UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO

André Luiz Ribeiro Arruda

Desenvolvimento de um dispositivo sensor para monitoramento de ninhos de tartarugas marinhas

Recife

Março de 2024

André Luiz Ribeiro Arruda

Desenvolvimento de um dispositivo sensor para monitoramento de ninhos de tartarugas marinhas

Artigo apresentado ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Universidade Federal Rural de Pernambuco – UFRPE
Departamento de Estatística e Informática
Curso de Bacharelado em Sistemas de Informação

Orientador: Victor Wanderley Costa de Medeiros

Recife
Março de 2024

Dados Internacionais de Catalogação na Publicação
Universidade Federal Rural de Pernambuco
Sistema Integrado de Bibliotecas
Gerada automaticamente, mediante os dados fornecidos pelo(a) autor(a)

A779d Arruda, André Luiz Ribeiro
Desenvolvimento de um dispositivo sensor para monitoramento de ninhos de tartarugas marinhas / André Luiz Ribeiro Arruda. - 2024.
25 f. : il.

Orientador: Victor Wanderley Costa de Medeiros.
Inclui referências.

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal Rural de Pernambuco, Bacharelado em Sistemas da Informação, Recife, 2024.

1. Monitoramento de ninhos de tartarugas marinhas. 2. Sensores IoT no monitoramento de tartarugas marinhas. 3. Preservação de tartarugas marinhas. 4. Dispositivos IoT. I. Medeiros, Victor Wanderley Costa de, orient. II. Título

CDD 004

ANDRÉ LUIZ RIBEIRO ARRUDA

DESENVOLVIMENTO DE UM DISPOSITIVO SENSOR PARA MONITORAMENTO DE NINHOS DE TARTARUGAS MARINHAS

Monografia apresentada ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Aprovada em: 10 de Março de 2024.

BANCA EXAMINADORA

Victor Wanderley Costa de Medeiros (Orientador)
Departamento de Estatística e Informática
Universidade Federal Rural de Pernambuco

Cicero Garrozi
Departamento de Estatística e Informática
Universidade Federal Rural de Pernambuco

Desenvolvimento de um dispositivo sensor para monitoramento de ninhos de tartarugas marinhas

André Luiz Ribeiro Arruda [1], **Victor Wanderley Costa de Medeiros** [2]

[1]Departamento de Estatística e Informática – Universidade Federal Rural de Pernambuco
Rua Dom Manuel de Medeiros, s/n - CEP: 52171-900 – Recife – PE – Brasil

[2]Centro de Informática – Universidade Federal de Pernambuco
Av. Jorn. Aníbal Fernandes, s/n - CEP: 50740-560 – Recife – PE – Brasil

andre.luizarruda@ufrpe.br, victor.wanderley@ufrpe.br

Resumo. *A preservação das tartarugas marinhas é essencial para a ecologia marinha e a biodiversidade global, dada sua relevância na manutenção dos ecossistemas costeiros e marinhos. Atualmente, algumas organizações não governamentais dedicadas a essa causa monitoram ninhos em praias da costa brasileira de forma manual, o que demanda tempo e recursos humanos. Para otimizar esse processo, este estudo propõe a implementação de um dispositivo IoT equipado com sensores para o monitoramento de ninhos de tartarugas marinhas. Utilizando a tecnologia de comunicação de longa distância e baixo consumo energético LoRa, o dispositivo visa aprimorar a eficiência e a precisão do monitoramento, contribuindo para a conservação dessas espécies vulneráveis.*

Abstract. *The preservation of sea turtles is crucial for marine ecology and global biodiversity, as they play important roles in maintaining coastal and marine ecosystems. Currently, some non-governmental organizations dedicated to this cause monitor nests on beaches along the brazilian coast manually, which requires time and human resources. To streamline this process, this study proposes the implementation of an IoT device equipped with sensors for monitoring sea turtle nests. Utilizing Long-Range, Low-Power (LoRa) communication technology, the device aims to enhance monitoring efficiency and accuracy, thereby contributing to the conservation of these vulnerable species.*

1. Introdução

As tartarugas marinhas são animais encantadores, conhecidos por suas majestosas migrações através dos oceanos e seu papel vital nos ecossistemas marinhos. No entanto, apesar de sua importância ecológica e cultural, as populações de tartarugas marinhas têm enfrentado ameaças crescentes, colocando em risco a sobrevivência de várias espécies. Elas podem ser prejudicadas em diversos estágios de sua vida, e de diferentes formas. De acordo com a reportagem de Diana Kurzeja [Kurzeja] apenas 1.000 de cada 10.000 filhotes sobrevivem.

Esses animais lutam pela vida já na fase de incubação, onde os ovos mais abaixo no ninho podem ser esmagados pelos filhotes que estão acima no processo de escavação para chegar à superfície. Os filhotes também podem perder o rumo do mar após sair dos ninhos, ao confundirem as luzes de residências nas proximidades da praia com a luz do sol

(como observado por [Witherington and Martin 2000]), o que consequentemente poderá causar sua morte por atropelamento ou desidratação devido ao período prolongado fora da água. Diante desse cenário preocupante, a conservação desses animais marinhos tornou-se uma questão urgente e complexa, demandando ações efetivas e comprometidas para protegê-los e garantir a perpetuação das espécies.

No período de reprodução, que acontece nos meses mais quentes do ano, as fêmeas adultas migram para as áreas de desova, saindo do mar à noite quando a chance de serem predadas é menor, procurando faixas de areia próximas à costa para enterrarem seus ovos a uma profundidade de cerca de 50 cm. Cada fêmea põe mais de 100 ovos. O período de incubação dura em média 50 dias, de acordo com [Zimmerman et al. 2014]. Após esse período, os ovos eclodem e os filhotes começam a cavar até chegar à superfície, utilizando uma combinação de instintos e dicas naturais, como a temperatura e o som do oceano. Os filhotes saem do ninho geralmente a noite, devido ao resfriamento da areia. Já na superfície, os filhotes serão guiados à água por meio da luz natural. Nesse período ficarão expostos a diversos predadores e perigos.

1.1. Problema e motivação

Com o objetivo de ajudar a diminuir a mortalidade nos ninhos, algumas organizações ambientais tem implantado medidas de proteção em diversas praias do litoral brasileiro. Estas organizações realizam medidas para preservar os locais de desova das tartarugas, protegendo-os do contato com banhistas e predadores, e ainda, implantando medidas que contribuam para a qualidade de vida das tartarugas no período de incubação.

Uma dessas medidas é o monitoramento contínuo dos ninhos, onde após o período de incubação, voluntários passam a verificar diariamente os ninhos, de forma manual, na tentativa de identificar movimentações na areia indicando a eclosão dos ovos. Tal medida contribui para a remoção dos filhotes em tempo hábil, evitando danos aos indivíduos.

O grande problema nesse processo se dá pelo modo como é feito e a baixa quantidade de voluntários disponíveis para realização do trabalho, que atualmente é realizado de forma manual. Surge então a oportunidade de automatizar parte deste processo através do uso de tecnologias como Internet das Coisas (IoT), sensores e comunicação de longa distância, como as redes LoRa.

Com o avanço da tecnologia, os sensores emergiram como componentes essenciais que sustentam uma grande quantidade de aplicações em diversas indústrias, e portanto, se apresentam como alternativa ideal para aplicação em estudos como este. Esses pequenos e sofisticados dispositivos são desenvolvidos para detectar e medir fenômenos físicos, convertendo-os em sinais elétricos que podem ser processados e analisados. Uma das tendências mais notáveis na tecnologia de sensores é o crescimento dos dispositivos de IoT. Esses sensores interconectados são embarcados em objetos e sistemas do cotidiano, permitindo-os coletar e compartilhar dados sem maiores problemas. Outra característica interessante de tais dispositivos é o baixo custo de implementação.

Neste contexto, este trabalho estudo propõe a implementação de um dispositivo sensor IoT equipado com sensores para o monitoramento em tempo real de atividades de ninhos de tartarugas marinhas. O dispositivo conta também com um mecanismo de comunicação baseado em redes LoRaWAN beneficiando-se do baixo consumo energético e do longo alcance inerente a esta tecnologia.

1.2. Objetivos

Este trabalho tem como principal objetivo a implementação de um dispositivo de monitoramento utilizando sensores e dispositivos IoT, com capacidade de comunicação de longo alcance e baixo consumo de energia para realizar o monitoramento dos ninhos de tartarugas marinhas, enviando dados para um gateway através da rede LoRa.

De maneira específica pretende-se:

1. Investigar a literatura científica relacionada à utilização de redes LoRa por dispositivos enterrados no solo.
2. Desenvolver um dispositivo IoT protótipo capaz de captar movimentos em ninhos de tartarugas marinhas e enviar dados via LoRa para um servidor na nuvem.
3. Dimensionar o consumo demandado pelo protótipo criado.
4. Determinar o consumo do protótipo utilizando o microcontrolador e acelerômetro em modo de economia de energia.
5. Realizar testes com o protótipo em ambiente simulado para ajustes de parâmetros.
6. Realizar testes com o protótipo em ambiente real.

2. Referencial Teórico

Abaixo elicitaremos alguns conceitos que serão amplamente explorados e utilizados nas seções seguintes.

2.1. IoT

Diferente da internet convencional utilizada por computadores e dispositivos, a Internet das Coisas (tradução do inglês “Internet of Things” - IoT) é a rede de objetos físicos que são conectados por sensores e software. Enquanto indivíduos usam a internet para acessar sites, os dispositivos conectados à rede IoT a utilizam para trocar informação com outros dispositivos da rede. Com a ajuda de sensores, software e meios de comunicação, como Wifi ou 5G, dispositivos formam uma rede chamada de IoT. Há uma série de benefícios referentes ao uso desses dispositivos, desde usuários tomando conhecimento de atividades que eles estão desenvolvendo até dispositivos sendo capazes de realizar tarefas autônomas.

2.2. LoRa e LoRaWAN

Os dois termos se referem a conceitos parecidos, porém com diferenças consideráveis entre eles. LoRa se refere a uma técnica de modulação de sinal wireless, capaz de transmitir dados a longa distância, porém a uma baixa taxa de bits, ou seja, carregando uma baixa quantidade de dados. As redes LoRa se tornam adequadas ao uso em IoT devido a essas características aliadas ao seu baixo consumo de energia, uma vez que em redes IoT dados de sensores não exigem muita capacidade de transmissão.

A Figura 1 demonstra a capacidade de propagação no solo de algumas das redes mais utilizadas em IoT. No estudo realizado por [Wan et al. 2017], pode-se perceber que a rede LoRa consegue entregar 100% dos pacotes a uma distância de 100m do receptor, com ambos os dispositivos sob o solo. Isso a torna uma boa alternativa à comunicação do dispositivo sensor.

LoRaWAN se refere a um protocolo de redes de área ampla (wide area network) que é construído sobre a técnica de modulação LoRa. Com LoRaWAN é possível criar

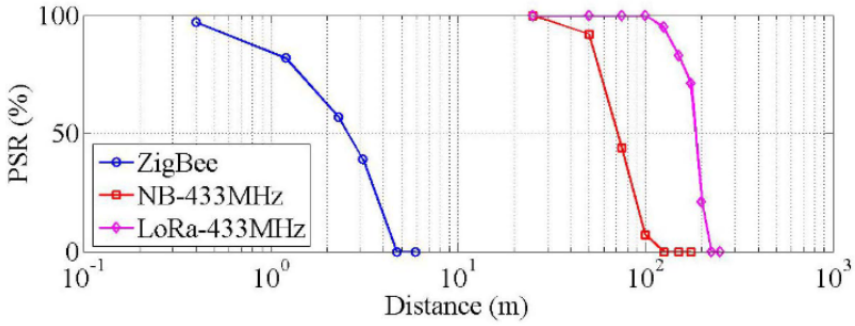

Figura 1. Taxa de Sucesso de Pacotes (PSR) das redes LoRa, ZigBee e Narrow-Band na propagação no solo. Fonte: [Wan et al. 2017].

uma rede sem fio que conecta dispositivos à internet e gerencia a comunicação entre dispositivos de ponta e gateways.

2.3. MQTT

Sigla para Messaging Queuing Telemetry Transport, é um protocolo de mensagens projetado para a troca eficiente de dados entre dispositivos ou aplicações em um ambiente distribuído. A principal característica desse protocolo é o sistema de publicação e subscrição de mensagens, onde usuário envia mensagens para um tópico, e todos os usuários subscritos naquele tópico receberão a mensagem.

2.4. Gateway

No ramo de telecomunicações, um gateway funciona como um nó que conecta duas redes com diferentes protocolos de transmissão. Os gateways funcionam como ponto de entrada e de saída de uma rede, onde todos os dados precisam passar ou se comunicar por eles a fim de serem roteados. Neste projeto o gateway possui um papel fundamental, que é interligar o dispositivo sensor com o servidor da TTN.

2.5. The Things Network (TTN)

É uma rede IoT aberta e descentralizada, construída e mantida por uma comunidade global de desenvolvedores, entusiastas e empresas que contribui com a implantação e operação de gateways, criando aplicações, e promovendo o uso de tecnologias IoT. Sua estrutura descentralizada permite a criação de redes IoT locais que podem servir comunidades ou regiões específicas sem a dependência de infraestrutura central.

No contexto desse trabalho, o gateway se comunica com uma rede criada localmente, pertencente ao laboratório Jualabs, da UFRPE. Criada com o intuito de utilização nas pesquisas do laboratório.

3. Trabalhos Relacionados

Esta Seção apresenta a análise de alguns trabalhos publicados que fizeram uso de sensores IoT ou redes LoRa em diferentes aplicações.

As redes LoRa tem sido bastante exploradas ao longo dos anos como um meio eficiente de transmissão de dados devido ao seu baixo custo e longo alcance. Também

se destacam por permitir a comunicação mesmo com os dispositivos enterrados no solo. No trabalho [Cardell-Oliver et al. 2019] a rede LoRa foi utilizada na comunicação de unidades de transmissão, contendo sensores de solo IoT, enterradas a uma profundidade de 50cm (2). Esses dados foram capturados no período de 10 meses, permitindo-se a criação de um *dataset* com informações relativas ao solo de um campo de trigo na Austrália que serviram para o monitoramento em tempo real das condições do solo, determinantes para o sucesso ou falha das decisões de cultivo e colheita.

O trabalho [Wan et al. 2017] se concentra na realização de testes de propagação das redes LoRa embaixo da terra considerando alguns cenários específicos, como a concentração de água no solo, a profundidade do dispositivo e o tamanho do *payload*. Tal estudo se mostrou bastante interessante devido a baixa quantidade de trabalhos relacionados ao uso de redes sem fio em ambientes subterrâneos. Redes sem fio subterrâneas (WUSN) são suscetíveis a perda de absorção de sinal de rádio, devido à profundidade que estão instaladas.

Jino Ramson [Ramson et al. 2021] em seu trabalho se concentrou em desenvolver um sistema para monitoramento da saúde do solo utilizando dispositivos IoT e implementando uma arquitetura LoRaWAN. No trabalho os autores utilizaram diferentes sensores para monitorar o solo em tempo real. A rede de dispositivos utilizou topologia estrela de estrelas em que várias unidades de monitoramento de solo (SHMU) se comunicam com seu próprio *gateway*, que por sua vez enviam as informações para serem armazenadas em um servidor na internet. O sistema se mostrou adequado para o uso na agricultura devido ao baixo consumo de energia, baixo custo de implementação e possibilidade de comunicação em campos amplos. Também foram avaliados outros protocolos de comunicação como SIGFOX e NB-IoT, que foram descartados devido ao alto custo de implementação, enquanto o protocolo LoRaWAN pode ser implementado gratuitamente.

Thomas Zimmerman em seu trabalho [Zimmerman et al. 2014] descreve a criação de um sistema semelhante empregado no monitoramento de ninhos de tartarugas marinhas utilizando acelerômetros e sensores de temperatura. Em seu artigo mais recente [Clabough et al. 2022] é mostrado como esse sistema é utilizado por meio do uso de dispositivos celulares para o envio das informações a um servidor, que constantemente as transforma em grafos que podem ser lidos e analisados por voluntários com acesso a aplicação.

Figura 2. Unidade de transmissão conectada a diversos sensores de solo. Fonte: [Cardell-Oliver et al. 2019].

4. Materiais e Métodos

A infraestrutura para construção dos dispositivos foi obtida através do laboratório de pesquisa Juá Labs da UFRPE. Abaixo estão descritos os materiais que foram utilizados na construção dos dispositivos, bem como seus respectivos modelos:

4.1. Dispositivo Sensor

Consiste no dispositivo que será instalado nos ninhos, responsável pela captação, processamento e envio dos dados para o *gateway* LoRa. É composto pela placa IoT Dev Kit (Figura 3) que abriga o microcontrolador ESP32 e os sensores de umidade e temperatura.

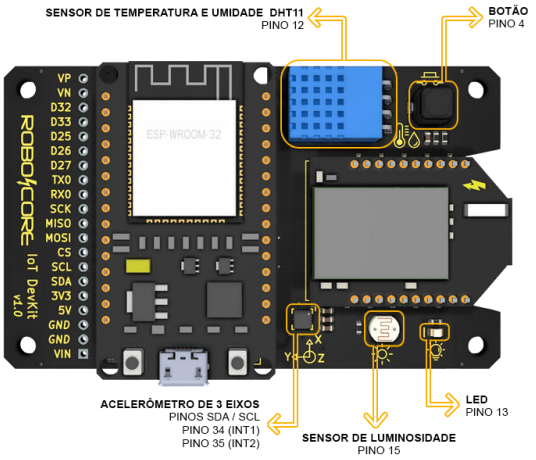

Figura 3. Placa IoT DevKit vendida pela RoboCore. Fonte: (https://s3-sa-east-1.amazonaws.com/robocore-lojavirtual/1255/iot-devkit-pinout.png).

O ideal é que sejam utilizados sensores externos e apropriados para o solo e sejam instalados em uma cápsula com formato idêntico ao dos ovos dentro do ninho, a aproximadamente 50cm de profundidade. O dispositivo sensor deve ser instalado próximo ao

ninho, conectado por meio de cabos e a uma distância segura dos sensores, com o objetivo de não interferir fisicamente na estrutura do ninho.

Este estudo foi realizado com o uso de dispositivo com sensores integrados visando a simplicidade, o qual chamamos de dispositivo sensor. Esse dispositivo é composto por:

- Placa Robocore IoT DevKit v1.0;
- Módulo microcontrolador ESP32 (ESP-WROOM-32) com Bluetooth e WiFi integrados;
- Sensor de temperatura e umidade do ar DHT11;
- Sensor de luminosidade LDR de 5mm;
- Acelerômetro de três eixos MMA8452;
- Módulo LoRaWAN Bee v2 (eLR100-UL-00);

O código desenvolvido neste trabalho e utilizado no dispositivo para detecção de movimento pode ser obtido no repositório https://gitlab.com/jualabs/research/sea-turtle-nest-monitoring/-/tree/main/deep_sleep_with_periodic_motion_counter.

4.1.1. Modos de energia do dispositivo sensor

O ESP32 dispõe de alguns modos de energia visando a eficiência energética do sistema. Os modos de energia permitem que o microcontrolador entre em um estado de economia de energia enquanto não estiver em uso, desligando componentes desnecessários e mantendo em funcionamento apenas componentes fundamentais, a depender do modo.

Modo Ativo O modo ativo é o modo padrão, em que todos os componentes permanecem ativos e, portanto, é o modo que apresenta o maior consumo de energia, conforme pode ser observado na Figura 4.

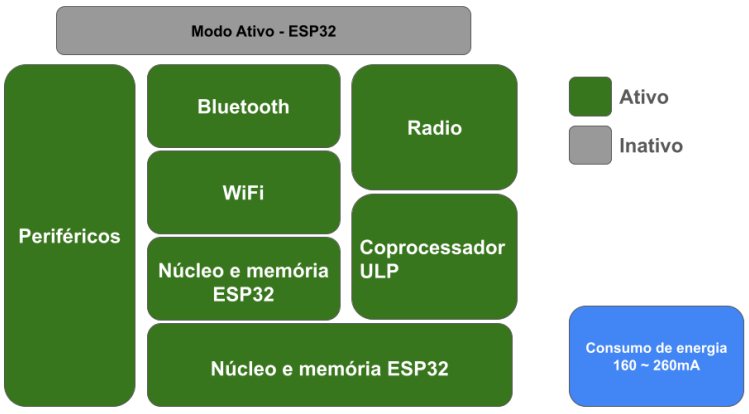

Figura 4. Consumo de energia do modo Ativo. Fonte: Autoria própria.

Modo Deep Sleep No modo Deep Sleep, a maioria dos componentes é desligado (Figura 5), mantendo-se ligados apenas a memória RTC e o coprocessador ULP. O coprocessador ULP é responsável por efetuar a leitura dos sensores e acordar a CPU quando necessário. O modo *deep sleep* é útil em aplicações onde a CPU precisa ser acordada por eventos externos, alarmes (timers), ou uma combinação destes eventos, enquanto mantém o mínimo consumo de energia.

O dispositivo sensor realiza a comunicação com o *gateway* através do protocolo LoRa. Como o envio de dados ocorre em intervalos de tempo determinados, o dispositivo sensor deve ficar em modo *Deep Sleep*, e somente ser acordado (entrando em modo Ativo) em duas situações: quando houver detecção de movimento ou ao disparo do alarme (timer) predeterminado para envio das informações. Essas mudanças de estado são ocasionadas pela geração de interrupções (4.1.2).

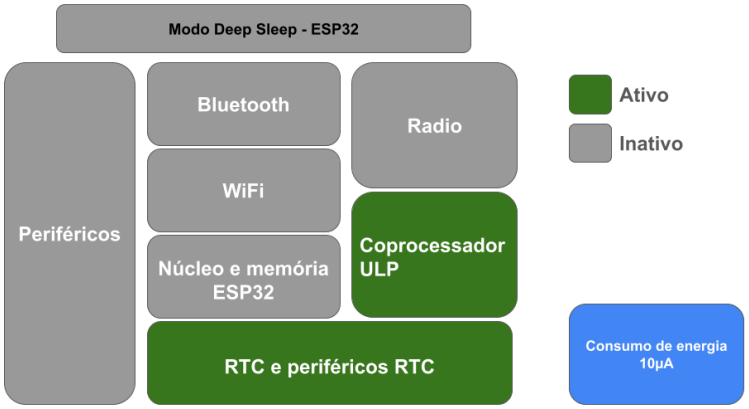

Figura 5. Consumo de energia do modo Deep Sleep. Fonte: Autoria própria.

4.1.2. Interrupções

Interrupções são mecanismos que permitem que o microcontrolador suspenda temporariamente seu caminho de execução atual para lidar com um evento ou condição específica que requer atenção imediata. Interrupções desempenham um papel crucial no gerenciamento de eventos em tempo real e na melhoria da responsividade e eficiência geral do microcontrolador.

4.2. Gateway LoRa

O Gateway LoRa é o dispositivo (Figura 6) responsável por receber as informações vindas do dispositivo sensor. Esse dispositivo recebe dados via LoRa e os envia para a The Things Network via WiFi. O dispositivo possui os seguintes componentes:

- Placa Raspberry Pi 3 Model B+ (Figura 7 inferior);
- Módulo LoRaMESH (Figura 7 superior).

Figura 6. Gateway LoRa. Fonte: Autoria própria.

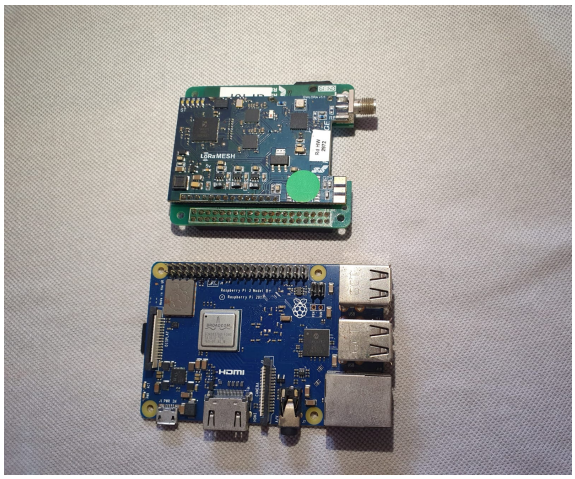

Figura 7. Componentes do Gateway LoRa. Fonte: Autoria própria.

4.3. Ferramentas

4.3.1. The Things Network (TTN)

A The Things Network é uma rede IoT aberta baseada no protocolo LoRaWAN, responsável por receber os dados vindos do Gateway LoRa e armazená-los. É a plataforma responsável pela criação da aplicação, dispositivo e gateway na nuvem. Permitindo que essas entidades se comuniquem com os dispositivos físicos. Cada aplicação na TTN disponibiliza um endpoint MQTT que pode ser acessado para publicação e subscrição em tópicos. Para acessar esse endpoint, o usuário deve criar uma chave de API na plataforma.

4.3.2. Thingsboard

É uma plataforma IoT para coleção, processamento, visualização de dados e gerenciamento de dispositivos. Essa ferramenta foi utilizada como plataforma para visualização de dados no trabalho de outros estudantes relacionado ao monitoramento de ninhos de tartarugas marinhas, permitindo a criação de dashboards interativos. A plataforma é alimentada via integração MQTT com a TTN.

Nome	Versão	Uso
Adafruit Unified Sensor	1.1.14	Comunicação dos sensores integrados
ArduinoJson	6.21.4	Serialização de dados
CayenneLPP	1.3.0	Formatação do payload
DHT sensor library	1.4.6	Sensor de temperatura e umidade do ar
RoboCore - SMW_SX1276M0	1.1.0	Comunicação da placa integrada
SparkFun MMA8452Q Accelerometer	1.4.0	Acelerômetro MMA8452

Tabela 1. Bibliotecas utilizadas

4.3.3. Arduino IDE

A IDE do Arduino, abreviação de Ambiente de Desenvolvimento Integrado, é um aplicativo de software projetado para auxiliar no desenvolvimento de projetos baseados em placas de microcontroladores Arduino. Esta IDE foi escolhida devido a sua simplicidade de uso e do gerenciamento de bibliotecas.

4.4. Bibliotecas e versões utilizadas

Para o correto funcionamento do projeto, é necessário utilizar as bibliotecas em suas versões corretamente. As versões de cada biblioteca utilizada podem ser conferidas na Tabela 1.

4.5. Arquitetura do sistema

Este projeto engloba toda a arquitetura descrita na Figura 8 e está sendo construído em conjunto com outros estudantes, porém, o escopo deste trabalho engloba apenas o uso do Dispositivo Sensor, o Gateway e a TTN, bem como a comunicação entre esses dispositivos.

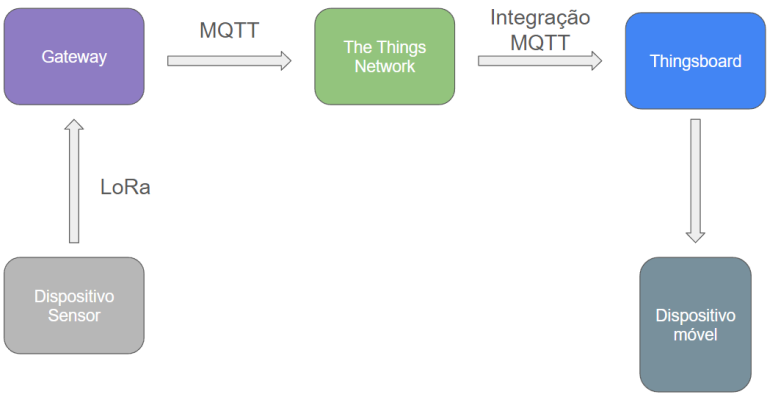

Figura 8. Esquema básico do sistema. Fonte: Autoria própria.

4.6. Consumo do dispositivo

No contexto dos dispositivos IoT, o consumo de energia desempenha um papel crucial, especialmente quando se trata de dispositivos destinados a áreas remotas, como praias. Nesses ambientes, onde a infraestrutura elétrica pode ser limitada ou inexistente,

a eficiência energética se torna uma consideração fundamental para garantir o funcionamento contínuo e confiável dos dispositivos. É fundamental que o dispositivo disponha de energia suficiente para ser utilizado durante todo o período de incubação, que dura entre 45 e 60 dias.

Os testes foram realizados com o dispositivo conectado diretamente à porta USB do computador, uma vez que não foram realizados testes em ambiente real. Em algumas ocasiões o dispositivo foi testado sendo alimentado por uma bateria do tipo Power Bank de 5200mAh da Elgin. Na seção 5.3.3 é feito o dimensionamento da bateria ideal para o dispositivo.

5. Resultados

O monitoramento eficaz dos ninhos de tartarugas marinhas desempenha um papel crucial na conservação dessas espécies ameaçadas. Compreender os padrões de incubação, a temperatura do ninho e outros parâmetros ambientais é essencial para garantir o sucesso reprodutivo e a sobrevivência das tartarugas marinhas em risco.

Neste estudo, apresentamos os resultados do desenvolvimento de um dispositivo sensor inovador projetado especificamente para monitorar ninhos de tartarugas marinhas. Este dispositivo foi concebido para fornecer dados precisos e em tempo real sobre uma variedade de parâmetros ambientais dentro do ninho, incluindo temperatura e umidade, além de uma variável referente à quantidade de movimento detectada nos últimos 60 segundos, importante para a tomada de decisão quanto à abertura do ninho para retirada dos filhotes.

Foram realizados vários testes manuais, onde o movimento foi simulado provocando o balanço do dispositivo sensor, diante da impossibilidade de efetuar testes de campo. Durante os testes o dispositivo foi mantido em modo *Deep Sleep* em intervalos de 60 segundos, passando para o modo ativo após esse tempo para enviar informações ou, após detectar movimento por meio de interrupção gerada por sinal externo em pino de propósito geral (GPIO) captado pelo acelerômetro.

De maneira prática, quando há a detecção de movimento, o acelerômetro gera uma interrupção, que é captada pelo microcontrolador, fazendo-o incrementar o contador de movimentos (variável *motionCounter*) que é salvo na memória RTC. Então na próxima vez em que o dispositivo for ativado para enviar dados para a TTN, esse contador será enviado junto com outros dados, e em seguida será zerado para contabilizar novos eventos de movimento durante a próxima janela de tempo, que no projeto foi definida para 60 segundos. A configuração da interrupção no acelerômetro é mostrada na seção 5.2.

Na Figura 9 são mostradas as mensagens impressas no serial monitor da Arduino IDE após o disparo das interrupções ocasionado pelo movimento, ou seja, causado pela interrupção de sinal externo.

Na Figura 10 são mostradas mensagens de disparo de interrupções, onde a última foi causada por *timer*. Ou seja, após o fim de uma janela de tempo de 60 segundos. Nesse momento os dados referentes aos eventos dos últimos 60 segundos são enviados para o *gateway*, via rede LoRa. E em seguida, o *gateway* os envia para a TTN, como podemos perceber na Figura 11.

Os resultados acima foram obtidos após realização das configurações da seção 5.1.

5.1. Configurações do dispositivo sensor

5.1.1. Configurações iniciais da placa ESP32

Antes de executar quaisquer códigos na placa ESP32, é necessário realizar algumas configurações, como as instalações de *drivers* e dos pacotes de suporte as placas utilizadas. Após esse procedimento, a placa será reconhecida através de uma interface serial RS-232. Após a instalação do *driver*[1], basta conectá-lo por meio do cabo USB ao com-

[1]Tutorial de instalação do driver - https://www.robocore.net/tutoriais/iot-devkit-configuracoes-iniciais

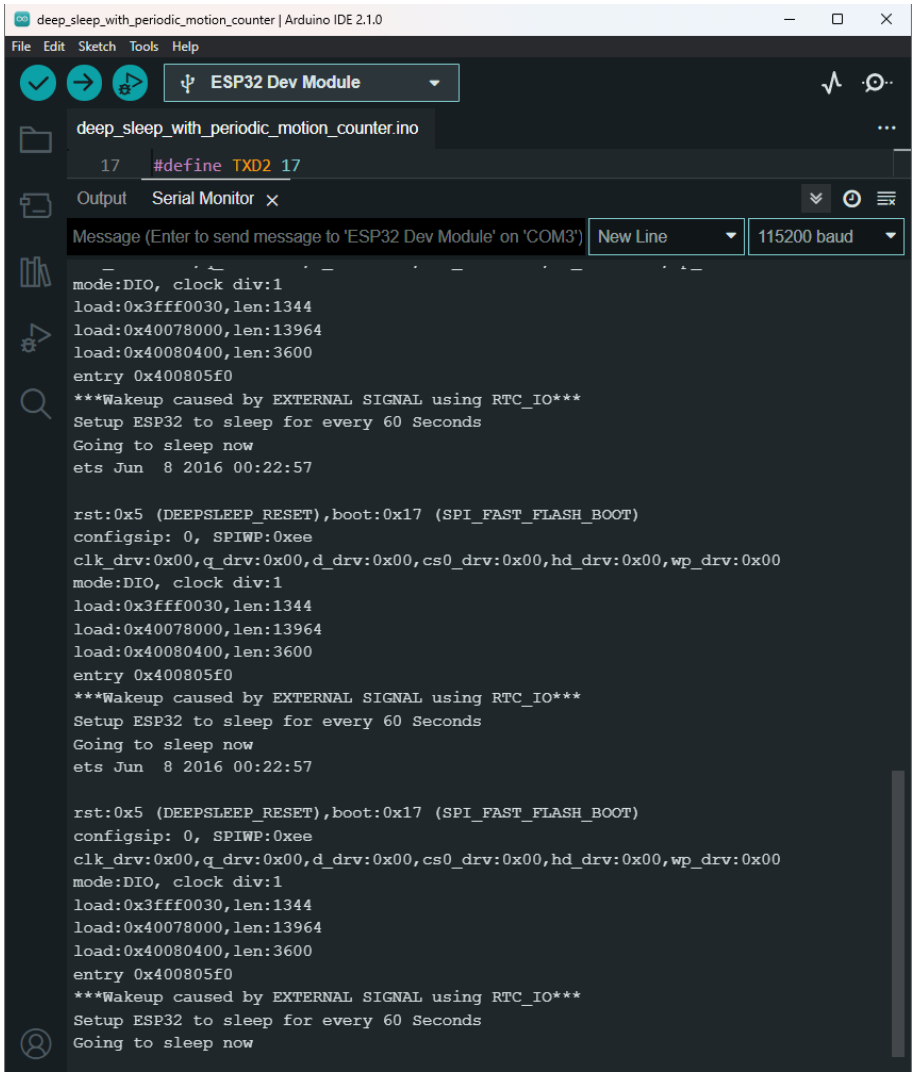

Figura 9. Monitor serial da Arduino IDE após evento de movimento detectado. Fonte: Autoria própria.

putador.

Para a instalação do pacote de placas, os seguintes passos devem ser seguidos:

1. Com a Arduino IDE aberta, clique em “File”;
2. Clique em “Preferences”;
3. Copie o endereço “https://dl.espressif.com/dl/package_esp32_index.json”;
4. Cole o endereço no campo “Additional boards manager URLs”;
5. Clique em “OK”;
6. Clique no menu “Tools” da Arduino IDE;
7. Clique em “Board” e em seguida em “Boards Manager”;
8. No campo para buscar, digite “esp32”;
9. Clique em instalar no pacote “esp32 by Espressif Systems”, conforme a figura 12;
10. Em “Tools” - “Board” - “esp32”, selecione a placa “ESP32 Dev Module”.

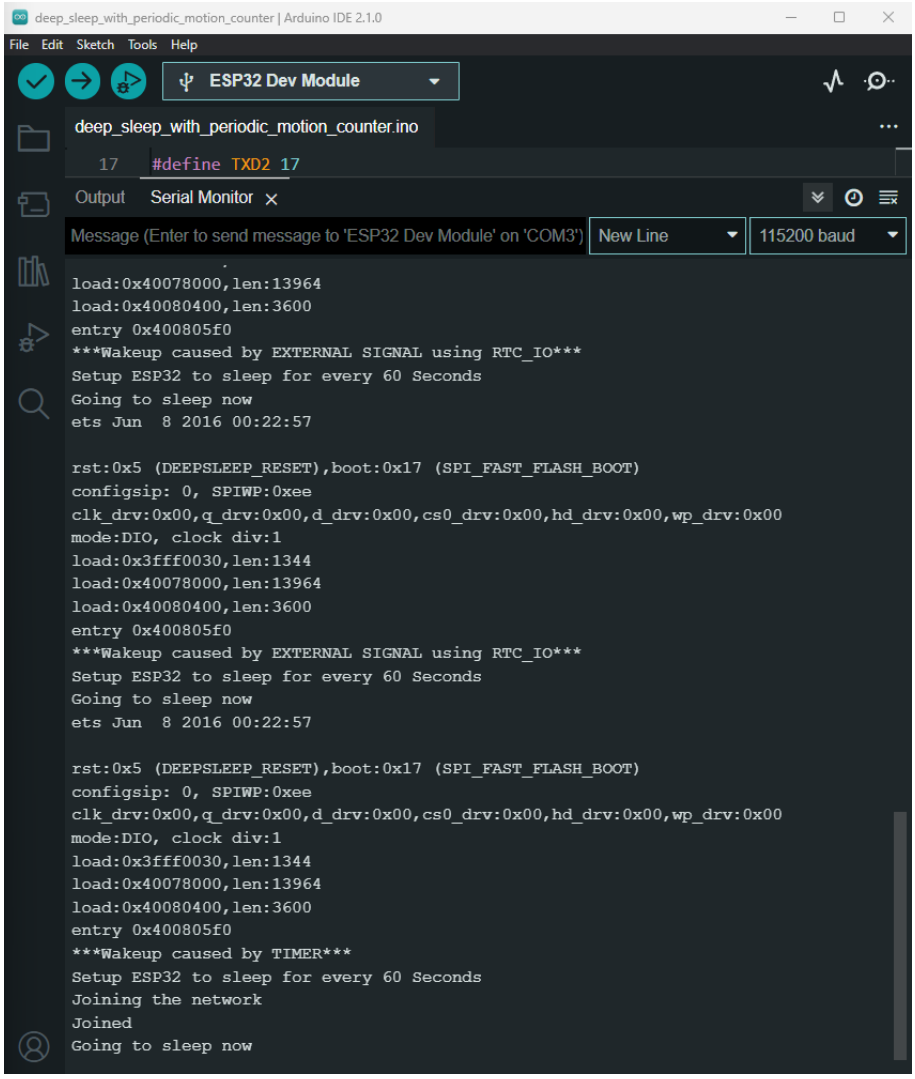

Figura 10. Monitor serial da Arduino IDE após evento causado por timer. Fonte: Autoria própria.

5.1.2. Configuração do módulo LoRa

Para habilitar o módulo LoRa é necessário compilar e executar um código responsável pela configuração adequada do node LoRa[2]. Após a execução, a configuração dos canais utilizados pelo módulo LoRa para comunicação via radio estará concluída.

5.1.3. Configuração do dispositivo na The Things Network

Para utilizar a plataforma TTN, é necessário criar uma conta. A conta utilizada no projeto foi uma conta gratuita. Em seguida deve-se criar uma aplicação e um *gateway* na plataforma.

Para a criação no *gateway* é necessário seguir os seguintes passos:

[2]Código de configuração do node LoRa - https://github.com/victorwcm/RoboCore_SMW-SX1276M0/blob/master/examples/AU915_FSB2_TTN/AU915_FSB2_TTN.ino

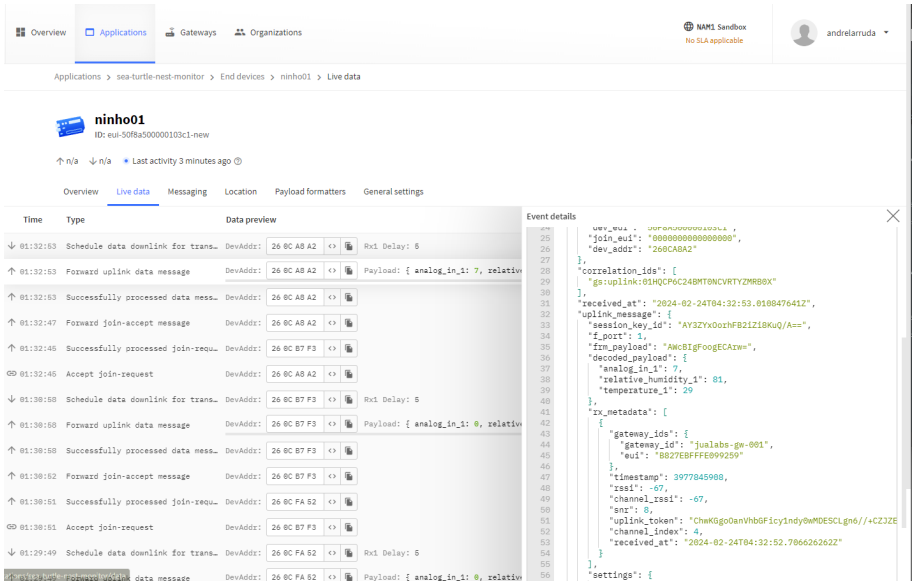

Figura 11. Informações chegando no servidor da TTN após disparo de evento no dispositivo. Fonte: Autoria própria.

1. Acesse https://console.cloud.thethings.network/ e crie uma conta;
2. Faça login na plataforma e escolha o cluster de rede “North America 1” (nam1);
3. Clique em criar Aplicação;
4. Defina um Application ID único e clique em “Create Application”.
5. Para criação do Gateway, clique em “Gateways”;
6. Clique em “Register gateway”;
7. Informe o Gateway EUI do seu dispositivo gateway.

Em seguida, deve-se registrar o dispositivo sensor (end device):

1. Acesse o endereço https://console.cloud.thethings.network/;
2. Faça login na plataforma com o email cadastrado;
3. Selecione o cluster de rede “North America 1”;
4. Clique em “Applications” e selecione o a aplicação criada anteriormente;
5. Na lateral esquerda, selecione “End Devices”;
6. Clique em “Register end device”;
7. Na seção “End device type”, selecione o método de entrada “Enter end device specifics manually”;
8. Em “Frequency plan” selecione o plano “Australia 915-928 MHz, FSB 2 (used by TTN)”;
9. Em “LoRaWAN version” selecione “LoRaWAN Specification 1.0.3”;
10. Em “Regional Parameters version” selecione “RP001 Regional Parameters 1.0.3 revision A”;
11. Definir o JoinEUI e AppEUI como: 0000000000000000;
12. Inserir o DevEUI encontrado na parte superior do módulo LoRa, no dispositivo sensor;
13. Gere um AppKey clicando em “Generate”;
14. Clique em “Register end device”;
15. Na tela seguinte, copie o AppKey (será necessário no código de detecção de movimento).

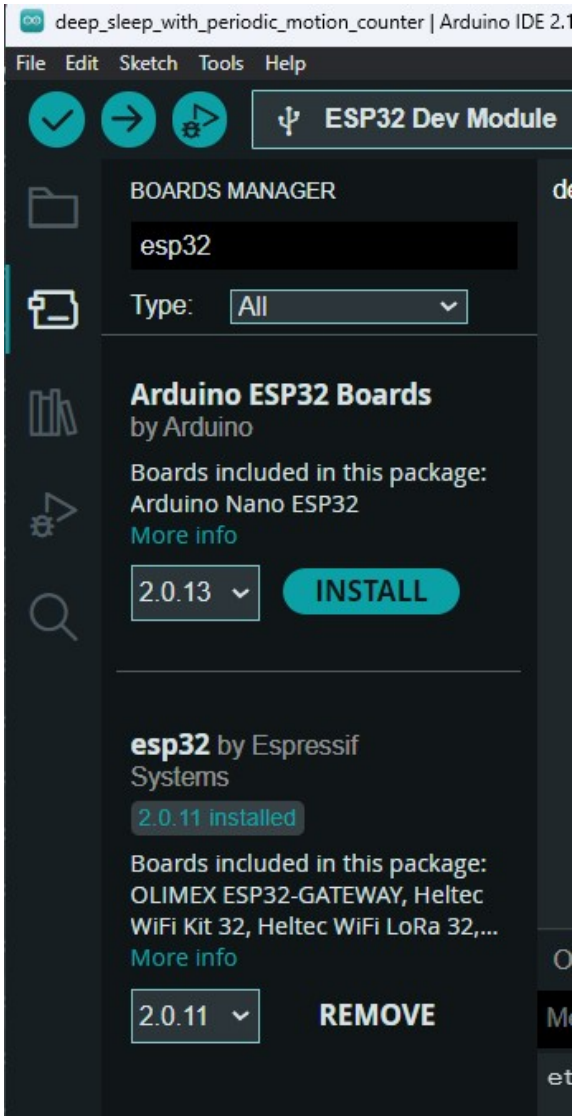

Figura 12. Instalação do pacote da ESP32. Fonte: Autoria própria.

5.2. Configurações da interrupção no acelerômetro

É necessário configurar o acelerômetro para disparar uma interrupção sempre que movimentos forem detectados, alertando a ESP32 que um evento ocorreu. De maneira prática, a detecção de movimento ocorrerá sempre que for identificado que o dispositivo ultrapassou um limite específico de aceleração. Esse evento pode ser ocasionado devido a uma inclinação ou a uma aceleração que excede um valor de um movimento linear, conforme pode ser visto na Figura (13). Uma vez configurada, basta configurar para que a placa acorde ao detectar uma interrupção externa (função *esp_sleep_enable_ext0_wakeup*) no pino 34, que é um dos pinos configurados para interrupção do acelerômetro (vide Figura 3).

Por padrão a biblioteca *SparkFun MMA8452Q Accelerometer* não permite acesso direto ao método de escrita em registradores (*writeRegister*). Portanto, antes de invocá-lo no código, é necessário alterar o seu modificador de acesso de privado para público, no código fonte da biblioteca (arquivo SparkFun_MMA8452Q.h).

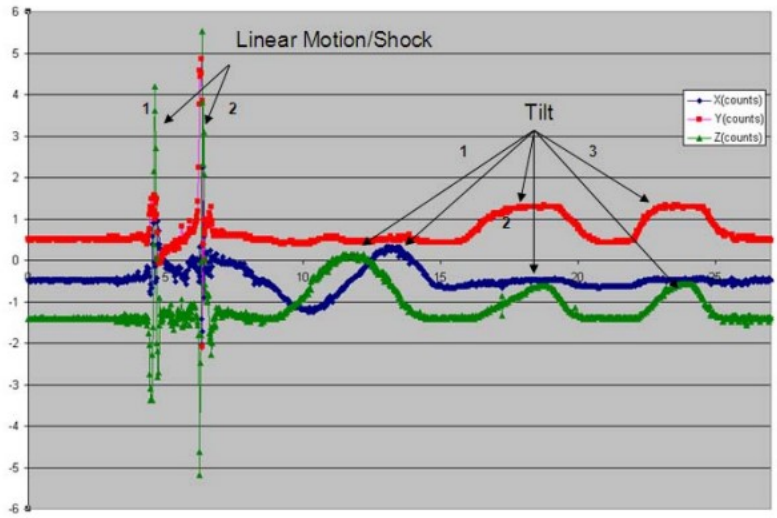

Figura 13. Detecção de movimento por inclinação ou aceleração linear. Fonte: (https://www.nxp.com/docs/en/application-note/AN4070.pdf).

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ELE	OAE	ZEFE	YEFE	XEFE	-	-	-

Tabela 2. Bits do registrador 0x15.

Os valores gravados nos registradores são representados por hexadecimais de 2 dígitos, onde cada dígito representa um conjunto de 4 bits, e cada bit tem um significado específico no registrador em que é inserido.

5.2.1. Registrador *FF_MT_CFG* (0x15)

Registrador responsável pela configuração de movimento ou queda livre. Determina quais eixos serão habilitados para monitoramento com relação a três condições: quais eixos estarão envolvidos; se o evento será uma queda livre linear ou movimento; se o evento detectado deve ser gravado ou não no registrador fonte (0x0C - *INT_SOURCE*). Na Tabela 2 podemos ver o significado de cada bit para este registrador.

Onde:

- **ELE**: Habilitação da flag de evento. Determina o modo como o bit EA - bit 7 do registrador 0x16 (*FF_MT_SRC*) - será limpo; caso o valor desse bit seja '0', o bit EA será limpo apenas depois do delay especificado no registrador *FF_MT_COUNT* (se DBCNTM = '0') ou assim que a condição de alto G que gerou o evento desaparecer (DBCNTM = '1'); caso o valor desse bit seja '1' o bit EA será limpo assim que o registrador *FF_MT_SRC* for lido.
- **OAE**: Define se o evento é detecção de movimento ('1') ou queda livre ('0');
- **ZEFE**: Determina se o eixo Z será levado em consideração na detecção de movimento;
- **YEFE**: Determina se o eixo Y será levado em consideração na detecção de movimento;

ELE	OAE	ZEFE	YEFE	XEFE	-	-	-
0	1	1	1	1	-	-	-

Tabela 3. Valores definidos para o registrador de configuração 0x15.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DBCNTM	THS6	THS5	THS4	THS3	THS2	THS1	THS0

Tabela 4. Bits do registrador 0x17.

- **XEFE**: Determina se o eixo X será levado em consideração na detecção de movimento.

Neste experimento os valores de configuração foram definidos de acordo com a tabela 3. Resultando no valor hexadecimal 0x78. Logo, em termos de código, temos:

```
accel.writeRegister(FF_MT_CFG, 0x78);
```

5.2.2. *FF_MT_THS* (0x17)

Esse registrador é responsável por definir o limite de aceleração para o evento de detecção de movimento. Ou seja, caso a intensidade de aceleração seja maior que o limite definido, houve detecção de movimento. Os bits desse registrador são definidos como segue na Tabela 4. Todos os limites são especificados em valores absolutos. O registrador tem um intervalo de 0 a 127 counts (bit 0 a 6 disponíveis). Quanto menor o limite, mais sensível à detecção de movimento o sensor ficará. Para calcular o valor adequado do limite (em termos de count) deve-se dividir a quantidade de G's por 0,063.

$$limite = \frac{g}{0,063} \tag{1}$$

Nesse registrador, o significado dos bits pode ser entendido da seguinte forma:

- **DBCNTM**: Seleção do modo do contador de debounce. Onde: '0' - Incrementa ou decrementa o debounce; '1': Incrementa ou limpa o contador.
- **THS[6:0]**: Limite do movimento, em 'counts'. Valor entre 0 e 127.

No experimento, o limite definido para esse registrador foi de 2G, e DBCNTM = '1', ou seja, aplicando na fórmula, o limite foi definido em 32 counts. Logo, o valor em hexadecimal pode ser representado como: 0xA0.

```
accel.writeRegister(FF_MT_THS, 0xA0);
```

5.2.3. *FF_MT_COUNT* (0x18)

Esse registrador define a quantidade de amostras de *debounce* para o disparo do evento. Ou seja, este registrador define o número mínimo de contagens de amostras de debounce

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
D7	D6	D5	D4	D3	D2	D1	D0

Tabela 5. Bits do registrador 0x18.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EA	-	ZHE	ZHP	YHE	YHP	XHE	XHP

Tabela 6. Bits do registrador 0x16.

que correspondem continuamente à condição de detecção selecionada pelo usuário para o evento de movimento.

Para esse registrador foi utilizado o valor padrão que é indicado no *datasheet* do controlador: 0x0A.

```
accel.writeRegister(FF_MT_COUNT, 0x0A);
```

5.2.4. *FF_MT_SRC* (0x16)

Este registrador mantém o controle do evento de aceleração que está acionando (ou acionou, no caso do bit ELE no registrador FF_M_CFG estar definido como '1') a *flag* do evento. Em particular, o bit EA é definido como '1' lógico quando a combinação lógica das flags de eventos de aceleração especificadas no registrador FF_MT_CFG é verdadeira.

Seguindo a tabela 6, o significado de cada bit pode ser entendido como segue:

- **EA**: Flag de evento ativo; '0': Nenhum evento detectado; '1': Uma ou mais flags de eventos foram detectadas.
- **ZHE**: Bit de indicação de evento no eixo Z; '0': Nenhum evento no eixo Z foi detectado; '1': Foi detectado evento no eixo Z.
- **ZHP**: Bit que indica polaridade do movimento no eixo Z; '0': evento foi g-positivo no eixo Z; '1': evento g-negativo no eixo Z.
- **YHE**: Bit de indicação de evento no eixo Y; '0': Nenhum evento no eixo Y foi detectado; '1': Foi detectado evento no eixo Y.
- **YHP**: Bit que indica polaridade do movimento no eixo Y; '0': evento foi g-positivo no eixo Y; '1': evento g-negativo no eixo Y.
- **XHE**: Bit de indicação de evento no eixo X; '0': Nenhum evento no eixo X foi detectado; '1': Foi detectado evento no eixo X.
- **XHP**: Bit que indica polaridade do movimento no eixo X; '0': evento foi g-positivo no eixo X; '1': evento g-negativo no eixo X.

Esse registrador não está sendo manipulado no código uma vez que é um registrador de somente leitura. Foi apenas destacado com o intuito de facilitar a compreensão de outros registradores que utilizaram um de seus bits.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ASLP	0	TRANS	LNDPRT	PULSE	FF_MT	0	DRDY

Tabela 7. Bits do registrador 0x2D.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ASLP	0	TRANS	LNDPRT	PULSE	FF_MT	0	DRDY

Tabela 8. Bits do registrador 0x2E.

5.2.5. *CTRL_REG4* (0x2D)

A função desse registrador é habilitar o tipo de evento que está disparando a interrupção, como pode ser visto na tabela 7. Portanto, como estamos interessados em habilitar a interrupção para detecção de movimento, o bit 2 deve ser habilitado.

```
accel.writeRegister(CTRL_REG4, 0x04);
```

5.2.6. *CTRL_REG5* (0x2E)

O registrador 5 é o responsável por configurar a interrupção que foi habilitada no registrador 4. Ou seja, informar se a interrupção deve ser roteada para o pino de interrupção 2 (caso '0') ou para o pino de interrupção 1 (caso '1'). Basta configurar o pino referente à interrupção que foi habilitada no registrador 4, seguindo a Tabela 8.

5.3. Medições de consumo

Apesar do baixo custo de implementação e da eficiência no uso de dispositivos IoT, essa revolução tecnológica vem com uma demanda significativa de energia que não pode ser negligenciada. O consumo de energia no campo dos dispositivos IoT é uma preocupação crucial que requer bastante atenção na aplicação. Uma das principais razões para o aumento do consumo de energia em dispositivos IoT é a necessidade constante de conectividade e transmissão de dados. Embora o dispositivo sensor não permaneça constantemente ligado e opere a maior parte do tempo em modo de economia de energia, esse consumo não pode ser desprezado. Abaixo analisaremos como o dispositivo se comportou durante as fases de funcionamento. Tanto em modo ativo quanto em modo de economia.

Os testes foram realizados utilizando-se uma ferramenta especial de medição de energia, chamado Power Profiler Kit II (PPK2) fabricado pela Nordic Semiconductor®. Durante o teste, o dispositivo sensor foi alimentado pelo PPK2, que por sua vez era alimentado pela porta USB do computador. O esquema de conexão pode ser visto na Figura 14. Os resultados puderam ser acompanhados pelo software que acompanha a ferramenta.

5.3.1. Modo Ativo

Na Figura 15 é possível analisar o que acontece com a energia durante todo o período em que o dispositivo sensor está em modo ativo.

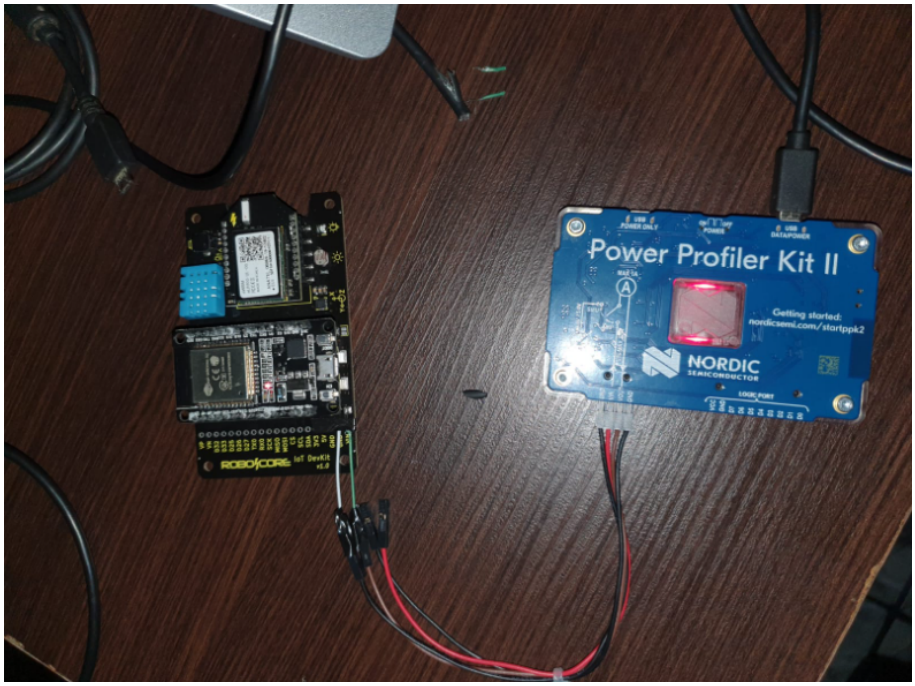

Figura 14. Dispositivo sensor conectado à placa PPK2, na medição de consumo. Fonte: Autoria própria.

O consumo é visivelmente mais alto em relação ao modo Deep Sleep (Figura 16), atingindo uma média de 50,40mA.

5.3.2. Modo Deep Sleep

Na Figura 16 pode-se perceber como a energia se comporta quando o dispositivo é acordado, entrando em modo Ativo, ao disparo da interrupção causada pelo timer.

O momento de maior consumo ocorre ao se utilizar o módulo LoRa para envio das informações ao Gateway. Nessa ocasião o consumo chega a aproximadamente 100mA, enquanto a média de consumo durante o período de 19,24s selecionado é de 50,84mA.

5.3.3. Dimensionamento da bateria

Para estimar o capacidade de carga da bateria para o dispositivo durante 60 dias (1440h), foi considerada a corrente média mostrada na seção 5.3.2 (43,79mA), que então pode ser aplicada na fórmula (2):

$$C = T \times i \tag{2}$$

Onde:
C = Capacidade da bateria (Ah); T = Duração da bateria (h); i = Consumo (A).

Portanto, realizando o cálculo da estimativa para o consumo médio, tem-se:

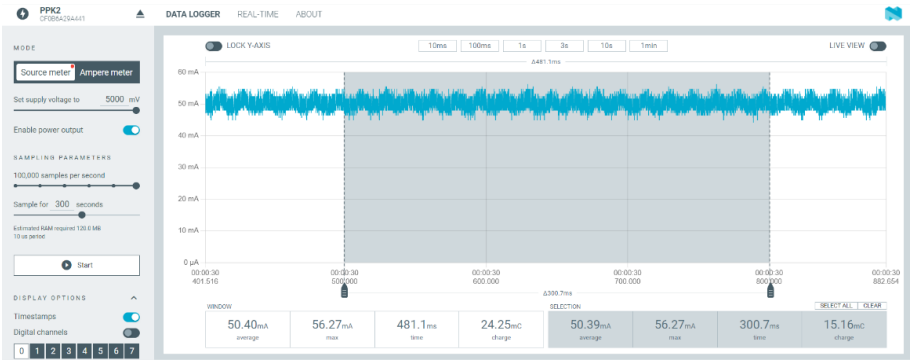

Figura 15. Perfil de consumo do dispositivo em modo Ativo. Fonte: Autoria própria.

Figura 16. Perfil de consumo do dispositivo em modo Deep Sleep. Fonte: Autoria própria.

$$C = 1440 \times 0,04379 \tag{3}$$

$$C \approx 63 \tag{4}$$

Logo, o ideal de bateria para o dispositivo sensor, considerando-se um período de uso de 60 dias, seria de 63.000mAh.

6. Discussões e Trabalhos futuros

Com base nos resultados obtidos neste experimento, pode-se observar que o dispositivo proposto demonstra ser uma ferramenta promissora para o monitoramento de ninhos de tartarugas, em virtude de sua eficiência energética, acessibilidade financeira, facilidade de implementação e capacidade de comunicação em longas distâncias.

A limitação imposta pela impossibilidade de realizar testes abrangentes relacionados à comunicação via tecnologia LoRa restringiu a profundidade da investigação nesse aspecto. Portanto, recomenda-se que estudos futuros explorem mais detalhadamente essa tecnologia, assim como conduzam testes em ambientes reais de monitoramento.

Em relação ao consumo energético, destaca-se a significativa discrepância entre os dois modos operacionais, Ativo e Deep Sleep. Enquanto o modo Deep Sleep mantém um consumo inferior a 20mA na maior parte do tempo, o modo Ativo demanda aproximadamente 50mA. Essa disparidade evidencia a importância da definição adequada do intervalo de tempo (timer), que influencia diretamente na frequência de transmissão de dados. É relevante notar que, embora o período de transmissão de dados seja breve, o uso da rede LoRa se destaca como o principal consumidor de energia, como ilustrado na Figura 16, atingindo um pico de aproximadamente 100mA.

Neste projeto a única estratégia para diminuir o consumo do dispositivo foi a utilização do modo Deep Sleep. Algumas outras medidas podem favorecer a diminuição do consumo, como por exemplo: adotar o uso de placa customizada, eliminando sensores e componentes integrados desnecessários; aumentar o timer, que vai contribuir com a redução da corrente média do dispositivo.

Em trabalhos futuros relacionados ao dispositivo proposto, diversos aspectos e variáveis podem ser explorados, como por exemplo:

1. Desempenho da Comunicação LoRa:
    - Investigação detalhada do desempenho e alcance da comunicação LoRa em diferentes ambientes e condições atmosféricas;
    - Estudo da otimização de parâmetros de comunicação, como taxa de transmissão, largura de banda e potência de transmissão, para maximizar a eficiência e confiabilidade da transmissão de dados.
2. Validação em Ambientes Naturais:
    - Realização de testes e validação do dispositivo em ambientes naturais onde ninhos de tartarugas marinhas são encontrados, considerando variações ambientais e sazonais.
3. Análise do impacto ambiental:
    - Uso de materiais sustentáveis na construção do dispositivo visando a não degradação do ninho.
4. Integração de sensores adicionais:
    - Integração de sensores adicionais para coletar dados complementares sobre o ambiente dos ninhos de tartarugas marinhas.

Referências

[Cardell-Oliver et al. 2019] Cardell-Oliver, R., Hübner, C., Leopold, M., and Beringer, J. (2019). Dataset: Lora underground farm sensor network. In *Proceedings of the 2nd Workshop on Data Acquisition To Analysis*, pages 26–28.

[Clabough et al. 2022] Clabough, E. B., Kaplan, E., Hermeyer, D., Zimmerman, T., Chamberlin, J., and Wantman, S. (2022). The secret life of baby turtles: A novel system to predict hatchling emergence, detect infertile nests, and remotely monitor sea turtle nest events. *Plos one*, 17(10):e0275088.

[Kurzeja] Kurzeja, D. Are sea turtles endangered?

[Ramson et al. 2021] Ramson, S. J., León-Salas, W. D., Brecheisen, Z., Foster, E. J., Johnston, C. T., Schulze, D. G., Filley, T., Rahimi, R., Soto, M. J. C. V., Bolivar, J. A. L., et al. (2021). A self-powered, real-time, lorawan iot-based soil health monitoring system. *IEEE Internet of Things Journal*, 8(11):9278–9293.

[Wan et al. 2017] Wan, X.-f., Yang, Y., Cui, J., and Sardar, M. S. (2017). Lora propagation testing in soil for wireless underground sensor networks. In *2017 Sixth Asia-Pacific Conference on Antennas and Propagation (APCAP)*, pages 1–3. IEEE.

[Witherington and Martin 2000] Witherington, B. E. and Martin, R. E. (2000). Understanding, assessing, and resolving light-pollution problems on sea turtle nesting beaches.

[Zimmerman et al. 2014] Zimmerman, T., Muiznieks, B., Kaplan, E., Wantman, S., Browning, L., and Frey, E. (2014). Design and field testing of a system for remote monitoring of sea turtle nests. In *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services*, pages 637–642.