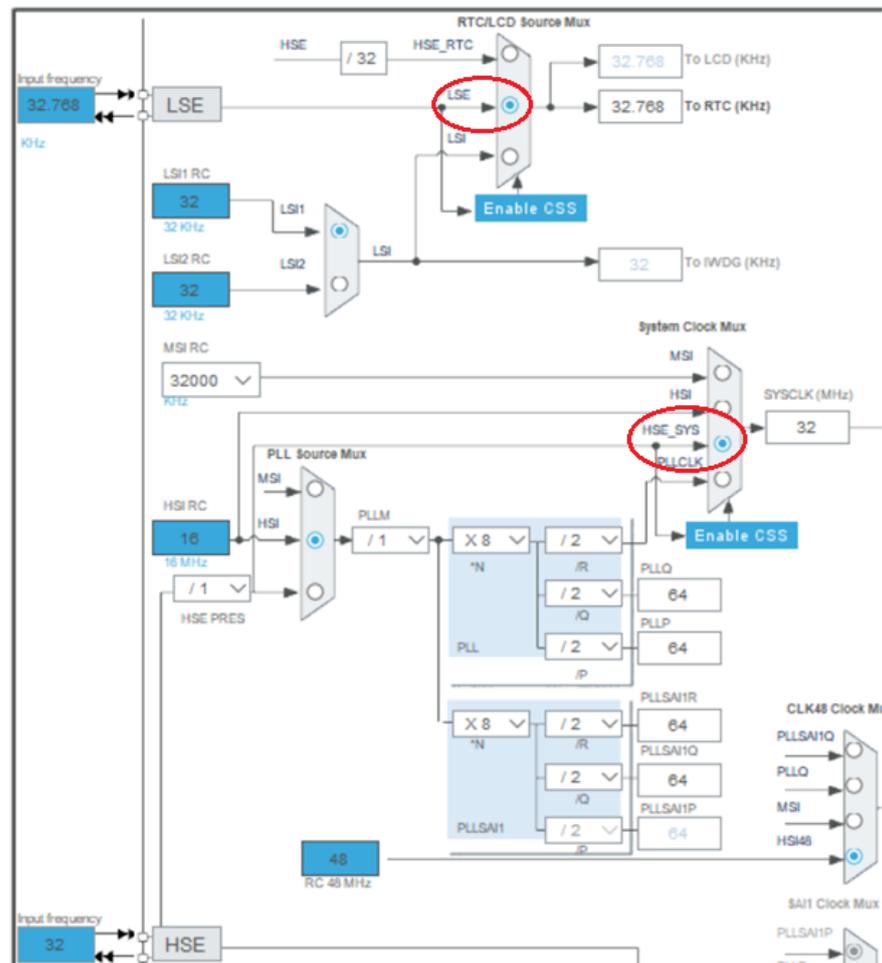
Pairing parameters

PAIRING_PARAMETERS	ON
CFG_BONDING_MODE	No-bonding mode(0x00)
CFG_USED_FIXED_PIN	Use a fixed pin (0x00)
CFG_FIXED_PIN	111111
CFG_ENCRYPTION_KEY_SIZE_MAX	16
CFG_ENCRYPTION_KEY_SIZE_MIN	8
CFG_SC_SUPPORT	Secure Connections Paring supported but optional (0x01)
CFG_BLE_IRK	12, 34, 56, 78, 9A, BC, DE, F0, 12, 34, 56, 78, 9A, BC, DE, F0
CFG_BLE_ERK	FE, DC, BA, 09, 87, 65, 43, 21, FE, DC, BA, 09, 87, 65, 43, 21
CFG_KEYPRESS_NOTIFICATION_SUPPORT	Keypress notification not supported (0x00)

30 - Finalizando as configurações dos periféricos, foi definido os pinos PB5, PA10 e PA3 manualmente como GPIO_Output, da forma que foi apresentado abaixo. Essa configuração é destinada para o acionamento dos LEDs pelos pinos PB5 e PA10, e para o controle da chave de carga pelo pino PA3, que gerencia a alimentação dos sensores, se estiver habilitado (nível logico alto) a alimentação para os sensores, se não tive habilitado (nível logico baixo) não há alimentação para os sensores.

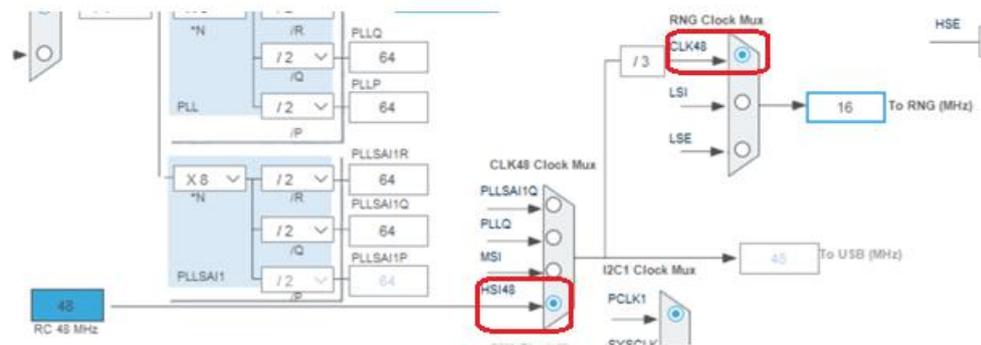
32 - Acima ao aceitar para que resolva de forma automática os erros de relógio, ainda não foi suficiente para que todas as configurações necessárias de relógio estejam feitas, então são necessárias mais algumas mudanças. Dentre essas mudanças é preciso no canto superior esquerdo, habilitar o cristal externo de baixa velocidade (32.768 KHz). Habilitando no *RTC/LCD Source Mux* a opção LSE. Em seguida é preciso habilitar o uso do cristal externo de alta velocidade (32 MHz) em *System Clock Mux*, habilitando a opção HSE_SYS.

Figura 58 - Configuração do LSE e HSE.



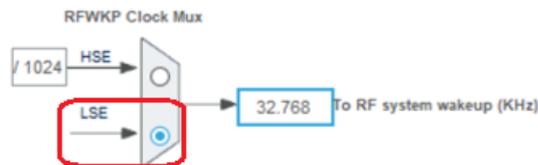
33 - Abaixo foi configurado o relógio ferente ao RNG, então foi necessário configurar para HSI48 no CLK48 Clock Mux, e em RNG Clock Mux configurado a opção CLK48.

Figura 59 - Configuração do relógio do RNG.



34 - Por fim, foi configurado o sistema de *wakeup* do RF, modificando em RFWKP Clock Mux para a opção LSE.

Figura 60 - Configuração do sistema de wakeup do RF.



35 - A última etapa é verificar as chamadas de função que vão ser geradas, sendo necessário que as configurações dos periféricos sejam habilitadas antes do middleware, isso está localizado na guia *Project Manager* -> *Advanced Settings*.

Figura 61 - Configurações avançadas das funções geradas.

Home > STM32WB56CEUX > FW_BEER_DENS_IOT.ioc - Project Manager > GENERATE CODE

Pinout & Configuration | Clock Configuration | **Project Manager** | Tools

Driver Selector: Search (Ctrl+F)

Project	Driver	HAL
	RCC	HAL
	GPIO	HAL
	ADC	HAL
	I2C	HAL
	IPCC	HAL
	RF	HAL
	RNG	HAL
	RTC	HAL
	STM32_WPAN	HAL
	HSEM	HAL

Generated Function Calls

Generate Code	Rank	Function Name	Peripheral Instance Name	Do Not Generate Function Call	Visibility (Static)
<input checked="" type="checkbox"/>	1	SystemClock_Config	RCC	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	2	MX_GPIO_Init	GPIO	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	3	MX_ADC1_Init	ADC1	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	4	MX_I2C1_Init	I2C1	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	5	MX_IPCC_Init	IPCC	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	6	MX_RF_Init	RF	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	7	MX_RNG_Init	RNG	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	8	MX_RTC_Init	RTC	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	9	APPE_Init	STM32_WPAN	<input type="checkbox"/>	<input type="checkbox"/>

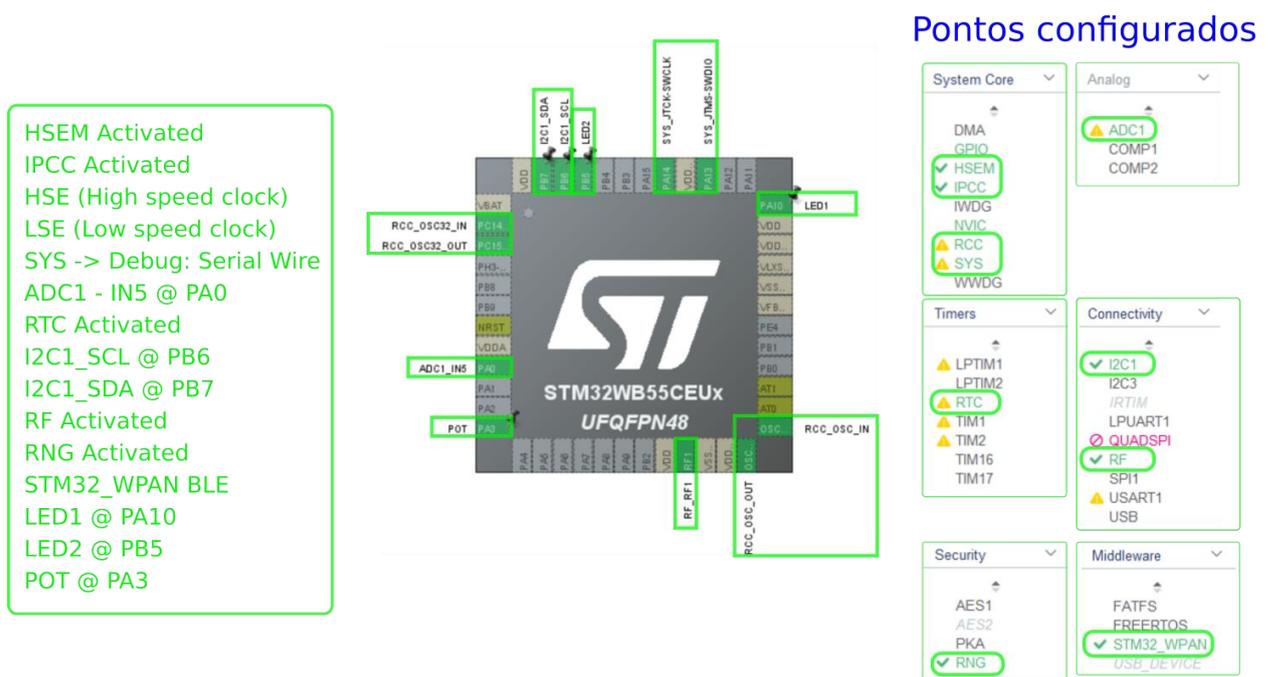
Register CallBack: Search

Register	Callback
ADC	DISABLE
COMP	DISABLE
CRYP	DISABLE
I2C	DISABLE
IRDA	DISABLE
LPTIM	DISABLE
PCD	DISABLE
PKA	DISABLE
QSPI	DISABLE
RNG	DISABLE
RTC	DISABLE
SAI	DISABLE
SMARTCARD	DISABLE
SMBUS	DISABLE
SPI	DISABLE
TIM	DISABLE
TSC	DISABLE
UART	DISABLE
USART	DISABLE
WWDG	DISABLE

Advanced Settings

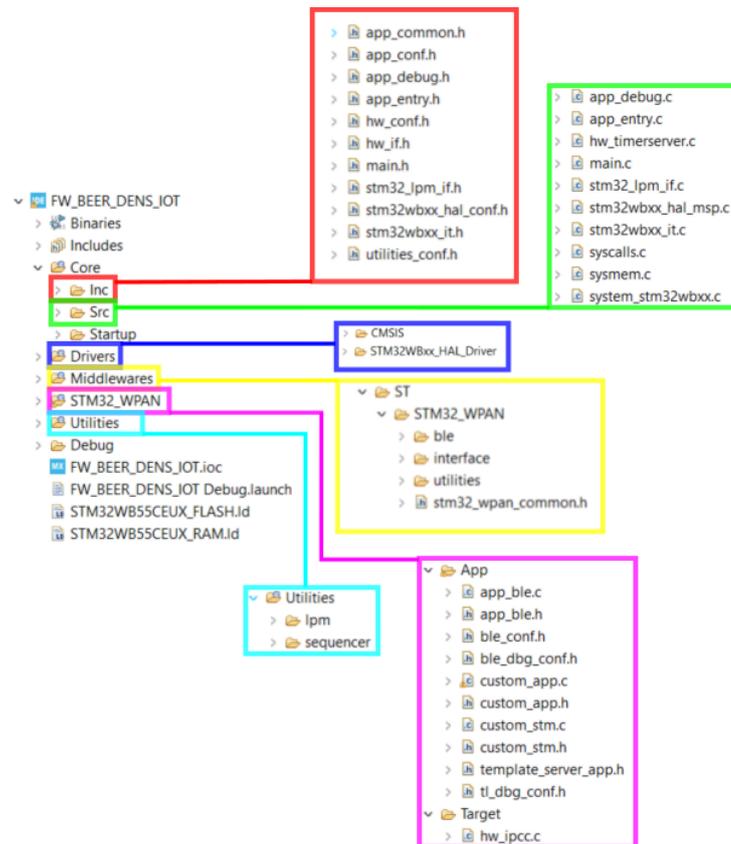
36 - Após verificar se o rank das chamadas de função que vão ser geradas na guia de gerenciamento de projeto estão na ordem correta, e todas as configurações abaixo já concluídas, então é salvo o projeto e é feito a geração do código de inicialização de todos os periféricos e do middleware. A configuração final ficou desta forma:

Figura 62 - Pontos de configuração.



37 - A partir do momento que foi feito a geração do código, então é gerada uma árvore de projeto no STM32CubeIde como essa abaixo:

Figura 63 - Árvore de projeto gerada no STM32CubeIde.



3.4.2 Implementação de firmware para os sensores

Foi implementado no firmware, drivers para comunicação entre o microcontrolador e os sensores, para aquisição de dados de acelerômetro e temperatura. Para o sensor inercial ICM_20689 foi utilizado o protocolo de comunicação serial I2C, para interface com o microcontrolador, para obtenção de dados de acelerômetro e temperatura.

Também foi implementando um driver para o sensor que é dedicado apenas para temperatura LMT86DCK, sendo utilizado o ADC do microcontrolador para o condicionamento do sinal analógico proveniente da saída sensor de temperatura.

3.4.2.1 Firmware sensor inercial

Na implementação do firmware do sensor inercial, foram desenvolvidas duas funções, uma para leitura do acelerômetro nos três eixos Figura 64 e outra para converter os dados do acelerômetro em inclinação Figura 65. Permitindo obter a inclinação do dispositivo desenvolvido em relação ao eixo y.

Figura 64 - Implementação de firmware para leitura de acelerômetro.

```

127 void ICM_20689_Read_Accel(I2C_HandleTypeDef *I2Cx, ICM_20689_t *DataStruct)
128 {
129     uint8_t Rec_Data[6];
130
131     HAL_I2C_Mem_Read(I2Cx, ICM_20689_ADDR, ACCEL_XOUT_H_REG, 1, Rec_Data, 6, i2c_timeout);
132
133     DataStruct->Accel_X_RAW = (int16_t) ((Rec_Data[0] << 8) | Rec_Data[1]);
134     DataStruct->Accel_Y_RAW = (int16_t) ((Rec_Data[2] << 8) | Rec_Data[3]);
135     DataStruct->Accel_Z_RAW = (int16_t) ((Rec_Data[4] << 8) | Rec_Data[5]);
136
137
138     DataStruct->Ax = DataStruct->Accel_X_RAW / 2048.0;
139     DataStruct->Ay = DataStruct->Accel_Y_RAW / 2048.0;
140     DataStruct->Az = DataStruct->Accel_Z_RAW / 2048.0;
141 }

```

Figura 65 - Implementação de firmware para leitura de inclinação.

```

153 void ICM_20689_Tilt(I2C_HandleTypeDef *I2Cx, ICM_20689_t *DataStruct)
154 {
155     ICM_20689_Read_Accel(I2Cx, DataStruct);
156
157     double gXYZ = sqrt(pow(DataStruct->Ax,2)+pow(DataStruct->Ay,2)+pow(DataStruct->Az,2));
158
159     // axis used for pitch is whichever is linear to hydrometer length
160     // generally this will be the Y axis
161     // get tilt angle (pitch) wrt to horizontal and convert to degrees
162     double ratio = (DataStruct->Ay) / gXYZ;
163
164     DataStruct->tiltAngle = (float)((ratio < 0.5) ? 90-fabs(asin(ratio)*RAD_TO_DEG) : fabs(acos(ratio)*RAD_TO_DEG));
165
166 }

```

3.4.2.2 Firmware sensor de temperatura do sensor ICM-20689

O sensor ICM-20689 além de conter um acelerômetro de 3 eixos e um giroscópio de 3 eixos, possui também um sensor de temperatura embutido. Então foi implementando um firmware capaz de fazer a leitura de temperatura e a converter na unidade de medida Grau Celsius, como apresentando na Figura 66.

Figura 66 - Implementação de firmware para leitura de temperatura.

```

197 void ICM_20689_Read_Temp(I2C_HandleTypeDef *I2Cx, ICM_20689_t *DataStruct)
198 {
199
200     uint8_t Rec_Data[2];
201     int16_t temp;
202
203     HAL_I2C_Mem_Read(I2Cx, ICM_20689_ADDR, TEMP_OUT_H_REG, 1, Rec_Data, 2, i2c_timeout);
204
205     temp = (int16_t) (Rec_Data[0] << 8 | Rec_Data[1]);
206     //TEMP_degC = ((TEMP_OUT - RoomTemp_Offset)/Temp_Sensitivity) + 25degC
207     DataStruct->Temperature = (float) (((int16_t) temp - 1648) / (float) 326.8 + (float) 25.0);
208 }

```

3.4.2.3 Firmware sensor de temperatura do sensor LMT86DCK

O sensor LMT86DCK possui uma saída analógica que muda a amplitude do nível de tensão conforme a variação de temperatura, então foi implementando um firmware para leitura da saída desse sensor utilizando o ADC do microcontrolador. Nesse firmware foi desenvolvido duas funções, Figuras 67 e 68, uma de aplicação responsável por executar as funções de hardware que são responsáveis por iniciar a aquisição do ADC, obter o valor lido no ADC e atribuir dentro de uma variável e logo em seguida parar o ADC.

Figura 67 - Leitura do ADC referente ao sensor de temperatura LMT86.

```

59 uint16_t app_read_temperature_adc_lmt86()
60 {
61     uint32_t adc1_val_ch5;
62     hw_adc_start();
63     hw_adc_read_channel(&adc1_val_ch5);
64     hw_adc_stop();
65
66     return adc1_val_ch5;
67 }

```

Figura 68 - Leitura do ADC referente ao sensor de temperatura LMT86.

```

53 bool hw_adc_read_channel(uint32_t *adc_val_ch5)
54 {
55     HAL_StatusTypeDef adc_hal_status;
56
57     while(1)
58     {
59         adc_hal_status = HAL_ADC_PollForConversion(&hadc1, 100);
60         if (adc_hal_status == HAL_OK)
61         {
62             *adc_val_ch5 = (uint16_t) HAL_ADC_GetValue(&hadc1);
63         }
64
65         else if (adc_hal_status != HAL_OK){
66             return false;
67         }
68     }
69 }

```

3.4.3 Implementação de firmware para o BLE

Com a implementação do firmware dos sensores já funcionando, então foi implementado a aplicação de envio dos dados dos sensores. Bastante coisa já foi implementada a partir das configurações do Middleware, que foram demonstradas anteriormente na configuração de inicialização dos periféricos, e o *stack* do rádio BLE já foi disponibilizado pela ST, no formato em binário no seu Github (<https://github.com/STMicroelectronics/STM32CubeWB>), contendo já os passos a passos para programar o *stack* sem fio dentro de dispositivo STM32WBxx.

A seguir vai ser apresentando as implementações e configurações necessárias da aplicação do BLE, para envio de dados dos sensores:

A primeira implementação foi referente aos dados de publicidade BLE, em que um vetor é definido, contendo determinados parâmetros de acordo com o protocolo BLE. Foi configurado os dados de publicidade com base na Tabela 4, que contém a formatação de campo específica do fornecedor do BlueST-SDK. Dentro dos dados de publicidade há uma estrutura predefinida, e ela corresponde a dados específicos do fabricante.

Tabela 4 - Formatação de campo específica do fornecedor do BlueST-SDK.

Comprimento	Nome	Valor
1	Comprimento	0x07/0x0D
1	Tipo de campo	0xff
1	Versão do protocolo	0x01
1	ID de dispositivo	0xXX
4	Máscara de recurso	0XXXXXXXXX
6	Dispositivo MAC (opcional)	0XXXXXXXXXXXXX

Fonte: Manual do usuário do protocolo BlueST (UM2496).

Nos dados de publicidade deve conter os valores correspondentes a máscara de bits da característica, que indicara ao dispositivo cliente que informação que está sendo transmitida. Como descrito na seção de configuração de inicialização de periféricos, foram configuradas características já mencionadas anteriormente como evento de acelerômetro, inclinação e temperatura, que no manual do protocolo BlueST (protocolo utilizado no aplicativo Android

ST BLE Sensor) menciona que essas características correspondem aos seguintes bits, apresentados na Tabela 5:

Tabela 5 - Bits referente as características utilizadas.

Bit	Característica
10	Evento de acelerômetro
18	Temperatura
28	Inclinação

Nos dados de publicidade apresenta 4 bytes destinados a máscara da característica, então foi usado os bits da Tabela 5, para aplicar a máscara de bits dentro desses 4 bytes (32 bits). Na Figura 69, apresenta como ficou a máscara de bits referentes as características utilizadas.

Figura 69 - Parâmetros da máscara dos bits das características.

```

app_ble.c
228 static const char local_name[] = { AD_TYPE_COMPLETE_LOCAL_NAME , 'B', 'e', 'e', 'r', '_', 'd', 'e', 'n', 's'};
229 uint8_t manuf_data[14] = {
230     //2,AD_TYPE_TX_POWER_LEVEL,0x00,/* 0dBm Transmission Power */
231     sizeof(manuf_data)-1, AD_TYPE_MANUFACTURER_SPECIFIC_DATA,
232     0x01,/*SKD version */
233     0x90, /* the device type */
234     0x10, /* GROUP A Feature */
235     0x04, /* GROUP A Feature */
236     0x04, /* GROUP B Feature */
237     0x00, /* GROUP B Feature */
238     0x00, /* BLE MAC start -MSB */
239     0x00,
240     0x00,
241     0x00,
242     0x00,
243     0x00, /* BLE MAC stop */
244 };

```

Na figura 70, foi alterado os dados de solicitação da publicidade BLE, adicionando como parâmetro nas linhas 777 e 778, informações sobre o nome local do dispositivo. E na linha 785 foi alterado o nome do vetor com os dados de publicidade.

Figura 70 – Alterações na função de solicitação de publicidade.

```

app_ble.c
765 static void Adv_Request(APP_BLE_ConnStatus_t New_Status)
766 {
767     tBleStatus ret = BLE_STATUS_INVALID_PARAMS;
768
769     BleApplicationContext.Device_Connection_Status = New_Status;
770     /* Start Fast or Low Power Advertising */
771     ret = aci_gap_set_discoverable(
772         ADV_TYPE,
773         CFG_FAST_CONN_ADV_INTERVAL_MIN,
774         CFG_FAST_CONN_ADV_INTERVAL_MAX,
775         CFG_BLE_ADDRESS_TYPE,
776         ADV_FILTER,
777         sizeof(local_name),
778         (uint8_t*) &local_name,
779         0,
780         NULL,
781         0,
782         0);
783
784     /* Update Advertising data */
785     ret = aci_gap_update_adv_data(sizeof(manuf_data), (uint8_t*) manuf_data);

```

O próximo passo foi fazer implementações na biblioteca custom_app.c como na Figura 71, primeiro foi feito a inclusão da biblioteca app.h na linha 32, uma biblioteca que contém as chamadas das funções destinadas a leitura dos sensores. Em seguida foi adicionado variáveis nas estruturas, as variáveis declaradas nas linhas 39,40 e 41 são destinadas aos status de notificação referente aos dados dos sensores. Já as variáveis declaradas nas linhas 45,46 e 47 são identificadores exclusivos de tempo de software.

Figura 71 – Inclusão de biblioteca e declaração de variáveis.

```

custom_app.c
30 /* Private includes ----- */
31 /* USER CODE BEGIN Includes */
32 #include "app.h"
33 /* USER CODE END Includes */
34
35 /* Private typedef ----- */
36 typedef struct
37 {
38     /* HW_SENS_W2ST_SERVICE */
39     uint8_t          Acceventchar_Notification_Status;
40     uint8_t          Dirarrival_char_Notification_Status;
41     /* ENV_SENS_SERVICE */
42     uint8_t          Temperaturechar_Notification_Status;
43
44     /* USER CODE BEGIN CUSTOM_APP_Context_t */
45     uint8_t          Update_Time_Acc_Event_Id;
46     uint8_t          Update_Time_Dir_Arrival_Id;
47     uint8_t          Update_Time_Temperature_Id;
48     /* USER CODE END CUSTOM_APP_Context_t */
49
50     uint16_t         ConnectionHandle;
51 } Custom_App_Context_t;

```

Na Figura 72, foi feita a declaração de variáveis para armazenar os dados obtidos dos sensores, para posteriormente serem enviados como notificação.

Figura 72 – Declaração de variáveis para armazenar os dados obtidos dos sensores.

```

custom_app.c
93 /* USER CODE BEGIN PV */
94
95 uint8_t AccEventCharData[2+3];
96 uint8_t DirArrivalCharData[2+2];
97 uint8_t TemperatureCharData[2+2];

```

Na Figura 73, mostra as definições dos tempos de envio da notificação nas linhas 59,60 e 61, referentes aos dados dos sensores. E nas linhas 63,66 e 71 foi feito macros para a ordenação dos bytes da notificação, para 16 ou 32 bits com padrões de ordenação *Little-Endian* e *Big-Endian*.

Figura 73 – Definições dos tempos de envio da notificação e definições de macros para ordenação dos bytes da notificação.

```

custom_app.c
57 /* Private defines -----*/
58 /* USER CODE BEGIN PD */
59 #define ACC_EVENT_CHANGE_PERIOD (uint32_t)(60000*1000/CFG_TS_TICK_VAL) /*60s*/
60 #define DIR_ARRIVAL_CHANCE_PERIOD (uint32_t)(60000*1000/CFG_TS_TICK_VAL) /*60s*/
61 #define TEMPERATURE_CHANCE_PERIOD (uint32_t)(60000*1000/CFG_TS_TICK_VAL) /*60s*/
62
63 #define STORE_LE_16(buf, val)  ( ((buf)[0] = (uint8_t) (val)  ), \
64                               ((buf)[1] = (uint8_t) (val>>8) ) )
65
66 #define STORE_LE_32(buf, val)  ( ((buf)[0] = (uint8_t) (val)  ), \
67                               ((buf)[1] = (uint8_t) (val>>8) ), \
68                               ((buf)[2] = (uint8_t) (val>>16) ), \
69                               ((buf)[3] = (uint8_t) (val>>24) ) )
70
71 #define STORE_BE_32(buf, val)  ( ((buf)[3] = (uint8_t) (val)  ), \
72                               ((buf)[2] = (uint8_t) (val>>8) ), \
73                               ((buf)[1] = (uint8_t) (val>>16) ), \
74                               ((buf)[0] = (uint8_t) (val>>24) ) )

```

Na biblioteca `app_conf.h` foram registrados tarefas a serem utilizadas para atualizar as características. Sendo registrada três tarefas, uma para notificação do evento de aceleração

(gravidade específica) na linha 588, outra para notificação da inclinação na linha 589, e a terceira para notificação da temperatura na linha 590.

Figura 74 - Tarefas para atualização das características.

```

app_conf.h
580 typedef enum
581 {
582     CFG_TASK_ADV_CANCEL_ID,
583 #if (L2CAP_REQUEST_NEW_CONN_PARAM != 0 )
584     CFG_TASK_CONN_UPDATE_REG_ID,
585 #endif
586     CFG_TASK_HCI_ASYNC_EVT_ID,
587     /* USER CODE BEGIN CFG_Task_Id_With_HCI_Cmd_t */
588     CFG_MY_TASK_NOTIFY_ACC_EVENT, //ACC_EVENT
589     CFG_MY_TASK_NOTIFY_DIRARRIVAL, // DIRECTION_ARRIVAL
590     CFG_MY_TASK_NOTIFY_TEMPERATURE, //TEMPERATURE
591     /* USER CODE END CFG_Task_Id_With_HCI_Cmd_t */
592     CFG_LAST_TASK_ID_WITH_HCICMD,
593 } CFG_Task_Id_With_HCI_Cmd_t;

```

Na Figura 75, apresenta as funções de *callback* criadas para os temporizados de software das tarefas.

Figura 75 - *Callback* para os temporizadores de software, referentes as tarefas.

```

custom_app.c
435 static void Custom_AccEvent_Change_Timer_Callback(void)
436 {
437     UTIL_SEQ_SetTask(1<<CFG_MY_TASK_NOTIFY_ACC_EVENT, CFG_SCH_PRIO_0);
438
439     return;
440 }
441
442 static void Custom_DirArrival_Change_Timer_Callback(void)
443 {
444     UTIL_SEQ_SetTask(1<<CFG_MY_TASK_NOTIFY_DIRARRIVAL, CFG_SCH_PRIO_0);
445
446     return;
447 }
448
449 static void Custom_Temperature_Change_Timer_Callback(void)
450 {
451     UTIL_SEQ_SetTask(1<<CFG_MY_TASK_NOTIFY_TEMPERATURE, CFG_SCH_PRIO_0);
452
453     return;
454 }

```

Em seguida na Figura 76, foi apresentado implementações na biblioteca `custom_app.c`, em que foram criadas inicializações das tarefas, na linha 289 para o evento de acelerômetro, na linha 294 para a inclinação e na linha 299 para temperatura. E nas linhas 292, 297 e 302 foram criados temporizados de software para atualização periódica das características. Já na linha 304

foram feitos a inicialização do sensor inercial e das variáveis de status das notificações como apresentados na Figura 77.

Figura 76 – Inicializações da aplicação BLE.

```

custom_app.c
286 void Custom_APP_Init(void)
287 {
288     /* USER CODE BEGIN CUSTOM_APP_Init */
289     UTIL_SEQ_RegTask(1<<CFG_MY_TASK_NOTIFY_ACC_EVENT, UTIL_SEQ_RFU, Custom_AccEvent_Update);
290
291     /* Create timer to get the AccEvent params and update characteristic */
292     HW_TS_Create(CFG_TIM_PROC_ID_ISR, &(Custom_App_Context.Update_Time_Acc_Event_Id),hw_ts_Repeated,Custom_AccEvent_Change_Timer_Callback);
293
294     UTIL_SEQ_RegTask(1<<CFG_MY_TASK_NOTIFY_DIRARRIVAL, UTIL_SEQ_RFU, Custom_DirArrival_Update);
295
296     /* Create timer to get the DirArrival params and update characteristic */
297     HW_TS_Create(CFG_TIM_PROC_ID_ISR, &(Custom_App_Context.Update_Time_Dir_Arrival_Id),hw_ts_Repeated,Custom_DirArrival_Change_Timer_Callback);
298
299     UTIL_SEQ_RegTask(1<<CFG_MY_TASK_NOTIFY_TEMPERATURE, UTIL_SEQ_RFU, Custom_Temperature_Update);
300
301     /* Create timer to get the Temperature params and update characteristic */
302     HW_TS_Create(CFG_TIM_PROC_ID_ISR, &(Custom_App_Context.Update_Time_Temperature_Id),hw_ts_Repeated,Custom_Temperature_Change_Timer_Callback);
303
304     Custom_APP_Context_Init();
305
306
307     /* USER CODE END CUSTOM_APP_Init */
308     return;
309 }

```

Figura 77 - Inicialização do sensor inercial e das variáveis de status de notificação.

```

custom_app.c
456 static void Custom_APP_Context_Init(void)
457 {
458     /* AccEvent sensor */
459     app_init_ICM_20689();
460     Custom_App_Context.Acceventchar_Notification_Status = 0;
461
462     /* Temperature sensor */
463     Custom_App_Context.Temperaturechar_Notification_Status = 0;
464 }

```

Na Figura 78, apresenta que foi feito a atribuição do status de notificação do evento de acelerômetro conforme se a notificação está habilitada ou desabilitada, nas linhas 159 e 168. E foi adicionado nas linhas 161 e 170, funções que permitem iniciar e parar o temporizador do software.

Figura 78 – Implementação na aplicação da notificação referente ao evento de acelerômetro.

```

custom_app.c
138= void Custom_STM_App_Notification(Custom_STM_App_Notification_evt_t *pNotification)
139 {
140     /* USER CODE BEGIN CUSTOM_STM_App_Notification_1 */
141
142     /* USER CODE END CUSTOM_STM_App_Notification_1 */
143     switch(pNotification->Custom_Evt_Opcode)
144     {
145         /* USER CODE BEGIN CUSTOM_STM_App_Notification_Custom_Evt_Opcode */
146
147         /* USER CODE END CUSTOM_STM_App_Notification_Custom_Evt_Opcode */
148
149         /* HW_SENS_W2ST_SERVICE */
150         case CUSTOM_STM_ACCEVENTCHAR_READ_EVT:
151             /* USER CODE BEGIN CUSTOM_STM_ACCEVENTCHAR_READ_EVT */
152
153             /* USER CODE END CUSTOM_STM_ACCEVENTCHAR_READ_EVT */
154             break;
155
156         case CUSTOM_STM_ACCEVENTCHAR_NOTIFY_ENABLED_EVT:
157             /* USER CODE BEGIN CUSTOM_STM_ACCEVENTCHAR_NOTIFY_ENABLED_EVT */
158             //MOTION_EXT_Set_Notification_Status(1);
159             Custom_App_Context.Acceventchar_Notification_Status = 1;
160
161             HW_TS_Start(Custom_App_Context.Update_Time_Acc_Event_Id, ACC_EVENT_CHANGE_PERIOD);
162             /* USER CODE END CUSTOM_STM_ACCEVENTCHAR_NOTIFY_ENABLED_EVT */
163             break;
164
165         case CUSTOM_STM_ACCEVENTCHAR_NOTIFY_DISABLED_EVT:
166             /* USER CODE BEGIN CUSTOM_STM_ACCEVENTCHAR_NOTIFY_DISABLED_EVT */
167             //MOTION_EXT_Set_Notification_Status(0);
168             Custom_App_Context.Acceventchar_Notification_Status = 0;
169
170             HW_TS_Stop(Custom_App_Context.Update_Time_Acc_Event_Id);
171             /* USER CODE END CUSTOM_STM_ACCEVENTCHAR_NOTIFY_DISABLED_EVT */
172             break;

```

Na Figura 79, apresenta que foi feito a atribuição do status de notificação de inclinação conforme se a notificação está habilitada ou desabilitada, nas linhas 182 e 190. E foi adicionado nas linhas 184 e 192, funções que permitem iniciar e parar o temporizador do software.

Figura 79 - Implementação na aplicação da notificação referente a inclinação.

```

custom_app.c
174     case CUSTOM_STM_DIRARRIVAL_CHAR_READ_EVT:
175         /* USER CODE BEGIN CUSTOM_STM_DIRARRIVAL_CHAR_READ_EVT */
176
177         /* USER CODE END CUSTOM_STM_DIRARRIVAL_CHAR_READ_EVT */
178         break;
179
180     case CUSTOM_STM_DIRARRIVAL_CHAR_NOTIFY_ENABLED_EVT:
181         /* USER CODE BEGIN CUSTOM_STM_DIRARRIVAL_CHAR_NOTIFY_ENABLED_EVT */
182         Custom_App_Context.Dirarrival_char_Notification_Status = 1;
183
184         HW_TS_Start(Custom_App_Context.Update_Time_Dir_Arrival_Id, DIR_ARRIVAL_CHANCE_PERIOD);
185         /* USER CODE END CUSTOM_STM_DIRARRIVAL_CHAR_NOTIFY_ENABLED_EVT */
186         break;
187
188     case CUSTOM_STM_DIRARRIVAL_CHAR_NOTIFY_DISABLED_EVT:
189         /* USER CODE BEGIN CUSTOM_STM_DIRARRIVAL_CHAR_NOTIFY_DISABLED_EVT */
190         Custom_App_Context.Dirarrival_char_Notification_Status = 0;
191
192         HW_TS_Stop(Custom_App_Context.Update_Time_Dir_Arrival_Id);
193         /* USER CODE END CUSTOM_STM_DIRARRIVAL_CHAR_NOTIFY_DISABLED_EVT */
194         break;

```

Na Figura 80, apresenta que foi feito a atribuição do status de notificação de temperatura conforme se a notificação está habilitada ou desabilitada, nas linhas 205 e 213. E foi adicionado nas linhas 207 e 215, funções que permitem iniciar e parar o temporizador do software.

Figura 80 - Implementação na aplicação da notificação referente ao evento de acelerômetro.

```

custom_app.c
196 /* ENV_SENS_SERVICE */
197 case CUSTOM_STM_TEMPERATURECHAR_READ_EVT:
198 /* USER CODE BEGIN CUSTOM_STM_TEMPERATURECHAR_READ_EVT */
199
200 /* USER CODE END CUSTOM_STM_TEMPERATURECHAR_READ_EVT */
201 break;
202
203 case CUSTOM_STM_TEMPERATURECHAR_NOTIFY_ENABLED_EVT:
204 /* USER CODE BEGIN CUSTOM_STM_TEMPERATURECHAR_NOTIFY_ENABLED_EVT */
205 Custom_App_Context.Temperaturechar_Notification_Status = 1;
206
207 HW_TS_Start(Custom_App_Context.Update_Time_Temperature_Id, TEMPERATURE_CHANCE_PERIOD);
208 /* USER CODE END CUSTOM_STM_TEMPERATURECHAR_NOTIFY_ENABLED_EVT */
209 break;
210
211 case CUSTOM_STM_TEMPERATURECHAR_NOTIFY_DISABLED_EVT:
212 /* USER CODE BEGIN CUSTOM_STM_TEMPERATURECHAR_NOTIFY_DISABLED_EVT */
213 Custom_App_Context.Temperaturechar_Notification_Status = 0;
214
215 HW_TS_Stop(Custom_App_Context.Update_Time_Temperature_Id);
216 /* USER CODE END CUSTOM_STM_TEMPERATURECHAR_NOTIFY_DISABLED_EVT */
217 break;

```

Na Figura 81, apresenta a função destinada a atualizar os dados adicionados na notificação do evento de acelerômetro, que são os valores do *timestamp*, do evento (número qualquer) e da densidade específica.

Figura 81 – Atualização dos dados que serão adicionados na notificação do evento de acelerômetro.

```

custom_app.c
466 static void Custom_AccEvent_Update()
467 {
468     app_on_pot();
469     HAL_Delay(100);
470     app_init_ICM2_20689();
471     /* Timestamp */
472     static uint16_t timer_AccEvent = 0;
473     timer_AccEvent = timer_AccEvent + 1;
474
475     STORE_LE_16(AccEventCharData, timer_AccEvent); //(HAL_GetTick() >>> 3));
476     AccEventCharData[2] = 0x08;
477
478     uint16_t tilt = app_read_SG();
479     app_off_pot();
480     STORE_LE_16(AccEventCharData+3, tilt);
481
482     Custom_AccEventChar_Send_Notification();
483 }

```

Na função apresentada na Figura 81, faz a chamada da função de leitura da densidade específica, que sua implementação está apresentada na Figura 82. A função obtida no processo de calibração, descrita no tópico 3.5 (processo de calibração), foi implementada dentro do firmware nas linhas 111 e 112 da função `app_read_SG()`.

Figura 82 - Função de leitura do SG (gravidade específica).

```

app.c
102=uint16_t app_read_SG()
103 {
104     float tilt3_soma = 0;
105     for(uint8_t i3_soma = 0; i3_soma < 100; i3_soma++)
106     {
107         ICM_20689_Tilt(&hi2c1, &ICM_20689);
108         tilt3_soma += ICM_20689.tiltAngle;
109     }
110     tilt3_avg = tilt3_soma/100;
111     SG = (0.8747713452757396 + 0.009772121703727848 *tilt3_avg-0.0002263336029309042 *tilt3_avg*tilt3_avg
112         + 0.00000184423694626061 *tilt3_avg*tilt3_avg*tilt3_avg);
113
114     return (uint16_t) (SG*1000);
115 }

```

Na Figura 83, apresenta a função destinada a atualizar os dados adicionados na notificação de inclinação que são os valores do *timestamp* e da inclinação.

Figura 83 - Atualização dos dados que serão adicionados na notificação de inclinação.

```

custom_app.c
485=static void Custom_DirArrival_Update()
486 {
487     app_on_pot();
488     HAL_Delay(100);
489     app_init_ICM2_20689();
490     /* Timestamp */
491     static uint16_t timer_DirArrival = 0;
492     timer_DirArrival = timer_DirArrival + 1;
493
494     STORE_LE_16(DirArrivalCharData, timer_DirArrival);//(HAL_GetTick()>>3));
495
496     int32_t tilt2 = app_read_tilt2();
497     app_off_pot();
498     STORE_LE_16(DirArrivalCharData+2,tilt2);
499
500     Custom_Dirarrival_char_Send_Notification();
501 }

```

Na Figura 84, apresenta a função destinada a atualizar os dados adicionados na notificação de temperatura que são os valores do *timestamp* e da temperatura.

Figura 84 - Atualização dos dados que serão adicionados na notificação de temperatura.

```

custom_app.c
516 static void Custom_Temperature_Update()
517 {
518     app_on_pot();
519     HAL_Delay(100);
520     app_init_ICM2_20689();
521     uint8_t BuffPos;
522
523     /* Timestamp */
524     static uint16_t timer_Temperature = 0;
525     timer_Temperature = timer_Temperature + 1;
526
527     STORE_LE_16(TemperatureCharData , timer_Temperature); //(HAL_GetTick()>>3));
528     BuffPos=2;
529
530     uint16_t temp = app_read_temperature_ICM_20689();
531     app_off_pot();
532     STORE_LE_16(TemperatureCharData+BuffPos,temp);
533     BuffPos+=2;
534
535     Custom_Temperaturechar_Send_Notification();
536 }

```

As funções apresentadas nas Figuras 85,86 e 87, são responsáveis pelo envio das notificações referentes aos dados de evento de acelerômetro (gravidade específica), inclinação e temperatura.

Figura 85 – Envio de notificação referente aos dados de evento de acelerômetro (gravidade específica).

```

custom_app.c
331 void Custom_Acceventchar_Send_Notification(void) /* Property Notification */
332 {
333     if(Custom_App_Context.Acceventchar_Notification_Status)
334     {
335         Custom_STM_App_Update_Char(CUSTOM_STM_ACCEVENTCHAR, (uint8_t *)AccEventCharData);

```

Figura 86 - Envio de notificação referente aos dados de inclinação.

```

custom_app.c
356 void Custom_Dirarrival_char_Send_Notification(void) /* Property Notification */
357 {
358     if(Custom_App_Context.Dirarrival_char_Notification_Status)
359     {
360         Custom_STM_App_Update_Char(CUSTOM_STM_DIRARRIVAL_CHAR, (uint8_t *)DirArrivalCharData);

```

Figura 87 - Envio de notificação referente aos dados de temperatura.

```

custom_app.c
382 void Custom_Temperaturechar_Send_Notification(void) /* Property Notification */
383 {
384     if(Custom_App_Context.Temperaturechar_Notification_Status)
385     {
386         Custom_STM_App_Update_Char(CUSTOM_STM_TEMPERATURECHAR, (uint8_t *)TemperatureCharData);

```

3.5 Processo de Calibração

A calibração consiste no processo de comparação entre um dispositivo padrão utilizado no mercado para medir densidade de bebidas alcoólicas como o densímetro, refratômetro em relação ao dispositivo desenvolvido que vai ser calibrado. Com objetivo de encontrar a relação de inclinação do dispositivo desenvolvido em função das medidas de densidade obtidas pelos dispositivos padrões. Permitindo encontrar uma equação que descreva o comportamento da densidade do líquido a partir de leituras de entrada de inclinação (0 a 90°).

Para o processo de calibração do dispositivo foi necessário equipamentos como balança Figura 88 para medição do peso de água e açúcar, para ter um controle da proporção na fabricação da solução de água e açúcar. Também foi necessário um equipamento de medição de gravidade específica do fluido, como o refratômetro Figura 89, para calibrar o dispositivo desenvolvido, sendo possível verificar qual a relação da inclinação do dispositivo desenvolvido com base nas medidas de SG do refratômetro. E sendo necessário também um recipiente Figura 90 para mistura da solução de água e açúcar.

Figura 88 - Balança digital.



Figura 89 - Refratômetro.



Figura 90 - Recipiente para mistura.



O método utilizado para calibração foi o de mistura de água com açúcar. Em que foi predefinido uma quantidade fixa de água, em torno de 1400 ml, e em seguida foi sendo adicionado açúcar, até que fosse obtido diversos valores de gravidade específica do fluido (SG), para ter uma boa resolução de dados, obtendo assim precisão na calibração do dispositivo. Foi estabelecido uma faixa de leitura da Gravidade específica do fluido entre 1,000 e 1,127, que é uma faixa que considera o SG da água pura até a bebidas alcólicas com maior concentração de álcool.

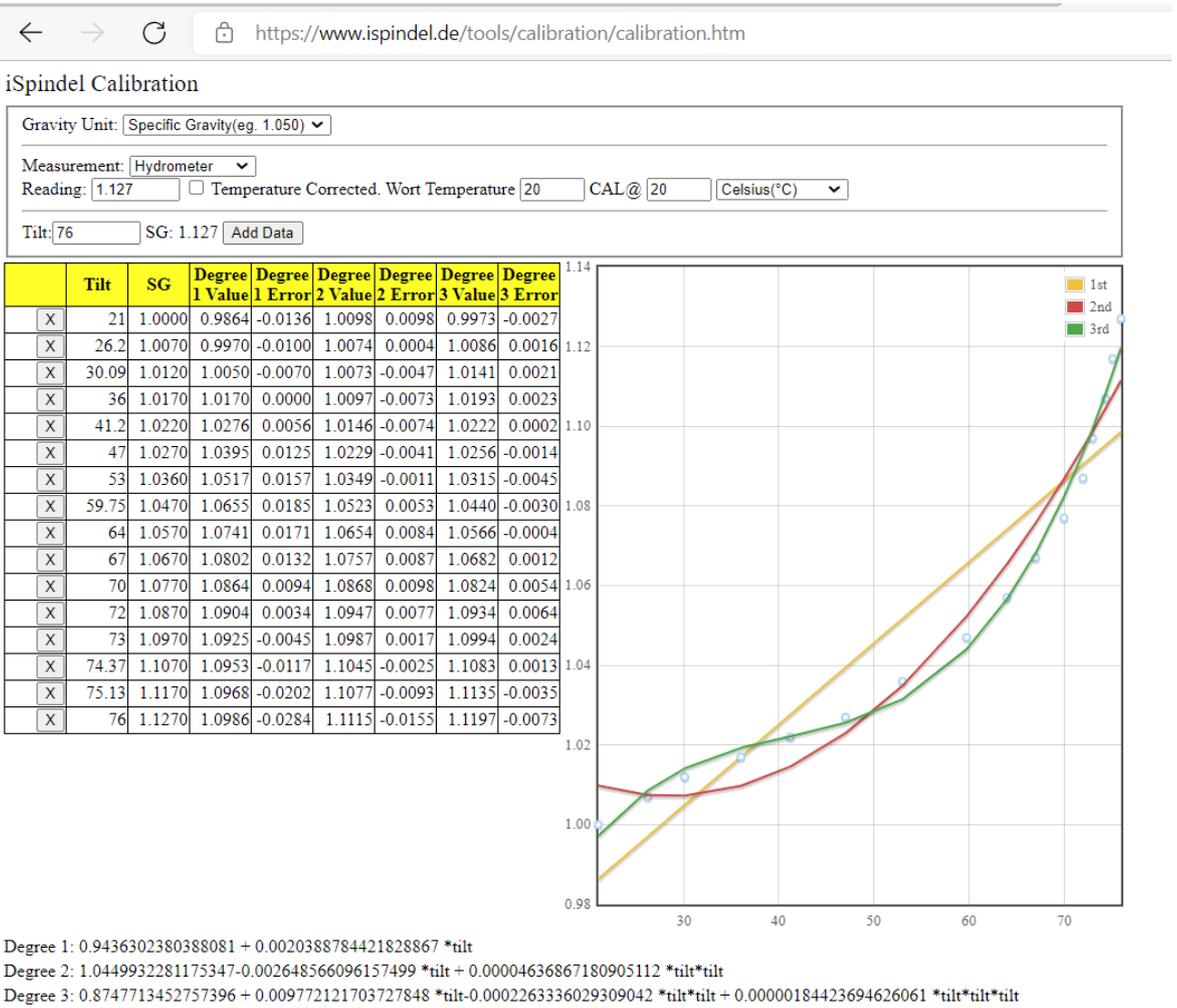
Na Tabela 6, representa todas as medidas aferidas, indicando o SG obtido pelo refratômetro e a inclinação correspondente a cada medida de SG, que foi obtido pelo dispositivo desenvolvido:

Tabela 6 - Dados de inclinação e SG obtidos na calibração do dispositivo.

Medida	Tilt (°C)	SG
1	20,66	1,000
2	76,00	1,127
3	75,13	1,117
4	74,37	1,107
5	73,00	1,097
6	72,00	1,087
7	70,00	1,077
8	67,00	1,067
9	64,00	1,057
10	59,75	1,047
11	53,00	1,036
12	47,00	1,027
13	41,20	1,022
14	36,00	1,017
15	30,09	1,012
16	26,20	1,007
17	21,00	1,000

Com os valores obtidos na Tabela 6, foi possível obter uma função polinomial que converte ângulo em gravidade específica. Para obtenção dessa função, foi utilizada uma calculadora online (<https://www.ispindel.de/tools/calibration/calibration.htm>) que é específica para obtenção da curva de calibração e da função polinomial. Então foram inseridos os dados na calculadora, correspondente ao ângulo de inclinação e o valor de SG. Após inserir os dados na calculadora, foi possível obter a curva de calibração do dispositivo e a fórmula que irá prever a densidade dado um valor de inclinação. Na Figura 91, está representando a curva de calibração e três funções que ordens diferentes, que foram obtidos através da calculadora. Foi escolhido a função polinomial de 3° ordem, para ser inserida no firmware, pois foi a que teve melhor aproximação em relação a curva de calibração do dispositivo.

Figura 91 - Calculadora para obtenção da curva de calibração do dispositivo.



4 RESULTADOS E DISCUSSÕES

No desenvolvimento desse projeto foi possível obter resultados bem significativos em relação ao sistema de monitoramento de bebidas alcoólicas. Como o dispositivo físico completamente finalizado e apresentando características finais de um produto funcional. Nas Figuras 92 e 93, apresenta os resultados reais, que foram abordados nas etapas de desenvolvimento de hardware e estrutura mecânica. Evidenciado toda a montagem física finalizada, com todos os componentes eletrônicos soldados na placa de circuito impresso, fixação da estrutura mecânica com suporte de pesos na base da placa de circuito impresso, e com o hardware dentro do encapsulamento plástico tubete.

Figura 92 - Hardware fixado na estrutura mecânica.

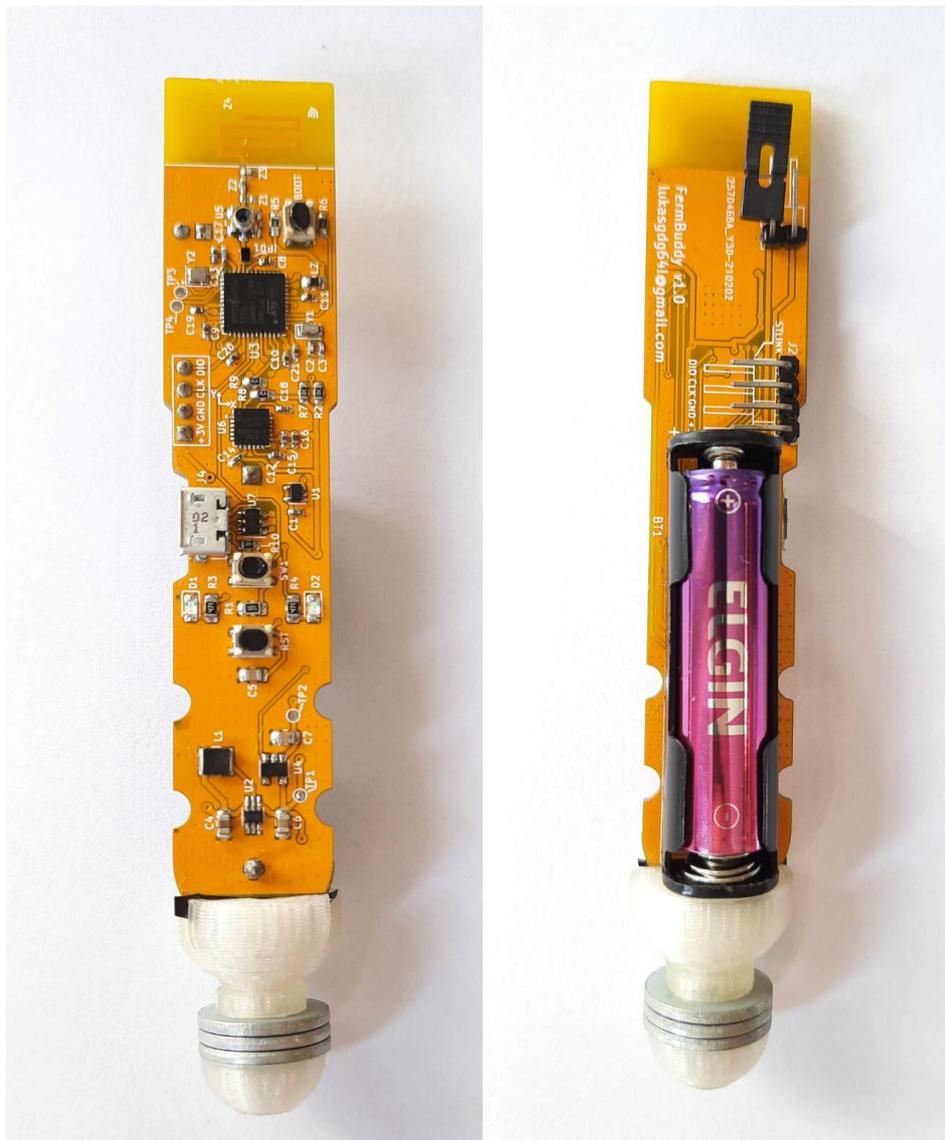


Figura 93 - Montagem completa do dispositivo de monitoração do processo de fermentação de bebidas alcoólicas.



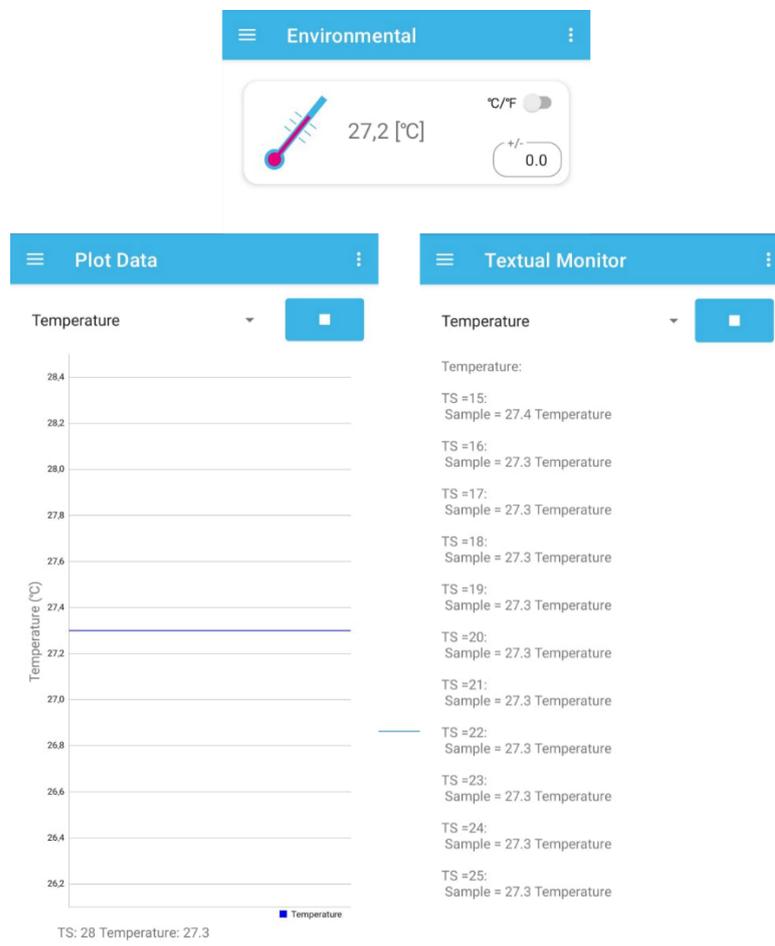
Para demonstração real de funcionamento do dispositivo, foi colocado o mesmo dentro de um recipiente com água Figura 94, e feito a coletas de dados de temperatura, inclinação, SG e RSSI, e apresentado esses dados coletados a partir de uma interface com o aplicativo Android ST BLE Sensor.

Figura 94 - Dispositivo imerso na água.



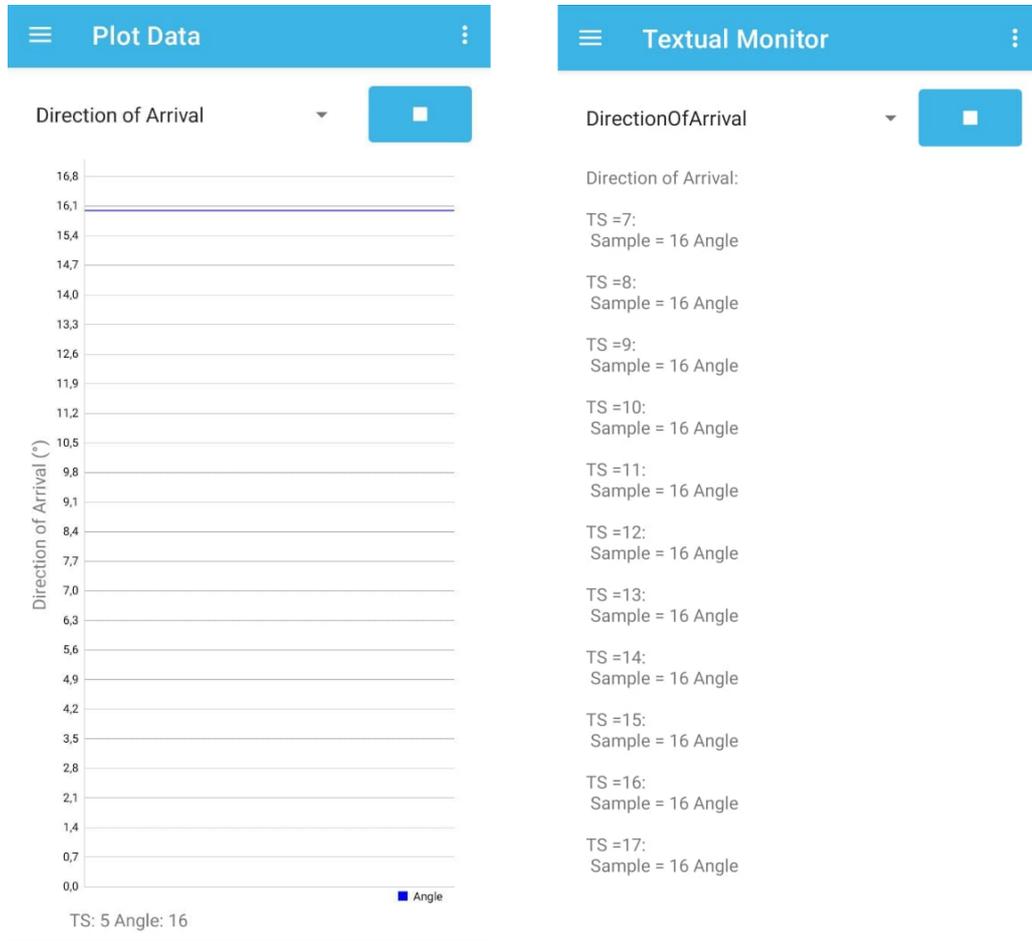
Na Figura 95 apresenta os dados de temperatura, em três configurações diferentes de visualização.

Figura 95 - Dados de temperatura.



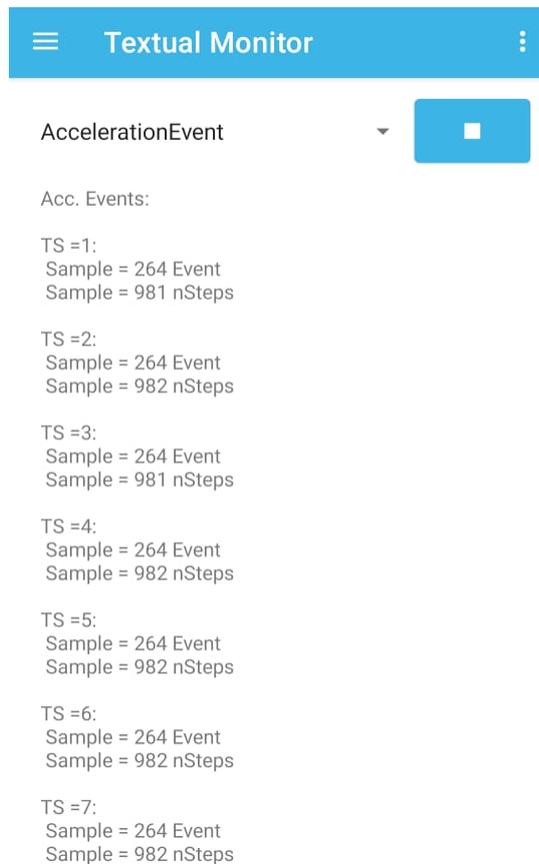
Na Figura 96 apresenta os dados de inclinação do dispositivo em relação a densidade da água, em duas configurações diferentes de visualização.

Figura 96 - Dados de inclinação.



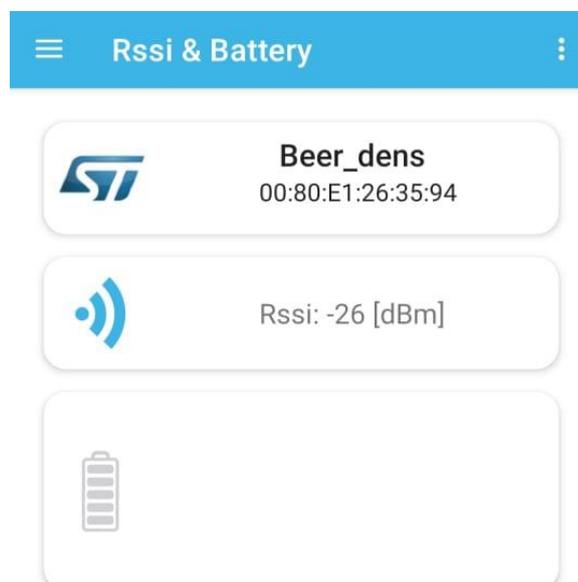
Na Figura 97 apresenta os dados de SG, o valor apresentado no aplicativo foi multiplicado por 1000 vezes o valor real, para obter pelo menos três casas decimais, isso foi feito por conta que o aplicativo é de terceiros e não foi possível fazer essa modificação no aplicativo, sendo adequado fazer essa alteração diretamente no firmware do dispositivo.

Figura 97 - Dados de SG.



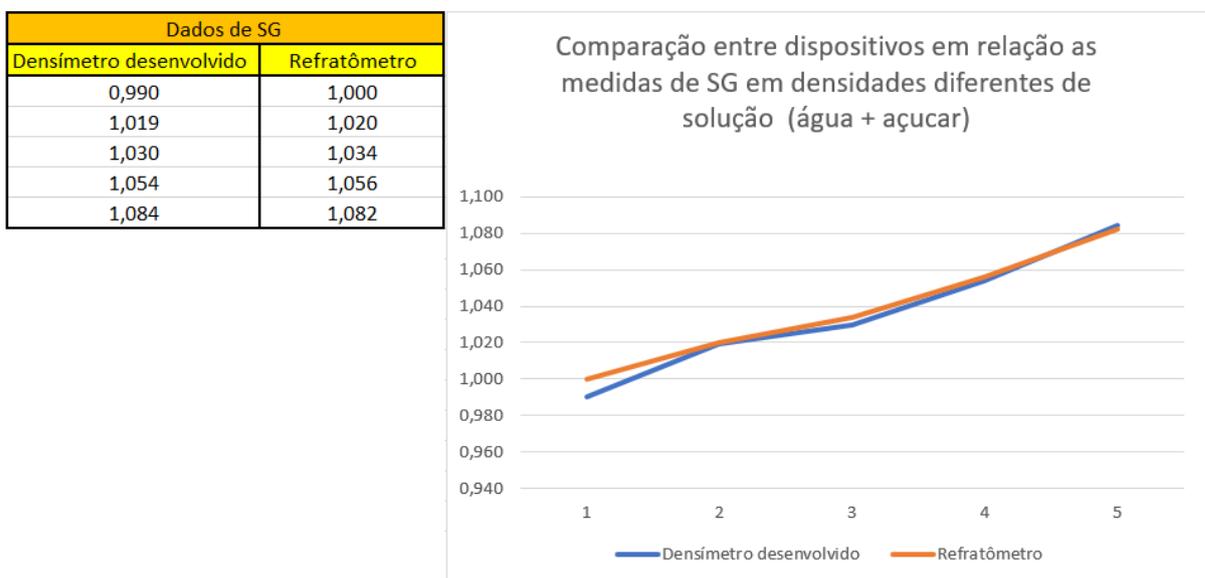
Na Figura 98 apresenta os dados de RSSI, que corresponde ao indicativo de intensidade do sinal recebido que é uma medida da potência presente em um sinal de rádio recebido

Figura 98 - Dados de RSSI.



Na validação do dispositivo desenvolvido, foi feita uma comparação de medições de gravidade específica em relação a um equipamento usual do mercado para medição de SG como o refratômetro. Para essa comparação foi feito cinco soluções de água e açúcar, com densidades diferentes. Nessa comparação foi possível obter para cada densidade de solução diferente, as leituras de SG tanto para o densímetro desenvolvido, quanto para o refratômetro. Os dados obtidos estão apresentados na Figura 99.

Figura 99 - Dados comparativos entre dois dispositivos na medição da densidade de soluções de água e açúcar.



Com base nos resultados apresentados acima, pode analisar que o dispositivo de monitoração de bebidas alcoólicas funcionou como o previsto, apresentando a coleta de dados de temperatura, inclinação e SG, e enviando esses dados para um dispositivo central (smartphone) por comunicação sem fio BLE. E apresentando na comparação com um dispositivo usual de medidas de densidade, valores bem próximos, mas ainda sendo necessário ainda validar esses resultados em um processo real de fermentação de bebida alcoólicas.

5 CONCLUSÃO

Nesse trabalho foi desenvolvido um dispositivo de monitoração do processo de bebidas alcoólicas, utilizando requisitos para o desenvolvimento de produtos. Evidenciando projetar um produto com o que tem de mais recente tecnologicamente e disponível no mercado, para produção em escala. E levando em consideração o custo acessível para produção, e confiabilidade de funcionamento. O funcionamento do produto permitiu evidenciar a validação da ideia, de um sistema capaz de monitorar de forma automática e robusta o processo de fermentação.

Durante o desenvolvimento do dispositivo foram encontradas dificuldades na seleção de componentes eletrônicos, que estariam disponíveis no mercado para o projeto de hardware, devido à crise mundial da falta de componentes eletrônicos durante o período de pandemia. Havendo um atraso inicial para obtenção do hardware finalizado em mãos, dificultando para o desenvolvimento de um firmware robusto e para criação de um aplicativo mobile. Também houve grandes desafios enfrentados em relação ao desenvolvimento do firmware do microcontrolador STM32WB55, devido ser um microcontrolador bastante recente no mercado, havendo poucas referencias de aplicação com esse chip, demandando então um grande tempo de estudo para desenvolvimento de um firmware funcional. Outro desafio enfrentado foi de projetar um dispositivo que coubesse dentro de um recipiente plástico com dimensões reduzidas, então foi bastante árduo estabelecer requisitos bem estabelecidos de dimensão de hardware, bateria e estrutura mecânica para que coubessem de forma adequada dentro desse recipiente.

Como trabalhos futuros há um planejamento de desenvolvimento de um aplicativo mobile próprio e focado para essa aplicação com esse dispositivo. Além disso, ter um foco maior na implementação de melhorias no firmware, para estratégias de consumo de energia. E realizar mais testes de validação do dispositivo, principalmente em uma situação real de fermentação de bebidas alcoólicas. E por fim estabelecer melhores estratégias na calibração do dispositivo, levando em consideração que no processo real há a geração de álcool, prevendo então medidas para correção na aquisição da densidade em relação ao álcool gerado durante o processo.

6 REFERÊNCIAS BIBLIOGRÁFICAS

Pintarelli, G. B; Marcondes, H. **DESENVOLVIMENTO DE UM DENSÍMETRO ELETRÔNICO PARA PRODUÇÃO DE CERVEJA**. Revista Ilha Digital, 2019.

Violino, S; Figorilli, S, Costa, C; Pallottino, Federico. **Internet of Beer: A Review on Smart Technologies from Mash to Pint**. Foods, 2020.

OLIVEIRA, B. D. M.; MELO FILHO, J. M.; AFONSO, J. C. **A densidade e a evolução do densímetro**. Revista Brasileira de Ensino de Física, v. 35, n. 1, p. 1 10, mar. 2013.

GITHUB. **iSpindel**. Disponível em: <<https://github.com/universam1/iSpindel>>. Acesso em 08 de dezembro de 2021.

WHITE, C.; ZAINASHEFF, J. **Yeast: the practical guide to beer fermentation**. [s.l.]: Brewers Publications, 2010.

BARON BREW EQUIPMENT. **Wireless Hydrometer and Thermometer**. Disponível em: <www.tilthydrometer.com>. Acesso em 08 de dezembro de 2021.

Real World Applications of Internet of Things (IoT)—Explained in Videos. Disponível em: <<https://www.analyticsvidhya.com/blog/2016/08/10-youtube-videos-explaining-the-real-world-applications-of-internet-of-things-iot/>>. Acesso em 08 de dezembro de 2021.

Nimbalkar, S.U.; Supekar, S.D.; Meadows, W.; Wenning, T.; Guo, W.; Cresko, J. **Enhancing Operational Performance and Productivity Benefits by Implementing Smart Manufacturing Technologies in Breweries**; Oak Ridge National Lab. (ORNL): Oak Ridge, TN, USA, 2019.

Baron, N. **CONTROLLING FERMENTATION RATE WITH PRESSURE**. Disponível em: <<https://tilthydrometer.com/blogs/news/controlling-fermentation-rate-with-pressure>>. Acesso em 08 de dezembro de 2021.

Zhao, F.M.; Chen, S.Y.; Yang, Z.J. **Design of Temperature Process Control System of Beer Fermentation Based on Internet of Things**. Comput. Technol. Dev. 2012, 7, 46

Tec5USA. **Optimization of the Beer Fermentation Process with NIR Spectroscopy.** Disponível em: <<https://www.tec5usa.com/optimization-of-the-beer-fermentation-process/>>. Acesso em 09 de dezembro de 2021.

Homebrew Fermentation Monitor. Disponível em <<https://www.hackster.io/135387/homebrew-fermentation-monitor-9df83e>>. Acesso em 09 de dezembro de 2021.

PANG, Z.; CHEN, Q.; HAN, W.; ZHENG, L. **Value-centric design of the internet-of-things solution for food supply chain: Value creation, sensor portfolio and information fusion.** Information Systems Frontiers, v. 17, n. 2, p. 289 319, 29 abr. 2015

Leonard, J. **The Internet of Beer: IoT for Breweries.** Disponível em: <<https://blog.nordicsemi.com/getconnected/the-internet-of-beer-iot-for-breweries>> Acesso em 08 de dezembro de 2021.

Agência Nacional de Vigilância Sanitária Anvisa. **Resolução de Diretoria Colegiada No 91, de 11 de maio de 2001.** Brasil, 2001.

STMicroelectronics, STM32WB55. **Ultra-low-power dual core Arm Cortex-M4 MCU 64 MHz, Cortex-M0+ 32 MHz com 1 Mbyte de memória Flash, Bluetooth LE 5.2, 802.15.4, Zigbee, Thread, USB, LCD, AES-256.** Disponível em: <<https://www.st.com/en/microcontrollers-microprocessors/stm32wb55rg.html>>. Acesso em 09 de dezembro de 2021.

Hobbybrew Main. Disponível em: <<http://hobbybrew.sourceforge.net>>. Acesso em 09 de dezembro de 2021.

OECD. Test No. 109: **Density of Liquids and Solids.** OECD Guidelines for the Testing of Chemicals, v. Section 1, out. 2012.

iSpindel–DIY electronic Hydrometer. Disponível em: <<http://www.ispindel.de/>>. Acesso em 09 de dezembro de 2021.

Thomas, D; Wilkie, E; Irvine, J. **Comparison of Power Consumption of WiFi Inbuilt Internet of Things Device with Bluetooth Low Energy.** World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering Vol:10, No:10, 2016

DAOUD, I. S.; SEARLE, B. A. **On-Line Monitoring Of Brewery Fermentation By Measurement Of Co 2 Evolution Rate.** *Journal of the Institute of Brewing*, v. 96, n. 5, p. 297 302, 10 set. 1990.

Tilt Hydrometer Home Page. Disponível em: <<https://tilthydrometer.com/>>. Acesso em 10 de dezembro de 2021.

IoT Beer Brewing Device Hops Over Human Error. Disponível em: <<https://www.engineersrule.com/iot-beer-brewing-device-hops-over-human-error/>>. Acesso em 10 de dezembro de 2021.

ELMEHDI, H. M.; PAGE, J. H.; SCANLON, M. G. **Evaluating Dough Density Changes During Fermentation by Different Techniques.** *Cereal Chemistry Journal*, v. 84, n. 3, p. 250 252, maio 2007.

MINISTÉRIO DA AGRICULTURA PECUÁRIA E ABASTECIMENTO. **Instrução Normativa No 5, de 31 de março de 2000**, 2000.

SACCO, Francesco. **10 mandamentos da PCB.** Brasil, 1 dez. 2015. Disponível em: <<https://www.embarcados.com.br/10-mandamentos-da-pcb/>>. Acesso em: 21 janeiro de 2022. Citado na página 43.

GITHUB. **BlueSTSDK_Android.** Disponível em: <https://github.com/STMicroelectronics/BlueSTSDK_Android>. Acesso em 18 de janeiro de 2022.

Android Developers. **Visão geral do Bluetooth de baixa energia.** Disponível em: <<https://developer.android.com/guide/topics/connectivity/bluetooth-le>>. Acesso em 18 de janeiro de 2022.

Afaneh, M. **Bluetooth GATT: How to Design Custom Services & Characteristics [MIDI device use case].** Disponível em: <<https://www.novelbits.io/bluetooth-gatt-services-characteristics/>>. Acesso em 7 de fevereiro de 2022.

STMicroelectronics. **Application note: Low cost PCB antenna for 2.4 GHz radio: meander design for STM32WB Series.** AN5129, 2019.

STMicroelectronics. **Application note: Development of RF hardware using STM32WB microcontrollers.** AN5165, 2020.