

**CAMILA MAYUMI RIBEIRO EGUCHI
GABRIEL FELIPE COELHO DE GOUVEIA
VINICIUS FLORENTINO MACEDO SOUZA
WILLIAM RYO TAKAHASHI**

**ENERGY HARVESTING PARA AUTONOMIA
ENERGÉTICA DE NÓS IOT**

São Paulo
2022

**CAMILA MAYUMI RIBEIRO EGUCHI
GABRIEL FELIPE COELHO DE GOUVEIA
VINICIUS FLORENTINO MACEDO SOUZA
WILLIAM RYO TAKAHASHI**

**ENERGY HARVESTING PARA AUTONOMIA
ENERGÉTICA DE NÓS IOT**

Trabalho apresentado à Escola Politécnica
da Universidade de São Paulo para obtenção
do Título de Engenheiro Eletricista.

São Paulo
2022

**CAMILA MAYUMI RIBEIRO EGUCHI
GABRIEL FELIPE COELHO DE GOUVEIA
VINICIUS FLORENTINO MACEDO SOUZA
WILLIAM RYO TAKAHASHI**

**ENERGY HARVESTING PARA AUTONOMIA
ENERGÉTICA DE NÓS IOT**

Trabalho apresentado à Escola Politécnica
da Universidade de São Paulo para obtenção
do Título de Engenheiro Eletricista.

Área de Concentração:

Eletrônica e Sistemas

Computação

Orientador:

Profa. Dra. Anarosa Alves Franco
Brandão

Co-orientador:

Profa. Dra. Laisa Caroline Costa de
Biase

São Paulo
2022

AGRADECIMENTOS

A todos aqueles que contribuíram, de alguma forma, para a realização deste trabalho.

A todos que participaram, direta ou indiretamente, do desenvolvimento deste trabalho de pesquisa, enriquecendo o processo de aprendizado.

Às professoras orientadoras, Dra. Laisa Caroline Costa de Biase e Dra. Anarosa Alves Franco Brandão por todos os conselhos, pela ajuda e pela paciência com a qual guiaram o nosso aprendizado.

Aos colaboradores, Geovane, Augusto e Matteus, pelo fornecimento de dados e materiais que foram fundamentais para o desenvolvimento da pesquisa que possibilitou a realização deste trabalho.

RESUMO

A integração entre tecnologias de energy harvesting com conceitos de Internet of Things (IoT) pode proporcionar autonomia energética para sensores de redes inteiras. Essas redes de sensores costumam ser aplicadas em regiões isoladas ou em condições que dificultam a manutenção, fazendo com que não seja possível utilizar energia cabeada e que não seja fácil, nem barato, realizar a manutenção para trocas de bateria. Dessa forma, a aplicação de circuitos de captação de energia aparecem como uma solução, capaz de aumentar a vida útil das baterias ou até eliminá-las. Este projeto faz uma análise de diferentes formas de energy harvesting com o intuito de construir um circuito de captação, e análise de consumo energético, além de fazer uma otimização do firmware da placa Pulga, do programa Caninos Loucos, para garantir a autonomia energética de nós IoT em uma aplicação de monitoramento florestal.

Palavras-Chave – Energy Harvesting. Internet of Things. Redes de sensores. Consumo energético.

ABSTRACT

The integration between energy harvesting technologies and concepts of Internet of Things (IoT) can provide energy autonomy for sensors of entire networks. These sensor networks are often used in remote locations or in conditions that make maintenance difficult, making it not possible to use the main power and not easy, nor cheap, to replace the batteries. In this way, energy harvesting circuits can provide a solution capable of increasing the lifespan of batteries or even eliminate them. This project reviews different energy harvesting techniques in order to build an energy harvesting circuit and analyzes energy consumption, in addition to optimize the Pulga board firmware (Caninos Loucos), to guarantee the energy autonomy of IoT nodes in a forest monitoring system.

Keywords – Energy Harvesting. Internet of Things. Sensor Networks. Energy Consumption.

LISTA DE FIGURAS

1	Pulga Core v2.0	13
2	Arquitetura básica de um EH-IoT [24]	17
3	Tabela comparativa de sistemas de micro-kernel	21
4	Cronograma macro	23
5	Pulga montada à dev-shield	25
6	Circuito de teste	26
7	Tabela de possíveis configurações de <i>power modes</i>	28
8	Tabela de possíveis configurações de <i>power modes</i>	29
9	Árvore de requisitos	31
10	Árvore de objetivos	32
11	Modelo macro de uma rede de sensores [24]	33
12	Efeito Fotovoltaico. Obtido em [6]	34
13	Circuito formado com uma célula fotovoltaica. Obtido em [7]	34
14	Princípio de funcionamento de um gerador CC. Obtido em [15]	35
15	Stack formado pela Pulga core e pela Pulga LoRa	36
16	Diagrama do nível 0	37
17	Circuito regulador de tensão para o painel solar	38
18	Tensão de saída quando varia-se a tensão de entrada	39
19	Gerador eólico	40
20	Arquitetura Power ORing para energy harvesting	41
21	Circuito de energy harvesting com múltiplas fontes	42
22	Corrente consumida pela Pulga	43
23	Estrutura do RIOT OS	44

24	Esquema de diretórios do RIOT OS	45
25	Bloco esquemático do contador RTC	47
26	Tensão gerada na placa 6x11 cm	50
27	Tensão gerada na placa 13x16 cm	50
28	Valores de corrente observados na bateria	51
29	Corrente consumida pela Pulga	52
30	Shell do RIOT apresentando a interface	53
31	Requisitos para o sistema durar entre 1, 2 e 5 anos	57

LISTA DE ABREVIATURAS E SIGLAS

CPU	Central Processing Unit
EH	Energy Harvesting
IB	Instituto de Biociências
IoT	Internet of Things
IMAZON	Instituto do Homem e Meio Ambiente da Amazônia
MPPT	Maximum Power Point Tracker
PMIC	Power Management Integrated Circuit
PMU	Power Management Unit
USP	Universidade de São Paulo
SO	Sistema Operacional
WBS	Work Breakdown Structure
WSN	Wireless Sensor Network

SUMÁRIO

1	Introdução	12
1.1	Monitoramento florestal na USP	13
1.1.1	O projeto	13
1.1.2	A Pulga	13
1.2	Objetivo	14
1.3	Motivação	14
1.4	Organização do Trabalho	14
2	Estado da Arte	16
2.1	Revisão da Literatura	16
2.2	Aspectos Conceituais	18
2.2.1	Energy Harvesting	18
2.2.2	Internet of Things	20
2.2.3	Sistemas Operacionais de Micro-Kernel	20
2.2.4	Power Management	21
2.3	Soluções existentes	22
3	Materiais e Métodos	23
3.1	Cronograma	23
3.2	Prova de Conceito	24
3.2.1	Objeto de estudo	24
3.2.2	Infraestrutura e configurações	24
3.2.3	Coleta e análise de dados	25
3.3	Estudo do CPU nRF52840	27

3.3.1	Sleep modes	27
3.4	Metodologia de desenvolvimento	28
4	Especificação de Requisitos do Sistema	30
4.1	Requisitos de hardware	30
4.2	Requisitos de firmware	30
4.3	Árvore de requisitos	31
4.4	Árvore de objetivos	32
4.5	Tecnologias Utilizadas	32
4.5.1	Redes de sensores sem fio (WSN)	32
4.5.2	Energy harvesters	33
4.5.2.1	Painel Solar	33
4.5.2.2	Gerador Eólico	35
4.5.3	O sistema operacional RIOT	35
4.5.4	LoRa	36
5	Projeto	37
5.1	Nível 0	37
5.2	Nível 1	38
5.3	Nível 2	38
5.3.1	Circuito da Fonte Solar	38
5.3.2	Múltiplas fontes de energia (energy harvesting)	40
5.3.2.1	Gerador eólico	40
5.3.2.2	Arquitetura do circuito	41
5.3.3	Análise de consumo energético	42
5.3.3.1	Consumo do firmware LoRa	43
5.4	Desenvolvimento do firmware	44
5.4.1	Arquitetura do RIOT OS	44

5.4.2	Power management no RIOT	45
5.4.3	Configuração e Implementação	46
5.4.3.1	Função de entrada ao Low-Power	46
5.4.3.2	Configuração do RTC	46
5.4.3.3	Definição do gerenciador	47
5.4.3.4	Implementação da Interface	48
6	Testes e resultados	49
6.1	Testes de captação	49
6.1.1	Capacidade de geração do painel solar	49
6.1.2	Corrente mínima e média fornecida pelo painel solar	50
6.1.3	Corrente na bateria quando auxiliada pelo painel solar	51
6.2	Testes de consumo	51
6.2.1	Testes pré-otimização	51
6.2.2	Testes pós-otimização	52
6.2.3	Interface de apresentação	53
6.3	Análise dos resultados	54
6.3.1	Circuito não otimizado e alimentado sem o circuito de captação	54
6.3.2	Circuito não otimizado e alimentado com o circuito de captação	55
6.3.3	Circuito otimizado e alimentado sem o circuito de captação	55
6.3.4	Circuito otimizado e alimentado com o circuito de captação	56
7	Considerações Finais	58
7.1	Conclusões do Projeto de Formatura	58
7.2	Contribuições	58
7.3	Perspectivas de Continuidade	59
	Referências	60

Apêndice A – Atividades (WBS)	63
Apêndice B – Valor dos resistores	64
Apêndice C – Circuito de energy harvesting montado	65
Apêndice D – Código Fonte	66
D.1 pm.c	66
D.2 pulga-interface.c	67
Anexo A – Pulga - Especificações	69
Anexo B – Pulga - Pinagem	70
Anexo C – LoRa	71
Anexo D – Pulga LoRa	72

1 INTRODUÇÃO

Nos últimos anos, os debates acerca do desmatamento vêm ganhando destaque, tendo em vista que os recordes de desmatamento são superados a cada ano. De acordo com o Instituto do Homem e Meio Ambiente da Amazônia (Imazon), foram destruídos mais de 10 mil km² de mata nativa [10], sendo que, de acordo com o projeto MapBiomas [16], quase 99% de maneira ilegal. Além disso, outros fatores, como incêndios florestais, contribuem de forma significativa para a devastação florestal.

As redes de sensores sem fio (WSNs) são sistemas que possuem grande aplicabilidade, e o campo do monitoramento florestal pode se aproveitar desta inovação. Os sensores captam dados sobre o ambiente e os transmitem, mantendo as autoridades informadas da situação atual local. Ainda, vale destacar que a detecção precoce é a melhor forma de proteger a floresta contra incêndios e desmatamentos [25].

Essas redes costumam ser energizadas por bateria. No entanto, em um ambiente isolado como é a floresta, a manutenção dessas baterias se torna inviável, tanto no âmbito do custo (baterias de boa qualidade e duradouras são caras) quanto no âmbito da sustentabilidade (é difícil descartar tantas baterias de forma correta). Desta forma, um circuito de captação de energia capaz de oferecer autonomia energética aos nós de IoT aparece como uma forma interessante de superar este problema.

Para o ambiente florestal, as fontes de EH mais promissoras são a solar (via placas solares) e a cinética (via material piezoelétrico). Devemos destacar que captar energia de forma confiável e eficiente é um grande desafio. Os dispositivos de captação atuais geram quantidades baixas de energia e dependem de fatores externos, tornando-as imprevisíveis. Logo, é necessário fazer uma avaliação crítica das tecnologias de conversão e dos circuitos.

Assim, neste projeto, buscaremos fazer uma análise de consumo energético e um estudo de diversas fontes de EH para selecionar as mais adequadas ao ambiente florestal. Além disso, será realizada a otimização do firmware visando máxima eficiência energética.

1.1 Monitoramento florestal na USP

O nosso trabalho objetiva garantir a autonomia energética das placas microcontroladoras (chamadas Pulga) que são utilizadas em um projeto de monitoramento florestal dentro da USP.

1.1.1 O projeto

O projeto em questão é uma parceria entre o Instituto de Biociências (IB) e a Escola Politécnica da USP. Nele, diversas dessas placas são espalhadas na mata do IB e o objetivo é transformá-la em um exemplo de floresta urbana, tornando-se um modelo de estudo da Mata Atlântica e também de outros biomas, como a Amazônia [13].

1.1.2 A Pulga

A placa utilizada neste projeto é a denominada Pulga Core, do programa Caninos Loucos.

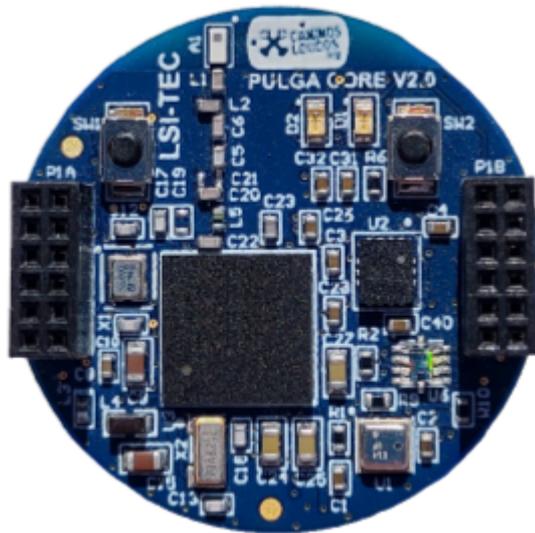


Figura 1: Pulga Core v2.0

É um projeto brasileiro de uma microcontroladora com uma variedade de sensores, apresentando baixo consumo e dimensões pequenas. A base board permite a conexão de diferentes fontes de captação de energia do ambiente (energy harvesting) e é ideal para aplicações IoT [5]. A modularidade da MCU permite que sejam formadas pilhas ou *stacks* de Pulgas com diferentes funções. As demais especificações da placa encontram-se no anexo A.

1.2 Objetivo

O trabalho multidisciplinar engloba duas áreas, a de Sistemas Eletrônicos, cujo objetivo é montar um circuito de coleta de energia por EH, contendo captadores, e uma bateria recarregável, como fonte de alimentação da Pulga, e a de Computação, cujo objetivo é desenvolver um sistema de gerenciamento e controle de distribuição energética dentro de seu sistema operacional.

O objetivo final, portanto, é atingir maior sustentabilidade através do mútuo trabalho entre aproveitamento de fontes externas e a redução de consumo energético. Para isso o trabalho consiste em implementar um sistema de EH integrando um *firmware* e um circuito de coleta de energia, utilizando a Pulga. Este trabalho fará parte de um projeto de monitoramento florestal e será aplicado na Reserva Florestal do campus da USP. O sistema consiste na placa microcontroladora conectada a um circuito de captação de energia, e com a implementação de um *firmware* de otimização energética.

1.3 Motivação

Redes de sensores conectados via Internet oferecem boa solução para monitoramento e coleta de dados em diversas aplicações, como na saúde, preservação ambiental e indústria. Uma grande barreira para sua adoção é a dificuldade de alimentar essas redes, pelo fato dos sensores estarem numa situação em que não há fonte de energia cabeada, nem uma infraestrutura de rede que seja de fácil instalação e manutenção apropriada para áreas isoladas, como florestas.

1.4 Organização do Trabalho

Todo o desenvolvimento está descrito neste documento em 7 capítulos: Introdução, Aspectos conceituais, Materiais e Métodos, Especificação de Requisitos, Projeto, Testes e resultados e por fim, Considerações Finais.

No primeiro capítulo são apresentados o problema, objetivos deste projeto e motivações.

O atual estado do projeto e a reflexão sobre as soluções já existentes encontram-se no capítulo 2.

O capítulo 3 contém a prova de conceito, o cronograma e estudos sobre os materiais

utilizados.

No capítulo 4 são especificados requisitos necessários para o desenvolvimento e as escolhas das tecnologias baseadas nestes requisitos.

O projeto e seu desenvolvimento, bem como o módulo de captação e o gerenciador de consumo, encontram-se no capítulo 5.

No capítulo 6, são apresentados todos os testes antes e depois das otimizações e estimativas de longevidade do sistema.

Por fim, nas no capítulo 7, apresenta-se as conclusões sobre o trabalho, bem como contribuições e as perspectivas de continuidade.

2 ESTADO DA ARTE

No capítulo em questão serão abordados os conceitos aplicados a este projeto. Primeiramente serão apresentados estudos com base em artigos de pesquisa sobre o tema, e depois serão explicados os conceitos aplicados neste projeto. Este capítulo contém uma introdução sobre EH e uma explicação do funcionamento de um módulo gerenciador de consumo, além disso serão também apresentados as soluções já existentes na literatura.

2.1 Revisão da Literatura

Durante os estudos de conceitos antes do desenvolvimento, foram encontradas pesquisas recentes acerca de projetos de EH com microcontroladores, discutidas nesta seção.

Nos últimos anos tivemos um grande aumento de aplicações de Internet of Things (IoT), que utilizam redes de sensores sem fio (WSN) nas mais diversas áreas, como automação residencial e industrial, infraestrutura civil, e monitoramento de regiões isoladas, como florestas. Essa tendência de crescimento deve se manter nos próximos anos, visto que estamos cada vez mais automatizando tarefas do dia a dia.

Apesar de aparecer como uma solução, captar energia de forma confiável e eficiente não é trivial, mas sim um grande desafio.

Os dispositivos de captação atuais geram quantidades baixas de energia e dependem de fatores externos, tornando-as imprevisíveis. Dessa forma, é necessário entender o consumo de energia do sensor, o funcionamento dos circuitos de captação e os melhores tipos de fontes externas para projetar algo confiável.

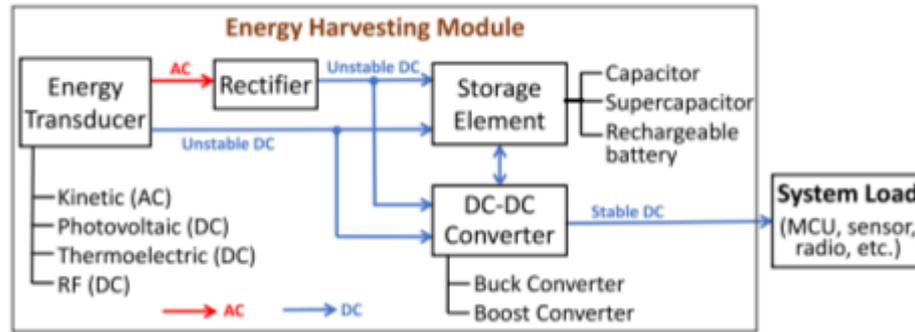


Figura 2: Arquitetura básica de um EH-IoT [24]

A figura 2 apresenta os componentes básicos de um módulo de EH, que substitui uma bateria convencional. Para simplificar, podemos separar este módulo em dois sub-componentes:

- **Transducer:** sua principal função é converter a energia disponível no ambiente (luminosa, cinética, térmica, entre outras) em energia elétrica capaz de alimentar os componentes eletrônicos.
- **Power management unit:** possui como principal função regular a tensão gerada pelo transdutor, com o objetivo de estabilizar o fornecimento de energia.

Por fim, como a geração pode variar ao longo do tempo, o PMU costuma ter um elemento de armazenamento, como supercapacitor e bateria renovável.

Já existem tanto na academia quanto no mercado alguns protótipos e produtos dessa categoria.

No artigo “Trinity: Enabling self-sustaining WSNs indoors with energy-free sensing and networking” [23], os autores apresentam um sistema de sensoriamento e controle alimentado com energia proveniente do movimento do ar. Isso foi alcançado instalando o sistema com um material piezoelétrico próximo à saída de um ar condicionado.

No artigo “Tethys: Collecting Sensor Data without Infrastructure or Trust” [11], os autores conseguiram criar um sensor sem uso de bateria, utilizando efeitos eletromagnéticos para captar energia a partir do fluxo de água de um chuveiro. Neste caso utilizando um chuveiro de fluxo de 3.8 L/min gerou uma potência média de 15.6 mW.

Em “Power management and energy harvesting techniques for wireless sensor nodes” [29], autores discutem a importância do gerenciamento de consumo energético para redes de sensores, alegando que apenas soluções de design e arquitetura de hardware

são insuficientes para um projeto sustentável. Além disso, apresentam técnicas de gerenciamento e EH que podem ser combinados para gerar um resultado e performance satisfatório.

Em "Recent advances and future prospects in energy harvesting technologies" [8], os autores apresentam pesquisas além das técnicas de EH convencionais, ou seja, quais são as com maior performance em cada cenário e o avanço tecnológico para se obter um melhor resultado. Estes novos avanços geram promessas e desafios que também são levantados neste artigo.

2.2 Aspectos Conceituais

2.2.1 Energy Harvesting

O princípio de Energy Harvesting é o processo de conversão de energia potencial provinda do meio ambiente em energia elétrica, que será armazenada para a alimentação de dispositivos autônomos de menor escala. Esse conceito já é presente em níveis industriais como o moinho de vento, usinas de painéis solares, captura de energia de ondas oceânicas e calor geotérmico.

No ramo de tecnologia de IoT, a escala de energia capturada é pequena, porém suficiente para alimentar dispositivos de consumo muito baixos. Estudos estão sendo feitos nesta área a fim de tornar o balanço de consumo e captura mais eficiente e então reduzir a demanda energia da bateria do sistema ou até tornar estes dispositivos totalmente autônomos [8]. Alguns exemplos de técnicas de EH são o piezoelétrica, termoelétrica, radiação eletromagnética e fotovoltaica, que se apresentam como as formas mais eficientes de captação de energia do ambiente.

Existem diversos tipos de fontes que podem ser utilizados para o energy harvesting dos nodes em florestas, entre elas: solar, cinética (por meio de materiais piezoelétricos ou vento), vibração, térmica e de ondas eletromagnéticas.

A conversão fotovoltaica de luz solar em energia elétrica já está bem estabelecida na indústria, e os painéis solares (para aplicações IoT) já possuem uma eficiência de 15% para conversão de energia. As células solares são classificadas de acordo com o nível de tensão na saída, o material usado e o local da aplicação (in ou outdoor). A energia gerada depende do tamanho da placa e da incidência de luz solar no local. Quanto maior esses dois fatores, mais energia será gerada [19].

Como os painéis solares só conseguem ser eficientes em alta concentração de luz solar e em um ambiente florestal a luz pode ser bloqueada por folhas e outras coisas, além do painel não funcionar de noite, são utilizados diferentes algoritmos para otimizar a aplicação da energia solar. O uso de um MPPT (Maximum power point tracking) para maximizar a eficiência é uma das formas mais comuns dessa otimização. A energia extra consumida pelo MPPT é facilmente compensada pela quantidade maior de energia que será coletada do ambiente. Um dos algoritmos utilizados para o MPPT é o “Perturb and Observe”, que é fácil de implementar e requer um menor tempo de processamento.

A forma mais comum de usar energia mecânica como fonte é por meio de materiais piezoelétricos, essa energia pode ser coletada a partir do movimento, cíclico ou não. Um exemplo desse tipo de energia é a passagem de veículos ou pessoas caminhando. Uma pesquisa feita por Starner [13] concluiu que a energia disponível variava de 7 mW (finger motion) até 67 W (lower limb movement). Foi constatado que uma pessoa pesando 60 kg aplica uma força de pelo menos 588 N para andar, e que esse valor pode aumentar até 3x durante a corrida. Assim, com uma deformação de 10 mm do tênis e a uma taxa de 2 passos por segundo, teríamos uma potência de 5.88 W por pé. A energia mecânica também pode ser coletada através do fluxo de ar do ambiente por meio de eletromagnetismo [24].

A energia por vibração está disponível em diversos ambientes e sua intensidade depende da amplitude e frequência da vibração (vale destacar que a razão entre a massa vibrante e a massa do equipamento de harvesting também afeta, uma vez que interfere na vibração). Para extrair essa energia, utiliza-se materiais piezoelétricos, que possuem a propriedade de gerar um campo elétrico proporcional à deformação que sofrem. A eficiência na conversão depende do coeficiente de acoplamento eletromecânico do material piezoelétrico [19].

A extração de energia térmica se dá pelo Efeito Seebeck e depende de um gradiente de temperatura. Esse gradiente irá determinar a eficiência da conversão pela seguinte expressão:

$$\eta = \frac{T_h - T_c}{T_h}$$

Na fórmula, T_h representa a temperatura (K) no lado quente e T_c representa a temperatura (K) no lado frio. Portanto, a eficiência da conversão será maior quanto maior for o gradiente.

Para gerar energia por meio de ondas eletromagnéticas do ambiente foram criadas antenas na Universidade de Georgia, nos Estados Unidos, que captam frequências de 100 MHz até 15 GHz. Essas antenas multibandas são capazes de gerar até 1 mW [3].

2.2.2 Internet of Things

Internet of Things, ou Internet das Coisas, refere-se a redes de dispositivos conectados e distribuídos que possibilitam a comunicação de dados com a nuvem em tempo real. Tais redes são caracterizadas por baixo consumo de energia, menores espaços de armazenamento de memória e a possibilidade de uso em grande escalas por meio dos nós.

A grande maioria dos dispositivos IoT são alimentados por baterias com durabilidade limitada, sendo necessária sua custosa substituição no futuro. Assim, mesmo no campo da telecomunicação, o fornecimento de energia para os sistemas distribuídos ainda é um grande desafio.

2.2.3 Sistemas Operacionais de Micro-Kernel

Um dos principais benefícios de um sistema operacional baseado em micro-kernel é a sua capacidade de funcionar de maneira eficiente e confiável com recursos limitados [20]. Isso é especialmente importante em dispositivos IoT, que muitas vezes são pequenos e operam com pouca energia. Além disso, a arquitetura de micro-kernel permite que o sistema seja altamente modular e flexível, o que significa que diferentes módulos podem ser adicionados ou removidos conforme necessário.

Alguns exemplos de sistemas operacionais micro-kernel para IoT incluem o Free RTOS¹, Tiny OS², Contiki OS³, RIOT OS⁴ e o Zephyr⁵. Os três últimos são os mais populares, já que além de oferecerem suporte à vasta maioria de processadores, são sistemas desenvolvidos recentemente e por isso, há uma grande comunidade que auxilia no desenvolvimento de *features*. Uma tabela comparativa entre os sistemas Tiny OS, Contiki OS e RIOT OS é apresentada abaixo:

¹<https://www.freertos.org>

²<http://www.tinyos.net>

³<http://www.contiki-os.org>

⁴<https://www.riot-os.org>

⁵<https://www.zephyrproject.org>

	Tiny OS	Contiki OS	RIOT
RAM, ROM	<1kb, <4kb	<2kb, <30kb	~1.5kb, 5kb
Multithreading	No	No But uses Protothreads	Yes
Modularity	No	Partial	Yes
Programing Language	NESC	C, C++	C, C++

Figura 3: Tabela comparativa de sistemas de micro-kernel

2.2.4 Power Management

O *Power Management* de um SO é um recurso de gerenciamento de consumo energético do dispositivo por meio de transições de diferentes estados ou *power modes*. Existem diversas formas de estados de economia de energia exploradas pelo recurso, dentre elas as mais comuns são:

- **Power Gating:** Uma técnica que envolve desligar a corrente nos blocos do circuito que não estão em uso. Esta técnica possui um mecanismo relativamente complexo que se relaciona com a determinação de como sequenciar os ciclos de desligamento e energização e se é possível antecipar a atividade do bloco com antecedência suficiente para realizar a sequência de energização.
- **Clock Gating:** Uma técnica popular usada em muitos circuitos síncronos para reduzir a dissipação de potência dinâmica, removendo o sinal de *clock* quando o circuito não está em uso.

O consumo total de energia de um dispositivo envolve três principais componentes: hardware (CPU, periféricos, componentes externos), SO e aplicativos. Em nível de hardware, o elemento de design central no sistema de gerenciamento de energia da CPU é o chamado *idle thread*. Este *thread* é criado durante a inicialização do sistema e é agendado quando nenhum outro precisa ser executado, ou seja, quando a CPU está ociosa. O gerenciamento de energia da CPU é construído em torno de uma única função, que a aciona para entrar no estado de energia mais baixo (*sleep mode*). Em nível de SO, há demandas de recursos combinados dos aplicativos e dispositivos. O gerenciamento de energia

existente no nível do SO tende a se concentrar em dispositivos individuais. No nível de aplicativos, é possível economizar energia tornando-os aplicativos com reconhecimento, ou seja diminuir seu consumo reduzindo sua atividade, além de poder gerar informações ao gerenciador de dispositivos ou alterar seu padrão de acesso ao dispositivo para criar oportunidades de economia de energia para o hardware.

Um dos maiores desafios de atingir a autonomia por meio de EH é obter o equilíbrio entre a energia captada e a energia consumida, sendo que, geralmente, a segunda possui peso consideravelmente maior que a primeira. Nesse sentido, o *Power Management* do SO possui um papel fundamental em um sistema de EH, de evitar desperdício deste consumo, administrando a forma como a energia disponível é distribuída no hardware. [29]

2.3 Soluções existentes

Já existem casos reais do uso de WSNs em larga escala para monitoramento florestal. Dentre eles, podemos destacar o sistema “Forest Guardian” [27], que monitora o som produzido por atividades de desmatamento, como o de motosserras, e envia notificações às autoridades, que podem agir rapidamente. Outro sistema proposto em [30] utiliza sensores de temperatura e de concentração de gás (monóxido de carbono) para identificar prováveis incêndios. Também há sistemas que utilizam da umidade relativa (além dos já mencionados sensores de temperatura e de fumaça) [22] e da vibração [18] para detectar incêndios.

3 MATERIAIS E MÉTODOS

Neste capítulo serão apresentadas análises para verificar se os objetivos propostos poderiam ser viabilizados. Para isso, foram realizados testes experimentais utilizando a Pulga e uma placa solar. Além disso, foi estabelecido um cronograma das atividades de modo a organizá-las para que sejam desenvolvidas dentro do tempo.

3.1 Cronograma

A elaboração de um planejamento em forma de cronograma de atividades foi imprescindível para o desenvolvimento principalmente por conta da questão da integração de diferentes áreas disciplinares. A seguir está o cronograma da divisão de tarefas e do prazo em que se espera cumprí-las, de um ponto de vista macro:

WBS	Nome da Tarefa	Maio	Junho	Julho	Agosto	Setembro	Outubro	Novembro	Dezembro
1	Circuito de captação de energia								
1.1	Estudo e pesquisa na literatura								
1.2	Definição dos tipos de fonte e seus parâmetros								
1.3	Análise do PMIC - Como ligar diferentes tipos de fontes?								
1.4	Projeto e design do circuito de captação								
1.5	Integração com a Pulga								
2	Análise de consumo energético								
2.1	Pesquisa na literatura - Como essa análise já foi feita?								
2.2	Criar um plano de testes para modelar o consumo de energia								
2.3	Fazer a análise de consumo do sistema pré-otimizado								
2.4	Analisar consumo energético da Pulga com o PM								
3	Otimização do Firmware								
3.1	Estudo do RIOT, PM e diferentes SOs								
3.2	Estudo de PM e sleep modes no nRF52840								
3.3	Implementações e testes de sleep na Pulga								
3.4	Melhorias e Implementação do timer de sleep								

Figura 4: Cronograma macro

O planejamento conta com três principais atividades distintas desenvolvidas paralelamente, como apresentado na figura anterior. Essas atividades são "Circuito de captação de Energia", "Análise de consumo energético" e "Otimização do *firmware*".

3.2 Prova de Conceito

A prova de conceito caracteriza-se pela implementação simples para validar a ideia proposta. Nesta etapa do trabalho buscamos mostrar a possibilidade de alimentar a Pulga utilizando uma placa solar.

3.2.1 Objeto de estudo

Para testar a viabilidade da nossa ideia, propomos 3 testes:

- **Tensão em aberto gerada pela placa solar:** buscamos observar a capacidade de geração da placa em situação de céu limpo e céu encoberto.
- **Tensão nos terminais da placa ao alimentar um led:** observação da capacidade de geração de corrente da placa solar.
- **Alimentação da Pulga apenas com a placa solar:** verificar se a placa solar sozinha é capaz de alimentar a Pulga.

A placa solar utilizada possui dimensões 11 cm x 6 cm (66 cm²) e é capaz de gerar até 7 V de tensão em aberto.

A placa solar é conectada a um gerenciador de energia (PMIC) da própria Pulga. O gerenciador em questão é o LTC3106, capaz de receber tensões de 0.3 V a 6.0 V na entrada, e produzir uma tensão estável de 3.3 V ou 5.0 V.

Vale destacar também que a Pulga conta com um módulo Bluetooth, o que será importante para verificarmos se a mesma encontra-se em funcionamento.

3.2.2 Infraestrutura e configurações

Antes de conectar a placa solar à Pulga, foi necessário colocar um divisor resistivo para limitar a tensão na entrada do PMIC.

Como a Pulga suporta até 6.0 V e a placa solar é capaz de gerar valores até em torno de 7 V, montamos um divisor resistivo para que a tensão máxima na entrada seja de 5 V. Dessa forma, caso ocorra algum pico de produção na placa solar, o valor continua abaixo de 6 V e não há risco de queimar a Pulga.

Os resistores utilizados foram de 2 k Ω e 5 k Ω . Os cálculos e a montagem estão no apêndice B.

A seguir estão, respectivamente, as figuras que mostram a Pulga montada em uma placa de desenvolvedor (usada para facilitar as conexões) e a imagem do circuito a ser testado.

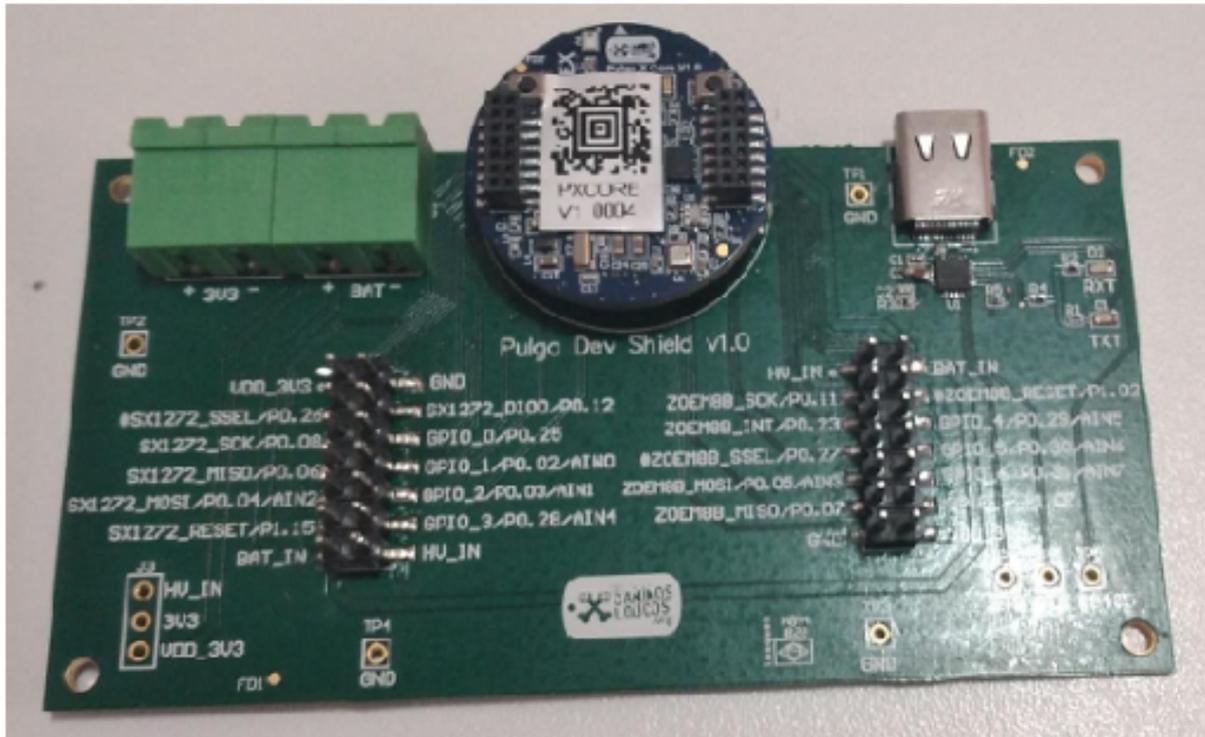


Figura 5: Pulga montada à dev-shield

3.2.3 Coleta e análise de dados

Para a coleta de informações utilizamos um multímetro (para coletar dados de tensão geradas pela placa) e um aplicativo chamado nRF Connect para verificar se o Bluetooth da placa encontra-se em funcionamento.

No primeiro teste, foi medida a tensão em aberto gerada pela placa em dois dias diferentes: um com o céu encoberto e outro com o céu limpo.

Na primeira situação, o céu encontrava-se nublado, com o sol totalmente coberto por nuvens brancas e a temperatura do ambiente era de 15^o C. Foi feita uma medição pontual, quando o multímetro registrou uma tensão de 2.06 V. Já na segunda situação, com o céu limpo, foi feita a medição ao longo do dia, sendo obtido o gráfico da figura 8.

Já o segundo teste foi realizado também na situação de céu limpo, à mesma temperatura. Foi registrado pelo multímetro uma tensão de 3.95 V nos terminais da placa solar, e o LED acendeu com brilho típico.

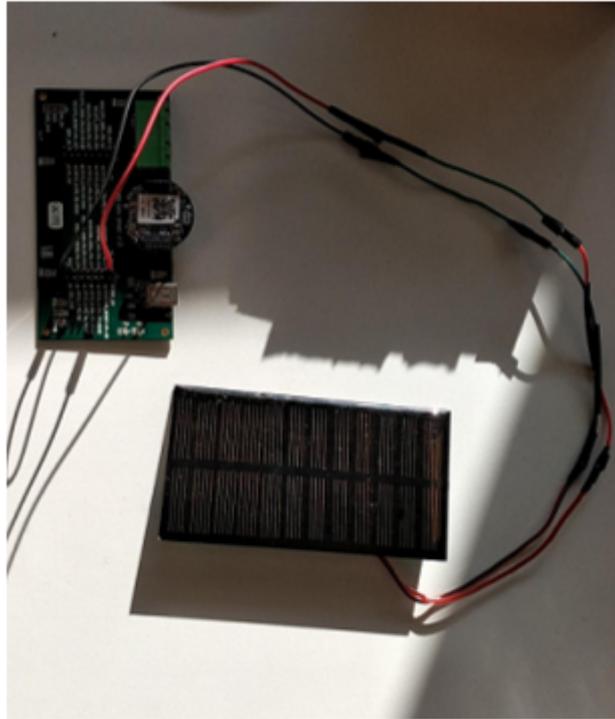


Figura 6: Circuito de teste

Por fim, para o último teste a placa solar foi conectada à Pulga, que rodava um firmware de bluetooth, para simular uma aplicação de baixo consumo.

Após conectar a Pulga à saída do divisor resistivo, notamos que a placa não ligava. Com o auxílio de um multímetro medimos a tensão na saída do divisor primeiro desconectado e então conectado à placa. Os valores obtidos foram os seguintes:

- Desconectado: 4.95 V
- Conectado: 1.30 V

Isso nos permitiu concluir que o divisor de tensão projetado não seria adequado, uma vez que a impedância de entrada da Pulga estava comprometendo a transferência de energia. Dessa forma, conectamos a saída da placa solar (sem divisor) direto à Pulga e a mesma ligou.

3.3 Estudo do CPU nRF52840

3.3.1 Sleep modes

O CPU nRF52840 conta com dois principais modos e dois submodos de economia, são eles **Sistema ON** e **Sistema OFF**, **Latência constante** e **Low power**, respectivamente.

O Sistema ON é o método mais utilizado. Acionando este modo, o PMU desliga todos os periféricos conectados que não estão em uso. O gerenciamento deles estão diretamente relacionados ao nível de atividade, ou seja, o PMU captura por meio de eventos e atividades gerados pelos periféricos. O Sistema ON conta com dois submodos:

- **Latência Constante:** Neste modo, a latência de ativação da CPU e a resposta da tarefa de periféricos são mantidas a valores constantes. Isso é garantido forçando um conjunto de recursos básicos a serem ativados durante o sono, tendo uma latência constante e previsível, ao custo de um maior consumo de energia.
- **Low power:** Ao contrário da Latência Constante, este modo garante um menor consumo de corrente ao custo de mais latência, variando reguladores e osciladores para on/off baseado na demanda de aplicativos e periféricos do sistema.

O Sistema OFF, por sua vez, é o modo com a máxima economia possível. O CPU é desligado e a maioria dos clocks e periféricos são completamente desabilitados. Algumas fontes de acordar o sistema desse modo são por GPIO, por comparador de voltagem e por *reset* de sistema.

A documentação do processador nRF52840 disponibiliza diversos cenários possíveis e suas respectivas configurações que podem ser implementados [1]. As condições são baseadas em sistema ON ou OFF, retenção ou não de memória e o modos para "acordar" o sistema, sendo elas, por GPIO, RTC, *reset*, ou por comparador de voltagem. A tabela com essas configurações é apresentada abaixo:

Symbol	Description	Min.	Typ.	Max.	Units
I _{ON_RAMOFF_EVENT}	System ON, no RAM retention, wake on any event		0.97		μA
I _{ON_RAMON_EVENT}	System ON, full 256 kB RAM retention, wake on any event		2.35		μA
I _{ON_RAMON_POF}	System ON, full 256 kB RAM retention, wake on any event, power-fail comparator enabled		2.35		μA
I _{ON_RAMON_GPIOTE}	System ON, full 256 kB RAM retention, wake on GPIOTE input (event mode)		17.37		μA
I _{ON_RAMON_GPIOTEPORT}	System ON, full 256 kB RAM retention, wake on GPIOTE PORT event		2.36		μA
I _{ON_RAMOFF_RTC}	System ON, no RAM retention, wake on RTC (running from LFRC clock)		1.50		μA
I _{ON_RAMON_RTC}	System ON, full 256 kB RAM retention, wake on RTC (running from LFRC clock)		3.16		μA
I _{OFF_RAMOFF_RESET}	System OFF, no RAM retention, wake on reset		0.40		μA
I _{OFF_RAMOFF_LPCOMP}	System OFF, no RAM retention, wake on LPCOMP		0.86		μA
I _{OFF_RAMON_RESET}	System OFF, full 256 kB RAM retention, wake on reset		1.86		μA
I _{ON_RAMOFF_EVENT_5V}	System ON, no RAM retention, wake on any event, 5 V supply on VDDH, REGO output = 3.3 V		1.29		μA
I _{OFF_RAMOFF_RESET_5V}	System OFF, no RAM retention, wake on reset, 5 V supply on VDDH, REGO output = 3.3 V		0.95		μA

Figura 7: Tabela de possíveis configurações de *power modes*

Apesar da grande gama de cenários a explorar, foram levados em consideração apenas alguns estados devido a aplicabilidade e dos requisitos que serão discutidos na seção a seguir

3.4 Metodologia de desenvolvimento

Antes de realizar qualquer desenvolvimento foi necessário seguir alguns passos de instalação e configuração na máquina. Para este projeto, as seguintes dependências foram instaladas:

- **Git** : Software de controle de versões de código, capturando alterações de código fonte com processo de criação de *branches* (ramificações) em um projeto, permitindo então, a programação colaborativa
- **Make** : Linha de código para compilação de código. Este comando lê um arquivo chamado "Makefile", que contém instruções sobre como compilar o código-fonte em um executável.
- **GCC para ARM** : Um conjunto de compiladores para linguagens de programação como C. Ele é utilizado em sistemas operacionais como Linux ¹ e é uma ferramenta essencial para compilar código-fonte em executáveis.

¹<https://www.linux.org/>

4 ESPECIFICAÇÃO DE REQUISITOS DO SISTEMA

Nesta seção, serão especificados requisitos para o desenvolvimento do projeto e apresentados a escolha das tecnologias. Devido a multidisciplinaridade, foi necessário separar os requisitos em duas partes, eletrônicos e computacionais. Além disso, serão apresentadas árvores de requisitos de marketing e de objetivos que auxiliaram no planejamento ao longo do desenvolvimento.

4.1 Requisitos de hardware

O mapeamento dos requisitos de *hardware* seguiram as propostas de acordo com o projeto de monitoramento florestal o qual o sistema Pulga fará parte. Dentre esses requisitos, estão o funcionamento a tempo integral, alto desempenho de coleta, baixo custo e leveza. O objetivo deste mapeamento será principalmente adaptar o sistema para uma produção, levando em consideração a rede de nós como a Pulga.

4.2 Requisitos de firmware

Da mesma forma, os requisitos funcionais de *firmware* também seguem o planejamento de instalação florestal, entretanto com mais foco na economia de consumo energético do sistema. É de suma importância que seu sistema operacional seja próprio para arquiteturas de pequeno porte e com baixo consumo. Além disso, deve possuir suporte ao processador da placa Pulga, o nRF52840, e possua um módulo gerenciador de consumo para que seja possível a implementação de *sleep modes*.

Levando em consideração a aplicação florestal, serão necessários dados retidos em memória e a transição de estados deve acontecer de forma automática. Portanto, foi descartado a opção *deep sleep* definido pelo estado SYSTEM OFF do processador.

4.3 Árvore de requisitos

Os requisitos de marketing são uma série de itens que descrevem as necessidades de um cliente. Ter esses requisitos bem claros logo no início do projeto é essencial para que o seu desenvolvimento seja feito corretamente e atenda o cliente.

Assim, o sistema de automação energética deve atender uma série de requisitos para atender à demanda de potenciais clientes interessados no projeto. Dentre as demandas principais, destacam-se:

- Custo benefício: deve ser o mais alto possível, apresentando baixo preço para o produto final, e baixo custo de manutenção.
- Confiabilidade energética: deve ser o melhor possível, garantindo que o sistema de monitoramento funcione sem interrupções e com alta disponibilidade energética.
- Portabilidade: o produto deve ser compacto e fácil de transportar e realizar manutenções.

A figura abaixo mostra a árvore de requisitos de marketing do produto a ser desenvolvido:

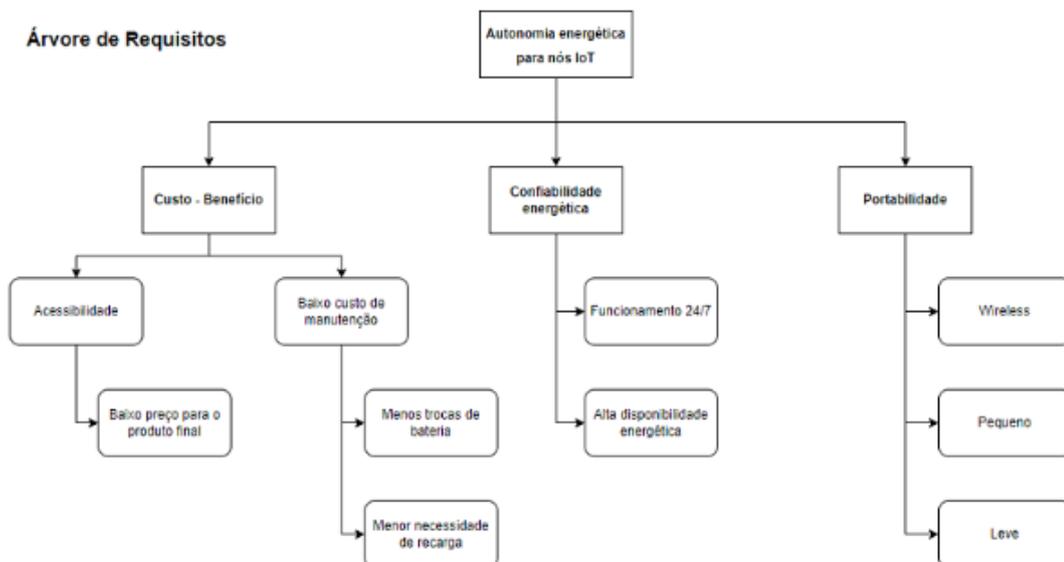


Figura 9: Árvore de requisitos

4.4 Árvore de objetivos

A figura a seguir apresenta a árvore de objetivos, isto é, a representação gráfica do objetivo central do projeto, dos meios para alcançá-lo (mostrado pelas "raízes"), e dos efeitos positivos resultantes (indicado pelas folhas).

Essa metáfora da árvore auxilia a visualização das fases de construção das ferramentas para o desenvolvimento do projeto. Além disso, o fato das fases estarem em ordem hierárquica nos permite notar a importância de cada fase.

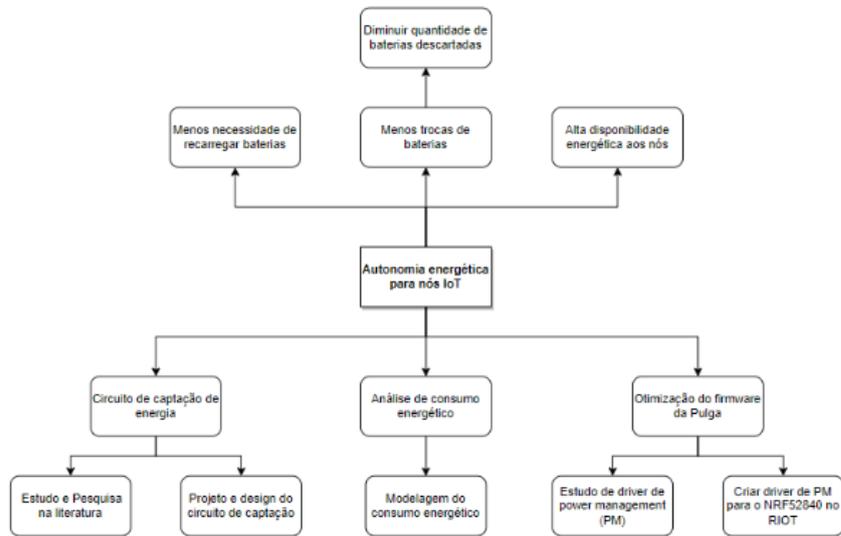


Figura 10: Árvore de objetivos

4.5 Tecnologias Utilizadas

Para a determinação das tecnologias a serem utilizadas, foram levados em conta os requisitos descritos nos subcapítulos acima. Entre as tecnologias relevantes para esse projeto temos as redes WSN, a escolha dos geradores de diferentes fontes e o sistema operacional RIOT e o *firmware* de comunicação LoRa.

4.5.1 Redes de sensores sem fio (WSN)

Uma das principais características das WSNs é que a rede pode ser modificada tanto em tamanho quanto em atuação a qualquer momento, assim não há um node principal na rede, qualquer um dos nodes pode ser o controlador/host da WSN [21].

A maioria dos nodes em uma WSN segue o modelo apresentado em [14], como está

representado na figura na seguir.

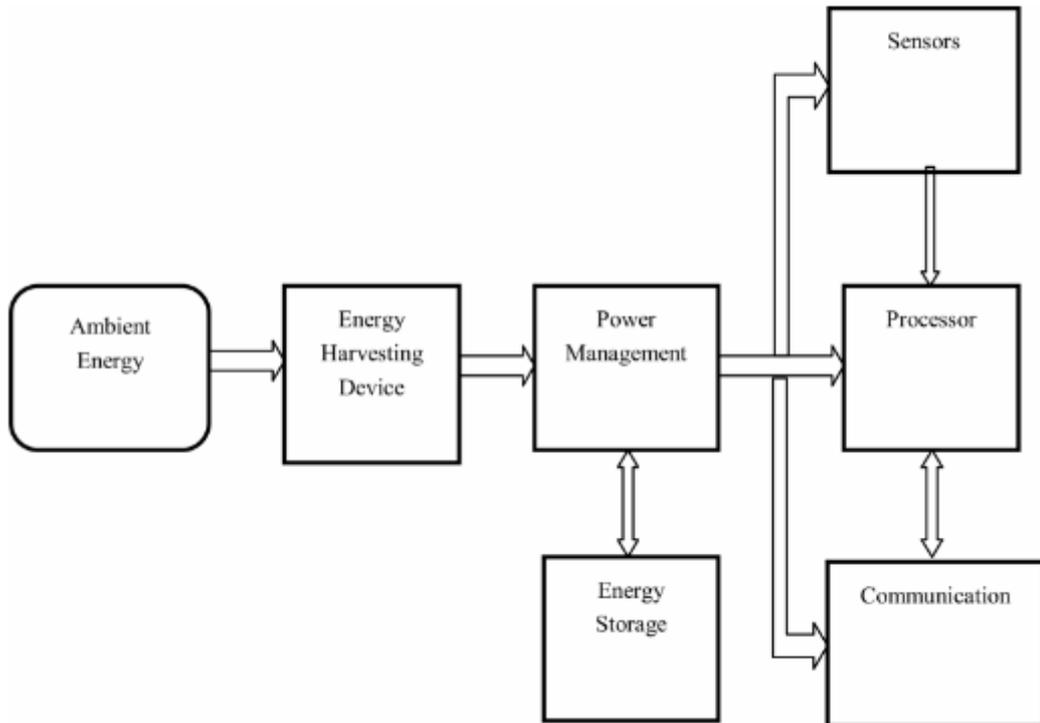


Figura 11: Modelo macro de uma rede de sensores [24]

Nos circuitos dos nodes da WSN, um componente fundamental é o conversor DC-DC, que converte a energia captada para a energia utilizada pelos nodes WSN [26]. Um destes conversores é o LTC3106, que é um conversor DC/DC do tipo buck-boost síncrono com duas entradas. Ele possui baixa corrente quiescente para uma longa faixa de tensões de entrada. O chip é capaz de gerar uma tensão regulada fixa na saída cujo valor é selecionado pelo usuário através de um indutor [2].

4.5.2 Energy harvesters

O projeto utilizará duas fontes de energia de captação, solar e eólica, sendo que a fonte primária será a solar. Assim, nesta seção vamos detalhar o princípio de funcionamento dessas tecnologias.

4.5.2.1 Painel Solar

O painel solar é o equipamento responsável por gerar a energia fotovoltaica. Ele é composto por um conjunto de células fabricadas com materiais semicondutores, como o silício [28]. O princípio de funcionamento desse equipamento é o efeito fotovoltaico, observado pela primeira vez por Alexandre-Edmond Becquerel, em 1839.

O efeito fotovoltaico é a criação de corrente elétrica após a exposição à luz. O material absorve fótons, fazendo com que os elétrons sejam transferidos da banda de valência para a banda de condução dentro do próprio material.

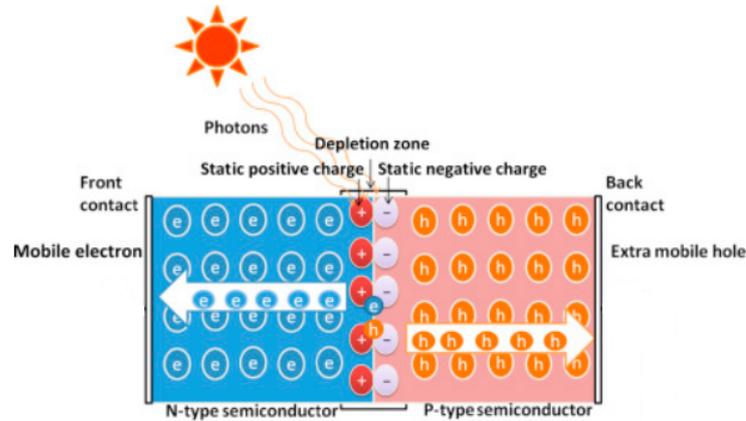


Figura 12: Efeito Fotovoltaico. Obtido em [6]

No caso do painel solar, as células voltaicas são construídas utilizando uma junção PN, isto é, uma parte com excesso de lacunas (P) e outra parte com excesso de elétrons (N). Quando essa junção recebe energia através dos fótons, cria-se uma corrente elétrica à medida em que os elétrons ejetados e as lacunas remanescentes são arrastados em direções diferentes pelo campo elétrico.

Com a corrente elétrica criada, o painel torna-se uma fonte, sendo que a região P é o eletrodo positivo, e a região N é o eletrodo negativo. Conectando-se uma carga nos contatos metálicos, temos um circuito completo:

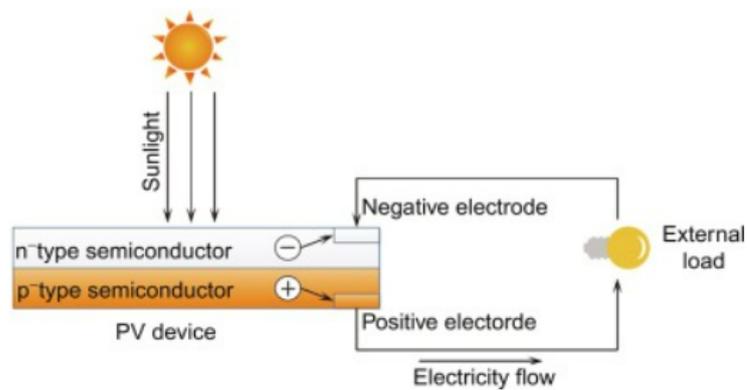


Figura 13: Circuito formado com uma célula fotovoltaica. Obtido em [7]

4.5.2.2 Gerador Eólico

A energia eólica é gerada através da conversão de energia cinética em energia elétrica, através de um aerogerador. Esse "aerogerador" pode ser as pás de uma grande turbina eólica ou, como é o nosso caso, as pás acopladas a um pequeno motor [12].

Os aerogeradores têm função fundamental para a geração da energia eólica. As pás desse equipamento bloqueiam a passagem do vento que, por sua vez, exercem força sobre as pás, forçando o movimento do rotor a que estão conectadas.

O movimento do rotor em relação ao campo magnético estático (material ferromagnético presente no estator) gera um fluxo magnético que, por sua vez, induz uma tensão nas bobinas do rotor. A tensão induzida na saída é proporcional à taxa de variação no fluxo magnético que atravessa as espiras da bobina. Isto significa que a velocidade de rotação do rotor afeta a tensão induzida na saída.

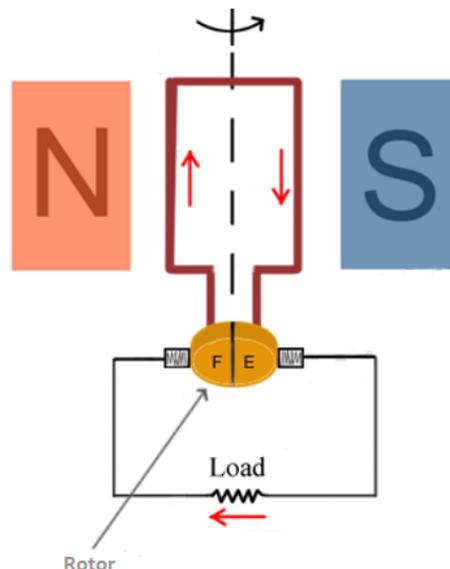


Figura 14: Princípio de funcionamento de um gerador CC. Obtido em [15]

4.5.3 O sistema operacional RIOT

O RIOT OS (*Real-Time Operating System*) [4] é um pequeno sistema operacional baseado em estrutura microkernel projetado para atender os requisitos de IoT, incluindo baixa utilização de memória (da ordem de alguns kilobytes), alta eficiência energética, recursos em tempo real, suporte para uma ampla variedade de hardware de baixo consumo, alta eficiência na comunicação de rede. Esse SO oferece um ambiente open-source, em que possibilita aos desenvolvedores programar o código fonte em linguagem C, C++ ou

Rust, conforme as necessidades do projeto. Seu uso é muito comum em dispositivos IoT por conta de sua eficiência energética, modularidade e pequena área de memória, além de oferecer suporte a diversos microcontroladores diferentes.

4.5.4 LoRa

O LoRa é uma tecnologia de comunicação utilizando rádio frequência com propriedades essenciais para aplicações no ramo de IoT [9]. O seu principal recurso vem do nome "Long Range", ou seja, a comunicação à longa distância, além disso a LoRa possui baixo custo de operação e baixo consumo energético.

Assim, foi importante a escolha desta tecnologia para este projeto, principalmente por atender aos requisitos de baixo consumo. A figura a seguir mostra o *stack* formado pela "Pulga Core" e pela "Pulga LoRa":



Figura 15: Stack formado pela Pulga core e pela Pulga LoRa

5 PROJETO

Nesta etapa detalhamos a realização do projeto, as decisões tomadas durante a escolha de alternativas e os procedimentos utilizados durante a realização de testes.

5.1 Nível 0

O objetivo do projeto é desenvolver um sistema de monitoramento florestal autônomo energeticamente. A base do sistema é composto pela Pulga, uma placa microcontroladora que conta com um processador, diversos sensores e suporte para circuitos de EH.

Para garantir a autonomia energética, utilizamos a coleta de energia de fontes do ambiente (EH), além de um firmware programado para economizar e até carregar a bateria.

O diagrama abaixo mostra o funcionamento simplificado do sistema completo.

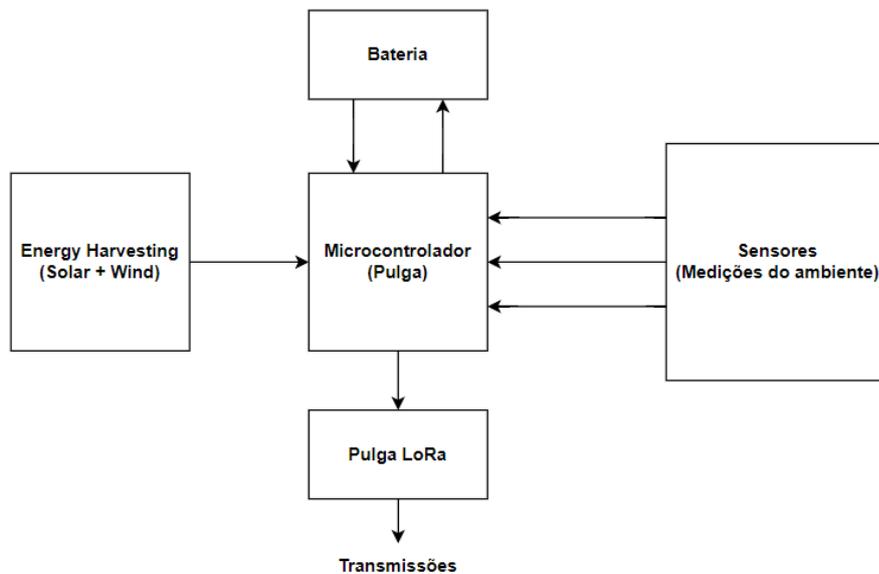


Figura 16: Diagrama do nível 0

5.2 Nível 1

Para alcançar nossos objetivos, foi necessário atuar em três frentes. O primeiro problema a ser estudado foi a **análise de consumo energético** do sistema completo. Para isso, foram feitos diversos testes com uma metodologia descrita no item 5.3.2.

Em seguida, projetamos o **circuito de energy harvesting** responsável por captar a energia do ambiente e auxiliar no fornecimento energético do sistema. Para isso, foram consideradas duas fontes: solar e eólica.

Por fim, foi necessário a **implementação do *firmware*** de modo a diminuir, ao máximo, o seu consumo de energia, principalmente no estado de sleep mode, que é o estado que o sistema permanece em grande parte do tempo, sem transmitir dados.

O item a seguir aprofunda mais cada uma dessas partes.

5.3 Nível 2

A seguir, aprofundamos o detalhamento de cada área destacada no item anterior.

5.3.1 Circuito da Fonte Solar

A primeira fonte utilizada é a solar. Esta será nossa principal fonte. Como o painel solar já produz tensão contínua, não é necessário o uso de retificadores. Dessa forma, o circuito de energy harvesting para esta fonte é apenas um regulador de tensão.

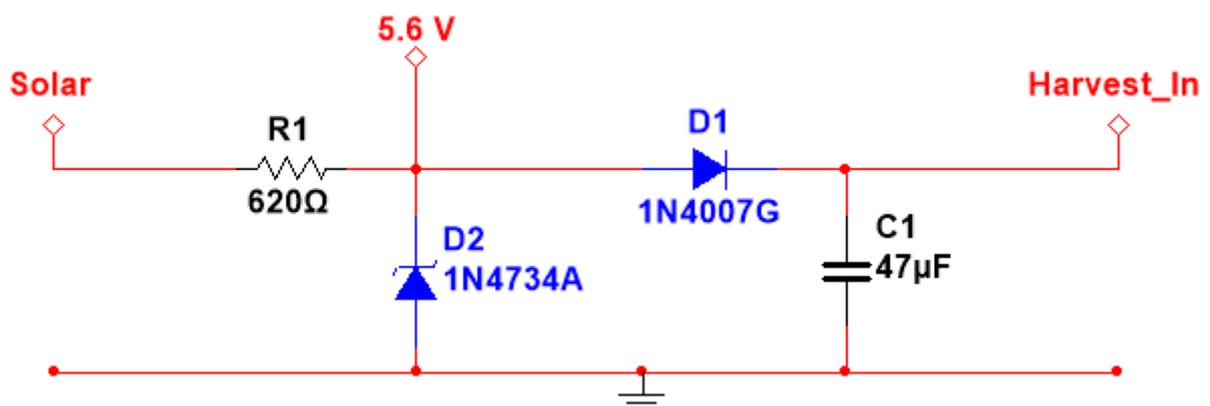


Figura 17: Circuito regulador de tensão para o painel solar

O painel solar é capaz de produzir tensões de até 7.5 V. No entanto, a Pulga só pode receber na entrada tensões de até 6.0 V. Dessa forma, um circuito regulador é necessário

para manter a tensão nessa faixa.

O circuito mostrado na figura é muito utilizado em produtos que dependem de reguladores, como carregadores de celular, e têm o seguinte funcionamento: o diodo zener (D2) mantém a tensão de saída em no máximo em 5.6 V, e o resistor (R1) consome a energia sobressalente. Um segundo estágio formado por um diodo comum (D1) e um capacitor (C1) funcionam como um filtro para estabilizar a saída.

Assim, quando a tensão no painel solar for menor que 5.6 V, a energia é toda transmitida para a saída (com uma queda de 0.6 V do diodo D2), e quando a tensão no painel é superior a 5.6 V, há queda de tensão no resistor de modo que a saída para a Pulga fique em 5.0 V aproximadamente.

O gráfico abaixo foi obtido no simulador de circuitos, Multisim, e mostra bem o funcionamento desejado. O eixo horizontal representa a tensão de entrada (Solar) e o eixo vertical mostra a tensão de saída (Harvest_In)

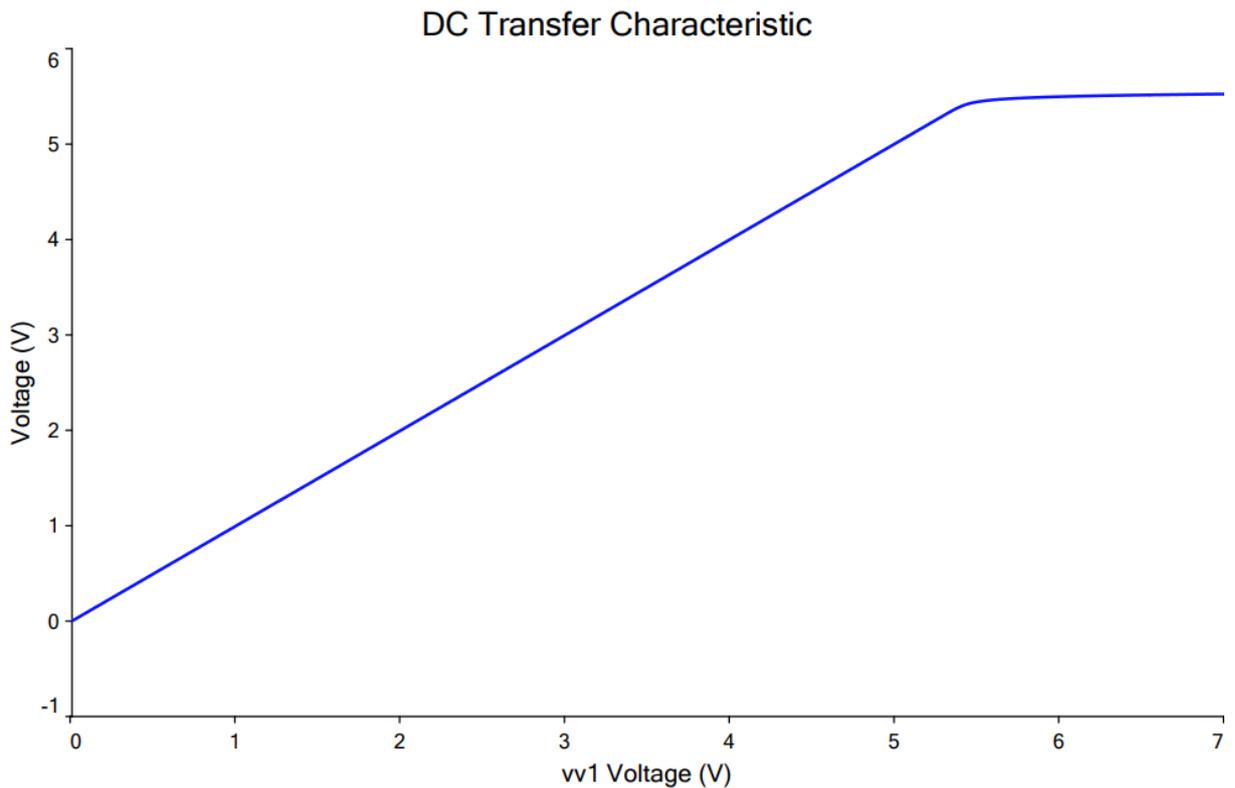


Figura 18: Tensão de saída quando varia-se a tensão de entrada

5.3.2 Múltiplas fontes de energia (energy harvesting)

Apesar dos testes iniciais mostrarem que o painel solar é uma promissora fonte de energia para nosso projeto, ele tem a limitação de depender das condições meteorológicas e, principalmente, de funcionar apenas durante o dia.

5.3.2.1 Gerador eólico

Dessa forma, foi necessário buscar uma outra fonte que seja capaz de auxiliar a alimentação do sistema quando o painel solar não conseguir fornecer corrente suficiente. Levando em consideração o ambiente florestal, a segunda fonte mais promissora foi a eólica.

Para isso, escolhemos um gerador eólico barato e eficiente, que consiga gerar os níveis de energia adequado ao nosso projeto.

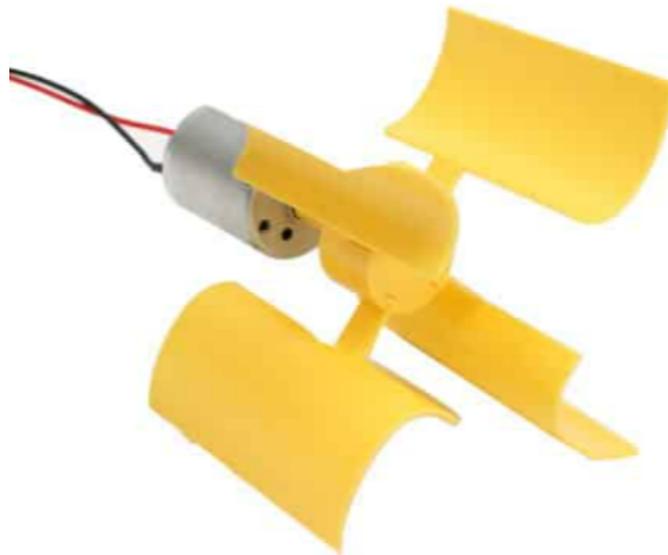


Figura 19: Gerador eólico

Esse gerador consiste em um motor com um eixo em que estão montadas 4 pás de 6 cm de comprimento por 3 cm de largura. De acordo com o fabricante, o gerador é capaz de gerar uma tensão média de 3 V e corrente média de 10 mA. Além disso, a taxa de rotação mínima é de 100 rpm, o que pode ser atingido facilmente com ventos leves.

5.3.2.2 Arquitetura do circuito

Após escolher o gerador, fica o desafio de como integrar ao nosso circuito de energy harvesting. Para isso, buscamos na literatura arquiteturas de circuitos que implementam o energy harvesting de múltiplas fontes.

O estudo publicado no " *Journal of Low Power Electronics and Applications* [17] faz uma comparação de diversas formas de realizar isso. Após analisar cada uma, decidimos adotar a arquitetura "Power ORing", que está indicada abaixo:

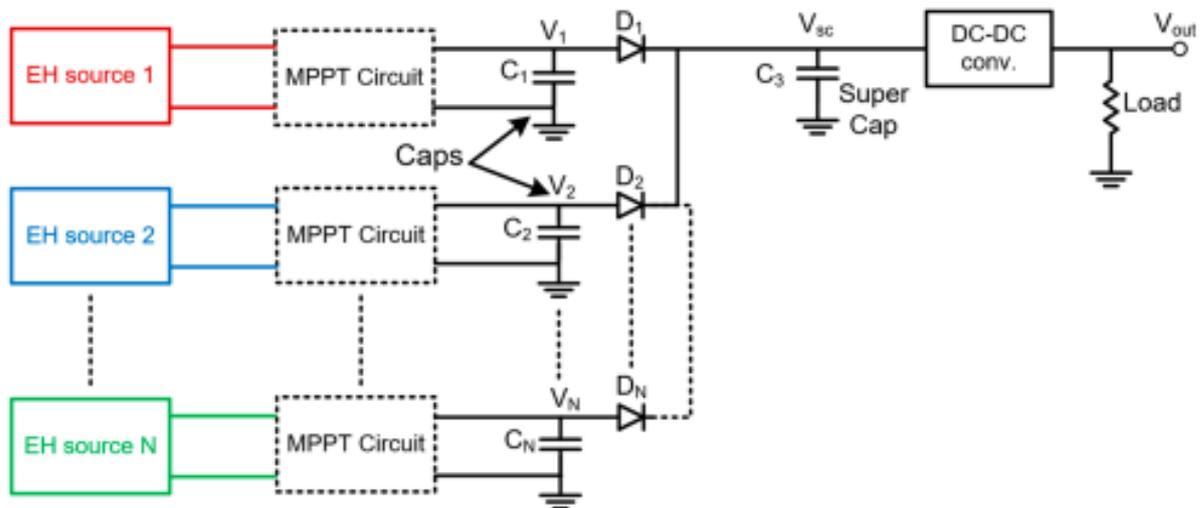


Figura 20: Arquitetura Power ORing para energy harvesting

A maior vantagem dessa arquitetura é que cada fonte pode atuar de maneira independente, além de que garante uma operação sincronizada, exigindo menos da unidade de gerenciamento de energia (PMU) do sistema.

Vale destacar que, nessa arquitetura, a energia de cada fonte não é somada. Apenas a fonte com maior tensão é selecionada (devido aos diodos) para a entrada do chip. Isso é desejável para nós pois o limite de tensão na entrada do PMIC da Pulga é baixo. Dessa forma, apenas uma fonte alimenta o circuito por vez.

Para exemplificar, ao longo do dia provavelmente temos a placa solar como fonte principal. No entanto, à medida em que vai ficando de noite o painel solar perde força e o gerador eólico pode atuar.

Por fim, temos nosso circuito de energy harvesting finalizado:

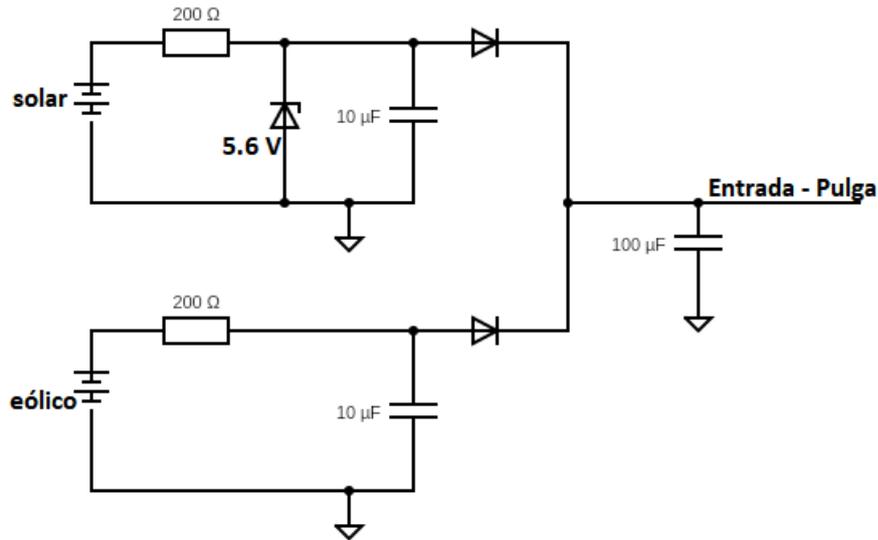


Figura 21: Circuito de energy harvesting com múltiplas fontes

5.3.3 Análise de consumo energético

A segunda etapa do projeto consistiu em fazer a análise de consumo energético da Pulga.

Para isso, nos baseamos em alguns artigos e materiais que mostram com detalhes diversas formas de analisar o consumo energético de uma placa como a Pulga. O método escolhido foi observar o consumo de corrente na bateria quando a placa está funcionando em todos os modos de operação.

A aplicação do projeto de monitoramento florestal consiste em manter a placa em baixíssimo consumo de corrente (sleep mode) até que ela " acorde" para fazer as medições e transmitir seus resultados.

Assim, esta etapa é extremamente importante para sabermos o quanto a placa está consumindo de corrente e, assim, podermos estimar o número de transmissões que a placa pode fazer e escolher a bateria mais adequada.

5.3.3.1 Consumo do firmware LoRa

Utilizando o método descrito, realizamos o teste de consumo energético do firmware. A figura a seguir mostra o comportamento da corrente consumida pela placa durante um ciclo de transmissão.

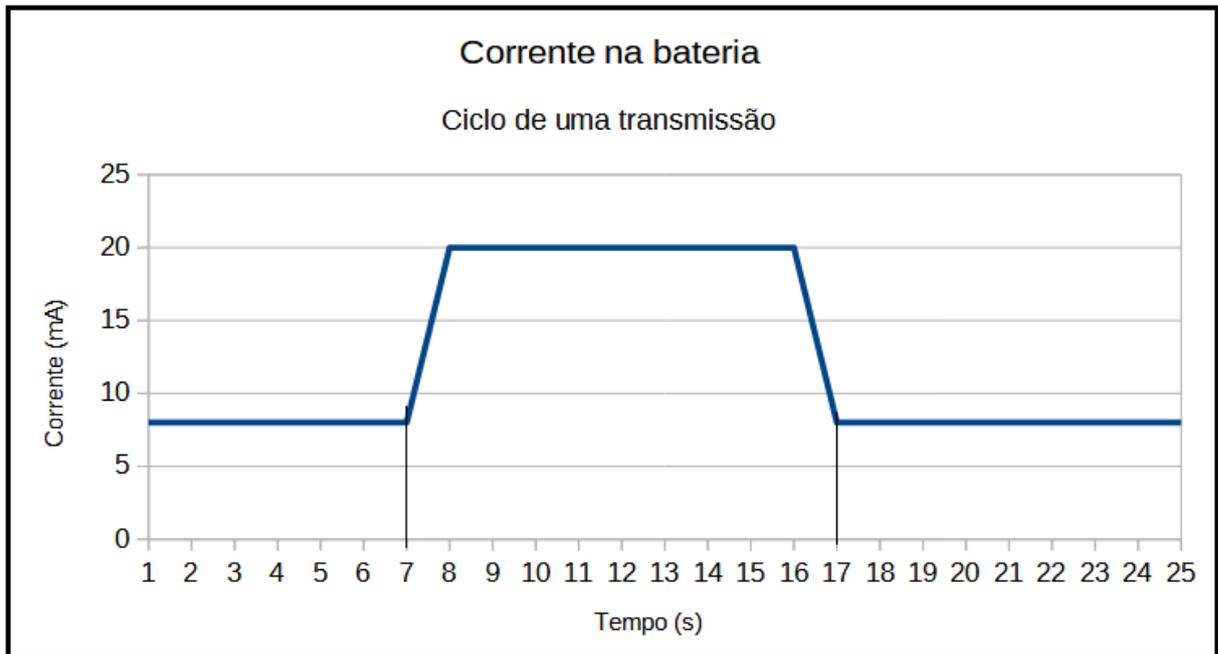


Figura 22: Corrente consumida pela Pulga

Como era esperado, a corrente fica estável em um certo valor e, durante a transmissão, atinge um pico. Nos experimentos, o valor de corrente fora da transmissão é de 8 mA, enquanto que na transmissão o valor chega a 20 mA. Vale destacar também que os intervalos de tempo indicados no gráfico foram superestimados para garantir confiabilidade. Os testes mostraram que o modo de transmissão dura por volta de 1 segundo.

No momento, essa corrente é totalmente consumida da bateria. Um dos objetivos do projeto é fazer com que a fonte harvesting seja capaz de assumir esta responsabilidade e, ainda, carregar a bateria. No entanto, com os valores de consumo nesse nível, isto não será possível. Isto porque, por ser um ambiente florestal, com mata fechada, o painel solar estará quase sempre na sombra. Logo, a corrente gerada por ele será de 2 mA, como mostram os testes, na seção seguinte.

Dessa forma, a parte de otimização do firmware será essencial para o sucesso do projeto. Diminuindo a corrente necessária para alimentar a placa, o painel solar poderá, pelo menos, assumir o controle energético durante os momentos de "sleep mode".

5.4 Desenvolvimento do firmware

A função deste SO no projeto será controlar a distribuição de energia. O consumo total de energia de um dispositivo é a soma da energia consumida por CPU, dispositivos conectados por meio de periféricos e outros componentes passivos externos a ele.

No projeto de SO para a Pulga, o sistema deve possuir estados ou *power modes* que incluem o estado *idle*, o modo de funcionamento ativo com disponibilidade e desempenho para realizar transmissões e o *sleep mode*, modos em que há economia de consumo, apenas o necessário para manter o sistema ligado. Além disso, o sistema deve ser capaz de transitar entre eles de acordo com as condições automaticamente. A ideia central será manter o sistema no *low-power mode* no maior tempo possível e entrar no modo ativo apenas durante certos períodos apenas para realizar transmissões de dados.

5.4.1 Arquitetura do RIOT OS

O RIOT é estruturado em módulos de software que são agregados em tempo de compilação, em torno de um *kernel* que fornece funcionalidade minimalista. Esta abordagem permite construir o sistema completo de forma modular, ou seja, incluindo apenas os módulos que são requeridos pelo caso de uso, reduzindo-se assim tanto o consumo de memória quanto a complexidade do sistema [4]. Um esquema representativo da estrutura do RIOT é apresentado abaixo:

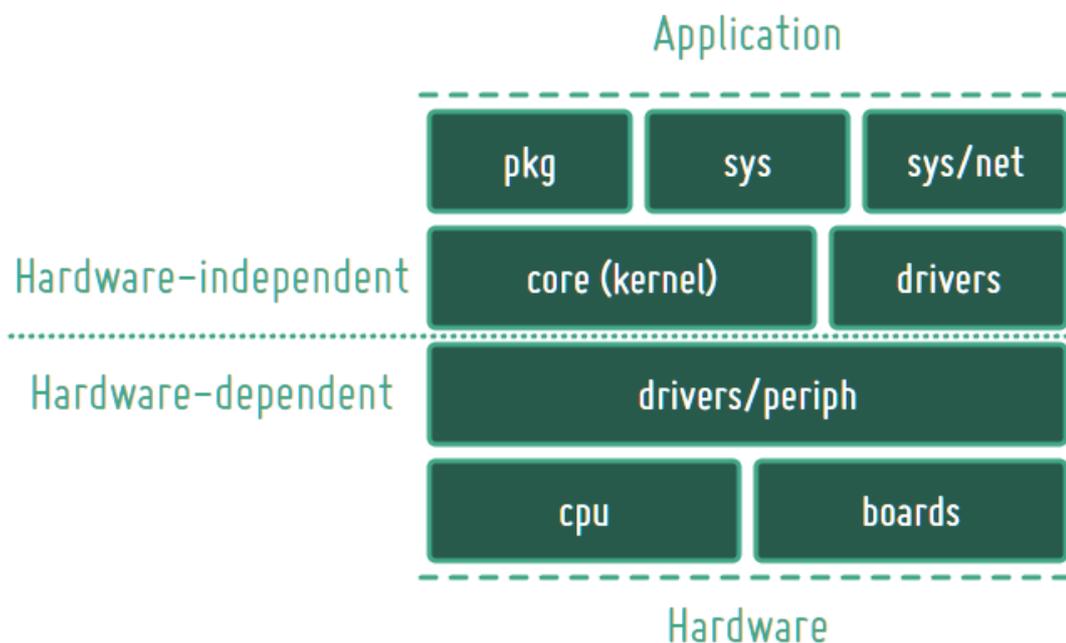


Figura 23: Estrutura do RIOT OS

Neste projeto, a implementação do *firmware* será feita no diretório *cpu* da estrutura, onde estão implementadas as configurações específicas de cada processador suportado pelo sistema, como no exemplo da figura abaixo, no caso da Pulga, o *nrf52840*. Essas configurações abrangem *power management*, tratamento de interrupção, código de inicialização, código de inicialização de *clock* e manipulação de *thread*.

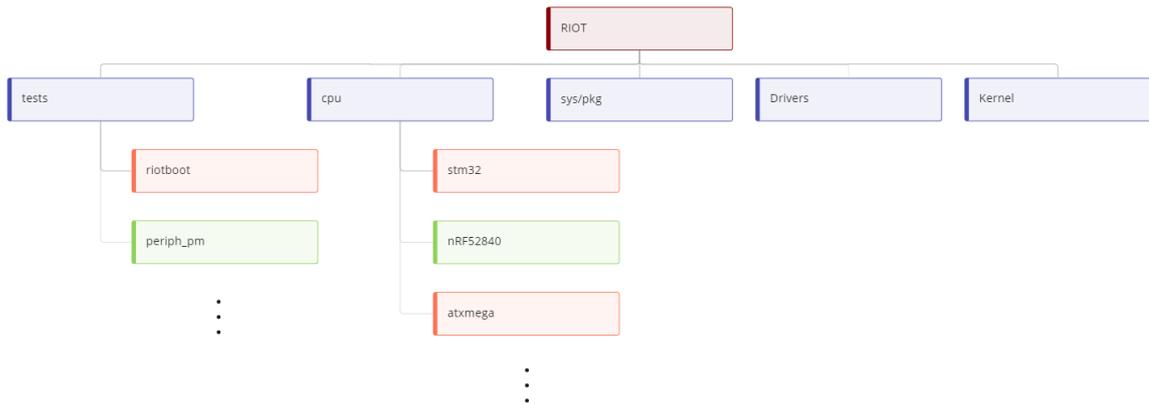


Figura 24: Esquema de diretórios do RIOT OS

A modalidade *open-source* do código do SO permite serem feitas modificações a fim de engajar-se em um projeto, implementando novas funcionalidades e suportes a diferentes processadores. O código fonte está hospedado no ambiente Github, uma plataforma que controla o versionamento de um repositório remoto por meio do *software* Git, permitindo que usuários contribuam em projetos privados.

5.4.2 Power management no RIOT

Um dos recursos disponibilizados pelo código *open-source* do RIOT OS é a interface para implementação do gerenciamento de energia. Essa interface é descrita no *kernel* do sistema e deve ser utilizada para implementação das diferentes configurações de gerenciamento de energia, para os variados tipos de CPUs. Essa interface é composta pelas seguintes funções:

- *pm set*: Função chamada para setar o estado para qual o sistema deve transitar.
- *pm off*: Desligar praticamente todos os componentes do MCU.
- *pm set lowest*: Transitar para o estado de máxima economia de energia.
- *pm block e unblock*: Respectivamente, bloquear e desbloquear estados.

5.4.3 Configuração e Implementação

Dentro do diretório `cpu/nrf52` do RIOT OS foi criado um arquivo em C "`pm.c`" onde foram feitas implementações das funções da interface acima. Na primeira delas, com uma lógica de *switch/case*, define-se um estado diferente para cada condição, de acordo com o parâmetro de entrada, um número inteiro de 0 à quantidade máxima de estados.

A configuração dos *power modes* foi feita com acessos aos registradores de controles de energia do CPU. Mais detalhes sobre o código-fonte encontra-se no Apêndice D deste documento.

5.4.3.1 Função de entrada ao Low-Power

Primeiramente define-se a função de entrada do processador, identificado pelo nome *nrf52 sys on*. As funções SEV (*Send Event*) e WFE (*Wait For Event*) contidas nesta definição servem como forma de colocar o processador em estado "sleep". A primeira, significa acionar uma instrução de evento, neste caso, ela é chamada no início a fim de "limpar" o registrador de eventos para a execução do WFE, uma instrução de entrar em modo de espera ou *Low-power*, pausando a execução do código, até que um evento ocorra.

5.4.3.2 Configuração do RTC

A fim de obter uma transição entre estados automático, foi escolhido implementar um contador RTC (*Real-Time Clock*), ou seja, um contador de segundos em tempo real a base de *clock* de baixa frequência (LFCLK) do processador. Este contador deve manter o estado *sleep* em um tempo predefinido até enviar um evento de *trigger* para acordar o sistema deste estado.

O módulo RTC do nRF52840 possui alguns parâmetros os quais devem ser inicialmente configurados. Um esquema extraído da documentação deste processador ilustra o bloco contador.

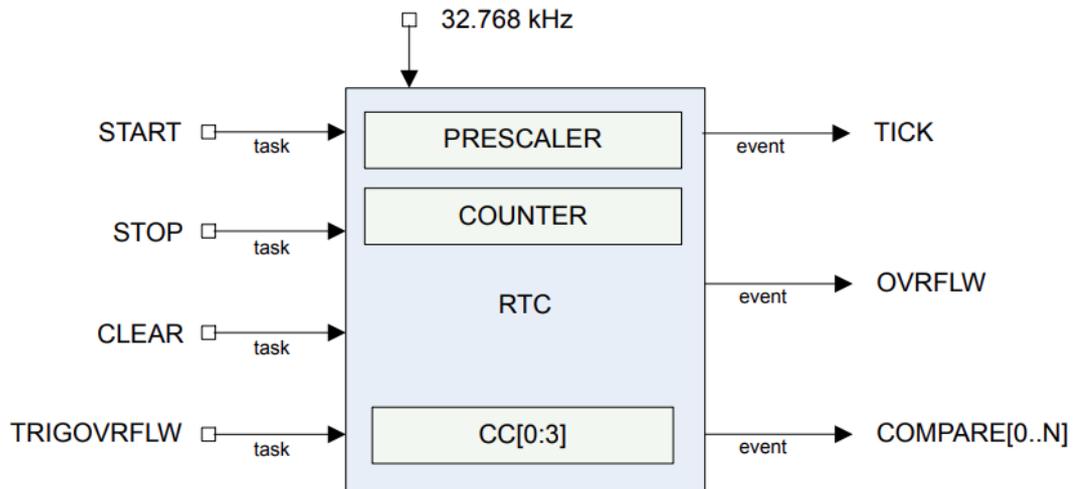


Figura 25: Bloco esquemático do contador RTC

No desenvolvimento para a Pulga, primeiramente define-se o STOP e o CLEAR, registradores de parar e reiniciar o contador, como 1, garantindo a contagem desde o zero. O PRESCALER é um registrador de escala em relação ao LFCLK que possui uma frequência base de 32768 kHz. Neste caso para obter uma contagem de 10 segundos por exemplo, pode-se definir o PRESCALE como 0, sem escala, e o registrador comparador CC[0] de valor como $10 * 32768$, isso significa que após este número de oscilações este enviará um evento COMPARE[0] na saída do bloco. Após todas estas definições, finalmente o contador é iniciado com o $START = 1$.

5.4.3.3 Definição do gerenciador

A função *pm set* é a principal função de comandos para realização de transições entre estados. Como descrito acima, foi utilizado uma lógica *switch/case* para transitar entre o *mode 0* (Sleep) e o modo 1 (Ativo). Com o intuito de o sistema realizar transições entre estados de forma automática, foi utilizado uma biblioteca *ztimer* para a contagem de segundos até a atribuição do novo estado. Vale notar que apenas no modo ativo utiliza-se o *ztimer*, já que este contador não gera *triggers* de evento, portanto coloca o sistema em um estado de espera, porém em seu desempenho máximo. O modo "sleep", por sua vez, utiliza o próprio *clock* do processador como forma de contagem, gerando *triggers* de evento e permitindo acordar do estado definido pela função *Wait for Event*.

Por fim, as funções *pm off* e *pm set lowest* são predefinidas de um diretório contendo características equivalentes da família de processadores nRF. São funções que definem o estado "deep sleep" do processador, no caso do nRF52840, o SYSTEM OFF.

5.4.3.4 Implementação da Interface

Acrescentando ao desenvolvimento do *firmware* gerenciador de consumo de energia, foi também criado um arquivo "pulga-interface.c" contendo uma interface para que seja possível a visualização do seu funcionamento. Para isso, foi aproveitado um recurso de *Shell* do RIOT OS que permite realizar comandos como *pm set*, apresentando ao usuário o estado cujo o dispositivo se encontra.

Esta interface permite chamadas de funções por meio uma simples digitação de uma linha como por exemplo *pm set 0*, ou seja, entrar no modo de parâmetro 0 do *power management*. Isso facilita tanto na execução testes quanto na visualização do funcionamento do programa principal.

6 TESTES E RESULTADOS

Nesta seção, serão apresentados os testes mais relevantes feitos para averiguar o funcionamento das diversas etapas do projeto, além de comparações e análises dos resultados. Esta etapa é separada em duas partes principais, os testes de captação e os testes de consumo energético.

6.1 Testes de captação

6.1.1 Capacidade de geração do painel solar

Este primeiro teste teve o objetivo de nos auxiliar a decidir qual painel solar iríamos utilizar. Haviam duas alternativas: um painel de dimensões 6x11 centímetros (utilizado para a prova de conceito) e outro de dimensões 13x16 centímetros. O desafio nesta escolha é conciliar dois requisitos conflitantes: portabilidade e capacidade de geração.

Os testes foram realizados em mesmas condições ambientes (céu limpo, com sol no período da tarde, após 13h). As figuras 14 e 15 mostram os gráficos da tensão gerada por cada painel. Vale destacar que, para o painel menor, de 6x11 cm, foi feita a aquisição da tensão nos terminais do painel, enquanto para o maior, de 13x16 cm, foi feita a aquisição na saída do regulador projetado.

Analisando os dois gráficos, fizemos os seguintes levantamentos: o painel de 13 x 16 cm, apesar de ser maior, é capaz de fornecer uma corrente capaz de ligar a placa mesmo quando não há sol diretamente, o que nos fornece maior confiabilidade. Afinal, devemos levar em conta que o painel solar ficará, em sua maior parte, sob a sombra. Dessa forma, o painel maior foi capaz de dar autonomia à placa por um período de 6 horas, enquanto que o painel menor só foi capaz de deixar a placa ligada por pouco mais de 2 horas.

Com esses levantamentos, concluímos que seria mais adequado utilizar o painel solar de 13x16 cm, já que nos fornecerá mais confiabilidade energética. Todos os testes seguintes foram feitos apenas com este painel.

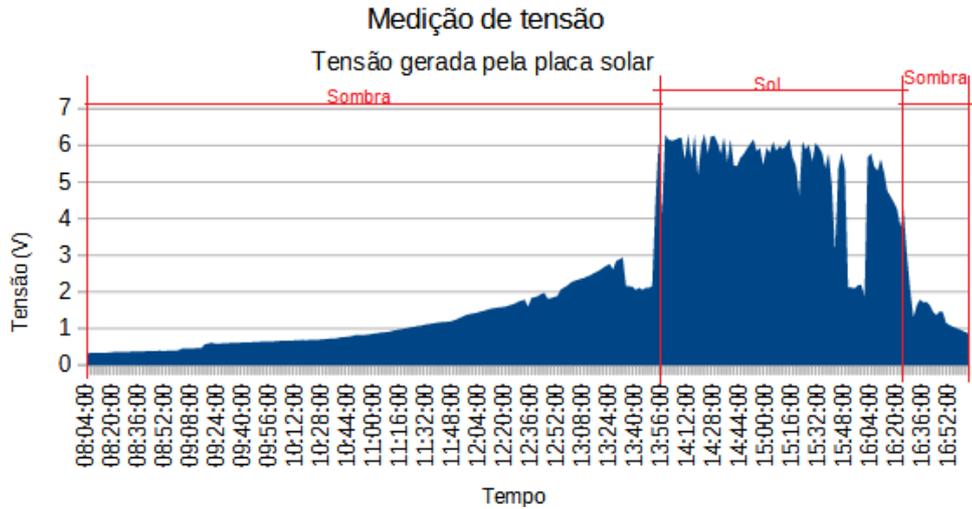


Figura 26: Tensão gerada na placa 6x11 cm



Figura 27: Tensão gerada na placa 13x16 cm

6.1.2 Corrente mínima e média fornecida pelo painel solar

O segundo teste teve o objetivo de identificar a corrente média fornecida pelo painel solar, a corrente mínima para ligar o painel e a corrente mínima que mantém a placa ligada. Nesse teste, utilizamos um firmware simples de baixo consumo (Bluetooth Low Energy - BLE).

Para isso, realizamos os seguintes passos: deixar o painel totalmente na sombra e lentamente expô-lo ao sol. Após a placa ligar, fazemos o movimento contrário, até a placa desligar.

Corrente mínima para ligar	7,0 mA
Corrente mínima para funcionar	3,2 mA
Corrente média na sombra (totalmente)	3,0 mA
Corrente média no sol (totalmente)	8,5 mA

Tabela 1: Valores de corrente observados no painel solar

A partir de observações dos testes, identificamos que a placa é capaz de fornecer os 7 mA mínimos quando uma pequena área da placa está exposta diretamente ao sol.

6.1.3 Corrente na bateria quando auxiliada pelo painel solar

Neste teste, conduzimos um experimento para averiguar a corrente na bateria enquanto o painel estava conectado. A partir dos dados na figura abaixo, podemos concluir que, mesmo quando o painel solar encontra-se totalmente na sombra, o consumo de corrente na bateria cai drasticamente. Além disso, quando há algum sol incidindo na placa, a bateria chega a carregar (valores negativos de corrente).

Situação	Corrente
Sem painel	1.91 mA
Painel na sombra	130 μ A
Painel parcialmente no sol	- 50 μ A
Painel totalmente no sol	- 240 μ A

Figura 28: Valores de corrente observados na bateria

6.2 Testes de consumo

Esta seção apresenta os testes feitos com o firmware mais próximo do real, isto é, aquele que será de fato utilizado no monitoramento florestal.

6.2.1 Testes pré-otimização

Inicialmente, foi testado o sistema com máxima taxa de consumo, ou seja, sem otimização de *firmware*, sem o circuito de captação e em processo de transmissão com LoRa,

para analisar os picos de correntes atingidos e comparar o consumo posteriormente com os testes após a otimização.

Como era esperado, a corrente forma um pulso (de 20 mA) quando é realizada a transmissão e um consumo mínimo de 8 mA quando a placa está em estado ativo. Como discutido na seção 5.3.3.1, esse consumo é muito alto e, se não fosse otimizado, drenaria toda a bateria em algumas semanas.

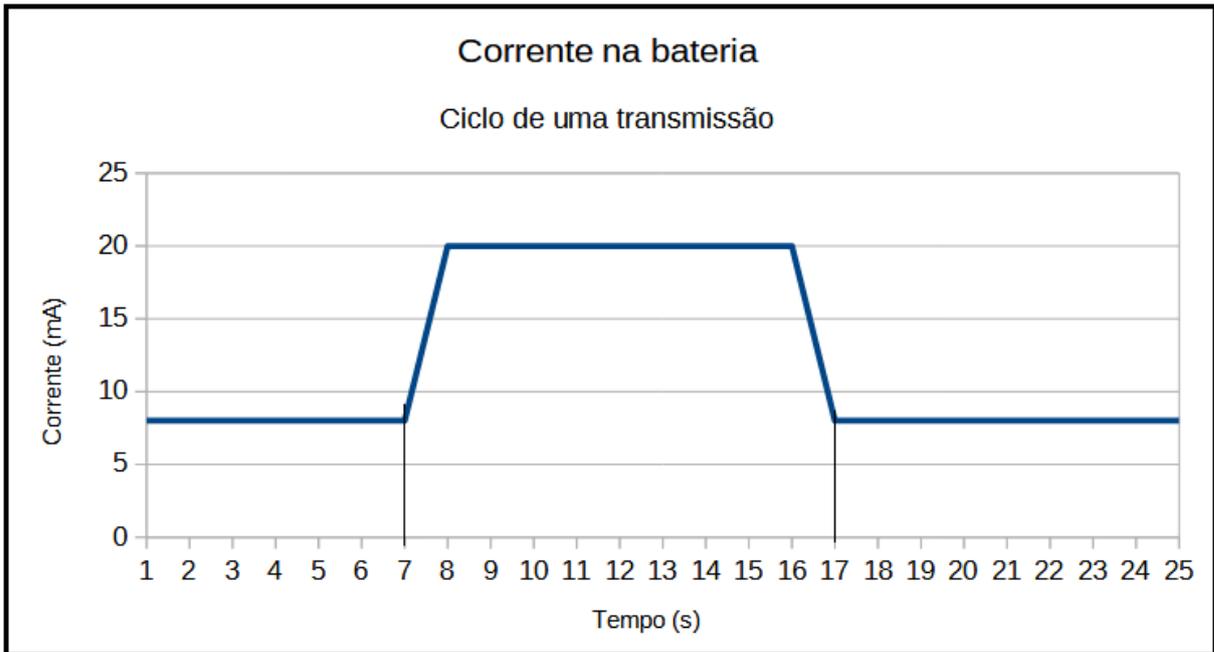


Figura 29: Corrente consumida pela Pulga

Já quando adicionamos o painel solar nas mesmas condições que esperamos na floresta, isto é, totalmente sob a sombra, temos uma diminuição do consumo da bateria em aproximadamente 2.5 mA. Os testes realizados com o painel solar em um dia ensolarado mostraram que o consumo na bateria cai para 5.5 mA no estado de espera, e para 16.9 mA no estado de transmissão.

6.2.2 Testes pós-otimização

Após o desenvolvimento do *firmware* responsável pela economia de consumo, foram realizadas medições de consumo do sistema utilizando o multímetro na escala de mA. Os *power modes* testados foram o *Idle*, SYSTEM ON (*Low power mode*) e o SYSTEM OFF. Uma tabela com dados de correntes medidas na Pulga em seus diferentes estados de economia enquanto alimentada somente com a bateria é apresentada abaixo:

Estado	Consumo
Idle	8 mA
Sleep mode	5,55 mA
SYSTEM OFF	0,88 mA

Tabela 2: Valores de corrente com otimização de firmware

É possível verificar na tabela acima uma considerável economia de aproximadamente trinta por cento do total consumido, quando em estado "sleep". Considerando uma rede de grande quantidade de sensores, esta redução é significativa e produziu um resultado satisfatório para o projeto.

6.2.3 Interface de apresentação

Conforme descrito no capítulo anterior, foi necessário o desenvolvimento de uma interface gráfica a fim de verificar o funcionamento do módulo gerenciador. A captura de tela a seguir mostra o resultado deste desenvolvimento:

```

2022-12-14 17:55:27,500 # Using BTM0 as wake-up source
> pm set 0
2022-12-14 17:55:30,969 # main(): This is RIOT! (Version: ad0e2-develop)
2022-12-14 17:55:30,970 # Pulga Low Power!
2022-12-14 17:55:30,971 # ZTIMER definido!
2022-12-14 17:55:30,975 # You are running RIOT on a(n) pulga board.
2022-12-14 17:55:30,978 # This board features a(n) nrf52 MCU.
2022-12-14 17:55:30,984 # This application allows you to test the CPU power management.
2022-12-14 17:55:30,989 # The available power modes are 0 - 2. Lower-numbered power modes
2022-12-14 17:55:30,995 # save more power, but may require an event/interrupt to wake up
2022-12-14 17:55:30,998 # the CPU. Reset the CPU if needed.
2022-12-14 17:55:30,998 #
2022-12-14 17:55:30,999 #
2022-12-14 17:55:31,001 # using BTN0 as wake-up source
>
2022-12-14 17:55:31,143 # pm set 0
2022-12-14 17:55:31,146 # CPU is entering power mode 0.
2022-12-14 17:55:31,149 # Now waiting for a wakeup event...
2022-12-14 17:55:31,150 # Entering sleep
2022-12-14 17:55:41,150 # Exiting sleep
2022-12-14 17:55:41,152 # Entering active
2022-12-14 17:55:51,152 # Entering sleep

```

Figura 30: Shell do RIOT apresentando a interface

Após o carregamento do código na Pulga através do J-link, o *Shell* do sistema é aberto e são apresentados respectivamente, a placa, MCU, número de *Power modes*, e a forma padrão de "acordar" o sistema.

Da figura acima, pode-se visualizar que após a chamada da função *pm set* do modo 0, o sistema entra em um ciclo de transição de estados a cada 10 segundos entre modos ativo e *sleep* sendo possível validar o funcionamento graças à interface. Para evitar qualquer

bloqueamento automático de *power management* por algum aplicativo ou periférico ativo no sistema, é necessário desbloquear com *pm unblock* de todos os estados.

6.3 Análise dos resultados

A partir dos resultados dos testes é possível fazer a análise de quanto a bateria que alimenta a Pulga duraria para os casos do sistema antes e após da otimização e também nas condições em que se emprega, ou não, o circuito de captação.

Em todas as análises será considerado que o painel solar estará totalmente sobre a sombra, e que ele consegue captar energia 8 horas por dia. Também será considerado um aumento linear da energia captada pelo painel solar ao compararmos um painel com uma outra área com o painel utilizado nos testes.

6.3.1 Circuito não otimizado e alimentado sem o circuito de captação

Como obtido nos testes, quando o sistema é alimentado sem o circuito de captação a corrente na bateria é de 8 mA quando está no estado *Idle* e de 20 mA durante a transmissão. Para essas estimativas, consideramos que a transmissão dura 1 segundo e que no resto do tempo o sistema fica no estado *Idle*. A seguinte fórmula foi utilizada para calcular quantos dias a bateria durará:

$$\frac{\text{Capacidade da bateria}}{8 * (24 - \frac{\text{transmissões diárias}}{3600}) + 20 * \frac{\text{transmissões diárias}}{3600}}$$

Aplicando a fórmula, foi possível levantar diversas estimativas. A tabela a seguir mostra quantos dias a bateria duraria variando-se o número de transmissões por dia e a capacidade da bateria.

Capacidade (mAh)	2	24	240	580	1440	8640
1000	5,21	5,21	5,19	5,16	5,08	4,53
2600	13,54	13,54	13,49	13,41	13,21	11,78
5000	26,04	26,03	25,93	25,78	25,41	22,64
10000	52,08	52,06	51,87	51,56	50,81	45,29

Tabela 3: Relação da duração da bateria e número de transmissões pré otimização

Na tabela podemos ver, por exemplo, que nessas condições a bateria duraria aproximadamente duas semanas fazendo 2 transmissões diárias. Além disso, podemos destacar

que a quantidade de transmissões não afeta significativamente a duração da bateria, o que já era esperado, visto que a transmissão é rápida.

6.3.2 Circuito não otimizado e alimentado com o circuito de captação

Neste segundo caso, agora utilizando o circuito de captação e considerando que o painel solar fique exposto por 8 horas diárias ao sol, a corrente na bateria diminui de 8 mA para 5,5 mA no estado *idle*, e diminui de 20 mA para 16,9 mA na transmissão.

Neste estágio podemos notar que o circuito de captação reduz o consumo na bateria, gerando uma economia média de 2 dias, o que ainda é muito baixo, destacando-se a importância da otimização.

Ainda, podemos fazer estimativas de duração de carga da bateria para diferentes tamanhos de painéis solares, uma vez que esta é nossa fonte de captação principal.

A tabela abaixo mostra esta estimativa de tempo de carga variando-se capacidade da bateria e a área do painel para um valor fixo de 24 transmissões diárias (uma vez por hora). Variando-se a quantidade de transmissões diárias, obtemos valores muito próximos entre si.

Capacidade (mAh)	104 cm ²	208 cm ²	416 cm ²	832 cm ²
1000	5,49	5,81	6,58	8,92
2600	14,28	15,11	17,10	23,20
5000	27,46	29,06	32,88	44,61
10000	54,92	58,11	65,76	89,23

Tabela 4: Relação da duração da bateria e tamanho do painel solar pré otimização

6.3.3 Circuito otimizado e alimentado sem o circuito de captação

Após a otimização, implementamos as mesmas análises para observar a redução do consumo obtida. Neste caso, temos um novo estado, o *sleep*. Neste estado o consumo de corrente na bateria é de 5,55 mA aproximadamente.

O sistema permanece em *sleep* até o momento da transmissão, em que primeiro entra no *idle* para depois sim transmitir.

Em relação à análise no item 6.3.1, obtivemos um aumento médio na duração da bateria de 44%. A tabela a seguir mostra quantos dias a bateria duraria nessa nova configuração.

Capacidade (mAh)	2	24	240	580	1440
1000	7,51	7,49	7,37	7,19	6,77
2600	19,52	19,48	19,17	18,69	17,59
5000	37,53	37,47	36,86	35,95	33,83
10000	75,06	74,94	73,73	71,90	67,66

Tabela 5: Relação da duração da bateria e número de transmissões pós otimização

6.3.4 Circuito otimizado e alimentado com o circuito de captação

Esta última configuração é a principal, uma vez que junta as duas frentes do trabalho, tanto a otimização quanto a captação de energia do ambiente.

Utilizando o circuito de captação e considerando que o painel solar fique exposto por 8 horas diárias ao sol, a corrente na bateria no estado *sleep* diminui de 5,55 mA para 3 mA, diminui de 8 mA para 5,5 mA no estado *idle*, e diminui de 20 mA para 16,9 mA na transmissão. Isso permitiu um aumento médio de 65% no tempo de vida da bateria.

Novamente, podemos fazer estimativas de duração de carga da bateria para diferentes tamanhos de painéis solares. A tabela abaixo mostra esta estimativa de tempo de carga variando-se capacidade da bateria e a área do painel para um valor fixo de 24 transmissões diárias (uma vez por hora).

Capacidade (mAh)	104 cm ²	208 cm ²	416 cm ²	832 cm ²
1000	8,10	8,82	10,70	18,71
2600	21,06	22,92	27,83	48,65
5000	40,50	44,08	53,51	93,57
10000	81,01	88,15	107,02	187,13

Tabela 6: Relação da duração da bateria e tamanho do painel solar pós otimização

Para a placa com consumo otimizado com painel solar ser viável para o projeto que ela irá participar é necessário que a sua bateria dure alguns anos, para isso será necessário aumentar ou o tamanho do painel solar, ou a capacidade da bateria ou ambos. As tabelas abaixo mostram combinações de capacidades de bateria e tamanhos de painéis solares para que a bateria dure 1, 2 e 5 anos. Para isso serão consideradas 2 transmissões por dia, que é o esperado para o projeto.

1 ANO:

tamanho do painel (cm ²)	208	416	832	1101	1243	1311
capacidade da bateria (mAh)	41354	34049	19439	10000	5000	2600
duração da bateria (dias)	365,25	365,25	365,25	365,25	365,25	365,25

2 ANOS:

tamanho do painel (cm ²)	208	416	832	1243	1314	1348
capacidade da bateria (mAh)	82707	68097	38877	10000	5000	2600
duração da bateria (dias)	730,5	730,5	730,5	730,5	730,5	730,5

5 ANOS:

tamanho do painel (cm ²)	208	416	832	1329	1357	1371
capacidade da bateria (mAh)	206768	170243	97193	10000	5000	2600
duração da bateria (dias)	1826,25	1826,25	1826,25	1826,25	1826,25	1826,25

Figura 31: Requisitos para o sistema durar entre 1, 2 e 5 anos

7 CONSIDERAÇÕES FINAIS

Esta seção apresenta conclusões e objetivos alcançados, estimativas feitas sobre a capacidade energética do sistema e uma reflexão sobre o que poderia ser melhorado em trabalhos futuros.

7.1 Conclusões do Projeto de Formatura

A partir de um balanço geral, o projeto conseguiu alcançar o objetivo e os requisitos propostos, mas com algumas ressalvas. Olhando para o que foi desenvolvido, o projeto resultou em um sistema com o firmware otimizado e compatível com mais de uma fonte de energia de captação, além da análise de consumo energético feito para o sistema. A partir dos testes realizados, notou-se que a otimização de firmware foi o principal fator para solucionar o problema de vida útil das baterias, contrariando expectativas, já que acreditava-se que inicialmente o energy harvesting teria maior impacto. Deve-se atribuir isso a todas as limitações impostas, como o fato do painel solar estar sob a sombra. Se fosse possível manter o painel parcialmente sob a luz solar, o consumo na bateria seria muito maior, chegando até a carregá-la, como o teste feito mostrou. Por fim, vale destacar que mesmo com as limitações, os requisitos propostos inicialmente foram alcançados, o que trouxe alguma viabilidade no lado da questão energética ao projeto inicial. Ainda que a longo prazo o sistema não se sustente, foi possível gerar um aumento médio de 41% no tempo útil da bateria mesmo sem o sistema de captação, e aumento de 65% quando a captação está presente.

7.2 Contribuições

No contexto de redes de sensores, principalmente focado na sua aplicação em florestas, o projeto trouxe algumas contribuições. Pode-se destacar todo estudo e análise de consumo energético dessas redes, o que possibilitou identificar e entender o funcionamento

do sistema em seus diferentes modos de operação, projetar o circuito de captação de energia e, principalmente, fazer estimativas e projeções para a operação do sistema a longo prazo. Além disso, foi possível contribuir com o software do projeto, uma vez que através do estudo do RIOT e de *drivers* já existentes de gerenciamento de energia, foi possível da criação do nosso próprio *driver* para o nRF52840 que implementa o modo *sleep*, cuja função é otimizar o consumo energético do sistema. Todos esses dados podem ser reaproveitados para outros projetos parecidos e até para futuro desenvolvimento deste mesmo sistema.

7.3 Perspectivas de Continuidade

Como perspectivas de continuidade, o projeto pode ser aperfeiçoado em alguns pontos. Do ponto de vista do consumo energético, é possível aprimorar o sistema de gerenciamento com estudo de novas técnicas e explorar diferentes formas de despertar, visto que cada cenário pode produzir um diferente resultado. Um exemplo seria um gerenciamento mais dinâmico, ou seja, transitar ao estado ativo apenas quando o módulo de EH estiver com alta eficiência. Além disso, seria interessante fazer um estudo de uma bateria mais adequada ao projeto, uma vez que a utilizada neste projeto não passou por análise, e foi escolhida por praticidade.

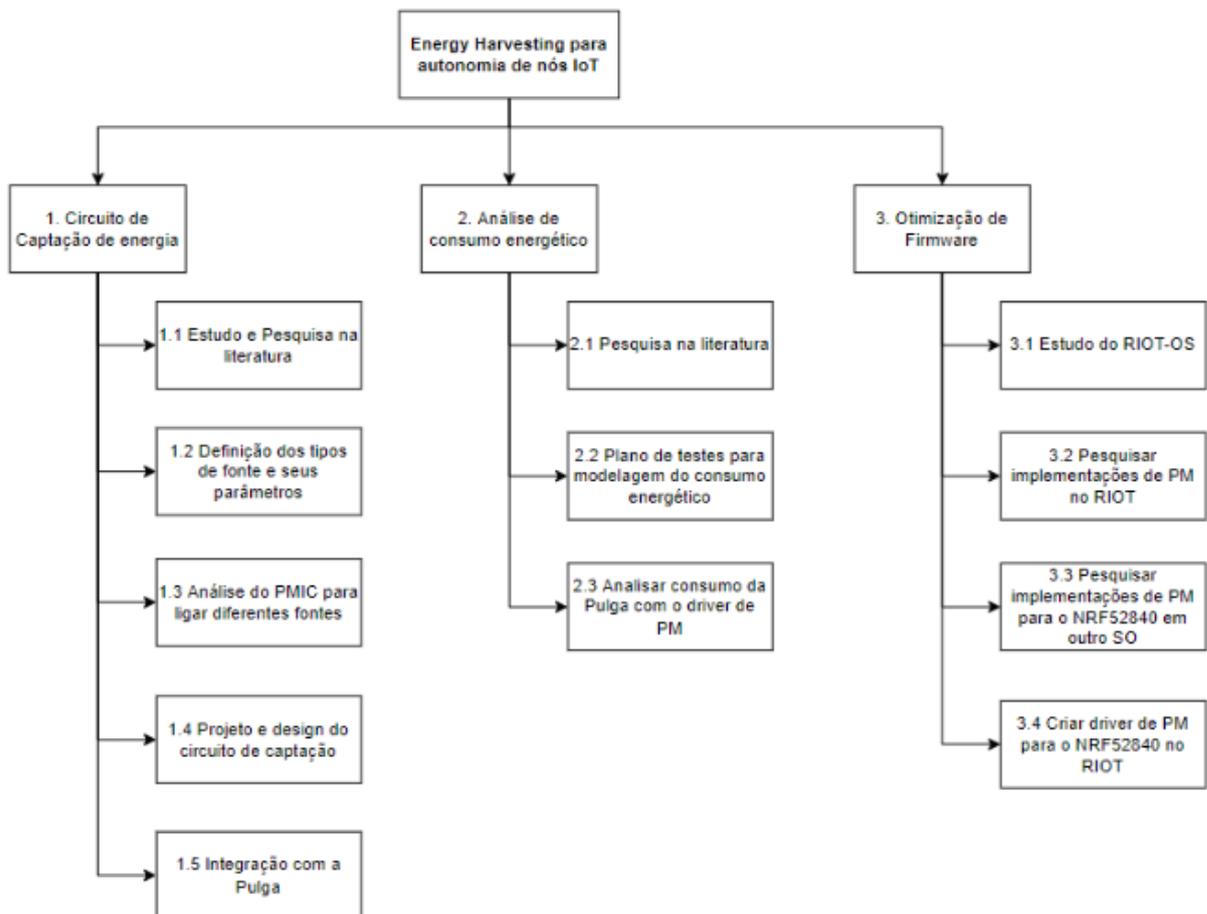
REFERÊNCIAS

- [1] **Datasheet do CPU nRF52840.** infocenter.nordicsemi.com/pdf/nRF52840_PS_v1.1.pdf.
- [2] **Datasheet do PMIC LTC3106.** <https://pdf1.alldatasheet.com/datasheet-pdf/view/1040972/LINER/LTC3106.html>.
- [3] **Energia do ar: ondas eletromagnéticas do ambiente viram fonte de energia.** <https://www.inovacaotecnologica.com.br/noticias/noticia.php?artigo=energia-ar-ondas-eletromagneticasid=010115110708>. Acessado em 18/07/2022.
- [4] **Informações e documentação do RIOT.** <https://www.github.com/RIOT-OS/RIOT>. Acessado em 18/07/2022.
- [5] **Programa Caninos Loucos.** <https://caninosloucos.org/pt/>. Acessado em 10-07-2022.
- [6] Chapter 41 - engineered nanomaterials for energy applications. In Chaudhery Mustansar Hussain, editor, *Handbook of Nanomaterials for Industrial Applications*, Micro and Nano Technologies, pages 751–767. Elsevier, 2018.
- [7] Chapter 7 - high efficiency plants and building integrated renewable energy systems. In Francesco Asdrubali and Umberto Desideri, editors, *Handbook of Energy Efficiency in Buildings*, pages 441–595. Butterworth-Heinemann, 2019.
- [8] Hiroyuki Akinaga. **Recent advances and future prospects in energy harvesting technologies.** *Japanese Journal of Applied Physics*, 59(11):110201, 2020.
- [9] Martin Bor, John Edward Vidler, and Utz Roedig. *Lora for the internet of things*. 2016.
- [10] CNN Brasil. **Desmatamento na Amazônia em 2021 é o maior dos últimos 10 anos.** <http://www.cnnbrasil.com.br/nacional/desmatamento-na-amazonia-em-2021-e-o-maior-dos-ultimos-10-anos/>, 2022. Acessado em 29-06-2022.
- [11] Holly Chiang, James Hong, Kevin Kinningham, Laurynas Riliskis, Philip Levis, and Mark Horowitz. **Tethys: Collecting Sensor Data without Infrastructure or Trust.** In *2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pages 249–254, 2018.
- [12] Mundo da Elétrica. **Gerador eólico, como funciona?** <https://www.mundodaeletrica.com.br/gerador-eolico-como-funciona/>, 2022. Acessado em 10-12-2022.

- [13] Agência Universitária de Notícias. **Mata do IB como exemplo de floresta urbana para o mundo**. <http://aun.webhostusp.sti.usp.br/index.php/2021/09/28/a-ideia-e-transformar-a-mata-do-ib-no-exemplo-de-floresta-urbana-para-o-restante-do-mundo-diz-diretor-do-instituto-de-biociencias-da-usp>, 2021. Acessado em 10-07-2022.
- [14] Renwei Ding and Bingbing Xing. **Comparative Research on the Way of Energy Harvesting of the Wireless Sensor Network Nodes**. In *2013 6th International Conference on Intelligent Networks and Intelligent Systems (ICINIS)*, pages 289–291, 2013.
- [15] Electrical Easy. **Basic Construction And Working Of A DC Generator**. <https://www.electricaleasy.com/2022/09/construction-and-working-of-dc-generator.html>, 2022. Acessado em 10-12-2022.
- [16] O Eco. **Relatório mostra que 99.8% dos desmatamentos no Brasil foram ilegais**. <https://oeco.org.br/reportagens/>, 2021. Acessado em 29-06-2022.
- [17] Johan Estrada-López, Amr Abuellil, Zizhen Zeng, and E. Sanchez-Sinencio. Multiple input energy harvesting systems for autonomous iot end-nodes. *Journal of Low Power Electronics and Applications*, 8:6, 03 2018.
- [18] Smita Gaikwad, Rajesh Patil, Ajay Khandare, and Anshuman Rai. **Design WSN node for protection of forest trees against poaching based on ZigBee**. In *2015 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, pages 1–4, 2015.
- [19] J. Gilbert and Farouk Balouchi. **Comparison of energy harvesting system for wireless sensor networks**. *International Journal of Automation and Computing*, 5:334–347, 10 2008.
- [20] David B Golub, Daniel P Julin, Richard F Rashid, Richard P Draves, Randall W Dean, Alessandro Forin, Joseph Barrera, Hideyuki Tokuda, Gerald Malan, and David Bohman. **Microkernel operating system architecture and Mach**. In *In Proceedings of the USENIX Workshop on Micro-Kernels and Other Kernel Architectures*, pages 11–30, 1992.
- [21] Zhang Huanan, Xing Suping, and Wang Jiannan. **Security and application of wireless sensor network**. *Procedia Computer Science*, 183:486–492, 2021. Proceedings of the 10th International Conference of Information and Communication Technology.
- [22] Z. Lei and Jianhua Lu. **Distributed coverage of forest fire border based on WSN**. In *2010 2nd International Conference on Industrial and Information Systems*, volume 1, pages 341–344, 2010.
- [23] Feng Li, Yanbing Yang, Zicheng Chi, Liya Zhao, Yaowen Yang, and Jun Luo. **Trinity: Enabling Self-Sustaining WSNs Indoors with Energy-Free Sensing and Networking**. 2018.
- [24] Dong Ma, Guohao Lan, Mahbub Hassan, Wen Hu, and Sajal K. Das. **Sensing, Computing, and Communications for Energy Harvesting IoTs: A Survey**. *IEEE Communications Surveys Tutorials*, 22(2):1222–1250, 2020.

- [25] Alina-Elena Marcu, George Suciu, Elena Olteanu, Delia Miu, Alexandru Drosu, and Ioana Marcu. **IoT System for Forest Monitoring**. In *2019 42nd International Conference on Telecommunications and Signal Processing (TSP)*, pages 629–632, 2019.
- [26] Young-Seok Noh, Jeong-Il Seo, Won-Jong Choi, Ji-Hwan Kim, Hoang Van Phuoc, Hyun-Sik Kim, and Sang-Gug Lee. **17.6 A Reconfigurable DC-DC Converter for Maximum TEG Energy Harvesting in a Battery-Powered Wireless Sensor Node**. In *2021 IEEE International Solid- State Circuits Conference (ISSCC)*, volume 64, pages 266–268, 2021.
- [27] Jozef Papán, Matós Jurečka, and Jana Púchyová. **WSN for forest monitoring to prevent illegal logging**. In *2012 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 809–812, 2012.
- [28] Portal Solar. **Como Funciona o Painel Solar Fotovoltaico**. <https://www.portalsolar.com.br/como-funciona-o-painel-solar-fotovoltaico.html>, 2022. Acessado em 10-12-2022.
- [29] Mile K Stojčev, Mirko R Kosanović, and Ljubiša R Golubović. **Power management and energy harvesting techniques for wireless sensor nodes**. In *2009 9th International Conference on Telecommunication in Modern Satellite, Cable, and Broadcasting Services*, pages 65–72. IEEE, 2009.
- [30] Huan Zhou, Arie Taal, Spiros Koulouzis, Junchao Wang, Yang Hu, George Suciu, Vlad Poenaru, Cees Laat, and Zhiming Zhao. **Dynamic Real-Time Infrastructure Planning and Deployment for Disaster Early Warning Systems**. pages 644–654, 06 2018.

APÊNDICE A – ATIVIDADES (WBS)



APÊNDICE B – VALOR DOS RESISTORES

A fórmula para o divisor resistivo é dada por

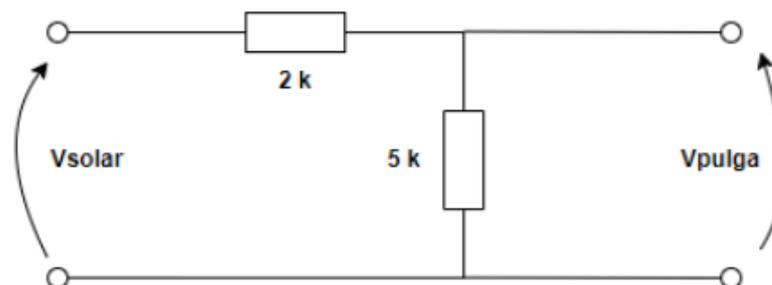
$$V_{Pulga} = V_{Solar} * \frac{R_2}{R_1 + R_2}$$

Como a tensão máxima na entrada pode ser de 7 V, e queremos no até 5 V, e escolhendo $R_2 = 5 \text{ k}\Omega$, temos:

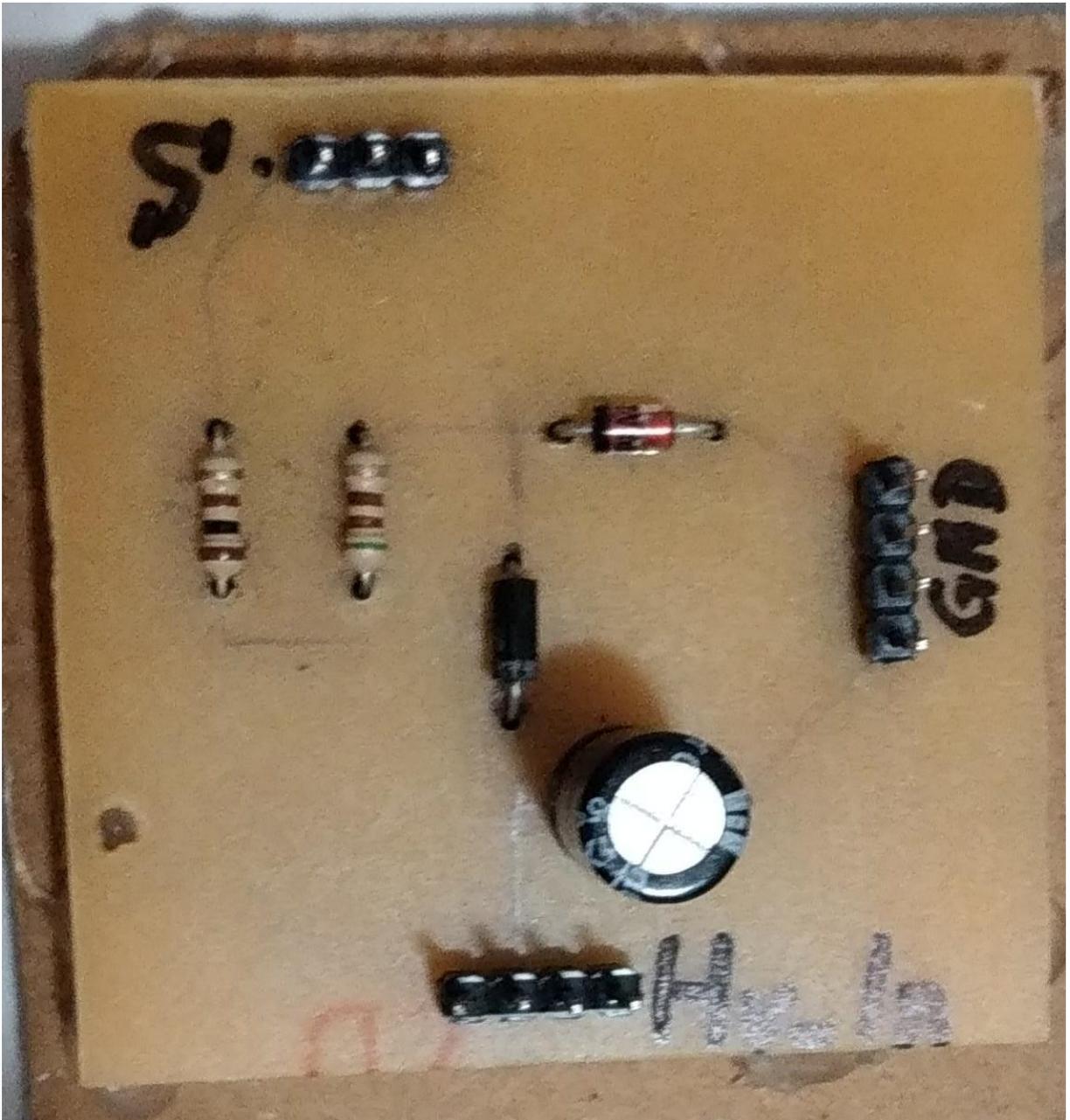
$$5 = 7 * \frac{5k}{R_1 + 5k}$$

$$R_2 = 2 \text{ k}\Omega$$

A seguir está o diagrama do divisor resistivo:



APÊNDICE C – CIRCUITO DE ENERGY HARVESTING MONTADO



APÊNDICE D – CÓDIGO FONTE

D.1 pm.c

```

void nrf52_sys_on(void) {
    NRF_POWER->TASKS_LOWPWR = 1; // Put the NRF52 in sleep mode
    /* Clear Event Register */
    __SEV();
    /* Wait for event */
    __WFE();
    /* Wait for event */
    __WFE();
}

void RTC0_IRQHandler(void)
{
    if (NRF_RTC0->EVENTS_COMPARE[0] == 1)
    {
        NRF_RTC0->EVENTS_COMPARE[0] = 0;
        // Clear the interrupt
        NRF_RTC0->INTENCLR = RTC_INTENCLR_COMPARE0_Msk;
    }
}

void rtc_config(void)
{
    RTC0_IRQHandler();
    NRF_RTC0->TASKS_STOP = 1;
    NRF_RTC0->PRESCALER = 0;
    NRF_RTC0->CC[0] = 10 * 32768;
    NRF_RTC0->INTENSET = RTC_INTENSET_COMPARE0_Msk;
    NRF_RTC0->TASKS_CLEAR = 1;
}

```

```
    NRF_RTC0->TASKS_START = 1;
}
void pm_set(unsigned mode)
{
    switch (mode) {
        case 0:
            //RTC wake up source sleep
            puts("Entering sleep");
            // Initialize the RTC
            rtc_config();
            nrf52_sys_on();
            puts("Exiting sleep");
            pm_set(1);
        break;
        case 1:
            //Ztimer active mode
            puts("Entering active");
            ztimer_sleep(ZTIMER_SEC, 10);
            pm_set(0);
        break;
        case 2:
            //No sleep
        break;
    }
}

void pm_off(void);

void pm_set_lowest(void);
```

D.2 pulga-interface.c

```
int main(void)
{
```

```
puts("Pulga Low Power!");
#ifdef MODULE_ZTIMER
puts("ZTIMER definido!");
#endif
printf("You are running RIOT on a(n) %s board.\n", RIOT_BOARD);
printf("This board features a(n) %s MCU.\n", RIOT_MCU);

printf("This application allows you to test the CPU power management.\n"
       "The available power modes are 0 - %d. Lower-numbered power modes\n"
       "save more power, but may require an event/interrupt to wake up\n"
       "the CPU. Reset the CPU if needed.\n\n\n",
       PM_NUM_MODES - 1);

#ifdef MODULE_PERIPH_GPIO_IRQ && defined(BTNO_PIN)
puts("using BTNO as wake-up source");
gpio_init_int(BTNO_PIN, BTNO_MODE, BTNO_INT_FLANK, btn_cb, NULL);
#endif

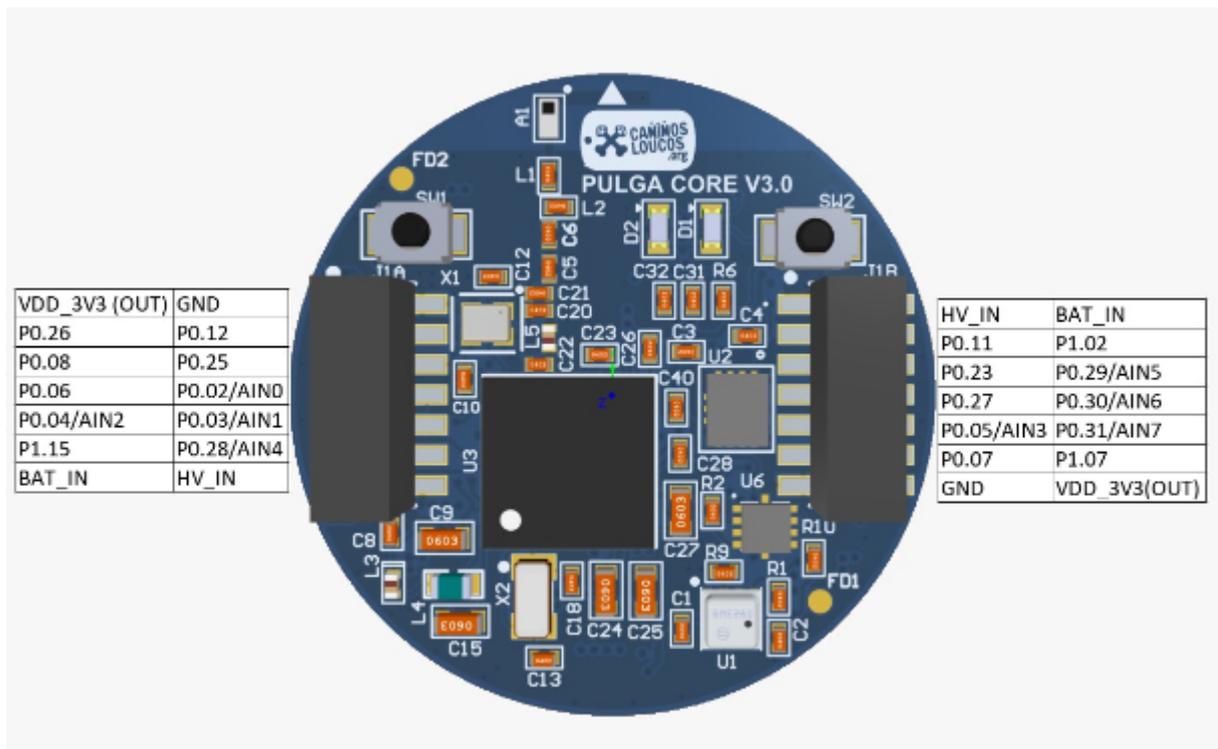
char line_buf[SHELL_DEFAULT_BUFSIZE];
shell_run(NULL, line_buf, SHELL_DEFAULT_BUFSIZE);

return 0;
}
```

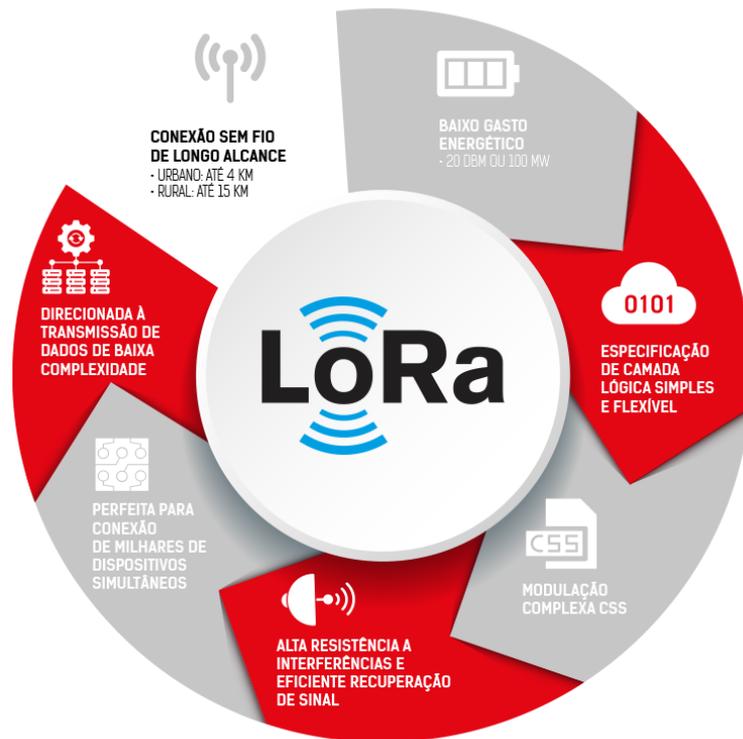
ANEXO A – PULGA - ESPECIFICAÇÕES

CPU	ARM Cortex-M4F
Disk	1024 kB
RAM	256 kB
Wireless	Bluetooth 5.0 Bluetooth Mesh
Segurança	Crypto Acceleration (ECC, AES, SHA)
Periféricos	I2C, I2S, SPI, UART, USART, GPIO Light/UV sensor 9-axis sensor (Gyroscope, Accelerometer, Magnetometer) Environmental sensor (Temperature, Humidity, Pressure)
LED	2 x programáveis (vermelho/laranja)
Botões	2 x programáveis
Alimentação	1~6 com capacidade de colheita
Expansão	24 I/O pins (Placa Principal) 58 I/O pins (Módulo)
Debug	SWD
Sistema Operacional	Mbed™ compatible
Dimensões	25mm diameter (Placa Principal) 20 x 20mm (Módulo)
Peso	aprox. 4g

ANEXO B – PULGA - PINAGEM



ANEXO C – LORA



ANEXO D – PULGA LORA

