

Lucas de Araujo Dantas

# **Desenvolvimento de um Sistema Genérico de Aquisição e Transmissão de Dados Sem Fio**

Uberlândia, MG

2023

Lucas de Araujo Dantas

# **Desenvolvimento de um Sistema Genérico de Aquisição e Transmissão de Dados Sem Fio**

Projeto de Trabalho de Conclusão de Curso da 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.

Universidade Federal de Uberlândia - UFU  
Faculdade de Engenharia Elétrica - FEELT

Orientador Prof. Dr. Marcelo Barros de Almeida

Uberlândia, MG

2023

Lucas de Araujo Dantas

## **Desenvolvimento de um Sistema Genérico de Aquisição e Transmissão de Dados Sem Fio**

Projeto de Trabalho de Conclusão de Curso da 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.

Uberlândia (MG), 28 de Novembro de 2023.

COMISSÃO EXAMINADORA

---

**Prof. Dr. Marcelo Barros de Almeida**  
Orientador

---

**Prof. Dr. Josué Silva de Moraes**  
Membro Avaliador

---

**Mestre Luís Ricardo Cândido Côrtes**  
Membro Avaliador

Uberlândia, MG  
2023

# Agradecimentos

Gostaria de expressar minha profunda gratidão à minha família, especialmente à minha mãe, Cristiane de Araujo, cujo constante estímulo e apoio foram fundamentais para o meu percurso acadêmico. Sua incansável dedicação em me incentivar a estudar e a nunca desistir moldou significativamente a minha jornada.

Quero estender os meus agradecimentos ao professor Marcelo Barros de Almeida, que desempenhou um papel fundamental ao me introduzir no universo científico. Sua confiança em meu potencial impulsionou-me a superar desafios e acreditar nas minhas capacidades.

Por fim, não posso deixar de expressar minha gratidão aos meus colegas de laboratório, Lukas Gabriel Dias Gomes e Adriel Luiz Marques. Sua colaboração e apoio ao longo do desenvolvimento deste trabalho foram extremamente importantes. A todos, o meu sincero agradecimento.



*“Even if you are not ready for the day, it cannot always be night.”*  
*(Gwendolyn Brooks, 1991)*

# Resumo

A evolução tecnológica tem impulsionado a busca por soluções eficientes e de baixo consumo energético em diversas áreas. Nesse contexto, o desenvolvimento de sistemas de aquisição e transmissão de dados utilizando protocolos de baixo consumo, como zigbee, LoRaWAN e bluetooth tem se mostrado uma área de grande relevância no mundo da internet das coisas. Este trabalho propõe o desenvolvimento de um sistema genérico de aquisição e transmissão de variáveis essenciais, capaz de adquirir dados analógicos/digitais e transmiti-los de forma precisa utilizando um sistema que utiliza componentes de baixo consumo de energia. Com isso, é apresentado um desenvolvimento de um sistema embarcado genérico e adaptável capaz de adquirir e transmitir dados. O hardware deste dispositivo foi desenvolvido em torno do microcontrolador ATSAMR21G17A-MF, que incorpora o rádio AT86RF233. Esse rádio é responsável pela transmissão de dados por meio do protocolo 802.15.4, utilizado para a comunicação Zigbee. Quanto ao firmware do sistema, ele foi programado em linguagem C/C++ na plataforma microchip studio com o objetivo de integrar a aquisição e a transmissão de dados.

**Palavras-chaves:** Baixo Consumo, Comunicação Sem Fio, Comunicação ZigBee, Internet das Coisas, Sistemas Embarcados.

# Abstract

Technological evolution has driven the search for efficient and low energy consumption solutions in several areas. In this context, the development of data acquisition and transmission systems using low-power protocols such as zigbee, LoRaWAN and bluetooth has proven to be an area of great relevance in the world of the internet of things. This work proposes the development of a generic system for acquisition and transmission of essential variables, capable of acquiring analogue/digital data and transmitting them accurately using a system that uses low energy consumption components. With this, a development of a generic and adaptable embedded system capable of acquiring and transmitting data is presented. The hardware of this device was developed around the ATSAMR21G17A-MF microcontroller, which incorporates the AT86RF233 radio. This radio is responsible for transmitting data through the 802.15.4 protocol, used for Zigbee communication. As for the system firmware, it was programmed in C/C++ language on the microchip studio platform with the aim of integrating data acquisition and transmission.

**Key-words:** Embedded Systems, Internet Of Things, Low Consumption, Wireless Communication, ZigBee Communication.

# Lista de ilustrações

Figura 1 – Exemplo de comunicação sem fio. . . . .	18
Figura 2 – Ferramenta de cálculo para o dimensionamento de largura de trilhas (KiCad). . . . .	21
Figura 3 – Ferramenta de cálculo para o dimensionamento de largura de trilhas (JLCPCB). . . . .	21
Figura 4 – Gráfico de consumo ZigBee. . . . .	23
Figura 5 – Arquitetura ZigBee. . . . .	24
Figura 6 – Exemplo de quantização. . . . .	26
Figura 7 – Esquema elétrico do regulador de tensão. . . . .	29
Figura 8 – Esquema elétrico do circuito USB. . . . .	30
Figura 9 – Esquema elétrico do circuito ADC. . . . .	31
Figura 10 – Esquema elétrico da memória RAM externa. . . . .	32
Figura 11 – Esquema elétrico dos pinos externados. . . . .	32
Figura 12 – Esquema elétrico do SKY66112-11. . . . .	34
Figura 13 – Esquema elétrico da antena. . . . .	34
Figura 14 – Esquema elétrico da alimentação do SKY66112-11. . . . .	34
Figura 15 – Esquema elétrico do microcontrolador. . . . .	36
Figura 16 – Layout da PCB (Todas as camadas). . . . .	38
Figura 17 – Layout da PCB (Primeira camada). . . . .	38
Figura 18 – Layout da PCB (Segunda Camada). . . . .	39
Figura 19 – Layout da PCB (Terceira Camada). . . . .	39
Figura 20 – Layout da PCB (Quarta Camada). . . . .	40
Figura 21 – Posicionamento de componentes na PCB. . . . .	41
Figura 22 – Cálculo da constante dielétrica efetiva. . . . .	44
Figura 23 – Cálculo da largura de trilha. . . . .	44
Figura 24 – Especificações da PCB. . . . .	45
Figura 25 – Opções de especificações avançadas. . . . .	46
Figura 26 – Placa fabricada pela JLCPCB (Vista da primeira camada). . . . .	48
Figura 27 – Placa fabricada pela JLCPCB (Vista da quarta camada). . . . .	48
Figura 28 – Placa fabricada pela JLCPCB e montada. . . . .	49
Figura 29 – Zoom no pino 3 no capacitor de desacoplamento C33. . . . .	50
Figura 30 – Diagrama do <i>firmware</i> . . . . .	51
Figura 31 – Diagrama de clock ATSAMR21G17A. . . . .	52
Figura 32 – Diagrama do DFLL48M. . . . .	52
Figura 33 – Disposição dos clocks. . . . .	53
Figura 34 – Configuração do timer (TCC). . . . .	55

Figura 35 – Configuração do timer (RTC). . . . .	57
Figura 36 – Configuração do timer (RTC). . . . .	59
Figura 37 – Modos de dormir. . . . .	60
Figura 38 – Registro DIPO. . . . .	62
Figura 39 – Registro DOPO. . . . .	62
Figura 40 – Pinos de <i>data</i> . . . . .	62
Figura 41 – Configuração do rádio (SPI). . . . .	63
Figura 42 – Configuração da RAM/ADC externo (SPI). . . . .	65
Figura 43 – Inicialização do ADC. . . . .	66

# Lista de tabelas

Tabela 1 – Tabela de preços produtos ZigBee. . . . .	23
Tabela 2 – Controle lógico SKY66112-11. . . . .	35
Tabela 3 – Disposição de camadas na PCB. . . . .	37
Tabela 4 – Parte 1 da lista de materiais. . . . .	70
Tabela 5 – Parte 2 da lista de materiais. . . . .	71

# Lista de abreviaturas e siglas

ADC	Analog-to-digital converter
APS	Application Support Sub-Layer
BOM	Bill of Materials
C.I	Circuito Integrado
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CSMA-CA	Carrier-Sense Multiple Access With Collision Avoidance
DSSS	Direct-Sequence Spread Spectrum
EMI	Electromagnetic Interference
FCS	Frame Check Sequence
I/O	Input/Output
ID	Identification
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
LNA	Low Noise Amplifier
LSB	Least Significant Bits
MAC	Media Access Control
NWK	Network
O-QPSK	Offset Quadrature Phase Shift Keying
PA	Power Amplifier
PAN	Personal Area Network
PCB	Printed Circuit Board
PHY	Physical Layer

RF	Radio Frequency
RTC	Real Time Clock
RX	Receive
SPI	Serial Peripheral Interface
TCC	Timer/Counter for Control Applications
THD	Total Harmonic Distortion
TX	Transmit
WFI	Wait for Interrupt
ZDO	ZigBee Device Object (ZDO)



# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>14</b>
<b>1.1</b>	<b>Justificativas</b>	<b>15</b>
<b>1.2</b>	<b>Objetivos</b>	<b>15</b>
1.2.1	Objetivos Específicos	15
<b>2</b>	<b>REFERENCIAIS TEÓRICOS</b>	<b>16</b>
<b>2.1</b>	<b>Levantamento de Tecnologias</b>	<b>16</b>
2.1.1	RF em PCB's	17
2.1.2	Protocolo ZigBee	21
2.1.3	Processos de Aquisição	25
<b>3</b>	<b>METODOLOGIA</b>	<b>28</b>
<b>3.1</b>	<b>Estudo de tecnologias</b>	<b>28</b>
<b>3.2</b>	<b>Desenvolvimento do Hardware</b>	<b>28</b>
3.2.1	Esquema elétrico	28
3.2.1.1	Alimentação da PCB	29
3.2.1.2	Circuito da USB	30
3.2.1.3	Circuito do ADC externo	30
3.2.1.4	Circuito da memória RAM externa	31
3.2.1.5	Conectores externos	32
3.2.1.6	Circuito de RF	33
3.2.1.7	Circuito do microcontrolador	35
3.2.2	Layout da PCB	36
3.2.2.1	Camadas da PCB	37
3.2.2.2	Posicionamento de componentes	40
3.2.2.3	Aterramento	41
3.2.2.4	Trilhas	42
3.2.2.5	Fabricação da Placa	44
3.2.2.6	Compra e montagem dos componentes na PCB	48
3.2.3	Testes de Hardware	49
<b>3.3</b>	<b>Desenvolvimento do Firmware</b>	<b>50</b>
3.3.1	Clock	51
3.3.2	Timers	53
3.3.3	Sleep	59
3.3.4	SPI SERCOM	61
3.3.5	Aplicação do Projeto	65

<b>4</b>	<b>CONCLUSÃO</b> . . . . .	<b>67</b>
	<b>REFERÊNCIAS BIBLIOGRÁFICAS</b> . . . . .	<b>68</b>
	<b>APÊNDICE A – TABELA BOM - MATERIAIS DA PCB</b> . . . . .	<b>70</b>

# 1 Introdução

Com o advento da quarta revolução industrial, caracterizada pela interconexão e digitalização de sistemas, a Internet das Coisas (IoT) emergiu como uma das principais impulsionadoras dessa transformação. A IoT representa a convergência entre a tecnologia digital e o mundo físico, permitindo a conectividade de dispositivos e a coleta de dados em tempo real (SCHWAB, 2016).

O conceito de Internet das Coisas (IoT) é literal, significa "coisas" ou "objetos" conectados a internet, essas "coisas" podem ser essencialmente tudo, um tablet, lâmpadas, sensores, computadores, tênis ou livros. Esses objetos se conectam por cabos, fios e tecnologia sem fio, incluindo satélites, redes celulares, Wi-Fi e Bluetooth (GREENGARD, 2015).

Nos últimos anos, o conceito de Internet das Coisas (IoT) tem se tornado gradualmente uma realidade presente em diversos setores da sociedade, essa presença vem se tornando cada vez mais comum devido ao contínuo aumento de processamento computacional e a queda nos preços de hardware, como sensores, microcontroladores, processadores, rádios e circuitos integrados (SCHWAB, 2016).

Com o avanço da tecnologia, os sensores estão se tornando cada vez menores, mais acessíveis e mais sofisticados. Estima-se que até 2025, o número de sensores conectados à Internet possa chegar a 1 trilhão. Eles desempenham um papel crucial na interação entre o mundo físico e a rede virtual, pois fornecem as variáveis físico-químicas do mundo físico, essenciais para o monitoramento e análise de dados (SCHWAB, 2016).

Para viabilizar o monitoramento real desses dados são utilizados protocolos de comunicação, que fornecem diretrizes de como a comunicação deve ser feita entre os dispositivos. Alguns dos protocolos de comunicação sem fio já utilizados no mundo *IOT*, são o Bluetooth, Wi-Fi, ZigBee e Z-Wave. Cada um desses protocolos possuem suas próprias vantagens e desvantagens, porém dentre estes protocolos citados o ZigBee se destaca pelo seu baixo consumo e na sua alta confiabilidade na transmissão de dados.

O ZigBee está se tornando cada vez mais presente no mercado devido ao seu baixo custo e à sua notável eficiência energética. Essa tecnologia está sendo amplamente aplicada em diversos setores, incluindo automação residencial, automação de edifícios comerciais, monitoramento de plantas industriais, telecomunicações e cuidados domésticos e de saúde pessoais (GISLASON, 2008).

Nesse contexto, é apresentada uma proposta de desenvolvimento de um sistema embarcado genérico de aquisição e transmissão de variáveis essenciais com foco no baixo consumo. O sistema utilizará o protocolo de comunicação sem fio ZigBee e contará com o

microcontrolador ATSAMR21G17A-MF, que incorpora o rádio AT86RF233, ademais o *firmware* será feito em linguagem C/C++.

## 1.1 Justificativas

A crescente demanda por dispositivos eletrônicos portáteis e autônomos tem impulsionado a necessidade de soluções eficientes em termos de consumo de energia. A eficiência energética tornou-se um fator essencial na concepção e desenvolvimento de sistemas eletrônicos, uma vez que afeta diretamente a duração da bateria, a sustentabilidade e a viabilidade econômica desses dispositivos.

Nesse contexto, o presente projeto tem como objetivo desenvolver um sistema genérico de aquisição e transmissão de variáveis essenciais de baixo consumo. A adoção de uma abordagem de baixo consumo energético se faz necessária para permitir o funcionamento prolongado dos dispositivos, minimizando a dependência de recargas frequentes ou substituições de baterias. Além disso, ao otimizar o consumo de energia, podemos contribuir para a redução do impacto ambiental, promovendo práticas mais sustentáveis.

## 1.2 Objetivos

O principal objetivo deste trabalho é o desenvolvimento de um sistema genérico de baixo consumo que faça aquisição e transmissão de variáveis essenciais. Para isso será utilizado o microcontrolador ATSAMR21G17A-MF, rádio AT86RF233 e o protocolo de comunicação sem fio ZigBee.

### 1.2.1 Objetivos Específicos

- Estudo e implementação do protocolo *ZigBee* em microcontroladores;
- Estudo do microcontrolador ATSAMR21G17A-MF, do seu rádio embutido AT86RF233 e sua plataforma de desenvolvimento e depuração MICROCHIP STUDIO;
- Desenvolvimento de um sistema embarcado genérico que possa integrar o microcontrolador com os tópicos estudados;
- Desenvolvimento de *firmware* para o sistema embarcado que possa adquirir essas variáveis de sensores diversos e que consiga fazer a transmissão dessas variáveis adquiridas pelos sensores;

## 2 Referenciais Teóricos

Neste capítulo são fundamentados conhecimentos necessários para o entendimento do trabalho, assim como tecnologias correlatas já existentes no mercado.

### 2.1 Levantamento de Tecnologias

Jonathan Valvano descreve um microcontrolador como um dispositivo integrado que encapsula um computador completo em um único chip. Funcionando como um microcomputador em escala reduzida, o microcontrolador abrange todos os componentes essenciais de um computador convencional, incluindo processador, memórias e I/O (entrada/saída). Esses microcontroladores encontram propósito por meio de atuadores/sensores externos. Portas de entrada possibilitam que informações do mundo externo sejam integradas ao computador, enquanto as portas de saída permitem que o microcontrolador envie informações de volta ao ambiente externo. As interfaces de I/O (entrada/saída) são classificadas em quatro categorias: paralelas, seriais, analógicas e por tempo (VALVANO, 2014).

Dito isso, dentre as opções do mercado, microcontroladores, microcomputadores e computadores, os microcontroladores possuem o menor consumo, pois eles são projetados para tarefas específicas com processamento limitado, eles geralmente têm um conjunto de instruções menor e frequências de *clock* mais baixas em comparação com computadores de uso geral. Além disso, os microcontroladores possuem periféricos integrados que reduzem a necessidade de periféricos externos, utilizam baixa tensão de operação e os microcontroladores também possuem modos de baixa potências podendo entrar em modo *sleep* quando não realizam nenhuma tarefa, reduzindo drasticamente o consumo de corrente.

Os microcontroladores possuem uma variedade de arquiteturas disponíveis, sendo uma das mais proeminentes no mercado a família de processadores ARM Cortex-M. Essa família de processadores é projetada especialmente para aplicações de baixo consumo de energia e desempenho moderado. A arquitetura ARM Cortex-M é conhecida por seu equilíbrio entre eficiência energética e capacidade de processamento (YIU, 2017).

Os processadores dessa família são projetados para oferecer um desempenho adequado para a maioria das aplicações de controle embarcado, enquanto minimizam o consumo de energia. Suportam opções, como modos de economia de energia avançados. Além do modo dormir (*sleep mode*) e do modo de hibernação profunda (*deepsleep*), os microcontroladores baseados em ARM Cortex-M podem oferecer modos de economia de energia mais avançados, como o modo de retenção (*retention mode*) ou o modo de

economia de energia ultra baixa (*ultra-low-power mode*). Esses modos permitem que o microcontrolador reduza ainda mais o consumo de energia, desligando ou reduzindo a energia fornecida a diferentes partes do circuito, enquanto retém o estado dos registradores e dados importantes (YIU, 2017).

### 2.1.1 RF em PCB's

A comunicação sem fio está se tornando cada vez mais presente em nossas vidas. Com a redução dos custos dos dispositivos eletrônicos, ficou mais acessível adquirir aparelhos sem fio, como aqueles equipados com tecnologia Wi-Fi e Bluetooth. Embora ainda existam alguns equipamentos que não utilizam a comunicação sem fio, é possível prever que, no futuro, praticamente tudo estará comunicando sem fio. Em conjunto com a evolução dos níveis de integração de tecnologias que utilizam a comunicação sem fio, houve também um aumento na rapidez dos circuitos RF (Radiofrequência) e uma diminuição do consumo de energia, sendo assim, viabilizando mais esse tipo de comunicação em aplicações reais (RAZAVI, 2012).

Nos circuitos de radiofrequência (RF), a utilização de transceptores é fundamental. Esses dispositivos eletrônicos desempenham o papel crucial de transmitir e receber informações por meio de ondas eletromagnéticas. Os transceptores são equipados com antenas que desempenham funções distintas. Na transmissão de dados, a antena atua como o ponto de acoplamento da onda eletromagnética no ar ou espaço livre. Por outro lado, na recepção de dados, as ondas eletromagnéticas são interceptadas por uma antena receptora, sendo então acopladas ao transceptor para a interpretação das informações (RAZAVI, 2012).

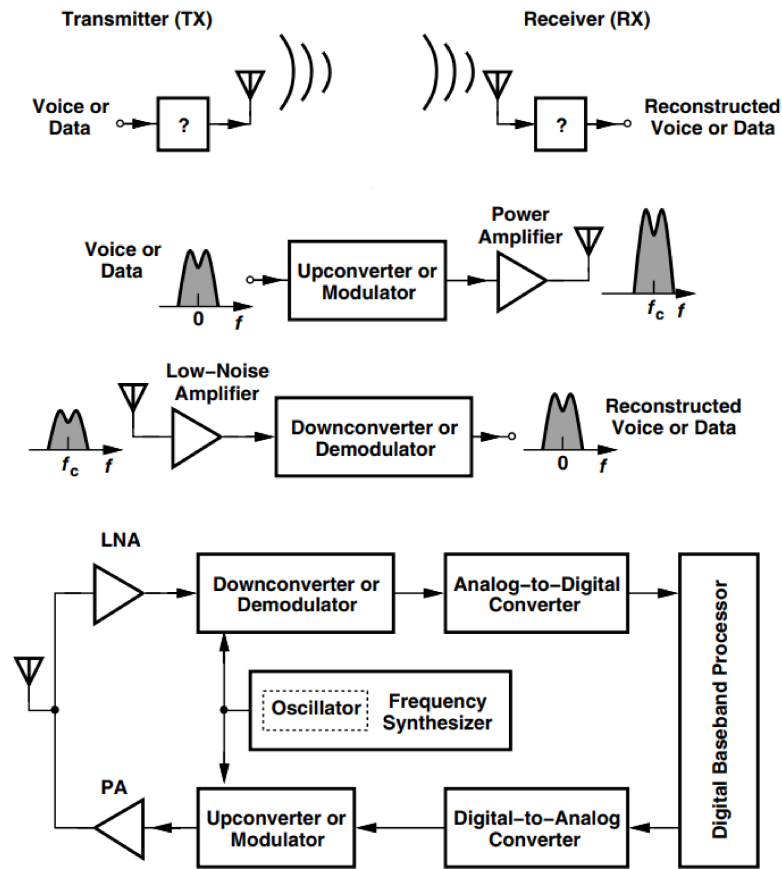
Tanto o transmissor (TX) quanto o receptor (RX) têm requisitos específicos para operarem eficientemente. O transmissor deve fornecer à antena uma quantidade adequada de energia, garantindo que o sinal transmitido seja suficientemente forte para percorrer longas distâncias. Enquanto isso, o receptor deve ser capaz de detectar sinais muito fracos. Para processar esses sinais, é necessário amplificá-los com um baixo nível de ruído (RAZAVI, 2012).

Na etapa de transmissão, a onda eletromagnética aciona a antena por meio de um "amplificador de potência"(PA). No lado do receptor, o sinal é detectado inicialmente por um "amplificador de baixo ruído"(LNA) e, em seguida, passa por um "conversor para baixo"ou "demodulador"(também conhecido como "detector") (RAZAVI, 2012).

A Figura 1 mostra um exemplo de como funciona o processo de transmissão e recepção de dados por meio de um transreceptor. A representação visual na figura oferece uma visão ilustrativa do fluxo de dados entre o transmissor e o receptor, destacando o papel crucial do amplificador de baixo ruído (LNA) e do amplificador de potência (PA) na

fase de recepção e transmissão de sinais.

Figura 1 – Exemplo de comunicação sem fio.



Fonte: Adaptado de (RAZAVI, 2012).

Nos circuitos RF, é crucial realizar o casamento de impedância. O casamento de impedância é um elemento fundamental no design e teste de radiofrequência (RF), onde a falta de casamento de impedância pode resultar em questões significativas. Essas questões podem incluir perda de potência, reflexões de sinal, distorção de sinal e menor eficiência do sistema. Portanto, garantir o casamento adequado de impedância é essencial para otimizar o desempenho e a integridade do sinal em aplicações de radiofrequência.

Nesse contexto, o transmissor desempenha o papel de fornecer potência à antena por meio da linha de transmissão. A eficácia dessa transferência de potência depende do grau de concordância de impedância. Nos casos em que o casamento de impedância é realizado de maneira correta, uma proporção predominante da potência se propaga sem obstáculos ao espaço livre. Em contrapartida, quando o casamento de impedância não atende aos parâmetros ideais, uma parte da potência transmitida é refletida de volta para a fonte, criando assim uma situação repleta de potenciais adversidades. Tais consequências adversas comumente envolvem efeitos deletérios na linha de transmissão e no próprio transmissor, precipitando o risco de comprometimento ou falha estrutural. A intrincada interação do

casamento de impedância, portanto, emerge como uma consideração indispensável na mitigação de reflexões de sinal e na garantia da robustez dos sistemas de RF.

Na maioria dos circuitos de RF, é uma prática comum projetar uma impedância de 50 ohms. Essa escolha se justifica pelo fato de que a impedância de 50 ohms é a referência em torno da qual a maioria dos sistemas de RF é desenvolvida. Dessa forma, o casamento de impedância torna-se crucial, exigindo compatibilidade entre a antena, o rádio e as trilhas que os conectam.

A harmonização de impedância é alcançada quando tanto a antena quanto o rádio, juntamente com as trilhas que os interligam, possuem impedâncias congruentes. Qualquer disparidade nas impedâncias do rádio, trilha e antena no circuito RF pode resultar em perda de potência e reflexão do sinal, conforme discutido anteriormente. Para assegurar uma integração eficiente, é fundamental que tanto o rádio quanto a antena apresentem uma impedância de 50 ohms.

Ao adotar uma abordagem de casamento de impedância, a trilha do circuito RF pode ser projetada para coincidir com essa impedância padrão. Isso implica ajustes na largura, comprimento e inclusão de elementos passivos e reativos, proporcionando um casamento eficaz entre os componentes do sistema.

No contexto do cálculo para o casamento de impedância, é imperativo considerar a noção de *"Critical length"* em RF (radiofrequência). Este termo refere-se ao comprimento crítico de uma linha de transmissão, marcando um ponto em que ocorre uma transição significativa no comportamento elétrico da linha. De maneira mais específica, trata-se do comprimento no qual a linha de transmissão deixa de se comportar predominantemente como uma linha de transmissão de corrente contínua e passa a adotar características mais alinhadas com uma linha de transmissão de onda de rádio. Essa transição assume relevância crucial, influenciando diretamente a eficácia do casamento de impedância. À medida que a linha se aproxima ou excede seu comprimento crítico, a interação entre a fonte, a linha de transmissão e a carga experimenta mudanças substanciais, afetando a transferência de potência e a reflexão de sinais no sistema RF (PETERSON, 2023b).

A importância do comprimento crítico reside no fato de que, abaixo desse valor, a linha de transmissão exibe propriedades elétricas diferentes em comparação com comprimentos superiores. Quando a linha é curta em relação ao comprimento crítico, os sinais tendem a se comportar mais como se estivessem viajando em uma linha de transmissão de impedância próxima à carga. Por outro lado, quando a linha é mais longa, ela passa a se comportar como se sua impedância estivesse mais próxima da impedância característica da linha (PETERSON, 2023b).

Essa transição de comportamento é crítica porque afeta a eficácia do casamento de impedância, que é vital para otimizar a transferência de potência entre a fonte, a linha de



transmissão e a carga. Se a linha for muito curta em relação ao comprimento crítico, podem ocorrer problemas de reflexão de sinal devido a um casamento inadequado de impedância. Portanto, entender e considerar a *critical length* é crucial para garantir o desempenho adequado de sistemas RF, especialmente em aplicações onde o casamento de impedância é fundamental para a integridade do sinal.

Para calcular esse comprimento crítico é utilizado as seguinte equações:

*Stripline (internal trace)*

$$L_c = \frac{c}{f} \cdot \frac{1}{\sqrt{\epsilon_r}} \cdot \frac{1}{12} \quad (2.1)$$

*Microstrip (trace on outer layer)*

$$L_c = \frac{c}{f} \cdot \frac{1}{\sqrt{\epsilon_{ff}}} \cdot \frac{1}{12} \quad (2.2)$$

Onde:

$L_c$  é o comprimento crítico resultante em mm.

$c$  é a velocidade da luz em m/s.

$f$  é a frequência de operação no caso do ZigBee (2,4GHz).

$\epsilon_r$  é a constante de permissividade relativa do dielétrico.

$\epsilon_{ff}$  é a constante efetiva do dielétrico.

No que diz respeito ao dimensionamento da largura da trilha para alcançar a impedância desejada, o KiCad e a JLCPCB oferecem ferramentas de cálculo dedicadas para o cálculo de controle de impedância. Essa ferramenta simplifica o processo ao permitir que os usuários preencham os parâmetros do *stack-up*, facilitando o cálculo preciso da largura da trilha necessária. Em outras palavras, ao fornecer as informações relacionadas à configuração do substrato, a ferramenta automaticamente determina a largura adequada da trilha para assegurar a impedância desejada.

Todos os parâmetros de substrato, essenciais para a caracterização do material utilizado na placa de circuito impresso (PCB), são disponibilizados pelo fabricante. Nesse sentido, é de suma importância atentar para esses dados, uma vez que constituem informações fundamentais o cálculo da impedância nas trilhas. A consideração desses parâmetros, tais como a constante dielétrica, a perda tangencial, a espessura do substrato e outros fatores relevantes, são cruciais para o cálculo situada na Figura 2.

Já a Figura 3 representa uma ferramenta de cálculo mais simplificada, exigindo menos parâmetros. Essa abordagem visa atender especificamente às necessidades do consumidor que requisitou o controle de impedância diretamente à JLCPCB.

Figura 2 – Ferramenta de cálculo para o dimensionamento de largura de trilhas (KiCad).

Fonte: Autoria própria (2023)

Figura 3 – Ferramenta de cálculo para o dimensionamento de largura de trilhas (JLCPCB).

### JLCPCB Impedance Calculator

The JLCPCB Impedance Calculator computes track width values and recommended stack-ups from user-input values of board layer, thickness, copper weight, target impedance, trace spacing (for edge-coupled pairs), and impedance trace to copper gap (coplanar waveguides). [Click here](#) to see the user guide of our impedance calculator.

\* The first stack-up calculated below is highly recommended as it has lowest cost and quickest turn-around.

Impedance (Ω)	Type	Signal Layer	Top Ref	Bottom Ref	Trace Spacing (mm)	Impedance trace to copper (mm)
50	Single Ended (Non coplanar)	L1	/	L2	/	/

Fonte: Autoria própria (2023)

## 2.1.2 Protocolo ZigBee

Uma das principais tecnologias IoT utilizadas na comunicação sem fio é o ZigBee. Ao contrário dos seus concorrentes o ZigBee foca em baixa taxa de dados, pois a tecnologia ZigBee visa pequenas pilhas que caibam em microcontroladores menos potentes, como o 8 bits. Enquanto as outras tecnologias, como Wi-Fi e bluetooth visam mais velocidade e mais definição, o ZigBee visa transmissão de informações sucintas e na sua confiabilidade na transmissão (GISLASON, 2008).

A comunicação sem fio por si só não é confiável, dependendo de vários fatores para o bom funcionamento, como potência do sinal e design da antena. O protocolo ZigBee possui várias maneiras de garantir essa confiabilidade de transmissão. Dentro da confiabilidade na transmissão do protocolo ZigBee, destacam-se alguns pontos importantes, entre eles:

- **IEEE 802.15.4 com O-QPSK e DSSS** - O ZigBee utiliza a especificação para comunicação sem fio de baixo alcance, IEEE 802.15.4, essa especificação é robusta e moderna, foi projetada ao longo de 40 anos de experiência pela IEEE. O IEEE 802.15.4 usa o O-QPSK e DSSS são tecnologias de comunicação sem fio que permitem transmitir dados com alta eficiência e robustez a ruídos. O-QPSK é uma forma de modulação de fase que permite transmitir dois bits por símbolo, enquanto o DSSS é um método de espalhamento espectral que ajuda a espalhar os dados em uma largura de banda maior, o que torna mais difícil para o ruído interferir na comunicação;
- **CSMA-CA** - O ZigBee possui o CSMA-CA, essa tecnologia permite que o ZigBee antes de transmitir verifique se o canal está apto a receber a transmissão, isso permite que não haja transmissões conflitantes;
- **16-bit CRCs** - O ZigBee utiliza 16 bit de CRC chamado FCS ele garante que os bits de dados estão corretos e caso haja erro de transmissão, cada pacote é retransmitido um determinado número de vezes;
- **Rede Mesh** - Uma das topologias que pode-se utilizar com o protocolo ZigBee é a rede Mesh, com ela é possível achar a melhor rota para o pacote, a informação de um nó pode ser transmitida para outros nós e caso um nó esteja com mau funcionamento o pacote é redirecionado automaticamente para outros nós;
- **Verificação fim-a-fim** - O ZigBee permite que sua aplicação saiba se o seu pacote foi recebido ou não por outro nó, o ZigBee também filtra pacotes repetidos de retransmissão automaticamente sem a necessidade de software para isso;

O ZigBee oferece confiabilidade na entrega de pacotes e é vantajoso devido ao seu baixo custo. Essa economia é possível devido à produção acessível do ZigBee e ao fato de seu rádio operar na frequência de 2.4GHz, amplamente utilizada globalmente sem a necessidade de licenças adicionais. A disponibilidade global e a ausência de restrições de licenciamento impulsionam a demanda pelo protocolo, resultando em redução de custos e ampla adoção de produtos ZigBee. A tabela 1 apresenta alguns produtos relacionados que utilizam o protocolo no ano de 2023, com informações de preços e *part numbers* retiradas do site da Mouser.

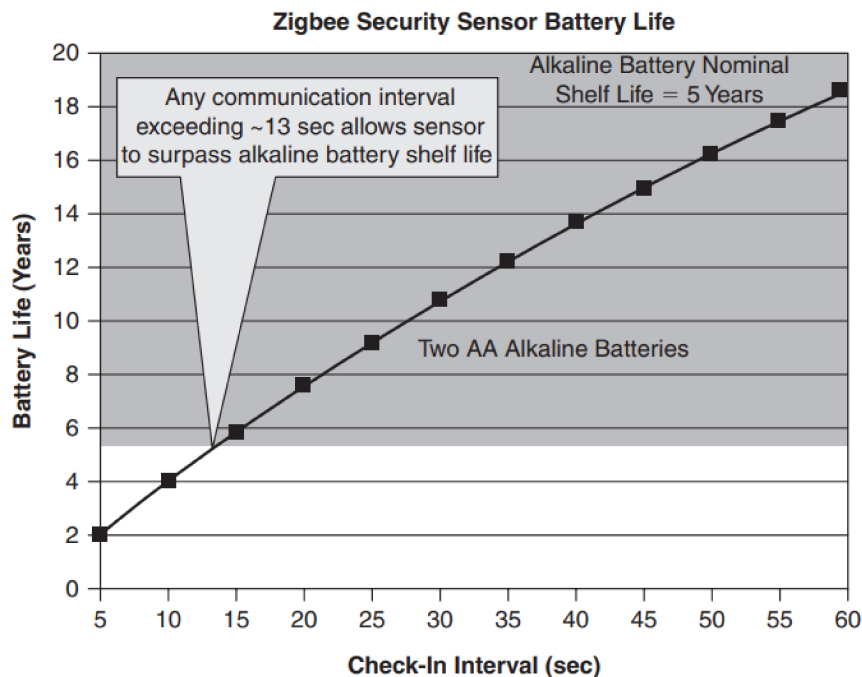
Tabela 1 – Tabela de preços produtos ZigBee.

Chip	Preço (Dólar)	Tipo
AT86RF232	\$3.04	Rádio
ATSAMR30M18AT	\$7.46	Módulo ZigBee
ATSAMR21B18	\$14.66	Módulo ZigBee
CYW20730A2KML2GT	\$2.02	MCU + Rádio
CC2630F128RHBR	\$2.62	MCU + Rádio
ATSAMR21G17A	\$6.30	MCU + Rádio

Fonte: A autoria própria (2023).

A Figura 4 destaca o gráfico de consumo de energia de um dispositivo ZigBee, enfatizando sua característica de baixo consumo. O ZigBee é amplamente reconhecido como um protocolo de baixo consumo de energia. Dispositivos que utilizam essa tecnologia podem ser alimentados por um par de pilhas AA e operar por longos períodos de tempo, podendo durar por anos sem a necessidade de trocar as pilhas. Essa eficiência energética é uma das vantagens significativas do ZigBee.

Figura 4 – Gráfico de consumo ZigBee.



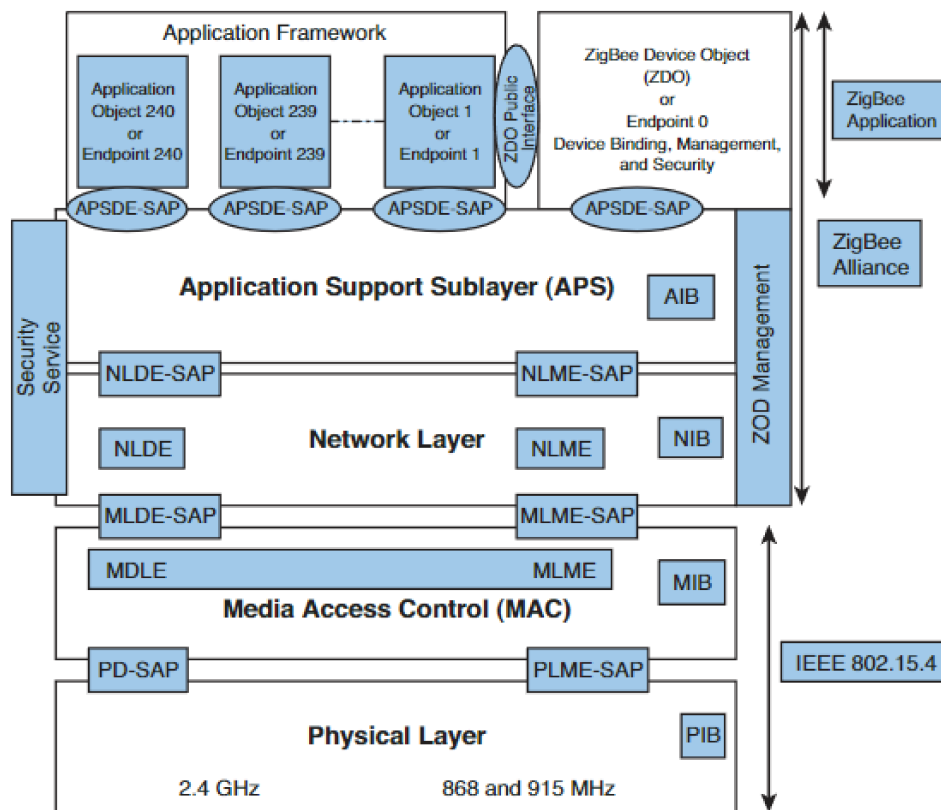
Fonte: Adaptado de (GISLASON, 2008).

O ZigBee é um protocolo de comunicação projetado para estabelecer uma rede de comunicação de área pessoal privada (PAN). A PAN permite a conexão de diversos dispositivos, e cada PAN possui um identificador único de 16 bits chamado PAN ID. Devido ao compartilhamento da frequência de 2.4GHz com tecnologias como Wi-Fi e Bluetooth,

como citado anteriormente, o ZigBee adota uma abordagem de baixa taxa de transferência de dados. O protocolo opera em 16 canais, com cada canal separado por uma frequência de 5MHz. Isso permite que o ZigBee alcance uma taxa de transmissão de aproximadamente 250 kbps, adequada para aplicações que requerem transferências de dados mais leves e eficientes (GISLASON, 2008).

A arquitetura do ZigBee é composta por três seções, a primeira seção é o IEEE 802.15.4 constituído pela camada MAC e PHY, a segunda seção é composta pelas camadas ZigBee que consistem na camada de comunicação, na *ZigBee device object* (ZDO), na subcamada de aplicação e na camada de gestão de segurança. Por fim, a terceira camada é composta pela camada de aplicação do fabricante, nessa camada é possível desenvolver o próprio perfil de aplicação ou utilizar perfis de aplicação ZigBee já pré-estabelecidos pela ZigBee Alliance (ELAHI, 2009). A Figura 5 ilustra a arquitetura do Zigbee, proporcionando uma representação visual clara e detalhada desse sistema.

Figura 5 – Arquitetura ZigBee.



Fonte: Adaptado de (ELAHI, 2009)

A camada física ou PHY é responsável por realizar modulação dos sinais de saída e demodulação dos sinais de entrada ela é responsável por transmitir e receber informações do transreceptor. De maneira sucinta, essa camada é responsável traduzir os pacotes em bits transmitidos pelo ar e e traduzir os pacotes recebidos pelo ar. Já a camada MAC (*Medium Access Control*), fornece o conceito de uma rede, incluindo um PAN

ID (Identificador de Rede Pessoal), ela é responsável por acessar a rede utilizando o Carrier-Sense *Multiple Access with Collision Avoidance* (CSMA/CA), transmitir quadros de beacon para sincronização e garantir a transmissão confiável (ELAHI, 2009).

A camada de rede (NWK) desempenha um papel fundamental na comunicação, sendo responsável por diversas funções essenciais. Ela permite a conexão entre dispositivos, controla a entrada e saída desses dispositivos na rede, realiza o roteamento eficiente dos dados, configura os dispositivos de forma adequada, determina as rotas mais adequadas a serem seguidas e também inclui comandos para garantir a segurança das informações transmitida (GISLASON, 2008).

A subcamada de suporte à aplicação desempenha um papel fundamental ao filtrar as aplicações em execução nos *endpoints*, simplificando a lógica dessas aplicações. Ela possui o entendimento dos conceitos de *clusters* e *endpoints*, permitindo verificar se um determinado *endpoint* pertence ao perfil de aplicação antes de encaminhar a mensagem. Além disso, essa camada também é responsável por filtrar mensagens repetidas na camada de rede (NWK). Em resumo, a camada de APS é responsável pela comunicação das aplicações, definindo uma estrutura de comunicação bem definida, que envolve perfis, *clusters* e *endpoints*. Por outro lado, o ZDO (Objeto de Dispositivo ZigBee) gerencia e controla as aplicações do objeto, fornecendo serviços para descobrir outros nós e serviços na rede. Ele é diretamente responsável pelo estado atual do nó na rede. O *Framework* de Aplicação contém a Biblioteca de *Clusters* ZigBee, proporcionando uma estrutura na qual as aplicações são executadas. Os *endpoints* são utilizados como mecanismo para distinguir uma aplicação da outra. Os serviços de segurança são amplamente utilizados por várias camadas e podem ser aplicados pelo ZDO, APS ou pela camada NWK (rede), por isso são separados em uma seção específica (GISLASON, 2008).

### 2.1.3 Processos de Aquisição

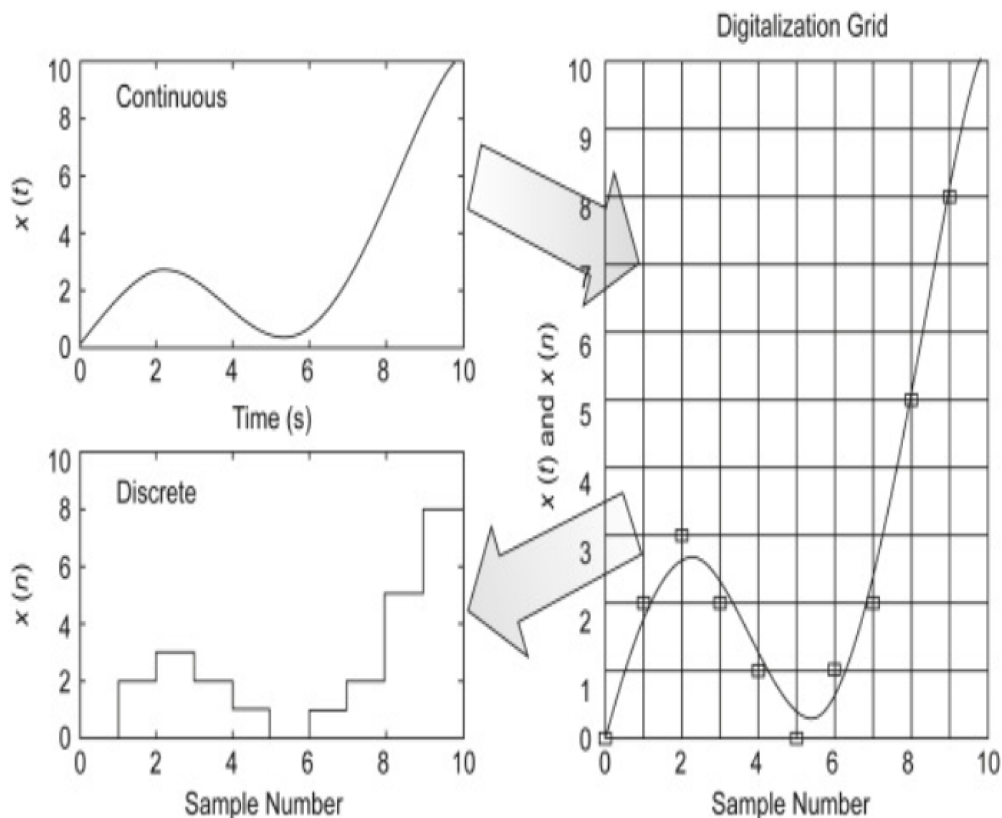
No processo de aquisição de dados, é utilizado um ADC (Conversor Analógico-Digital). O ADC tem a função de converter sinais analógicos, como sinais de tensão, em sinais digitais. Isso é especialmente útil em microcontroladores, pois eles não são capazes de ler diretamente sinais analógicos, apenas sinais digitais. Normalmente, os microcontroladores possuem ADCs integrados, mas também é possível utilizar um ADC externo através de algum protocolo de comunicação serial. Os ADCs geralmente são utilizados para controle e monitoramento de variáveis, essas variáveis são sensores que transcrevem processos físico-químicos em tensão, alguns dos sensores mais utilizados nesse tipo de processos são os de temperatura, pressão, força, inclinação etc.

$$\text{Valores Discretos} = 2^{\text{Número de Bits}} - 1 \quad (2.3)$$

$$\text{LSB} = \frac{V_{\text{ref (+)}} - V_{\text{ref (-)}}}{2^{\text{Número de Bits}} - 1} \quad (2.4)$$

A precisão dos ADCs está diretamente relacionada ao processo de quantização. A quantização envolve a subdivisão da amplitude do sinal analógico em vários níveis discretos. O número de bits do ADC determina a quantidade de valores discretos possíveis. Por exemplo, um ADC de 12 bits, utilizando a equação 2.3, resultaria em 4096 níveis de quantização. Isso significa que o valor analógico seria subdividido em 4096 segmentos distintos. Se a amplitude de tensão lida pelo ADC, for de 0 a 5V, temos que  $5/4096 = 0.0012\text{V}$  por cada seção das 4095 partes (SEMMLOW, 2018). A Figura 6 ilustra o processo de quantização, uma etapa essencial na conversão analógico-digital (A/D). Este processo transforma o sinal de amplitude contínua e tempo discreto em um sinal de amplitude discreta e tempo discreto.

Figura 6 – Exemplo de quantização.



Fonte: Adaptado de (SEMMLOW, 2018)

Outro aspecto importante nos ADCs é o *Least Significant Bit (LSB)*, que representa o menor intervalo que um ADC pode converter. Geralmente, esse valor é fornecido pelo fabricante na folha de dados do produto. É importante ressaltar que um ADC nunca atingirá o nível máximo de seu  $V_{\text{ref}}$ , pois sua escala de tensão é representada pela equação 2.5. Devido ao sistema ser binário e a contagem começar do número zero, o ADC contará

sempre a partir de  $V_{\text{ref}}(-)$  até  $V_{\text{ref}}(+)$  menos o valor do LSB. Portanto, quanto menor o valor do LSB, maior será a resolução do ADC.

$$\text{Escala ADC} = V_{\text{ref}} \times \frac{2^n - 1}{2^n} \quad (2.5)$$



## 3 Metodologia

Neste capítulo, é abordado a análise das tecnologias empregadas no projeto, detalharemos o desenvolvimento tanto do hardware quanto do *firmware*, e também trataremos da sua aplicação prática e dos testes finais realizados.

Visando atingir os objetivos do trabalho, o desenvolvimento do projeto foi dividido em algumas partes: (i) Estudo de tecnologias, (ii) Desenvolvimento do *Hardware*, (iii) Desenvolvimento do *Firmware*, (iv) Testes e (v) Aplicação.

### 3.1 Estudo de tecnologias

Na etapa inicial, inicia-se com uma análise abrangente das tecnologias disponíveis. A escolha recaiu sobre o microcontrolador ATSAMR21G17A-MF devido à sua ampla disponibilidade no mercado, à arquitetura ARM-Cortex M0+ que oferece uma gama de possibilidades, entre elas um robusto modo de baixo consumo, preço acessível e à inclusão do rádio AT86RF233 diretamente no componente. Esse rádio é especialmente otimizado para suportar diversos protocolos, sendo o ZigBee, peça central no projeto, um deles.

Embora o microcontrolador apresente um Conversor Analógico-Digital (ADC) integrado, decidiu-se empregar um ADC externo, o AD7091R-4BRUZ, para a aquisição de dados. Essa escolha se justifica pela sua resolução de bits superior, o que contribuí significativamente para a qualidade da aquisição de dados no projeto. Para a elaboração do esquemático elétrico e de layout da placa de circuito impresso (PCB), é utilizado a ferramenta CAD KiCAD 6 e por fim, para a depuração do *firmware*, utilizou-se o J-LINK como ferramenta de escolha.

### 3.2 Desenvolvimento do Hardware

Nesta etapa do projeto, é apresentado de forma detalhada a execução do esquema elétrico e o desenvolvimento do *layout* da placa de circuito impresso (PCB) utilizando a plataforma KiCAD 6. Isso abrange a concepção das conexões elétricas e a disposição e a escolha dos componentes na PCB, largura de trilhas, *stack-up* entre outros parâmetros garantindo uma análise completa do processo de design.

#### 3.2.1 Esquema elétrico

O esquema elétrico será subdividido em blocos para facilitar a compreensão. É importante destacar que todos os circuitos seguem rigorosamente as especificações das

folhas de dados e os manuais de referência de seus respectivos componentes. Dessa forma, os componentes ativos e passivos dos circuitos foram criteriosamente escolhidos com base nessas informações contidas em suas respectivas folhas de dados.

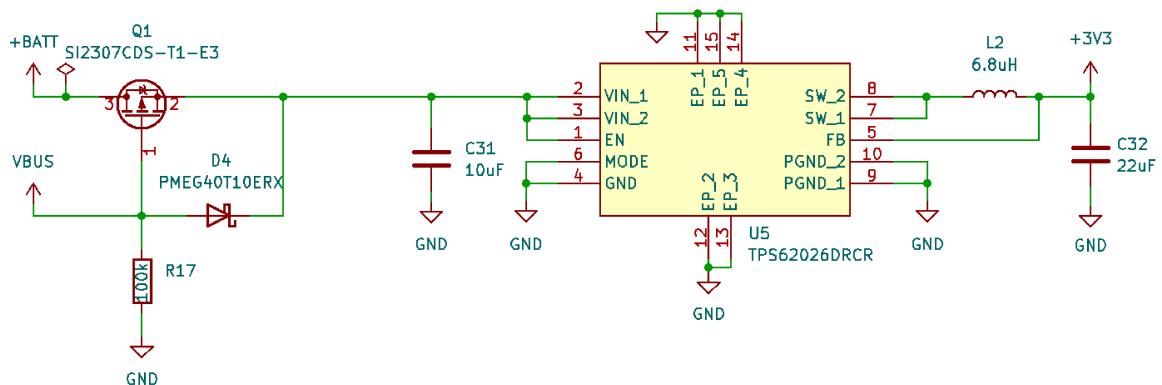
### 3.2.1.1 Alimentação da PCB

O circuito é alimentado por uma tensão de 5V proveniente da porta USB ou por uma bateria externa. Quando alimentado pela bateria, um MOSFET canal P é empregado para evitar a conexão reversa da bateria ao sistema. Além disso, um diodo de Schottky (D4) foi incorporado como medida de proteção a entrada do circuito, este diodo é bastante utilizado devido à sua baixa queda de tensão e sua praticidade ao ser utilizado com pequenas tensões de entrada. Já o resistor (R17) atua como um resistor de *pull-down*, assegurando que não haja tensão indevida no gate do MOSFET, garantindo seu correto funcionamento somente quando polarizado adequadamente.

O microcontrolador ATSAMR21G17A-MF, utilizado no circuito, opera com uma faixa de tensão entre 1,8V a 3,6V. Uma vez que as tensões da bateria e da USB são superiores a essa faixa, é necessário empregar um regulador de tensão para adequar a tensão. Neste caso, é utilizado um circuito integrado da Texas Instruments chamado TPS62026DRCR. Este componente é responsável pela alimentação efetiva da PCB, pois, com a configuração definida no esquemático, possui uma entrada variável de 3,6V a 6V e uma saída fixa de 3,3V dentro da faixa de operação do microcontrolador e do conversor analógico-digital.

Neste circuito regulador é utilizado alguns componentes passivos, como os capacitores C31 e C32 utilizados como capacitores de desacoplamento do regulador de tensão, estes capacitores agem como filtros de tensão, pois são responsáveis por evitar picos nas tensões de entrada e saída do C.I (TEXAS INSTRUMENTS, 2004). A Figura 7, ilustra o esquema elétrico utilizado para o regulador de tensão na PCB.

Figura 7 – Esquema elétrico do regulador de tensão.

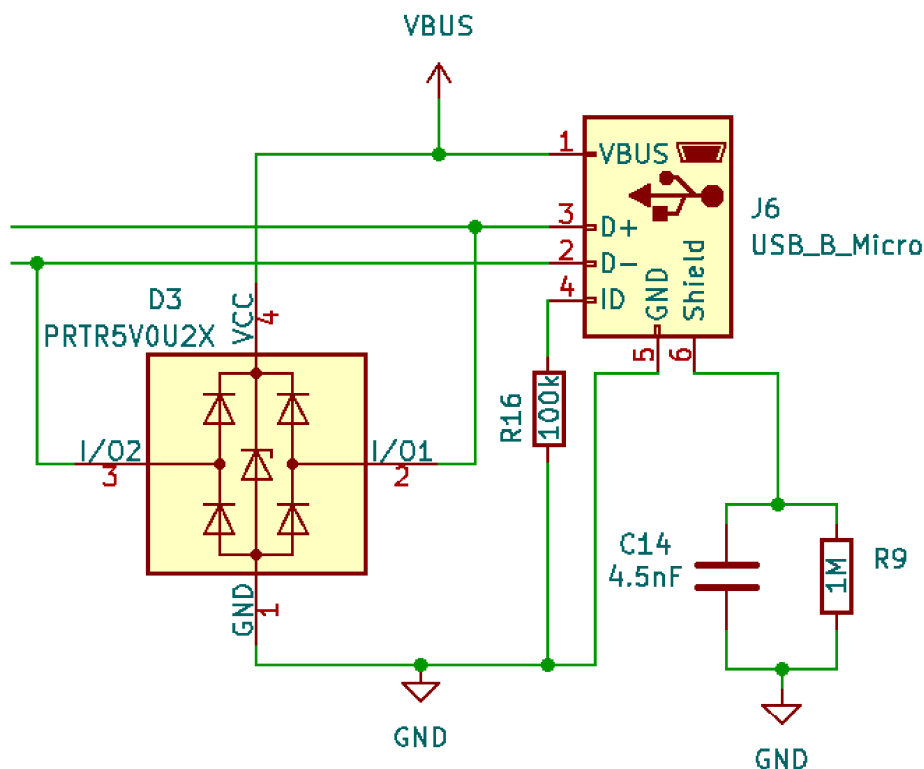


### 3.2.1.2 Circuito da USB

O circuito USB é constituído por uma porta USB micro, denominada 10103594-0001LF. Este circuito desempenha um papel crucial, fornecendo a alimentação de 5V Vbus e gerenciando um par diferencial que é responsável pela comunicação de dados D+ e D-. Adicionalmente, foi implementado um filtro passa-faixa no *shield* para mitigar interferências parasitas na faixa de frequência de RF (AMPHENOL, 2010).

Adicionalmente, foi implementado um circuito de proteção contra descargas eletrostáticas, conhecido como PRTR5V0U2X, nas linhas de comunicação de dados D+ e D-. Esse componente desempenha um papel fundamental na salvaguarda do circuito, uma vez que o C.I em questão está equipado com uma ponte de diodos TVS, que contribui para a proteção contra surtos eletrostáticos que podem danificar o circuito, o pino de ID foi aterrado com um resistor de *pull-down* R16, pois essa identificação não será necessária no contexto desta placa (NEXPERIA, 2023). A Figura 8 ilustra o esquema elétrico do circuito da USB.

Figura 8 – Esquema elétrico do circuito USB.



Fonte: Autoria própria (2023)

### 3.2.1.3 Circuito do ADC externo

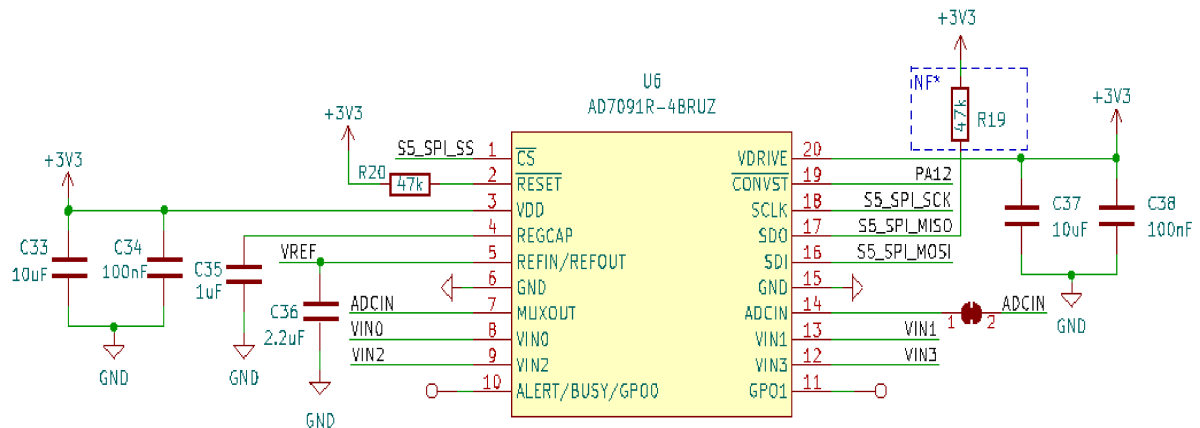
O circuito ADC é composto pelo ADC AD7091R-4BRUZ e por elementos passivos, como resistores e capacitores. Os resistores R19 e R20 atuam como resistores de *pull-up*,

enquanto os capacitores C33, C34, C35, C36, C37 e C38 funcionam como capacitores de desacoplamento. Esses capacitores desempenham a função de filtros de tensão, evitando picos de tensão na entrada do ADC e saída do regulador interno (ANALOG DEVICES, 2015). Além disso, as trilhas SPI SCK, SPI MISO, SPI MOSI e SPI SS estão diretamente conectadas ao microcontrolador para comunicação serial SPI.

As trilhas VIN0, VIN1, VIN2 e VIN3 representam os canais do ADC, que estão conectados a um barramento de pinos externo, estes canais são utilizados para a medição dos sensores. Outra função muito importante do ADC é o *reset*, o *reset* do ADC garante a inicialização correta do C.I, entretanto por falta de pinos no microcontrolador essa função será realizada por meio de *firmware* ao invés de *hardware*, por esta razão o pino 2 de *reset* possui uma tensão de 3v3 em conjunto com um resistor de *pull-up* R20, isso garante uma tensão de fixa de nível lógico alto neste pino, permitindo que o *reset* seja feito por pulsos no pino 19 (ANALOG DEVICES, 2015).

No circuito, é incorporado um *jumper* normalmente fechado no pino 14, além de pontos de teste disponíveis nos pinos 10 e 11. A Figura 9 representa o esquema elétrico do circuito do ADC, conforme discutido ao longo do texto.

Figura 9 – Esquema elétrico do circuito ADC.



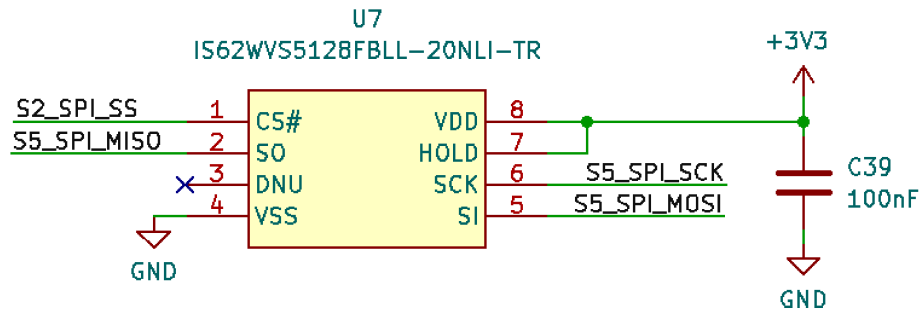
Fonte: Autoria própria (2023)

#### 3.2.1.4 Circuito da memória RAM externa

Devido aos requisitos do projeto, foi necessário a adição de um circuito da memória RAM externa, este circuito necessita de elementos passivos para seu funcionamento correto, portanto foi necessário colocar um capacitor de desacoplamento C39, que atua como um filtro de tensão para evitar picos na tensão de entrada do circuito integrado (ISSI, 2021). Além disso, por falta de pinos, foi utilizado as mesmas conexões SPI do ADC externo na RAM externa, ou seja, compartilham o mesmo barramento de SPI com a conexão das

trilhas SPI SCK, SPI MISO, SPI MOSI e SPI SS, que estão diretamente conectadas entre os C.I's e o microcontrolador. A Figura 10 representa o esquema elétrico do circuito da RAM externa,

Figura 10 – Esquema elétrico da memória RAM externa.

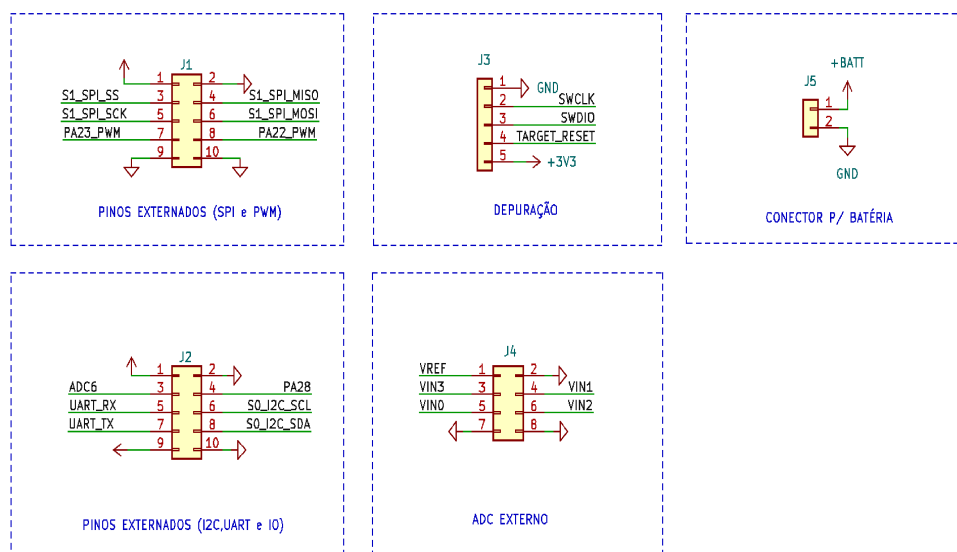


Fonte: Autoria própria (2023)

### 3.2.1.5 Conectores externos

Dentro do projeto, alguns pinos do microcontrolador foram disponibilizados externamente. Esses pinos incluem interfaces de comunicação serial, como SPI, I2C e UART, pinos relacionados a timers, I/O's, pinos destinados à depuração através do JLINK com trilhas para SWDIO E SWCLK, conector para bateria externa, entrada para o ADC interno do microcontrolador e canais do ADC externo onde serão utilizados os sensores. Na Figura 11 é ilustrado todos os pinos externados no microcontrolador,

Figura 11 – Esquema elétrico dos pinos externados.



Fonte: Autoria própria (2023)

### 3.2.1.6 Circuito de RF

No circuito RF, incorporamos um balun, um dispositivo que desempenha a função de converter um sinal de corrente alternada não balanceado (fim único) em um sinal de corrente alternada balanceado (diferencial) (JOHANSON, 2016). No contexto do nosso projeto RF, os terminais RF P e RF N representam um sinal balanceado que está sendo convertido em um sinal não balanceado (fim único). Esse sinal unificado é então direcionado ao módulo RF SKY66112-11, que desempenha o papel de um amplificador de potência (PA) e de um amplificador de baixo ruído (LNA) integrado.

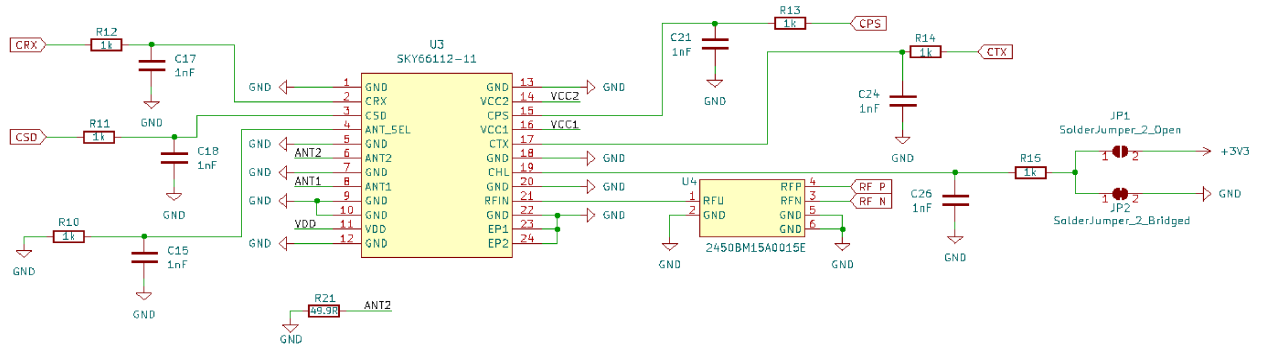
O objetivo principal desse módulo é melhorar a qualidade do sinal na transmissão e na recepção de dados, contribuindo para uma melhor performance do sistema, nesse contexto, no circuito integrado SKY66112-11, são empregados elementos passivos, como resistores e capacitores, nas entradas dos pinos do C.I, esses elementos passivos tem o propósito de criar filtros passa-baixa (C17 e R12, C18 e R11, C15 e R10, C21 e R13, C24 e R14, e por fim C26 e R15). Esses filtros passivos desempenham um papel importante ao serem posicionados nas entradas dos pinos digitais do C.I., atuando como salvaguardas para prevenir a reflexão de ondas de radiofrequência (RF) de volta ao microcontrolador. Isso ajuda a preservar a integridade do sinal e a evitar interferências indesejadas, adicionalmente estes pinos digitais ditam algumas configurações do circuito, como mostrado na tabela 2 (SKYWORKS, 2022).

Ademais foram inseridos dois *jumpers* abertos no pino 19, o JP1 conectado a 3v3 e o JP2 conectado ao GND, foi optado pela utilização de *jumpers* nesse pino, para a economia de pinos no microcontrolador. Os pinos de saída do C.I são ANT1 e ANT2. O pino ANT2 é conectado ao terra através de um resistor de *pull-down* de 49,9 Ohms, este valor de resistência foi utilizado para manter a impedância projetada de 50Ω. Quanto ao sinal proveniente do pino ANT1, ele passa pelo C.I 902-9040, que é um conector coaxial manual utilizado para avaliar a qualidade do sinal em um analisador de espectro. Além disso, o indutor L1 e os capacitores C16 e C19 são empregados para controlar a impedância da trilha em 50Ω. Por fim, a antena cerâmica 2450AT42B100E é utilizada para aprimorar a transmissão do sinal, garantindo uma melhor qualidade de comunicação, conforme a Figura 13.

O circuito integrado (C.I) recebeu alimentação por meio de uma tensão de 3,3V aplicada nos pinos VDD, VCC1 e VCC2, conforme ilustrado na Figura 14. Com o intuito de assegurar uma alimentação estável, foram estrategicamente empregados diversos capacitores de desacoplamento, a saber, C20, C22, C23, C25, C27, C28, C29 e C30 (SKYWORKS, 2022).

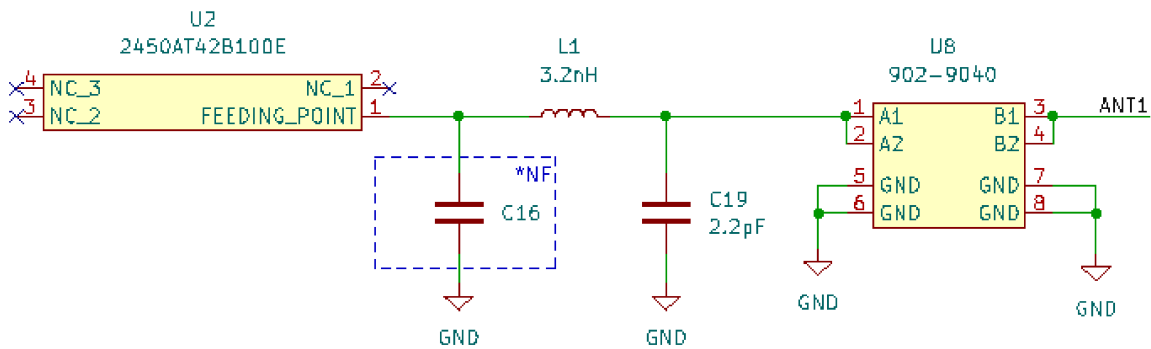
Esses componentes capacitivos desempenham a crucial função de filtros de tensão, incumbindo-se de prover a voltagem necessária ao funcionamento do circuito integrado.

Figura 12 – Esquema elétrico do SKY66112-11.



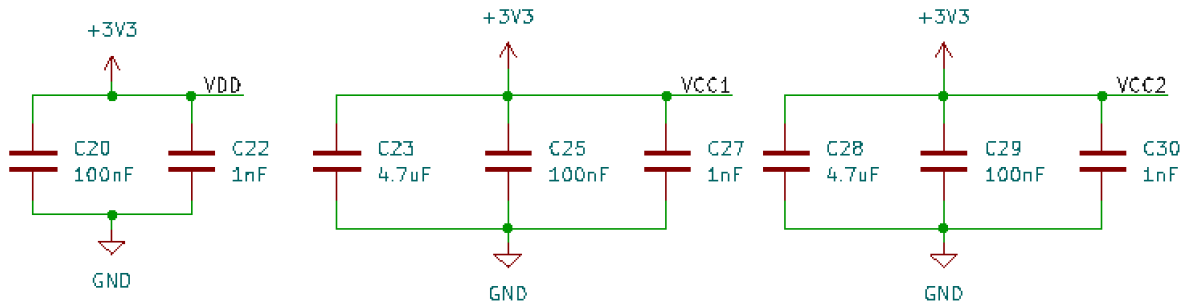
Fonte: Autoria própria (2023)

Figura 13 – Esquema elétrico da antena.



Fonte: Autoria própria (2023)

Figura 14 – Esquema elétrico da alimentação do SKY66112-11.



Fonte: Autoria própria (2023)

Tabela 2 – Controle lógico SKY66112-11.

Modo	Descrição	CSD	CPS	CRX	CTX	CHL	ANT SEL
0	All off (sleep mode)	0	X	X	X	X	X
1	Receive LNA mode	1	0	1	0	X	X
2	Transmit high-power mode	1	0	X	1	1	X
3	Transmit low-power mode	1	0	X	1	1	X
4	Receive bypass mode	1	1	1	0	X	X
5	Transmit bypass mode	1	1	X	1	X	X
6	All off (sleep mode)	1	X	0	0	X	X
-	ANT1 port enable	X	X	X	X	X	0
-	ANT2 port enable	X	X	X	X	X	1

### 3.2.1.7 Circuito do microcontrolador

O microcontrolador depende de vários componentes externos para garantir seu funcionamento adequado. Entre esses componentes, destacam-se os cristais osciladores. No circuito, existem dois cristais osciladores externos: Y1, que é um cristal oscilador de 32,768kHz, em conjunto com os capacitores em paralelo C2 e C7 (ECS, 2019). Esses capacitores desempenham o papel de estabilização da frequência do cristal. Da mesma forma, o cristal externo Y2 de 16MHz é utilizado como oscilador externo para o rádio interno AT86RF233, com os capacitores C1 e C4 cumprindo a mesma função que os capacitores C2 e C7, contribuindo para a estabilidade do cristal (ABRACON, 2019).

Os capacitores C3, C5, C6, C8, C9, C10, C11, C12 e C13 desempenham um papel crucial como capacitores de desacoplamento no circuito em questão. Esses componentes desacopladores são essenciais para mitigar flutuações indesejadas nas fontes de alimentação, garantindo a estabilidade operacional do sistema. Além disso, o componente FB1, constituído por uma ferrite, desempenha uma função adicional de desacoplamento da tensão de entrada no pino 6, contribuindo significativamente para a redução de ruído em faixas de frequência específicas. A inclusão de resistores de *pull-up* nos pinos 15 e 16 é estrategicamente realizada para a facilitação da comunicação serial I2C, enquanto o resistor R18 é designado para assegurar a funcionalidade adequada da linha de SWCLK durante fases de depuração. Da mesma forma, o resistor R4 atua como um resistor de *pull-up* na linha de reset, contribuindo para a estabilidade dessa funcionalidade essencial no contexto do sistema em análise (MICROCHIP TECHNOLOGY, 2016).

Por outro lado, os resistores R1, R5, R6, R7 e R8 desempenham um papel na limitação da corrente nos pinos correspondentes, desse modo exercendo controle preciso sobre o fluxo de energia no circuito. A inclusão desses resistores visa garantir operações seguras e estáveis. Adicionalmente, a presença dos LEDs D1 e D2 no circuito serve como um meio visual para fornecer indicações perceptíveis de potenciais eventos ou estados operacionais ao longo do desenvolvimento do projeto. Esses indicadores visuais contribuem





transmissão de variáveis. Nesta subseção, exploraremos em detalhes a abordagem adotada no projeto da placa para garantir o desempenho ideal do sistema.

### 3.2.2.1 Camadas da PCB

Para esta placa de circuito impresso (PCB), optou-se por uma estrutura composta por quatro camadas de cobre, pois essas placas oferecem maior flexibilidade no design, permitindo que as trilhas e os componentes sejam organizados de maneira mais eficiente.

As placas de circuito impresso de quatro camadas são comumente escolhidas quando se busca um projeto mais complexo ou quando é necessário acomodar um grande número de componentes eletrônicos em um espaço limitado.

As quatro camadas de cobre em uma PCB típica são organizadas da seguinte forma:

- **Camada Superior (*Top Layer*):** Esta é a camada mais externa da placa e é onde a maioria dos componentes é montada. Trilhas de cobre são usadas para conectar os componentes entre si e com outros pontos da placa.
- **Camada Interna 1:** Esta é a primeira camada interna, localizada abaixo da camada superior. Foi utilizada apenas como aterramento.
- **Camada Interna 2:** Esta é a segunda camada interna, localizada abaixo da camada interna 1. Foi utilizada como aterramento e trilhas de alimentação.
- **Camada Inferior (*Bottom Layer*):** Esta é a camada mais interna da PCB e é semelhante à camada superior. Ela foi utilizada para conectar os componentes e trilhas que estão localizados na parte inferior da placa.

Na Tabela 3, é apresentado a disposição de todas as camadas na placa:

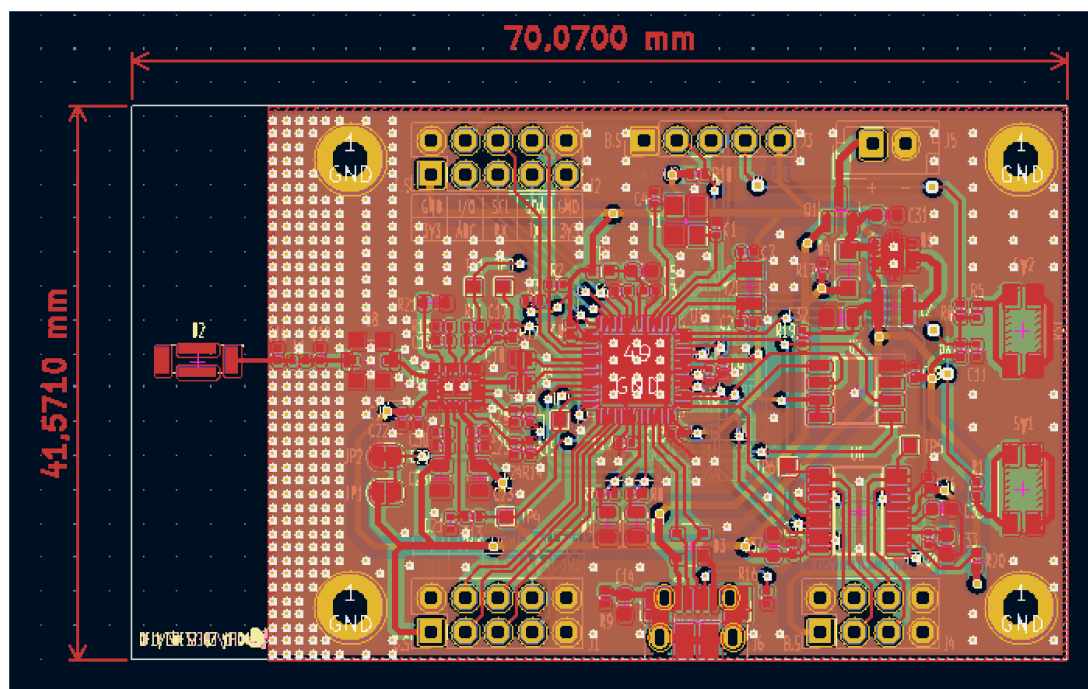
Tabela 3 – Disposição de camadas na PCB.

Camada	Nome	Tipo	Material	Espessura (mm)	Função
Camada 1	F.cu	Cobre	-	0,035	Sinal/Terra
Dielétrico	Dielétrico 1	PrePreg	FR4	0,2104	-
Camada 2	In1.cu	Cobre	-	0,0152	Terra
Dielétrico	Dielétrico 2	Núcleo	FR4	1,065	-
Camada 3	In2.cu	Cobre	-	0,0152	Alimentação/Terra
Dielétrico	Dielétrico 3	PrePreg	FR4	0,2104	-
Camada 4	B.cu	Cobre	-	0,035	Sinal/Terra

Fonte: Autoria própria (2023)

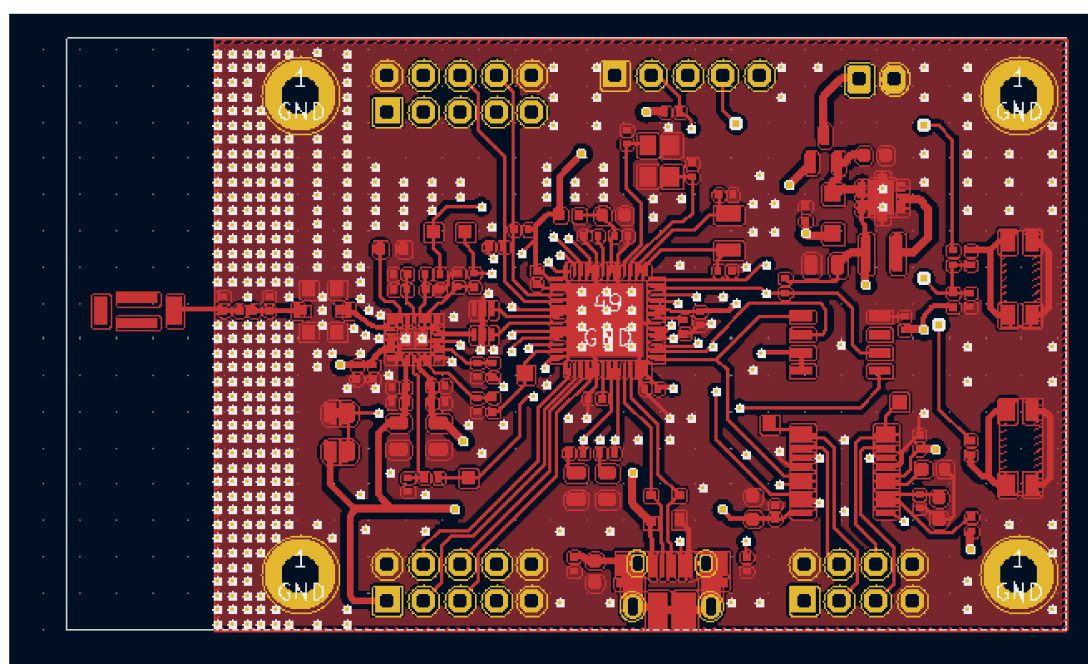
A seguir, é apresentado os *layouts* da PCB correspondentes às quatro camadas de cobre (F.cu, In2.cu, In3.cu e B.cu), conforme especificado na Tab. 3:

Figura 16 – Layout da PCB (Todas as camadas).



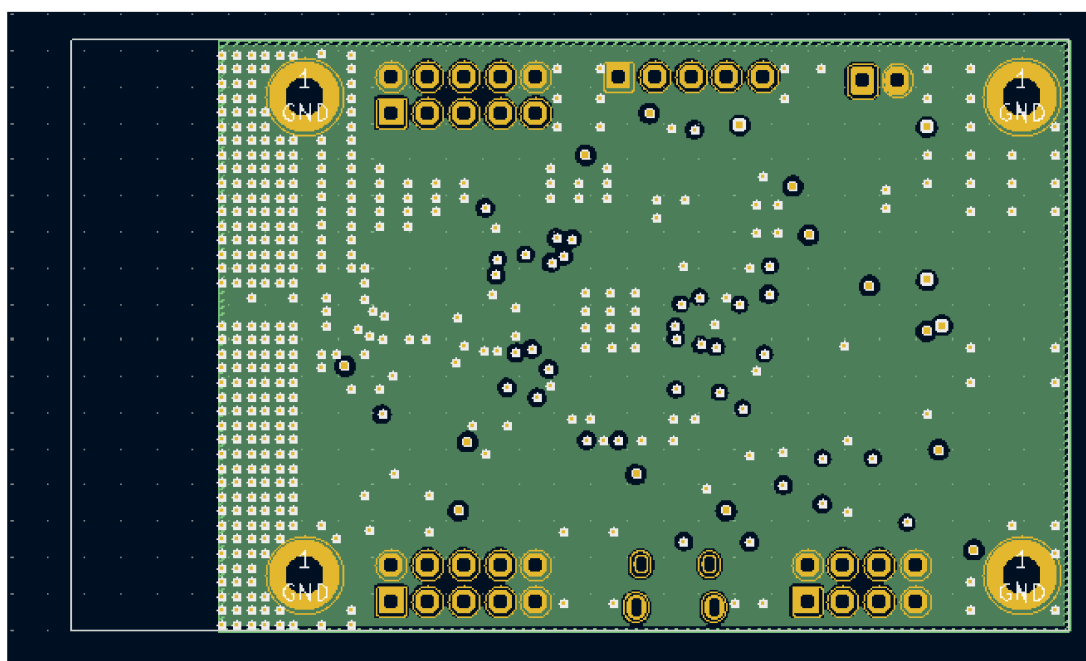
Fonte: Autoria própria (2023)

Figura 17 – Layout da PCB (Primeira camada).



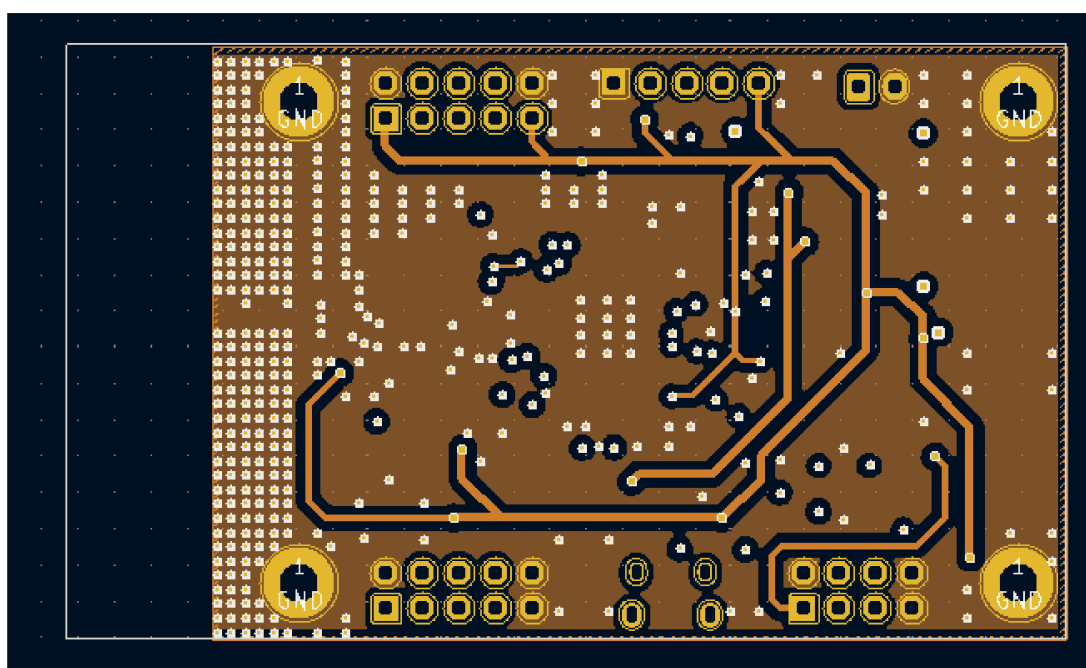
Fonte: Autoria própria (2023)

Figura 18 – Layout da PCB (Segunda Camada).



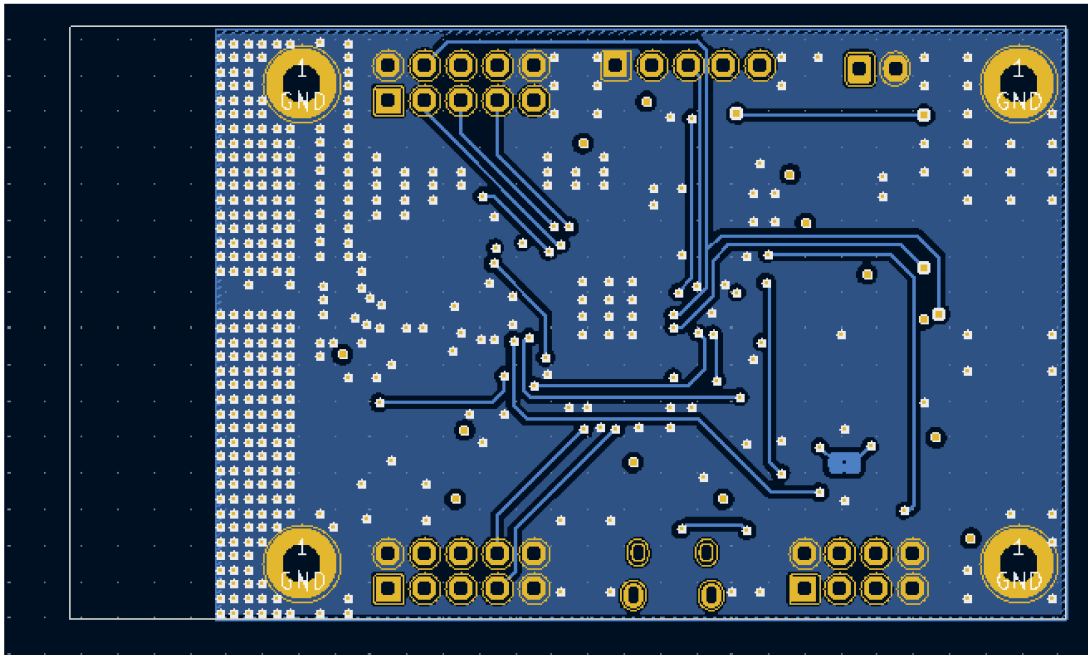
Fonte: Autoria própria (2023)

Figura 19 – Layout da PCB (Terceira Camada).



Fonte: Autoria própria (2023)

Figura 20 – Layout da PCB (Quarta Camada).



Fonte: Autoria própria (2023)

Conforme evidenciado na Tabela 3, as Figuras 17, 18, 19 e 20 representam as quatro camadas de cobre da PCB. A Figura 17 corresponde à primeira camada da PCB, conhecida como a camada superior (*top*). Esta camada é responsável pela distribuição de sinal e terra. A Figura 18 representa a segunda camada, que é exclusivamente dedicada ao aterramento e está localizada logo abaixo da primeira camada de cobre. Em seguida, na Figura 19, temos a terceira camada, que foi projetada para distribuir a alimentação da placa, incluindo tensões de 5V, 3V3 e também funções de aterramento.

Por fim, a Figura 20 ilustra a última camada de cobre, conhecida como camada 4 (*bottom*). Nesta camada, ocorre a distribuição de sinal e aterramento. Este arranjo de camadas tem o intuito de diminuir ao máximo as interferências eletromagnéticas da PCB e garantir a integridade do circuito, permitindo a segregação eficaz das diferentes funções e conexões da placa.

### 3.2.2.2 Posicionamento de componentes

O posicionamento dos componentes desempenha um papel fundamental no *layout*, pois determina a disposição das conexões, trilhas e vias na PCB. Portanto, o posicionamento de cada componente foi planejado, considerando suas funções em relação ao microcontrolador. Como ilustrado na Figura 16, o circuito foi dividido em seções distintas.

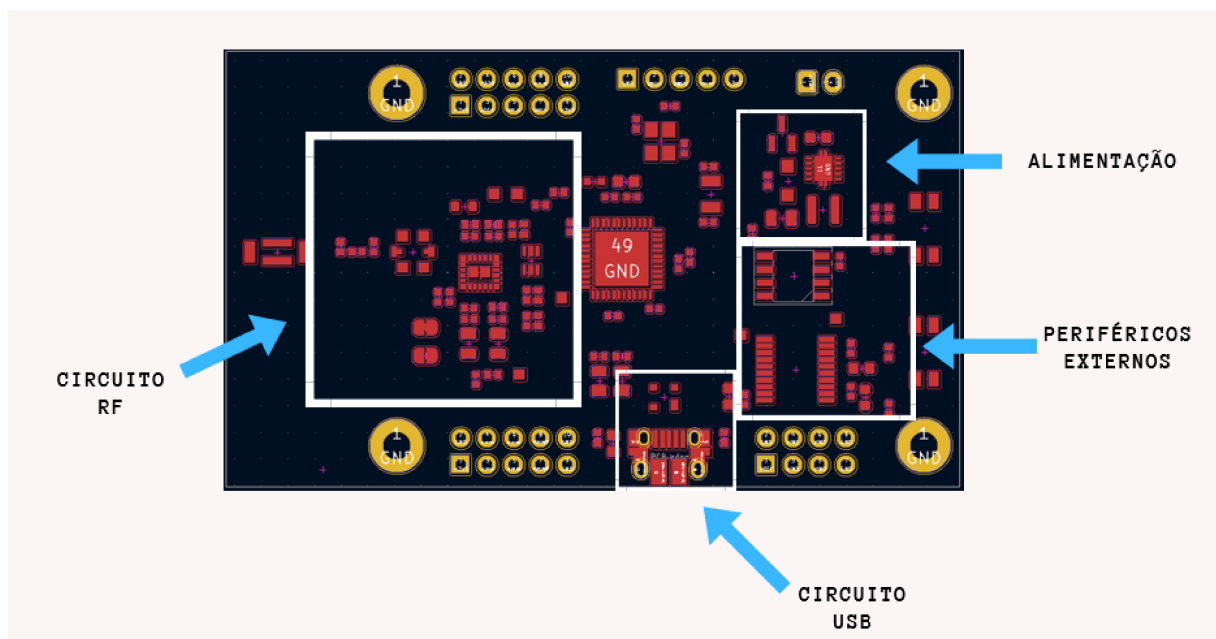
O microcontrolador é situado na parte central da PCB, com seus pinos 18 e 19 (RF P e RF N) localizados à esquerda da placa, resultando na alocação de todos os componentes relacionados à RF nessa região. Da mesma forma, os pinos 33 e 34 (USB DM e USB DP)

estão na parte inferior da PCB, levando à disposição de toda a circuitaria USB ao sul do microcontrolador.

Na região leste do microcontrolador, encontram-se os periféricos externos, como RAM e ADC. Essa disposição foi determinada pela limitação de espaço e pela necessidade de compartilhar o mesmo barramento SPI, tornando importante a proximidade física. Também nessa área, os componentes de regulagem de tensão foram posicionados. Embora pudesse ser colocado em qualquer lugar da placa, optou-se por alocá-lo afastado da parte de RF devido à natureza chaveada do regulador de tensão.

A zona norte da PCB foi reservada para os pinos externados do microcontrolador, incluindo as conexões de *debug*, entrada de tensão e especialmente os pinos externados de comunicação serial do microcontrolador. A Figura 21 evidencia o posicionamento dos componentes na PCB.

Figura 21 – Posicionamento de componentes na PCB.



Fonte: Autoria própria (2023)

### 3.2.2.3 Aterramento

O aterramento desempenha um papel crítico e fundamental no design de uma PCB, sendo um dos pilares essenciais. Um sistema de aterramento bem concebido desempenha um papel crucial na redução das interferências eletromagnéticas (EMI) e no controle do *crosstalk*, contribuindo de forma significativa para o desempenho e a confiabilidade do circuito.

A criação de um aterramento eficaz requer a implementação de planos de terra e vias de terra, como ilustrado nas Figuras 17, 18, 19 e 20. Em todas as camadas da PCB, há a presença desses planos de terra e eles são fundamentais para otimizar o sistema de

aterramento. As vias de terra desempenham um papel vital na PCB, pois através delas é possível permitir um caminho de baixa impedância para que a corrente retorne ao ponto de referência além de interligar os planos.

A importância disso se reflete na prevenção de diferenças de potencial entre os vários componentes da PCB. Reduzindo essas diferenças de potencial, o risco de EMI é substancialmente diminuído, o que é fundamental para evitar perturbações no funcionamento de outros dispositivos eletrônicos e interferências em comunicações sem fio. É por esse motivo que optamos por utilizar um número considerável de vias de terra nas proximidades do circuito de RF, onde a sensibilidade a EMI é particularmente crítica.

Além disso, um sistema de aterramento adequado desempenha um papel fundamental na minimização do *crosstalk*, que se refere à interferência entre sinais que percorrem trilhas adjacentes na PCB. Esse aspecto é especialmente crucial em PCB's com alta densidade de componentes e em circuitos sensíveis, como aqueles operando em altas frequências.

#### 3.2.2.4 Trilhas

As trilhas desempenham um papel crítico na PCB, pois são responsáveis por entrelaçar os componentes eletrônicos. A largura e espessura dessas trilhas são determinadas pela equação apresentada na Equação 3.1. Com base em parâmetros de entrada, como a corrente, a variação de temperatura e o comprimento do condutor, podemos calcular as dimensões ideais das trilhas em pontos específicos do circuito.

Neste projeto, que envolve baixa potência e operação em temperatura ambiente, a maioria das trilhas segue um padrão semelhante. No entanto, algumas trilhas relacionadas à alimentação foram dimensionadas com margem de segurança. É importante destacar que trilhas mais estreitas contribuem para uma menor impedância no sistema, o que, por sua vez, ajuda a reduzir as interferências eletromagnéticas no sistema. De acordo com a norma IPC-2221A *Generic Standard on Printed Board Design*, temos a Equação 3.1 (IPC-2221A, 2023). Vale salientar que esses cálculos são válidos para correntes de 35A (externo) ou 17,5A (interno), aumentos de temperatura de até 100 graus Celsius e larguras com até 400 mil (10 mm).

$$I = k \cdot \Delta T^{0.44} \cdot (W \cdot H)^{0.725} \quad (3.1)$$

Onde:



$I$  é a corrente em ampères.

$k$  é uma constante 0,024 para trilhas internas ou 0,048 para trilhas externas.

$\Delta T$  é o aumento da temperatura acima do ambiente em graus Celsius.

$W$  é a largura da trilha em mils.

$H$  é a altura da trilha em unidades correspondentes à largura da trilha.

É de extrema importância sublinhar que, em virtude da presença do circuito RF, a PCB incorpora trilhas com controle de impedância. Esse controle desempenha um papel primordial ao assegurar uma correspondência precisa aos 50 ohms projetados, um requisito essencial para minimizar perdas na transmissão e recepção de dados.

No tocante à largura das trilhas de RF, é imperativo considerar as especificações escolhidas para a PCB. Nas Figuras 24 e 25, são evidenciados os parâmetros de fabricação da PCB, sendo esses os mesmos que serão empregados na calculadora de impedância da JLCPCB, conforme ilustrado na Figura 3. A partir desses dados de fabricação, foi possível determinar que a largura da trilha na seção de RF seria aproximadamente 0,3493mm para a impedância projetada de 50 ohms. Na Figura 23, todos os parâmetros utilizados são apresentados, com a largura de trilha resultante destacada em verde para uma compreensão visual clara.

Ao obter a informação sobre a largura da trilha, pode-se avançar no processo de cálculo da constante dielétrica efetiva. Essa etapa crucial proporciona uma compreensão mais profunda das propriedades dielétricas do material utilizado na PCB e será conduzida por meio de uma calculadora online especializada (PASTERNAK, 2023). Com esse valor, torna-se possível calcular a "critical length" ao utilizar a Equação 2.2.



Figura 22 – Cálculo da constante dielétrica efetiva.

Note: Units for width and height must be consistent

**Calculation**

Dielectric Constant: 4.6

Width: 0.349 Millimeters

Height: 0.2 Millimeters

**CALCULATE**

---

**Result:**  
 Width/Height: 1.745  
 Effective Dielectric Constant: 3.441  
 Impedance: 51.96 Ω

Fonte: Aatoria própria (2023)

Figura 23 – Cálculo da largura de trilha.

**JLCPCB Impedance Calculator**

The JLCPCB Impedance Calculator computes track width values and recommended stack-ups from user-input values of board layer, thickness, copper weight, target impedance, trace spacing (for edge-coupled pairs), and impedance trace to copper gap (coplanar waveguides). [Click here](#) to see the user guide of our impedance calculator.

\* The first stack-up calculated below is highly recommended as it has lowest cost and quickest turn-around.

Layers: 4 PCB Thickness: 1.6 Inner Copper Weight: 0.5oz Outer Copper Weight: 1oz Unit: mm

**Impedance Configure** + New Impedance Duplicate Impedance Calculate

Impedance (Ω)	Type	Signal Layer	Top Ref	Bottom Ref	Trace Spacing (mm)	Impedance trace to copper (mm)
50	Single Ended (Non coplanar)	L1	/	L2	/	/

[JLC04161H-7628\(Standard\)](#)
[JLC04161H-3313](#)
[JLC04161H-3313A\(Special\)](#)
[JLC04161H-7628B\(Special\)](#)
[JLC04161H-7628A\(Special\)](#)
[JLC04161H-2116A\(\)](#)

Impedance (Ω)	Type	Signal Layer	Top Ref	Bottom Ref	Trace Width	Trace Spacing	Impedance trace to copper
50	Single Ended (Non coplanar)	L1	/	L2	0.3493	/	/
Layer	Material	Thickness (mil)		Thickness (mm)			
L1	Outer Copper Weight1oz	1.38		0.0350			
Prepreg	7628, RC 49%, 8.6 mil	8.28		0.2104			
L2	Inner Copper Weight	0.60		0.0152			
Core	1.1mm HHOZ with copper	41.93		1.0650			
L3	Inner Copper Weight	0.60		0.0152			
Prepreg	7628, RC 49%, 8.6 mil	8.28		0.2104			
L4	Outer Copper Weight1oz	1.38		0.0350			

Fonte: Aatoria própria (2023)

### 3.2.2.5 Fabricação da Placa

A placa de circuito impresso foi fabricada na empresa JLCPCB *factory*. A JLCPCB é uma empresa chinesa conhecida no mundo inteiro como uma das maiores fabricantes globais de PCB e uma fabricante de alta tecnologia especializado em protótipos rápidos de

PCB, produção de pequenos lotes de PCB e impressão 3D. Ao fazer o pedido de uma PCB na JLCPCB é necessário preencher algumas especificações da placa, essas especificações estão situadas nas Figuras. 24 e 25.

Figura 24 – Especificações da PCB.

The image shows a screenshot of the 'PCB Specifications' form on the JLCPCB website. The form is organized into several rows, each with a label on the left and a set of options on the right. The options are as follows:

- Different Design:** A row of five buttons labeled 1, 2, 3, 4, and an empty box. Button 1 is selected.
- Delivery Format:** Three buttons: 'Single PCB' (selected), 'Panel by Customer', and 'Panel by JLCPCB'.
- PCB Thickness:** A row of seven buttons labeled 0.4, 0.6, 0.8, 1.0, 1.2, 1.6, and 2.0. Button 1.6 is selected.
- PCB Color:** A row of seven buttons with colored circles: Green, Purple, Red, Yellow, Blue (selected), White, and Black.
- Silkscreen:** A row of two buttons: 'White' (selected) and an empty box.
- Material Type:** A row of three buttons: 'FR4-Standard TG 135-140' (selected), 'FR-4 TG155', and 'FR-4 TG170'.
- Surface Finish:** A row of three buttons: 'HASL(with lead)' (selected), 'LeadFree HASL', and 'ENIG'.

Fonte: Autoria própria (2023)

Na figura 24, é possível identificar alguns parâmetros das especificações de fabricação da PCB:

- *Different Design:* Este parâmetro possibilita a escolha entre designs distintos de PCB presentes nos meus arquivos de fabricação. Esta opção é relevante quando há múltiplos layouts de placas de circuito impresso disponíveis (JLCPCB, 2023b).
- *Delivery Format:* Este termo refere-se à configuração na qual as PCBs são entregues. No presente caso, como foi produzida apenas uma placa nos arquivos de fabricação, a opção selecionada foi "single PCB"— ou seja, a entrega de uma única placa de circuito impresso (JLCPCB, 2023b).
- *PCB Thickness:* Neste parâmetro, optamos pela definição da espessura da placa de circuito impresso, estabelecendo-a em 1,6 mm conforme as camadas finais da disposição. A escolha dessa medida é crucial para atender aos requisitos específicos do projeto (JLCPCB, 2023b).
- *PCB Color:* A coloração da placa de circuito impresso foi selecionada como azul, uma escolha estratégica para proporcionar destaque adicional à serigrafia na PCB. Esta decisão visa não apenas a estética, mas também a facilidade de identificação e leitura das marcações presentes na superfície da placa (JLCPCB, 2023b).
- *Silkscreen:* Esta opção determina a cor da serigrafia. Neste caso, foi selecionado a cor branca (JLCPCB, 2023b).

- *Material Type*: Este parâmetro define o tipo de dielétrico da PCB. O dielétrico escolhido foi o FR-4, ele é uma designação de grau NEMA para material laminado epóxi reforçado com vidro. No caso desse dielétrico ele possui o sufixo "TG", a característica de resistência ao calor do FR4 é indicada por esse parâmetro "TG"(temperatura de transição vítrea), ele determina onde ocorre a mudança do estado sólido para um estado macio e emborrachado (JLCPCB, 2023b).
- *Surface Finish*: O acabamento superficial da PCB é aplicado às áreas de cobre expostas para protegê-las da oxidação, o que prejudicaria a capacidade da placa de formar conexões com a solda derretida. O acabamento escolhido foi o HASL, que é um tipo de acabamento comum usado em placas de circuito impresso (PCBs). Para o acabamento da nivelamento da solda por ar quente, a placa será imersa em um banho de solda derretida e, em seguida, passará por uma esteira com facas de ar quente que removerá o excesso de solda (JLCPCB, 2023b).

Na Figura 25, são apresentadas as opções de especificações avançadas da PCB. Essas opções podem determinar os seguintes parâmetros da PCB:

Figura 25 – Opções de especificações avançadas.

The image shows a screenshot of the 'High-spec Options' configuration page on the JLCPCB website. The page is organized into a list of settings, each with a question mark icon and a set of buttons or input fields. The selected options are highlighted in blue. The settings include:

- Outer Copper Weight**: 1 oz
- Inner Copper Weight**: 0.5 oz
- Impedance Control**: Yes (with a link to 'Impedance calculator >')
- Layer Stackup**: JLC04161H-7628A
- Via Covering**: Tented
- Min via hole size/diameter**: 0.3mm/0.4/0.45mm
- Board Outline Tolerance**: ±0.2mm(Regular)
- Confirm Production file**: No
- Remove Order Number**: No
- Flying Probe Test**: Fully Test
- Gold Fingers**: No
- Castellated Holes**: No

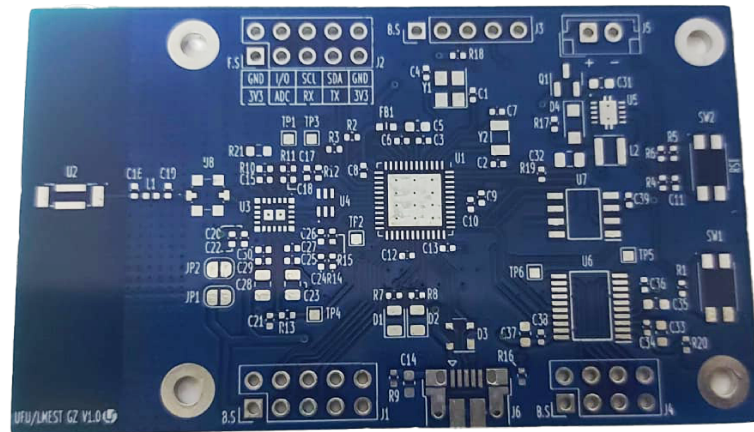
Fonte: Autoria própria (2023)

- *Outer Copper Weight*: Refere-se ao peso da camada externa de cobre em onças. Neste caso, foi selecionada uma espessura de 1 oz (JLCPCB, 2023b).

- *Inner Copper Weight*: Indica o peso da camada interna de cobre em onças. Neste caso, foi escolhida uma espessura de 0,5 oz (JLCPCB, 2023b).
- *Impedance Control*: Este parâmetro determina se a PCB requer controle de impedância. Devido ao uso de circuitos de RF nesta PCB e à necessidade de uma correspondência de impedância de  $50\Omega$  nessas trilhas, a opção marcada foi "sim"(JLCPCB, 2023b).
- *Layer Stackup*: Este parâmetro determina a disposição das camadas da PCB. A disposição escolhida foi a *stack-up JLC04161H-7628*, a espessura e os materiais desse *stack-up* podem ser observados na Tabela 3 (JLCPCB, 2023b).
- *Via Covering*: Este parâmetro determina se as vias serão cobertas ou não pela máscara de solda. No caso desta PCB, elas serão cobertas (JLCPCB, 2023b).
- *Min via hole size/diameter*: Este parâmetro define o diâmetro mínimo das vias na PCB (JLCPCB, 2023b).
- *Board Outline Tolerance*: Este parâmetro define a tolerância para componentes, trilhas e vias nas bordas da PCB (JLCPCB, 2023b).
- *Confirm Production File*: Os engenheiros da JLCPCB criarão arquivos de produção para a PCB. Neste caso, esta opção não foi necessária para a confecção desta PCB (JLCPCB, 2023b).
- *Remove Order Number*: Determina se o número do pedido de produção estará escrito na placa (JLCPCB, 2023b).
- *Flying Probe Test*: A própria JLCPCB realiza um teste para verificar se há curtos-circuitos ou circuitos abertos na PCB (JLCPCB, 2023b).
- *Gold Finger*: São colunas banhadas a ouro ao longo das bordas de conexão da PCB. Não foi necessário para esta PCB (JLCPCB, 2023b).
- *Castellated Holes*: Este parâmetro determina se há necessidade de criar furos castelados na PCB. No caso desta PCB, não foi necessário, pois não foram utilizados furos castelados (JLCPCB, 2023b).

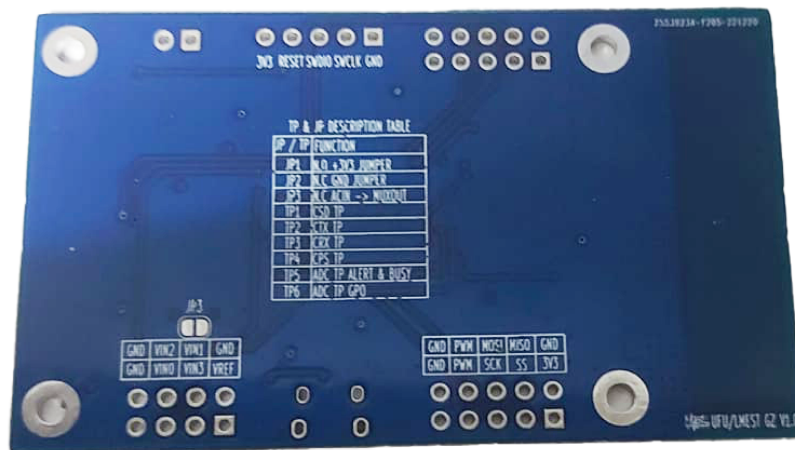
Após a encomenda das PCB's via JLCPCB com os parâmetros selecionados de acordo com as Figuras 24 e 25. As figuras 26 e 27 evidenciam o resultado final da PCB fabricada (JLCPCB, 2023b).

Figura 26 – Placa fabricada pela JLCPCB (Vista da primeira camada).



Fonte: Autoria própria (2023)

Figura 27 – Placa fabricada pela JLCPCB (Vista da quarta camada).



Fonte: Autoria própria (2023)

### 3.2.2.6 Compra e montagem dos componentes na PCB

Após a conclusão do esquema elétrico e do layout na PCB, o próximo passo é a elaboração do "BOM- *Bill of Materials*", que consiste fundamentalmente em uma relação de todos os componentes planejados para a PCB. Esta relação é apresentada em duas tabelas, denominadas Tabela 4 e Tabela 5, nas quais são fornecidas informações detalhadas sobre a referência, quantidade e preço dos componentes.

É importante ressaltar que todos os componentes passivos e ativos mencionados nas Tabelas 4 e 5 já foram previamente detalhados na subseção de esquema elétrico no início do Capítulo 3.

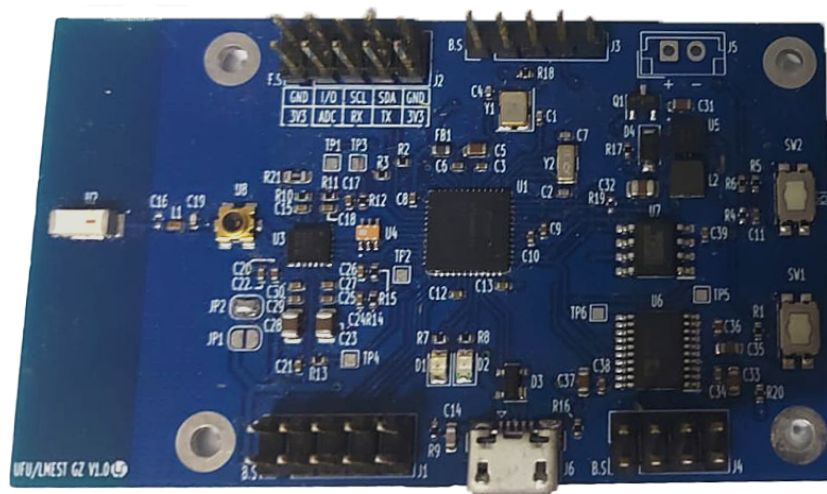
A aquisição e a integração dos componentes na placa de circuito impresso (PCB) foram conduzidas por um parceiro externo, a *Tardigrado Comércio Varejista de Compo-*

*mentes Eletrônicos*. Esta decisão de terceirização foi tomada com base na especialização e experiência da empresa em lidar com as complexidades e requisitos específicos da montagem de placas de circuito.

Para garantir a qualidade e disponibilidade dos componentes necessários, optamos por fazer compras em fornecedores renomados e confiáveis, como a *Mouser* e a *DigiKey*. A escolha dessas empresas se deveu à sua vasta seleção de componentes eletrônicos, bem como à reputação de entrega pontual e produtos de alta qualidade.

Ao confiar a compra e a montagem dos componentes a parceiros especializados, pode-se otimizar tempo e recursos internos, concentrando os esforços em outras etapas críticas do projeto. Isso permitiu garantir a eficiência e a precisão na construção da PCB, ao mesmo tempo em que é assegurado a qualidade dos componentes utilizados, proporcionando um resultado final confiável e de alto desempenho. Na Figura 28 é ilustrado a PCB das Figura 26 e 27 montada com os componentes das Tabelas 4 e 5.

Figura 28 – Placa fabricada pela JLCPCB e montada.



Fonte: Autoria própria (2023)

### 3.2.3 Testes de Hardware

Após receber a placa de circuito impresso (PCB) montada, foi realizada uma série de testes de *hardware* cruciais para garantir o funcionamento correto da placa. Durante esses testes, foi identificado um problema de conexão no circuito do Conversor Analógico-Digital (ADC), especificamente no pino 3 (VDD). Neste pino, há dois capacitores de desacoplamento, C33 e C34. No entanto, o capacitor C33 não estava devidamente conectado à alimentação de 3,3V.

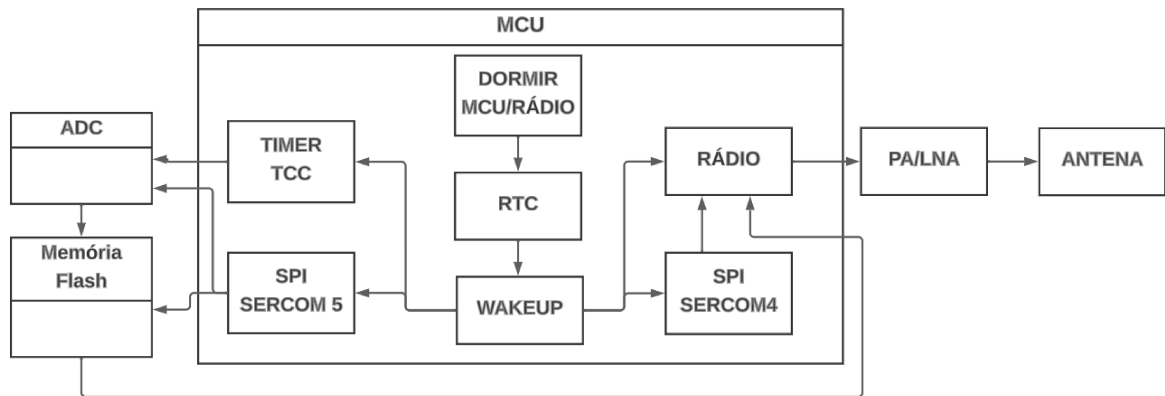
Para solucionar esse problema, foi necessário realizar uma adaptação na PCB. Foi soldado um *jumper* entre o pino 3 e o ponto de alimentação mais próximo, assegurando





contribuindo para a funcionalidade e desempenho do *firmware*. A seguir, discutiremos em detalhes os *clocks*, *timers* de controle, RTC (Relógio de Tempo Real), memória externa e o ADC (Conversor Analógico-Digital), explorando suas implicações no contexto do projeto.

Figura 30 – Diagrama do *firmware*.



Fonte: Autoria própria (2023)

### 3.3.1 Clock

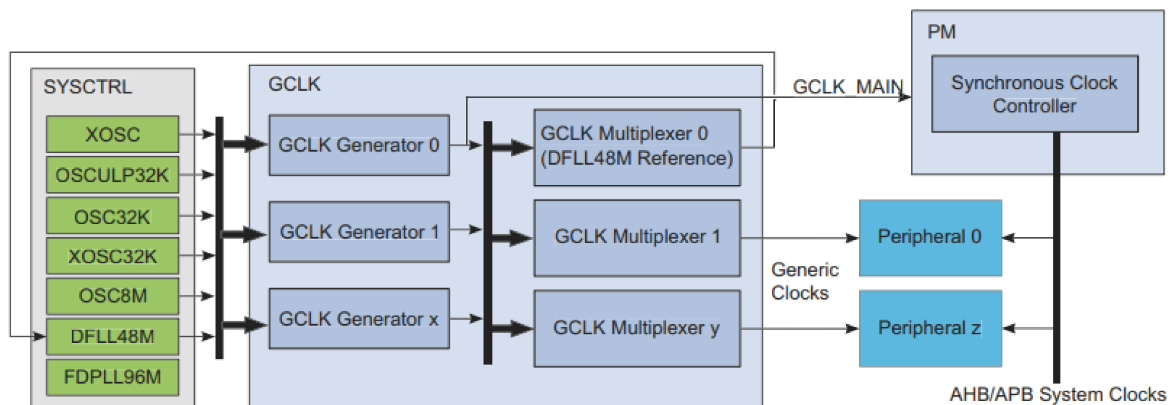
Para o entendimento dos tópicos seguintes, é necessário compreender o funcionamento da árvore de *clock* utilizada no ATSAMR21G17A, na Figura 31 é possível ver o funcionamento dessa árvore de *clock*. Em suma, as fontes de *clock* são os sinais que fornecem a base de tempo para o sistema (MICROCHIP TECHNOLOGY, 2016).

- As fontes de *clock* são controladas pelo SYSCTRL. Algumas fontes de *clock* contidas no microcontrolador são o oscilador interno de 8 MHz (OSC8M), o oscilador de cristal externo (XOSC) e o loop bloqueado de frequência digital (DFLL48M). (MICROCHIP TECHNOLOGY, 2016)
- Já o *Generic Clock Controller (GCLK)* é um circuito que controla a distribuição de relógios no sistema. Ele é composto por três partes principais:
- Geradores de relógio genérico: Esses geradores podem ser programados para usar qualquer fonte de relógio do sistema como sua fonte de origem. Eles também podem ser programados para dividir o relógio de origem por um fator específico (MICROCHIP TECHNOLOGY, 2016).
- Relógios genéricos: Esses relógios são as entradas de relógio para os periféricos do sistema. Eles podem ser conectados a qualquer um dos geradores de relógio genérico (MICROCHIP TECHNOLOGY, 2016).



- Gerenciador de energia (PM): O PM controla os relógios síncronos do sistema, incluindo a CPU, relógios de barramento e interfaces de usuário síncronas dos periféricos (MICROCHIP TECHNOLOGY, 2016).
- O GCLK permite que os periféricos do sistema sejam alimentados com relógios de diferentes frequências, o que pode ajudar a economizar energia e melhorar o desempenho (MICROCHIP TECHNOLOGY, 2016).

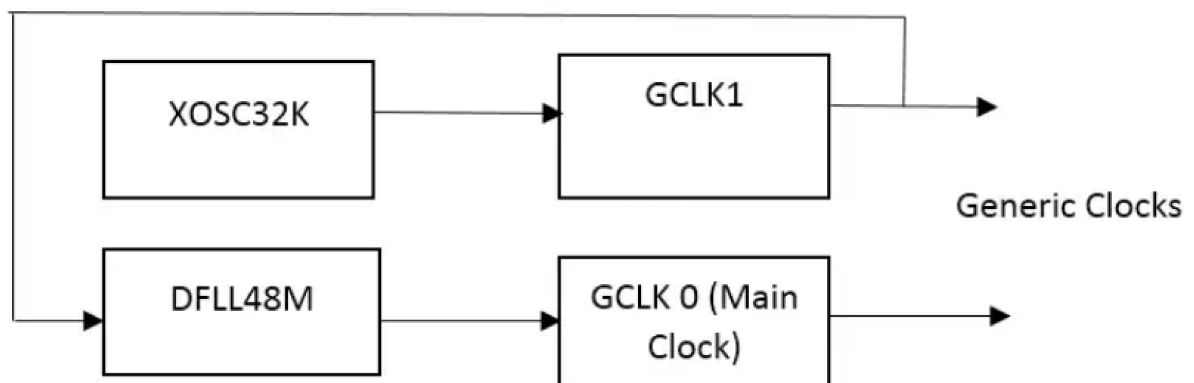
Figura 31 – Diagrama de clock ATSAMR21G17A.



Fonte: Adaptado de (MICROCHIP TECHNOLOGY, 2016).

A partir desses conceitos, foi escolhido que o *clock* principal responsável por alimentar o microcontrolador é o DFLL48M. Sua configuração pode ser visualizada no diagrama da Figura 31. O DFLL48M opera em modos de circuito aberto e fechado. Em sua configuração de circuito fechado, é necessário fornecer um *clock* de baixa frequência (com mínimo de 732 Hz e máximo de 33 kHz) como referência para o DFLL48M, e ajustar um fator de multiplicação para gerar a saída de 48 MHz. A definição desses *clocks* é crucial para os *timers* que serão utilizados a seguir.

Figura 32 – Diagrama do DFLL48M.



Fonte: Adaptado de (MICROCHIP, 2023b).

No *firmware*, adotou-se o seguinte conjunto de configurações para estabelecer o *clock*. O *clock* principal foi configurado utilizando o DFLL48M em modo de circuito fechado, onde a realimentação provém do GLCK GENERATOR 1, alimentado pelo cristal externo XOSC32K. Por último, o *clock* do RTC foi sincronizado com o cristal interno OSC32K.

Figura 33 – Disposição dos clocks.

```

/* Set this to true to configure the GCLK when running clocks_init. If set to
 * false, none of the GCLK generators will be configured in clocks_init(). */
# define CONF_CLOCK_CONFIGURE_GCLK          true

/* Configure GCLK generator 0 (Main Clock) */
# define CONF_CLOCK_GCLK_0_ENABLE          true
# define CONF_CLOCK_GCLK_0_RUN_IN_STANDBY  false
# define CONF_CLOCK_GCLK_0_CLOCK_SOURCE    SYSTEM_CLOCK_SOURCE_DFLL
# define CONF_CLOCK_GCLK_0_PRESCALER      1
# define CONF_CLOCK_GCLK_0_OUTPUT_ENABLE   false

/* Configure GCLK generator 1 */
# define CONF_CLOCK_GCLK_1_ENABLE          true
# define CONF_CLOCK_GCLK_1_RUN_IN_STANDBY  false
# define CONF_CLOCK_GCLK_1_CLOCK_SOURCE    SYSTEM_CLOCK_SOURCE_XOSC32K
# define CONF_CLOCK_GCLK_1_PRESCALER      1
# define CONF_CLOCK_GCLK_1_OUTPUT_ENABLE   false

/* Configure GCLK generator 2 (RTC) */
# define CONF_CLOCK_GCLK_2_ENABLE          true
# define CONF_CLOCK_GCLK_2_RUN_IN_STANDBY  false
# define CONF_CLOCK_GCLK_2_CLOCK_SOURCE    SYSTEM_CLOCK_SOURCE_OSC32K
# define CONF_CLOCK_GCLK_2_PRESCALER      32
# define CONF_CLOCK_GCLK_2_OUTPUT_ENABLE   false

```

Fonte: Autoria própria (2023)

### 3.3.2 Timers

Timer/Counter para aplicações de controle (TCC) consiste em um contador, um pré-divisor, canais de comparação/captura e lógica de controle. O contador pode ser ajustado para contar eventos ou pulsos de *clock*. Juntamente com os canais de comparação/captura, o contador pode ser configurado para registrar instantes de eventos de entrada, permitindo a captura de frequência e largura de pulso. Além disso, ele pode realizar geração de forma de onda, como geração de frequência e modulação de largura de pulso (MICROCHIP TECHNOLOGY, 2016).

As extensões de forma de onda são destinadas a aplicações de controle de motor, lastro, LED, ponte H, conversores de energia e outros tipos de aplicações de controle de potência. Elas possibilitam a saída em *low-side* e *high-side* com inserção opcional de tempo morto. Também é possível gerar um padrão de bits sincronizado nas saídas de forma de

onda. As opções de falha oferecem proteção contra falhas para um manuseio seguro e determinístico, desabilitando e/ou desligando *drivers* externos, quando necessário.

Um *clock* genérico (GCLK TCCx) é necessário para cronometrar o TCC. O GCLK TCCx, ou Clock Genérico do Contador/Comparador x (onde 'x' representa o número específico do contador/ comparador), é vital para temporizar as operações do TCC (*Timer/Counter for Control Applications*). Este *clock* é fornecido pelo GCLK 0 que são os 48MHz do DFLL48M (MICROCHIP TECHNOLOGY, 2016).

A equação que rege o funcionamento do *timer* é a seguinte:

$$f_{\text{PWM}_{\text{SS}}} = \frac{f_{\text{GCLK}_{\text{TCC}}}}{N(\text{TOP} + 1)} \quad (3.2)$$

Onde:

$f_{\text{PWM}_{\text{SS}}}$  é a frequência resultante.

$f_{\text{GCLK}_{\text{TCC}}}$  é a frequência fornecida ao clock, nesse caso de 48MHz.

$N$  é o pré-escalar que pode ser (1, 2, 4, 8, 16, 64, 256, 1024).

$\text{TOP}$  é o período controlado pelo registro PER.

O procedimento de configuração do temporizador, conforme ilustrado na Figura 34, é empregada a variável "period timer". Essa variável é responsável por calcular o período em milissegundos com base no tempo requisitado pelo usuário. Em essência, ela utiliza a Equação 3.2, isola o valor de TOP, e converte a frequência para um período em milissegundos. Esse ajuste permite que o temporizador chame uma função de retorno (*callback*) por meio de interrupção em intervalos definidos pelo usuário.

O *timer* do RTC compartilha o mesmo propósito que o *timer* TCC, ativar uma interrupção após o período determinado pelo usuário, mas diverge na fonte do gerador de clock, conforme ilustrado na Figura 33. Este *timer* é alimentado por um fonte de *clock* de 32 kHz. Além disso, sua equação é distinta daquela do TCC. Sua equação é pode ser observada na Equação 3.3.

$$f_{\text{CLK}_{\text{RTC}_{\text{CNT}}}} = \frac{f_{\text{GCLK}_{\text{RTC}}}}{2^{\text{PRESCALER}}} \quad (3.3)$$

Onde:

$f_{CLK\_RTC\_CNT}$  é a frequência resultante.

$f_{GCLK\_RTC}$  é a frequência fornecida ao clock, nesse caso de 32khz.

PRESCALER é o pré-escalar que pode ser (1, 2, 4, 8, 16, 64, 128, 256, 512 e 1024).

Para emular a funcionalidade do Contador de Tempo de Controle (TCC), conforme representado na Figura 34, o usuário tem a prerrogativa de selecionar um intervalo de tempo em milissegundos para a função de retorno de chamada (*callback*) do Relógio de Tempo Real (RTC). Esse procedimento é regido pela aplicação da Equação 3.3, com os ajustes necessários realizados para alcançar a medida de tempo desejada em milissegundos. Importante destacar que o RTC é o único temporizador que opera no modo de *STANDBY MODE*.

Figura 34 – Configuração do timer (TCC).

```

void hw_timer_tcc_init(uint32_t time_ms)
{
    uint32_t period_timer = (uint32_t)((float)CLOCK_FREQUENCY *
↪ ((float)time_ms/1000.0f))/1024.0f) - 1;

    tcc_get_config_defaults(&config_tcc, PWMCTRL_MODULE);

    config_tcc.counter.period = period_timer;
    config_tcc.counter.clock_source = GCLK_GENERATOR_0;
    config_tcc.counter.clock_prescaler = TIMER_PRESCALER;
    config_tcc.compare.wave_generation =
↪ TCC_WAVE_GENERATION_SINGLE_SLOPE_PWM;
    config_tcc.pins.enable_wave_out_pin[2] = true;
    config_tcc.pins.wave_out_pin[2] = PWMTCC_OUT_PIN;
    config_tcc.pins.wave_out_pin_mux[2] = PWMTCC_OUT_MUX;

    tcc_init(&tcc_instance, PWMCTRL_MODULE, &config_tcc);
    tcc_register_callback(&tcc_instance, hw_timer_tcc_cbk,
↪ TCC_CALLBACK_OVERFLOW);
}

void hw_timer_tcc_start(void)
{
    tcc_enable(&tcc_instance);
    tcc_enable_callback(&tcc_instance, TCC_CALLBACK_OVERFLOW);
}

void hw_timer_tcc_stop(void)
{
    tcc_disable(&tcc_instance);
    tcc_disable_callback(&tcc_instance, TCC_CALLBACK_OVERFLOW);
}

```

## Função `hw_timer_tcc_start(void)`

Esta função é responsável por iniciar o timer TCC. Os principais passos são:

## Função `hw_timer_tcc_init(uint32_t time_ms)`

Esta função tem como objetivo inicializar o *timer* TCC com base em um período de tempo fornecido em milissegundos. Aqui estão os passos principais:

1. Calcula o valor do período do *timer* com base na frequência do relógio (*CLOCK\_FREQUENCY*), o tempo desejado em milissegundos (*time\_ms*), e outros fatores.
2. Configura as opções do *timer* usando a estrutura *config\_tcc*.
3. Define a fonte de *clock*, pré-divisor e outras configurações necessárias.
4. Configura o *timer* para operar no modo de geração de onda única (*single slope PWM*).
5. Habilita a saída de onda na terceira saída do pino do *timer* (*PWMTCC\_OUT\_PIN*).
6. Inicializa o *timer* com as configurações especificadas.

## Função `hw_timer_tcc_start(void)`

Esta função é responsável por iniciar o *timer* TCC. Os principais passos são:

1. Ativa o *timer* TCC.
2. Ativa o retorno de chamada (*callback*) associado ao estouro do *timer*. O *callback* parece ser *hw\_timer\_tcc\_cbk*, associado ao evento de overflow do timer.

## Função `hw_timer_tcc_stop(void)`

Esta função é responsável por parar o *timer* TCC. Os principais passos incluem:

1. Desativa o *timer* TCC.
2. Desativa o retorno de chamada associado ao estouro do *timer*.

### Observações Gerais:

- O código usa uma estrutura de configuração com valores já pré-estabelecidos *config\_tcc* e uma instância do *timer tcc\_instance*.

- O *timer* é configurado para operar em um modo específico de geração de onda (*single slope PWM*).
- Há o uso de um *callback* associado ao evento de overflow do timer (*TCC\_CALLBACK\_OVERFLOW*).
- Detalhes específicos, como a definição de pinos (*PWMTCC\_OUT\_PIN*, *PWMTCC\_OUT\_MUX*) e outras constantes, são definidos em um arquivo de configurações, mas correspondem aos pinos de saída do *timer*.

Figura 35 – Configuração do timer (RTC).

```

void hw_timer_rtc_set_period(uint16_t time_ms)
{
    uint16_t period_timer_rtc =
↪ (uint16_t)(((float)time_ms)/((float)CLOCK_FREQUENCY_RTC/65536.0f));
    rtc_count_set_period(&rtc_instance, period_timer_rtc);
}

void hw_timer_rtc_init(uint16_t time_ms)
{
    rtc_count_get_config_defaults(&config_rtc_count);

    config_rtc_count.prescaler      = RTC_COUNT_PRESCALER_DIV_16;
    config_rtc_count.mode          = RTC_COUNT_MODE_16BIT;

    #ifdef FEATURE_RTC_CONTINUOUSLY_UPDATED
        config_rtc_count.continuously_update = true;
    #endif

    rtc_count_init(&rtc_instance, RTC, &config_rtc_count);

    hw_timer_rtc_set_period(time_ms);

    rtc_count_register_callback(&rtc_instance, hw_timer_rtc_cbk,
↪ RTC_COUNT_CALLBACK_OVERFLOW);
}

void hw_timer_rtc_start(void)
{
    rtc_count_enable(&rtc_instance);
    rtc_count_enable_callback(&rtc_instance, RTC_COUNT_CALLBACK_OVERFLOW);
}

void hw_timer_rtc_stop(void)
{
    rtc_count_disable(&rtc_instance);
    rtc_count_disable_callback(&rtc_instance, RTC_COUNT_CALLBACK_OVERFLOW);
}

```

Fonte: Autoria própria (2023)

`hw_timer_rtc_set_period(uint16_t time_ms)`

1. Define o período do *timer RTC* com base no tempo em milissegundos (*time\_ms*) e a frequência do relógio *RTC* (*CLOCK\_FREQUENCY\_RTC*). Usa *rtc\_count\_set\_period*.

`hw_timer_rtc_init(uint16_t time_ms)`

1. Obtém configurações padrão (*config\_rtc\_count*).
2. Configura *prescaler* e modo.
3. Inicializa o contador usando *rtc\_count\_init*.
4. Define o período com *hw\_timer\_rtc\_set\_period*.
5. Registra *callback* de estouro (*RTC\_COUNT\_CALLBACK\_OVERFLOW*).

`hw_timer_rtc_start(void)`

1. Inicia o *timer RTC*:
2. Ativa o contador usando *rtc\_count\_enable*.
3. Ativa *callback* de estouro usando *rtc\_count\_enable\_callback*.

`hw_timer_rtc_stop(void)`

1. Para o *timer RTC*:
2. Desativa o contador usando *rtc\_count\_disable*.
3. Desativa *callback* de estouro usando *rtc\_count\_disable\_callback*.

### Observações Gerais:

- Usa *config\_rtc\_count* e uma instância do contador *RTC*.
- *Prescaler* e modo são configurados conforme necessário.
- Usa *callback* de estouro (*RTC\_COUNT\_CALLBACK\_OVERFLOW*).

### 3.3.3 Sleep

No microcontrolador ATSAMR21G17A, estão disponíveis vários modos de economia de energia. O modo de suspensão pode ser ativado através da instrução Wait For Interrupt (WFI). Os bits *Idle* no registrador *Sleep Mode* (SLEEP.IDLE) e o bit SLEEPDEEP no registrador de Controle do Sistema da CPU são empregados como parâmetros para escolher o nível do modo de suspensão (MICROCHIP TECHNOLOGY, 2016).

Existem dois principais tipos de modo de suspensão:

- Modo IDLE: Nesse modo, a CPU é interrompida, e opcionalmente, alguns domínios de *clock* síncronos são interrompidos, dependendo do argumento *IDLE* especificado. O regulador opera no modo normal (MICROCHIP TECHNOLOGY, 2016).
- Modo STANDBY: Este modo envolve a interrupção de todas as fontes de *clock*, exceto aquelas em que o bit RUNSTDBY está ativado. O regulador opera em modo de baixa potência. Antes de entrar no modo de espera, o usuário deve garantir que uma quantidade significativa de *clocks* e periféricos estejam desativados, evitando uma sobrecarga no regulador de voltagem (MICROCHIP TECHNOLOGY, 2016).

Figura 36 – Configuração do timer (RTC).

Sleep Mode	CPU Clock	AHB Clock	APB Clock	Oscillators				Main Clock	Regulator Mode	RAM Mode
				ONDEMAND = 0		ONDEMAND = 1				
				RUNSTDBY=0	RUNSTDBY=1	RUNSTDBY=0	RUNSTDBY=1			
Idle 0	Stop	Run	Run	Run	Run	Run if requested	Run if requested	Run	Normal	Normal
Idle 1	Stop	Stop	Run	Run	Run	Run if requested	Run if requested	Run	Normal	Normal
Idle 2	Stop	Stop	Stop	Run	Run	Run if requested	Run if requested	Run	Normal	Normal
Standby	Stop	Stop	Stop	Stop	Run	Stop	Run if requested	Stop	Low power	Low power

Fonte: Adaptado de (MICROCHIP TECHNOLOGY, 2016).

O modo de suspensão do ATSAM R21 está intrinsecamente vinculado ao uso de temporizadores, uma vez que no modo de economia de energia conhecido como *STANDBY MODE*, é possível alcançar um consumo ainda mais baixo em comparação com outros modos, devido à desativação de mais *clocks*. Entretanto, é importante salientar que, neste modo, somente o temporizador do RTC (*Real-Time Clock*) possui a capacidade de despertar o sistema em situações normais, sem a necessidade de modificar os bits RUNSTDBY. Essa característica é essencial para o funcionamento adequado da máquina de estados, conforme representado no diagrama da Figura 30.

A seguir a implementação dos modos de dormir no *firmware*:



Figura 37 – Modos de dormir.

```

void hw_cpu_power_enter_sleep_mode(uint8_t sleep_mode)
{
    if ((sleep_mode >= 0) && (sleep_mode <= 2))
    {
        hw_cpu_sleep_mcu(sleep_mode);
    }
    else if (sleep_mode == SYSTEM_SLEEPMODE_STANDBY)
    {
        hw_cpu_sleep_all();
    }
    else
    {
        hw_cpu_sleep_mcu(0);
    }
}

void hw_cpu_sleep_mcu(uint8_t sleep_mode)
{
    system_set_sleepmode(sleep_mode);
    system_sleep();
}

void hw_cpu_sleep_all(void)
{
    // Disable voltage references
    system_voltage_reference_disable(SYSTEM_VOLTAGE_REFERENCE_BANDGAP);
    system_voltage_reference_disable(SYSTEM_VOLTAGE_REFERENCE_TEMPSENSE);

    // Put TRX into deep sleep
    rf_hw_deep_sleep_radio();

    // Disable SPI
    hw_at86_disable_radio_spi();

    // Disable BOD33
    while (!(SYSCTRL->PCLKSR.bit.B33SRDY)) ;
    SYSCTRL->BOD33.bit.ENABLE = 0;

    // Prep MCU for standby mode
    hw_cpu_sleep_mcu(SYSTEM_SLEEPMODE_STANDBY);
}

```

Fonte: Aatoria própria (2023)

`hw_cpu_power_enter_sleep_mode(uint8_t sleep_mode)`

1. Verifica se `sleep_mode` está entre 0 e 2.

- Se verdadeiro, chama `hw_cpu_sleep_mcu(sleep_mode)`.
- Se `sleep_mode == SYSTEM_SLEEPMODE_STANDBY`, chama `hw_cpu_sleep_all()`.
- Se falso, chama `hw_cpu_sleep_mcu(0)`.

`hw_cpu_sleep_mcu(uint8_t sleep_mode)`

1. Configura o modo de sono do sistema usando `system_set_sleepmode(sleep_mode)`.
2. Coloca o sistema para dormir usando `system_sleep()`.

`hw_cpu_sleep_all(void)`

1. Desativa as referências de tensão:
2. Desativa a referência de banda (`SYSTEM_VOLTAGE_REFERENCE_BANDGAP`).
3. Desativa a referência de temperatura (`SYSTEM_VOLTAGE_REFERENCE_TEMPSENSE`).
4. Coloca o rádio TRX em modo de sono profundo usando `rf_hw_deep_sleep_radio()`.
5. Desativa a SPI usando `hw_at86_disable_radio_spi()`.
6. Desativa o BOD33 (Brown-out Detector 3.3V).
7. Prepara o MCU para o modo de espera (`hw_cpu_sleep_mcu(SYSTEM_SLEEPMODE_STANDBY)`).

### Observações Gerais:

- As funções utilizam a API do sistema para configuração e gestão de modos de sono.
- A função `hw_cpu_power_enter_sleep_mode` é o ponto de entrada para o gerenciamento de modos de sono, chamando funções específicas com base no argumento.
- A função `hw_cpu_sleep_mcu` configura o modo de sono do sistema e coloca o sistema para dormir.
- A função `hw_cpu_sleep_all` desativa várias funcionalidades (referências de tensão, rádio TRX, SPI, BOD33) antes de colocar o MCU em modo de espera.

### 3.3.4 SPI SERCOM

Na PCB é utilizado três dispositivos com SPI, o rádio, ADC externo e a RAM externa. A SPI utiliza o transmissor e receptor SERCOM. Cada lado, mestre e escravo, representa um SPI separado contendo um registrador de deslocamento, um buffer de transmissão e dois *buffers* de recepção. Além disso, o mestre SPI utiliza o gerador de taxa de *baud* SERCOM, enquanto o escravo SPI pode utilizar a lógica de correspondência de endereço do SERCOM (MICROCHIP TECHNOLOGY, 2016).

A configuração destes dispositivos dependem da sua ligação no hardware. Como o rádio AT86 possui ligação interna dentro do chip com o microcontrolador, o *datasheet* já

aponta sua configuração. Importante salientar que no caso do microcontrolador ATSAMR21, os pinos de MISO, MOSI e SCLK são dependentes da configuração de DIPO e DOPO (MICROCHIP TECHNOLOGY, 2016).

Figura 38 – Registro DIPO.

DIPO[1:0]	Name	Description
0x0	PAD[0]	SERCOM PAD[0] is used as data input
0x1	PAD[1]	SERCOM PAD[1] is used as data input
0x2	PAD[2]	SERCOM PAD[2] is used as data input
0x3	PAD[3]	SERCOM PAD[3] is used as data input

Fonte: Adaptado de (MICROCHIP TECHNOLOGY, 2016).

O registro DIPO, no contexto do microcontrolador, desempenha uma função ao definir qual pino de dados (MISO) na interface SERCOM será utilizado. Ao configurar o valor adequado neste registro, é possível especificar de maneira precisa e flexível qual dos pinos de entrada/saída da SERCOM será designado como a linha de saída de dados mestre (MISO).

Figura 39 – Registro DOPO.

DOPO	DO	SCK	Slave_SS	Master_SS
0x0	PAD[0]	PAD[1]	PAD[2]	System configuration
0x1	PAD[2]	PAD[3]	PAD[1]	System configuration
0x2	PAD[3]	PAD[1]	PAD[2]	System configuration
0x3	PAD[0]	PAD[3]	PAD[1]	System configuration

Fonte: Adaptado de (MICROCHIP TECHNOLOGY, 2016).

De maneira análoga, o registro DOPO desempenha uma função semelhante ao DIPO, contudo, sua finalidade é determinar qual pino de dados (MOSI) na interface SERCOM será utilizado como linha de saída de dados mestre.

Figura 40 – Pinos de *data*.

DOPO	DO	SCK	Slave_SS	Master_SS
0x0	PAD[0]	PAD[1]	PAD[2]	System configuration
0x1	PAD[2]	PAD[3]	PAD[1]	System configuration
0x2	PAD[3]	PAD[1]	PAD[2]	System configuration
0x3	PAD[0]	PAD[3]	PAD[1]	System configuration

Fonte: Adaptado de (MICROCHIP TECHNOLOGY, 2016).

Vale destacar que a gestão desses registros está sob o controle do *firmware*, conferindo uma versatilidade às funções dos pinos. No contexto da configuração dos pinos associados

à SPI, é imperativo ajustar esses dois registros de acordo com as interconexões realizadas no esquemático e no layout da placa de circuito impresso (PCB). Essa prática assegura uma correspondência precisa entre a configuração do hardware e as expectativas do *firmware*, contribuindo para o correto funcionamento e desempenho otimizado do sistema em diferentes cenários de uso.

Figura 41 – Configuração do rádio (SPI).

```
void hw_config_radio_spi(void)
{
    struct spi_config config_spi_master;
    struct spi_slave_inst_config slave_dev_config;

    spi_slave_inst_get_config_defaults(&slave_dev_config);
    slave_dev_config.ss_pin = SLAVE_SELECT_PIN;
    spi_attach_slave(&slave, &slave_dev_config);

    spi_get_config_defaults(&config_spi_master);

    config_spi_master.mux_setting = CONF_MASTER_MUX_SETTINGS;
    config_spi_master.mode_specific.master.baudrate = CONF_MASTER_BAUDRATE;
    config_spi_master.pinmux_pad0 = CONF_MASTER_PINMUX_PAD0;
    config_spi_master.pinmux_pad1 = CONF_MASTER_PINMUX_PAD1;
    config_spi_master.pinmux_pad2 = CONF_MASTER_PINMUX_PAD2;
    config_spi_master.pinmux_pad3 = CONF_MASTER_PINMUX_PAD3;

    spi_init(&spi_master_instance, CONF_MASTER_SPI_MODULE, &config_spi_master);

    spi_enable(&spi_master_instance);

    struct extint_chan_conf eint_chan_conf;

    extint_chan_get_config_defaults(&eint_chan_conf);
    eint_chan_conf.gpio_pin = AT86RFX_IRQ_PIN;
    eint_chan_conf.gpio_pin_mux = AT86RFX_IRQ_PINMUX;
    eint_chan_conf.gpio_pin_pull = EXTINT_PULL_UP;
    eint_chan_conf.filter_input_signal = false;
    eint_chan_conf.detection_criteria = EXTINT_DETECT_FALLING;
    extint_chan_set_config(AT86RFX_IRQ_CHAN, &eint_chan_conf);
}
```

Fonte: Autoria própria (2023)

## hw\_config\_radio\_spi(void)

### 1. Configuração do dispositivo escravo SPI:

- Obtém as configurações padrão (escravo *spi\_slave\_inst\_get\_config\_defaults*).
- Define o pino de seleção de escravo (*SLAVE\_SELECT\_PIN*) e anexa o escravo usando *spi\_attach\_slave*.

## 2. Configuração do mestre SPI:

- Obtém as configurações padrão para o mestre usando *spi\_get\_config\_defaults*.
- Configura a taxa de baud (*CONF\_MASTER\_BAUDRATE*) e os pinos usando *config\_spi\_master*.
- Inicializa o mestre SPI com as configurações usando *spi\_init*.
- Ativa o mestre SPI usando *spi\_enable*.

## 3. Configuração do pino de interrupção externa (EINT):

- Configurações padrão para o canal EINT usando *extint\_chan\_get\_config\_defaults*.
- Define o pino EINT (*AT86RFX\_IRQ\_PIN*), o mux do pino EINT (*AT86RFX\_IRQ\_PINMUX*), e outras configurações.
- Configura o canal EINT com as configurações usando *extint\_chan\_set\_config*.

## Observações Gerais:

- Usa a API de configuração do SPI e EINT para configurar o dispositivo escravo e mestre SPI, bem como o pino de interrupção externa.
- Configurações específicas, como os pinos, a taxa de baud e os critérios de detecção, são definidas nas macros de configuração.
- A função encapsula todas as configurações necessárias para o SPI e EINT associados ao rádio.

Na Figura 41, temos uma visão abrangente que aborda a configuração dos *pads*, a taxa de *baud*, o procedimento de inicialização da SPI, e ainda a especificação do pino responsável por desencadear as interrupções no dispositivo de rádio.

Destaca-se a relevância dessa representação visual, pois proporciona uma compreensão integrada de diversos parâmetros cruciais para o funcionamento eficaz da SPI nesse contexto específico. A configuração dos *pads*, por exemplo, é um componente essencial para garantir a correta comunicação entre os dispositivos, enquanto a taxa de *baud* define a velocidade de transmissão, influenciando diretamente o desempenho geral da interface.

É válido ressaltar que, devido à utilização compartilhada do mesmo barramento pela RAM e pelo ADC externo juntamente com a SPI, ambas as entidades adotam as mesmas configurações, com a única exceção sendo o pino de seleção de chip.

Após a configuração dos periféricos, é necessário iniciá-los, a maioria dos periféricos não necessitam de procedimentos para a inicialização correta, a não ser o ADC, o ADC externo possui uma sequência de inicialização.

Figura 42 – Configuração da RAM/ADC externo (SPI).

```
void hw_config_spi(void)
{
    struct spi_config config_spi_master_adc;

    spi_get_config_defaults(&config_spi_master_adc);

    config_spi_master_adc.mode_specific.master.baudrate = RAM_SPI_BAUDRATE;
    config_spi_master_adc.mux_setting = RAM_SPI_SERCOM_MUX_SETTING;
    config_spi_master_adc.pinmux_pad0 = RAM_SPI_SERCOM_PINMUX_PAD0;
    config_spi_master_adc.pinmux_pad1 = PINMUX_UNUSED;
    config_spi_master_adc.pinmux_pad2 = RAM_SPI_SERCOM_PINMUX_PAD2;
    config_spi_master_adc.pinmux_pad3 = RAM_SPI_SERCOM_PINMUX_PAD3;

    spi_init(&spi_master_instance_adc, RAM_SPI_MODULE, &config_spi_master_adc);
    spi_enable(&spi_master_instance_adc);
}
```

Fonte: Autoria própria (2023).

Feita por software, essa sequência de inicialização consiste em excitar o pino CONVST 66 vezes, e fazer a configuração inicial via SPI. Para essa inicialização foi seguido a risca passo-a-passo da rotina de *power on* contido no *datasheet* do AD7091R4-BRUZ (ANALOG DEVICES, 2015), seu procedimento pode ser observado na Figura 43.

### 3.3.5 Aplicação do Projeto

Com a inicialização de todos os periféricos, é possível observar que a implementação do projeto segue a proposta delineada no diagrama apresentado na Figura 30. Inicialmente, tanto o microcontrolador quanto o rádio encontram-se em estado de dormência, com o microcontrolador no modo *STANDBY* e o rádio em *deepSleep* (sono profundo). Posteriormente, é estabelecido um período de amostragem, durante o qual o *callback* do RTC é acionado para despertar o microcontrolador. Nesse intervalo, o *timer* TCC é empregado para manter a regularidade das amostras do ADC. Se o usuário programar um intervalo de 200 ms, as amostras serão coletadas a cada 200 ms. Após essa etapa, os dados são armazenados na RAM externa e transmitidos via rádio para um outro dispositivo externo (*bridge* USB-ZigBee), que obtém o dado do módulo sensor que o envia para a serial do computador, com isso, é possível exibir estes dados na tela de um computador.

Figura 43 – Inicialização do ADC.

```

void ad9071r_init(uint8_t channels, uint32_t sampling_time_ms, ad9071r_cbk_t cbk)
{
    write_regs[0][1] = channels;
    bool ch0_enabled = (channels & (AD7091R_CHANNEL_0 | AD7091R_CHANNEL_DEFAULT))
↪ != 0;
    bool ch1_enabled = channels & AD7091R_CHANNEL_1;
    bool ch2_enabled = channels & AD7091R_CHANNEL_2;
    bool ch3_enabled = channels & AD7091R_CHANNEL_3;

    num_configured_channels = (ch0_enabled? 1 : 0) + (ch1_enabled? 1 : 0) +
↪ (ch2_enabled? 1 : 0) + (ch3_enabled? 1 : 0);

    period_timer = (uint32_t)(((float)CLOCK_FREQUENCY *
↪ ((float)sampling_time_ms/1000.0f))/1024.0f) - 1;

    tcc_get_config_defaults(&config_tcc, PWMCTRL_MODULE);

    config_tcc.counter.period = period_timer;
    config_tcc.counter.clock_source = GCLK_GENERATOR_0;
    config_tcc.counter.clock_prescaler = TIMER_PRESCALER;
    config_tcc.compare.wave_generation =
↪ TCC_WAVE_GENERATION_SINGLE_SLOPE_PWM;
    config_tcc.pins.enable_wave_out_pin[2] = true;
    config_tcc.pins.wave_out_pin[2] = PWMTC_OUT_PIN;
    config_tcc.pins.wave_out_pin_mux[2] = PWMTC_OUT_MUX;
    tcc_init(&tcc_instance, PWMCTRL_MODULE, &config_tcc);

    tcc_register_callback(&tcc_instance, callback_timer, TCC_CALLBACK_OVERFLOW);
↪ // Register the callback function

    user_cbk = cbk;

    delay_ms(100);

    for(uint8_t i = 0; i<66; i++)
    {
        port_pin_set_output_level(ADC_CONVST,true);
        port_pin_set_output_level(ADC_CONVST,false);
    }

    port_pin_set_output_level(ADC_CONVST, true);

    for (uint8_t i = 0; i < sizeof(write_regs) / sizeof(write_regs[0]); i++)
    {

        hw_write_header(write_regs[i], i);

    }
}

```

Fonte: Autoria própria (2023)

## 4 Conclusão

Neste projeto, foi concebido e implementado um sistema versátil de aquisição e transmissão de variáveis essenciais. A proposta de desenvolver um dispositivo capaz de adquirir e transmitir dados desde o nível de *hardware* até o *firmware* foi integralmente realizada. Como resultado deste trabalho, tornou-se possível utilizar um protótipo funcional capaz de realizar a aquisição e transmissão de dados provenientes de qualquer sensor com um alcance determinado do conversor analógico-digital (ADC). Além disso, é viável efetuar a reconstrução destes dados com sucesso. Essa abordagem proporciona uma solução robusta e flexível para aquisição de dados, permitindo a adaptação a diferentes contextos e requisitos de sensores.

Não obstante o sucesso alcançado pelo projeto, é importante ressaltar que enfrentou-se consideráveis desafios ao longo do desenvolvimento. A curva de aprendizado, tanto no que concerne ao *hardware* e ao *layout* da PCB, quanto na esfera do *firmware*, apresentou-se como um obstáculo significativo. No âmbito do *hardware*, a confecção de um layout que utiliza RF revelou-se desafiadora, demandando um esforço substancial para superar as complexidades inerentes a essa etapa. No que diz respeito ao *firmware*, embora tenha-se optado pela linguagem C, a assimilação da IDE da Microchip e suas funcionalidades singulares impôs uma curva de aprendizado adicional. Este aspecto ressalta a complexidade e a exigência técnica do processo de desenvolvimento, ressaltando, ao mesmo tempo, a necessidade de habilidades específicas para lidar com os desafios associados a cada fase do projeto.

Além disso, identificam-se oportunidades significativas para aprimoramentos no método de transmissão de dados, particularmente no contexto da retransmissão de pacotes, cujo impacto reverbera diretamente na onda reconstruída e na análise de dados. Adicionalmente, há potencial para otimizações substanciais na automatização de processos internos do *firmware*. No que tange à aquisição de dados, observa-se uma notável melhoria na qualidade da onda ao aumentar a frequência de aquisição. Contudo, é crucial salientar que tal incremento na frequência acarreta um aumento na quantidade de *frames* a serem enviados. Destarte, torna-se evidente uma interdependência entre esses dois desafios, destacando a necessidade de aprimoramentos em projetos futuros para endereçar eficazmente tais questões.



# Referências Bibliográficas

ABRACON. *Automotive and Industrial Grade Ceramic Base SMD Crystal*. [S.l.], 2019. Citado na página 35.

ALTIUM; HARTLEY, R. *How to Achieve Proper Grounding - Rick Hartley - Expert Live Training (US)*. Disponível em: <<https://www.youtube.com/live/ySuUZEjARPY?si=VzL3VUNlggGWk-r5>>. Nenhuma citação no texto.

AMPHENOL. *Micro USB 2.0 Connectors*. [S.l.], 2010. Rev. A. Citado na página 30.

ANALOG DEVICES. *Ultralow Power, 12-Bit SAR ADC*. [S.l.], 2015. Rev. C. Citado 2 vezes nas páginas 31 e 65.

BOGATIN, E. *Bogatin's Practical Guide to Prototype Breadboard and PCB Design*. Local da Editora: Artech House Publishers., 2022. Nenhuma citação no texto.

ECS. *ECX-34R SMD CRYSTAL*. [S.l.], 2019. Citado na página 35.

ELAHI, A. *ZigBee Wireless Sensor and Control Network*. Local da Editora: Prentice Hall, 2009. Citado 2 vezes nas páginas 24 e 25.

FERANEC, R.; BOGATIN, E. *What Every PCB Designer Should Know - Crosstalk Explained (with Eric Bogatin)*. Disponível em: <<https://www.youtube.com/watch?v=EF7SxgcDfCo>>. Nenhuma citação no texto.

GISLASON, D. *ZigBee Wireless Networking*. Local da Editora: Newnes, 2008. Citado 5 vezes nas páginas 14, 21, 23, 24 e 25.

GREENGARD, S. *The Internet of Things*. Local da Editora: MIT Press, 2015. Citado na página 14.

IPC-2221A. *Generic Standard on Printed Circuit Board Design*. 2023. Disponível em: <[https://www.eng.lbl.gov/~shuman/NEXT/CURRENT\\_DESIGN/TP/MATERIALS/IPC-2221A\(L\).pdf](https://www.eng.lbl.gov/~shuman/NEXT/CURRENT_DESIGN/TP/MATERIALS/IPC-2221A(L).pdf)>. Citado na página 42.

ISSI. *512Kx8 LOW VOLTAGE, SERIAL SRAM with SPI, SDI and SQI INTERFACE*. [S.l.], 2021. Rev. A8. Citado na página 31.

JLCPCB. *PCB Manufacturing Assembly Capabilities*. 2023. Disponível em: <<https://jlcpcb.com/capabilities/pcb-capabilities>>. Nenhuma citação no texto.

JLCPCB. *Standard PCB/PCBA*. 2023. Disponível em: <<https://cart.jlcpcb.com/quote/?orderType=1&stencilLayer=2&stencilWidth=100&stencilLength=100>>. Citado 3 vezes nas páginas 45, 46 e 47.

JOHANSON. *INTEGRATED PASSIVE COMPONENTS*. [S.l.], 2016. Citado na página 33.

LINDON, J. C. *Encyclopedia of Spectroscopy and Spectrometry*. Local da Editora: ISBN-10, 2016. Nenhuma citação no texto.

- MICROCHIP. *Low Power, 2.4GHz Transceiver for ZigBee, RF4CE, IEEE 802.15.4, 6LoWPAN, and ISM Applications*. [S.l.], 2014. Nenhuma citação no texto.
- MICROCHIP. *Embedded Software Solution for Atmel Flash MCUs*. 2023. Disponível em: <<https://asf.microchip.com/docs/latest/>>. Nenhuma citação no texto.
- MICROCHIP. *How to configure DFLL48M oscillator as main clock in ASF project for SAMR21/SAMD21/SAMD20/SAMD10/SAMD11?* 2023. Disponível em: <<https://microchip.my.site.com/s/article/How-to-configure-DFLL48M-oscillator-as-main-clock-in-ASF-project-for-SAMR21-SAMD21-SAMD20-SAMD10-SAMD11>>. Citado na página 52.
- MICROCHIP TECHNOLOGY. *SMART ARM-Based Wireless Microcontroller*. [S.l.], 2016. Rev. G. Citado 8 vezes nas páginas 35, 51, 52, 53, 54, 59, 61 e 62.
- NEXPERIA. *Ultra low capacitance double rail-to-rail ESD protection diode*. [S.l.], 2023. Rev. 4. Citado na página 30.
- PANASONIC. *4.7 mm×3.5 mm SMD Light Touch Switches*. [S.l.], 2017. Citado na página 36.
- PASTERNAK. *Pasternack's Microstrip Calculator*. 2023. Disponível em: <<https://www.pasternack.com/t-calculator-microstrip.aspx>>. Citado na página 43.
- PETERSON, Z. *The Mysterious 50 Ohm Impedance: Where It Came From and Why We Use It*. 2023. Disponível em: <<https://resources.altium.com/p/why-there-transmission-line-critical-length>>. Nenhuma citação no texto.
- PETERSON, Z. *Why is There a Transmission Line Critical Length?* 2023. Disponível em: <<https://resources.altium.com/p/why-there-transmission-line-critical-length>>. Citado na página 19.
- RAZAVI, B. *RF Microelectronics*. Local da Editora: Paul Boger, 2012. Citado 2 vezes nas páginas 17 e 18.
- SCHWAB, K. *A Quarta Revolução Industrial*. Cidade da Editora: Editora Schwab, 2016. Citado na página 14.
- SEMMLOW, J. *Circuits, Signals and Systems for Bioengineers*. Local da Editora: Academic Press, 2018. Citado na página 26.
- SKYWORKS. *SKY66112-11: 2.4 GHz Front-End Module for Zigbee Technology Applications / Thread / Bluetooth® Applications*. [S.l.], 2022. Citado na página 33.
- TEXAS INSTRUMENTS. *600 mA/1.25 MHz HIGH-EFFICIENCY STEP-DOWN CONVERTER*. [S.l.], 2004. SLVS076C. Citado na página 29.
- VALVANO, J. W. *Introduction to Embedded Systems: Using the ARM Cortex-M Microcontrollers*. Local da Editora: Independent Publishing Platform., 2014. Citado na página 16.
- YIU, J. *ARM Cortex-M for Beginners*. Local da Editora: Independent Publishing Platform., 2017. Citado 2 vezes nas páginas 16 e 17.

# APÊNDICE A – Tabela BOM - Materiais da PCB

Tabela 4 – Parte 1 da lista de materiais.

Ref	Quantidade	Mouser	Preço Individual	Preço total (Dólar)
C1, C2, C4, C7	4	GRM1555C1H120JA01D	0.1	0.4
C3, C6, C8, C9, C10, C11, C12, C13, C20, C25, C29, C34, C38, C39	14	GCM155R71H104KE02J	0.1	1.4
C5, C31, C33, C37	4	GRM188R61E106KA73D	0.28	1.12
C14	1	C1608C0G1E472J080AA	0.18	0.18
C15, C17, C18, C21, C22, C24, C26, C27, C30	9	GCM155R71H102KA37D	0.1	0.9
C16	1	NM		0
C19	1	GJM1555C1H2R2WB01D	0.18	0.18
C23, C28	2	GRM21BZ71H475KE15K	0.74	1.48
C32	1	GRM219R61C226ME15K	0.47	0.47
C35	1	0603YC105KAT2A	0.18	0.18
C36	1	GRT155R61E225KE13D	0.27	0.27
D1, D2	2	D1 -> APTD2012LCGCK, D2- >APTD2012LSYCK	0.41	0.82
D3	1	PRTR5V0U2X	0.53	0.53
D4	1	PMEG40T10ERX	0.52	0.52
FB1	1	BLM18PG471SN1D	0.12	0.12
H1, H2, H3, H4	4			0

Tabela 5 – Parte 2 da lista de materiais.

Ref	Quantidade	Mouser	Preço Individual (Dólar)	Preço total (Dólar)
J1, J2	2	10129381- 910001BLF	0.22	0.44
J3	1	10129378- 905001BLF	0.17	0.17
J4	1	10129381- 908001BLF	0.21	0.21
J5	1			0
J6	1	10103594- 0001LF	0.76	0.76
JP1	1			0
JP2	1			0
JP3	1			0
L1	1	LQP15MN3N2W02D	0.16	0.16
L2	1	LQH3NPH6R8MMEL	0.4	0.4
Q1	1	SI2307CDS- T1-E3	0.67	0.67
R1, R5, R6	3	RC0402FR- 0739RL	0.1	0.3
R2, R3	2	CRCW04024K70FKEDC	0.1	0.2
R4, R16, R17	3	CRCW0402100KFEDC	0.1	0.3
R7, R8	2	CRCW0402330RFKEDC	0.1	0.2
R9	1	CRCW04021M00FKEDC	0.1	0.1
R10, R11, R12, R13, R14, R15, R18	7	CRCW04021K00FKEDC	0.11	0.77
R19, R20	2	CRCW040247K0FKEDC	0.1	0.2
R21	1	CRCW060349R9FKEAHP	0.16	0.16
SW1, SW2	2	EVQ- P9W02M	0.67	1.34
TP1, TP2, TP3, TP4, TP5, TP6	6			0
U1	1	ATSAMR21G17A- MF	7.13	7.13
U2	1	2450AT42B100E	0.71	0.71
U3	1	SKY66112- 11	3.1	3.1
U4	1	2450BM15A0015E	0.98	0.98
U5	1	TPS62026DRCR	1.96	1.96
U6	1	AD7091R- 4BRUZ	8.28	8.28
U7	1	IS62WVS5128FBLL- 20NLI-TR	5.74	5.74
U8	1	902-9040	3.05	3.05
Y1	1	ABM8AIG- 16.000MHZ- 12-2Z-T3	1.19	1.19
Y2	1	520-X-.327- CDX2096	0.73	0.73
Componentes totais:	99			47.82