



Universidade Federal de Pernambuco  
Centro de Tecnologia e Geociências  
Departamento de Eletrônica e Sistemas

Graduação em Engenharia Eletrônica

**Monitoramento da Distância de Navios  
em Relação ao Cais Utilizando o  
iMCP-HT32SX e a Rede Sigfox**

Gabriel Lopes de Albuquerque Dutra

Trabalho de Graduação

Recife

Outubro de 2021

Universidade Federal de Pernambuco  
Centro de Tecnologia e Geociências  
Departamento de Eletrônica e Sistemas

Gabriel Lopes de Albuquerque Dutra

## **Monitoramento da Distância de Navios em Relação ao Cais Utilizando o iMCP-HT32SX e a Rede Sigfox**

*Trabalho apresentado ao Programa de Graduação em Engenharia Eletrônica do Departamento de Eletrônica e Sistemas da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Engenharia Eletrônica.*

Orientador: *João Marcelo Teixeira*

Recife

Outubro de 2021

# Agradecimentos

*A Deus, por ter permitido que eu tivesse saúde e determinação para ultrapassar todos os obstáculos encontrados ao longo desse percurso.*

*À minha família que sempre me incentivou e me deu condições para percorrer esse caminho. Por todo amor e confiança depositados em mim,  
meus sinceros agradecimentos.*

*Ao professor João Marcelo, por ter sido meu orientador e ter desempenhado tal função com dedicação e amizade.*

*Aos amigos Thiago, Daniel e Paulo, que contribuíram com o desenvolvimento do projeto que resultou nesse trabalho.*

*Aos meus colegas de curso, com quem convivi intensamente durante os últimos anos, pelo companheirismo e pela troca de experiências que me permitiram crescer não só como profissional, mas também como pessoa.*

# Resumo

Dentro da operação portuária, uma das atividades que mais demandam controle e acompanhamento é o processo de carga e descarga de um navio. A segurança na operação de embarque e/ou desembarque de cargas nos navios é um elemento crucial nas atividades portuárias, e para este fim existem uma série de normativas e regulamentos que determinam como esses procedimentos devem ser realizados. Um dos principais procedimentos é a amarração das embarcações aos berços de atracação. Uma amarração incorreta pode resultar em um afastamento excessivo ou em uma aproximação excessiva do navio em relação ao cais, devido a fatores naturais como ventos, ondas e correntes marítimas. Por exemplo, caso um navio se afaste muito do cais durante a operação de carga e descarga, podem ocorrer rompimentos de amarras, que podem chicotear estruturas ou pessoas próximas à embarcação. O rompimento de amarras também traz prejuízos à estabilidade do navio, fazendo com que este possa inclinar caso haja algum desequilíbrio de cargas sobre a embarcação. Em um caso como esse, o navio pode ser conduzido de encontro a outras embarcações que estão em seu entorno, ocasionando acidentes, que podem gerar sérios danos às estruturas de um porto e resultar na perda de vidas. Já a aproximação excessiva do navio em relação à estrutura física do terminal marítimo deteriora mais rapidamente as anteparas de colisão chamadas de defensas, que servem como um dissipador de energia, fazendo com que o berço de atracação não sofra tanto com colisões diretas do navio. Visando o monitoramento do estado de atracação dos navios, esse trabalho tem por objetivo propor uma solução utilizando sensores de distância fixados ao cais conectados ao iMCP-HT32SX, que processará as informações continuamente referentes ao distanciamento do navio em relação ao cais e enviará notificações de alerta via rede Sigfox às equipes de operação.

**Palavras-chave:** *IoT*, processamento de dados, monitoramento de terminais marítimos, segurança operacional em portos.

# Abstract

In port operations, one of the activities that most demand control and monitoring is the process of loading and unloading a ship. Safety in loading and/or unloading cargo on ships is a crucial element in port activities, and for this purpose there are a series of rules and regulations that determine how these procedures must be carried out. One of the main procedures is the mooring of vessels to berths. Incorrect mooring can result in the ship moving too far or too close to the pier, due to natural factors such as winds, waves and tidal currents. For example, if a ship strays too far from the pier during the loading and unloading operation, breakage of moorings can occur, which can whip structures or people close to the vessel. The breakage of moorings also harms the ship's stability, causing the ship to tilt in the event of any unbalance of loads on the vessel. In this situation, the ship can be driven into other vessels that are nearby, causing accidents, which can cause serious damage to the structures of a port and result in deaths. The excessive approach of the ship in relation to the physical structure of the maritime terminal, on the other hand, deteriorates more quickly the collision bulkheads called fenders, which serve as an energy dissipator, making the mooring cradles not suffer so much from direct collisions from the ship. Aiming at monitoring the berthing of ships, this work aims to implement a prototype using distance sensors attached to the pier connected to the iMCP-HT32SX, which will continuously process information regarding the distance of the ship from the pier and send alert notifications via the Sigfox network to the operating teams.

**Keywords:** IoT, data processing, monitoring of maritime terminals, operational security in ports.

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Contextualização e Motivação	1
1.2	Objetivos	2
1.3	Organização do Texto	2
<b>2</b>	<b>Fundamentação Teórica</b>	<b>5</b>
2.1	IoT	5
2.2	Sistemas Embarcados	5
2.3	Microcontrolador	6
2.3.1	Comunicação Serial	7
2.3.2	Comunicação $I^2C$	7
2.3.3	Comunicação Via Rádio Frequência	7
2.3.4	<i>Firmware</i>	8
<b>3</b>	<b>Metodologia</b>	<b>9</b>
3.1	Ferramentas Utilizadas	10
3.1.1	Componentes do Hardware	10
3.1.1.1	HT Micron iMCP-HT32SX	10
3.1.1.2	ESP32	11
3.1.1.3	Sensor Ultrassônico JSN-SR04T	12
3.1.1.4	Sensor Laser VL53l0/1x v2	13
3.1.1.5	Antena 2.4Ghz 3dBi	14
3.2	STM Cube IDE	14
3.3	Rede Sigfox	15

3.4	TagoIO	16
<b>4</b>	<b>Implementação</b>	<b>17</b>
4.1	Aquisição de dados	18
4.2	Processamento de dados	20
4.3	Visualização de dados	20
<b>5</b>	<b>Resultados</b>	<b>23</b>
<b>6</b>	<b>Conclusão</b>	<b>27</b>

# Lista de Figuras

3.1	Fluxograma do projeto.	9
3.2	HT Micron iMCP-HT32SX.	11
3.3	ESP32.	11
3.4	Módulo sensor ultrassônico JSN-SR04T.	12
3.5	Sensor Laser VL5310/1x v2.	13
3.6	Antena 2.4Ghz 3dBi.	14
4.1	Ilustração da embarcação atracada no cais.	17
4.2	Arquitetura do sistema.	19
4.3	Mensagens recebidas no <i>Back-end</i> Sigfox.	21
4.4	Mensagens recebidas no TagoIO.	22
5.1	Protótipo do sistema.	24
5.2	Posicionamento do protótipo às margens da piscina.	25
5.3	Testes do protótipo com o navio em escala.	26
6.1	Células de carga e seus respectivos preços.	27



# **Lista de Tabelas**

6.1	Componentes necessários para o protótipo e seus respectivos custos.	28
-----	---	----

## Lista de Acrônimos

<b>RF</b>	rádio frequência . . . . .	7
<b>D</b>	distância entre o navio e o cais . . . . .	9
<b>d</b>	distância entre a defesa e o cais . . . . .	9
<b>IoT</b>	Internet das Coisas . . . . .	5
<b>MCU</b>	microcontrolador . . . . .	6
<b>CI</b>	circuito integrado . . . . .	6
<b>SOC</b>	<i>System on a Chip</i> . . . . .	6
<b>SW</b>	<i>software</i> . . . . .	5
<b>HW</b>	<i>hardware</i> . . . . .	7
<b>FW</b>	<i>firmware</i> . . . . .	8
<b>SCL</b>	<i>Serial Clock</i> . . . . .	7
<b>SDA</b>	<i>Serial Data</i> . . . . .	7
<b>SE</b>	sistema embarcado . . . . .	5
<b>I<sup>2</sup>C</b>	<i>Inter-Integrated Circuit</i> . . . . .	7
<b>SiP</b>	<i>System in Package</i> . . . . .	10

## CAPÍTULO 1

# Introdução

### 1.1 Contextualização e Motivação

Durante as operações de carga e descarga, é imprescindível que o navio permaneça atracado no cais sem grandes variações de movimento. Devido a isso, é fundamental que a atracação dos navios seja bem feita. Uma atracação é feita utilizando amarras, que são cordas ou cabos de aço que prendem o navio ao cais. Essas cordas são fixadas no cais em uma estrutura chamada de cabeços.

Como essas amarrações são realizadas por colaboradores da equipe de operação dos portos, podem acontecer falhas humanas no processo, como uma amarração mal feita, ou até mesmo falhas estruturais, como por exemplo uma amarra desgastada se romper. Falhas desse tipo podem fazer com que o navio passe a oscilar muito à margem do cais, devido à ação dos ventos e da correnteza, podendo causar acidentes dentro do porto, seja se chocando com outras embarcações que estão operando em seu entorno, ou ocasionar algum acidente durante a operação de carga e descarga feitas por maquinários pesados que retiram e colocam cargas no navio.

Hoje o levantamento de que o navio continua bem amarrado ao berço de atracação é feito de maneira manual, onde um tripulante do navio atracado fica responsável pela checagem das amarras, que servem de garantia para que o navio esteja imóvel. O sistema desenvolvido nesse trabalho visa tornar esse processo automático e por assim dizer mais seguro, de forma a se ter a garantia de que o navio se encontra em uma zona de segurança em relação ao cais, munindo a equipe de operações portuárias com informações para que possam tomar as devidas providências para prevenir acidentes.

O Brasil hoje conta com 175 instalações portuárias, onde cada porto possui uma certa quantidade de berços de atracação. A quantidade de berços por porto varia, devido ao tamanho do

porto e a sua finalidade. Por exemplo, no porto de Santos, um dos portos mais movimentados do mundo e o mais movimentado do Brasil, existem 65 berços de atracação, ao passo que o porto de Suape conta com apenas 5 terminais marítimos em atividade. Esses números trazem uma noção da dimensão de aplicabilidade do protótipo produzido neste trabalho[1][2].

Esse projeto foi realizado com o apoio do Laboratório de Pesquisa e Desenvolvimento de Sistemas Embarcados (E.S.T.U.F.A), do Centro de Informática da Universidade Federal de Pernambuco (*Cin - UFPE*) através de mentoria e acompanhamento da professora Edna Barros, e o auxílio e orientação do professor João Marcelo Teixeira.

## **1.2 Objetivos**

Visando o monitoramento do estado de atracação dos navios, esse trabalho visa implementar um protótipo utilizando sensores de distância fixados ao cais, conectados ao iMCP-HT32SX, um circuito integrado projetado para fornecer uma solução de conectividade pronta para uso para aplicações em *IoT*, que processará as informações continuamente, referentes ao distanciamento do navio em relação ao cais e o distanciamento das defensas em relação ao cais, que são dispositivos que asseguram a proteção adequada entre navio e a estrutura de atracação, e enviará notificações de alerta via rede Sigfox às equipes de operação, caso haja um distanciamento excessivo, seja de aproximação ou de afastamento, durante as atividades de carga e descarga. Dessa forma pode-se garantir que, caso a amarração de um navio não esteja dentro do esperado e o navio passe a oscilar demais, rapidamente os responsáveis sejam notificados e acionados para que o problema seja sanado de forma imediata.

## **1.3 Organização do Texto**

O presente trabalho é dividido em 6 capítulos. Ao final, estão listadas as referências ordenadas em citações durante o texto.

**Capítulo 2 - Fundamentação Teórica.** Contém a fundamentação teórica com detalhes de conceitos utilizados para o entendimento e composição do projeto.

**Capítulo 3 - Metodologia.** Apresenta uma descrição das ferramentas e programas utilizados para a execução deste trabalho, os microcontroladores e sensores que configuram o protótipo, além das ferramentas de recepção e visualização dos dados adquiridos.

**Capítulo 4 - Implementação.** Descreve os processos implementados para a realização desse projeto, com foco na arquitetura da solução proposta e fluxo do processamento de dados.

**Capítulo 5 - Resultados.** Descreve a análise dos resultados obtidos a partir do sistema desenvolvido, mostrando um teste realizado em escala.

**Capítulo 6 - Conclusão.** Expõe as considerações finais referentes ao trabalho desenvolvido, além de propor futuros trabalhos.

## Fundamentação Teórica

### 2.1 IoT

A Internet das Coisas (IoT) descreve a rede de “objetos físicos” incorporados a sensores, *software* (SW) e outras tecnologias com o objetivo de conectar e trocar dados com outros dispositivos e sistemas pela internet[3]. Esses dispositivos variam de objetos domésticos comuns a ferramentas industriais sofisticadas. Com mais de 7 bilhões de dispositivos IoT conectados hoje, os especialistas esperam que esse número cresça para 10 bilhões em 2020 e 22 bilhões em 2025. Nos últimos anos, IoT se tornou uma das tecnologias mais importantes do século XXI. Agora que podemos conectar objetos do cotidiano - eletrodomésticos, carros, termostatos, básculas eletrônicas - à Internet por meio de dispositivos incorporados, é possível uma comunicação perfeita entre pessoas, processos e outras coisas.

Por meio da computação de baixo custo, nuvem, *big data*, análise avançada e tecnologias móveis, coisas físicas podem compartilhar e coletar dados com o mínimo de intervenção humana. Nesse mundo hiperconectado, os sistemas digitais podem gravar, monitorar e ajustar cada interação entre itens conectados. O mundo físico encontra o mundo digital - e eles cooperam entre si[4].

### 2.2 Sistemas Embarcados

Um sistema embarcado (SE) é um sistema microprocessado no qual o computador é completamente encapsulado ou dedicado ao dispositivo ou sistema que ele controla[5]. Diferentemente de computadores de propósito geral, como o computador pessoal, um SE realiza um conjunto de tarefas predefinidas, geralmente com requisitos específicos. Já que o sistema é

dedicado a tarefas específicas, através de engenharia pode-se otimizar o projeto reduzindo tamanho, recursos computacionais e custo do produto[6][7]. Em geral tais sistemas não podem ter sua funcionalidade alterada durante o uso. Caso queira-se modificar o propósito é necessário reprogramar todo o sistema.

## 2.3 Microcontrolador

O microcontrolador (MCU) é um pequeno computador *System on a Chip* (SOC) num único circuito integrado (CI) o qual contém um núcleo de processador, memória e periféricos programáveis de entrada e saída. A memória de programação pode ser RAM, NOR flash ou PROM a qual, muitas vezes, é incluída no chip. Os MCU's são concebidos para aplicações embarcadas, em contraste com os microprocessadores utilizados em computadores pessoais ou outras aplicações de uso geral.

Os MCU's são usados em produtos e dispositivos automatizados, como os sistemas de controle de automóvel, dispositivos médicos implantáveis, controles remotos, máquinas de escritório, eletrodomésticos, ferramentas elétricas, brinquedos e outros sistemas embarcados. Ao reduzir o tamanho e o custo em comparação a um projeto que usa um dispositivo microprocessado, microcontroladores tornam-se econômicos para controlar digitalmente dispositivos e processos. MCU's de sinal misto são comuns, integrando componentes analógicos necessários para controlar sistemas eletrônicos não digitais.

O seu consumo de energia é relativamente baixo, normalmente, na casa dos miliwatts e possui habilidade para entrar em modo de espera (Sleep ou Wait) aguardando por uma interrupção ou evento externo, como, por exemplo, o acionamento de uma tecla, ou um sinal que chega via uma interface de dados. O consumo destes MCU's em modo de espera pode chegar na casa dos nanowatts, tornando-os ideais para aplicações onde a exigência de baixo consumo de energia é um fator decisivo para o sucesso do projeto[8].

### 2.3.1 Comunicação Serial

Comunicação serial é o processo de enviar dados transmitindo um bit de cada vez, sequencialmente, em um canal de comunicação ou barramento. É diferente da comunicação paralela, em que todos os bits de cada símbolo são enviados juntos. A comunicação serial é usada em toda comunicação de longo alcance e na maioria das redes de computadores, onde o custo de cabos e as dificuldades de sincronização tornam a comunicação paralela impraticável. Para curtas distâncias, barramentos em série estão se tornando cada vez mais comuns devido ao ponto em que as desvantagens dos barramentos paralelos, densidade de interconexão, superam as suas vantagens de simplicidade[9].

### 2.3.2 Comunicação $I^2C$

O protocolo *Inter-Integrated Circuit* ( $I^2C$ ) descreve o funcionamento de um barramento de comunicação serial que utiliza apenas dois fios. Inventado pela Philips no início da década de 90, este protocolo é muito utilizado para conectar periféricos de baixa velocidade a placas-mãe, MCU's e afins. Tanto a unidade de controle quanto os periféricos devem possuir implementação e suporte  $I^2C$ , seja via *hardware* (HW) no próprio SOC, seja utilizando CI's externos ou até mesmo via SW, onde o funcionamento do protocolo é emulado bit a bit. O barramento  $I^2C$  é composto de dois fios, *Serial Data* (SDA) e *Serial Clock* (SCL), e alimentação (VDD), tipicamente de 3,3V ou 5V. A linha SCL é responsável pelo clock do barramento, e a linha SDA pela transmissão de dados[10].

### 2.3.3 Comunicação Via Rádio Frequência

Comunicação via rádio frequência (RF) é um recurso tecnológico das telecomunicações utilizado para propiciar comunicação por intermédio da transcepção de informações previamente codificadas em sinal eletromagnético que se propaga através do espaço. Para que a comunicação aconteça entre dois pontos são necessários 3 elementos básicos da comunicação: O transmissor, o meio de transmissão e o receptor. O transmissor converte sinais sonoros, analógicos



ou digitais em ondas eletromagnéticas, enviando-os para o espaço através de uma antena transmissora, para serem recebidos por um receptor. O receptor é responsável pela decodificação dos sinais eletromagnéticos recebidos do espaço, captados pela antena, transformando-os em ondas sonoras, sinais digitais e/ou analógicos. Basicamente existem três modos de transmissão: Simplex, quando a comunicação é unidirecional no meio da transmissão, Semi-Duplex, quando a transmissão é bidirecional, porém é executada alternadamente em cada sentido, ou Duplex, quando a comunicação é bidirecional e simultânea. Acerca dos modos de transmissão fundamentalmente existem dois: O Direcional, que privilegia um destinatário em detrimento dos outros e o Omnidirecional, que distribui o sinal para o maior número de usuários[11].

#### **2.3.4 *Firmware***

O *firmware* (FW) é uma classe específica de SW que fornece controle de baixo nível para o HW específico do dispositivo. O FW é armazenado permanentemente num CI de memória de HW, como uma ROM, PROM, EPROM ou ainda EEPROM e memória flash, no momento da fabricação do componente. Muitos aparelhos simples possuem FW, entre eles: controles-remotos, calculadora de mão, instrumentos científicos e robôs. Outros aparelhos mais complexos como celulares, câmeras digitais, sintetizadores, entre outros, também possuem um FW para a execução de suas tarefas[12][13][14].

Não há fronteiras rígidas entre FW e SW, pois ambos são termos descritivos bastante livres. Contudo, o termo FW foi originalmente inventado para contrastar com SW de alto nível que poderia ser alterado sem a troca de um componente de HW, e o FW está normalmente envolvido com operações muito básicas de baixo nível sem as quais um dispositivo seria completamente não funcional.

## CAPÍTULO 3

# Metodologia

A metodologia do desenvolvimento do protótipo é baseada em 5 fundamentos. Inicialmente, realiza-se a **Aquisição de Dados**, que consiste na aquisição das distâncias lidas pelos sensores, referentes à distância entre o navio e o cais (D) e à distância entre a defesa e o cais (d). Em seguida realiza-se a **Eliminação dos Outliers**, que envolve a eliminação de possíveis valores obtidos através de erros de leitura dos sensores. Logo após, realiza-se o **Processamento dos Dados**, que consiste em calcular uma média das distâncias aferidas pelos sensores. Em seguida podem ser realizados **Envios de Alerta via Rede Sigfox**, caso a média dos valores das distâncias tenha saído de limiares pré-definidos. Por fim, efetua-se a **Visualização dos Dados**, em um *Dashboard* utilizando o site do TagoIO[15].

A Figura 3.1 demonstra o fluxograma desse projeto.

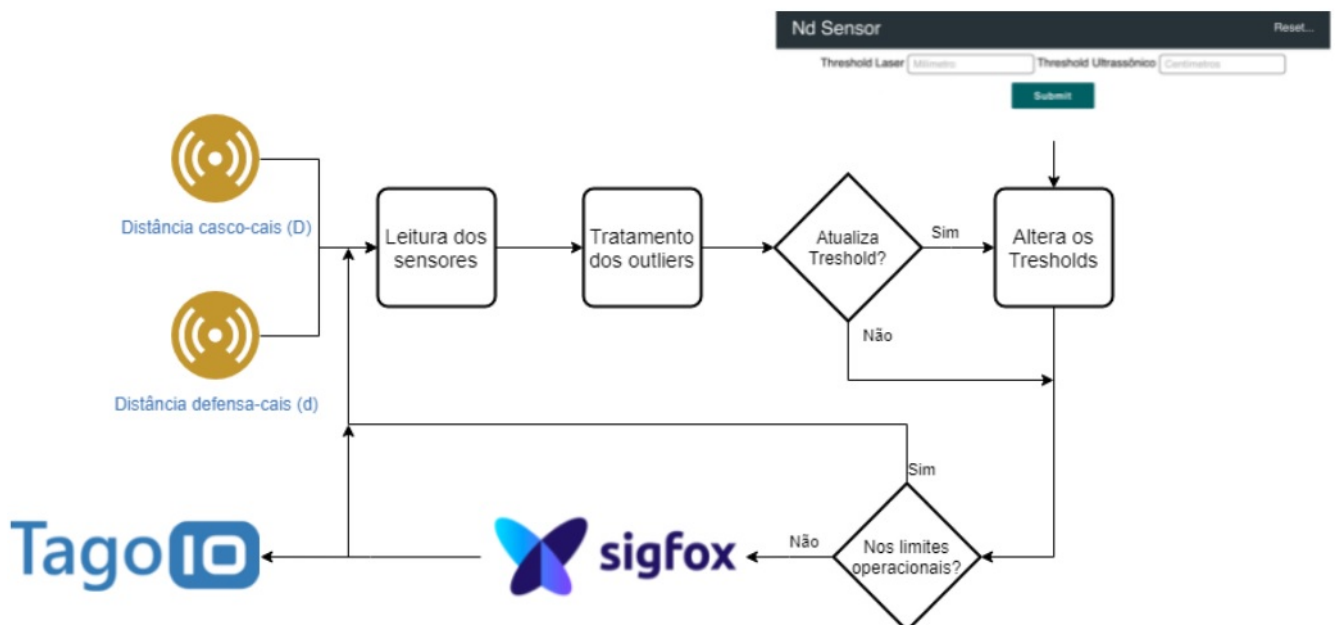


Figura 3.1 Fluxograma do projeto.

## 3.1 Ferramentas Utilizadas

Esta seção descreve as ferramentas utilizadas para a realização deste trabalho. São apresentados todos os componentes de HW, a IDE (*Integrated Development Environment*) utilizada para o desenvolvimento do FW e a aplicação web utilizada como *front-end* para visualização dos *dashboards*.

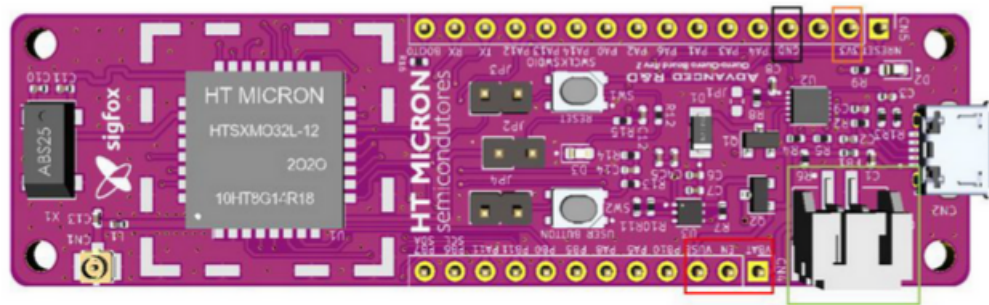
### 3.1.1 Componentes do Hardware

Utilizou-se dois MCU's, um sensor laser, um sensor ultrassônico, e uma antena para transmissão dos dados coletados. O FW foi desenvolvido no STM Cube IDE[16]. Utilizou-se o site tago.io para visualizar os dashboards, que foi integrado junto a rede Sigfox para o recebimento dos dados. Todos esses componentes citados serão descritos em detalhes a seguir.

#### 3.1.1.1 HT Micron iMCP-HT32SX

O iMCP-HT32SX é um *System in Package* (SiP) IoT projetado e fabricado no Brasil, o chip utiliza o protocolo Sigfox e tecnologia Monarch, que permitem a comunicação em todos os países com rede Sigfox e mudança automática de região. O iMCP-HT32SX é um CI multicomponente e foi projetado para fornecer uma solução de conectividade pronta para uso para aplicativos IoT usando a rede Sigfox. Ele fornece comunicações de uplink (transmissão) e downlink (recebimento), e suas pequenas dimensões (13mm x 13mm), alto desempenho e baixo consumo de energia têm como alvo melhorar a experiência e minimizar o tempo de projeto para os times de desenvolvedores de produtos de IoT. O SiP combina um MCU ARM Cortex M0+ 32 bits, um transceiver de baixa potência STM S2-LP e um amplificador de potência, agregando todas as vantagens, integração e conveniência da mais avançada tecnologia de encapsulamento de semicondutores em um único chip. É o primeiro chip a ser certificado como Sigfox Monarch, ou seja, ele conecta-se à rede Sigfox de forma automática em qualquer país do mundo que ofereça acesso a redes Sigfox. Para aplicações simples como rastreamento, a solução final precisará apenas de uma antena e de uma bateria conectadas ao chip. Para apli-

cações mais complexas, pode ser utilizada a memória interna do chip e os GPIOs disponíveis para conectar sensores ou outros dispositivos[17]. Na figura 3.2 pode-se visualizar a placa de desenvolvimento, com o iMCP-HT32SX integrado, utilizada nesse trabalho.



**Figura 3.2** HT Micron iMCP-HT32SX.

### 3.1.1.2 ESP32

O módulo ESP32 possui um módulo de alta performance para aplicações envolvendo wifi, contando com um baixíssimo consumo de energia. Com 4 MB de memória flash, o ESP32 permite criar variadas aplicações para projetos de IoT, acesso remoto, webserver e dataloggers, entre outros. Na placa utilizada tem-se o chip ESP32 com antena embutida, uma interface usb-serial e um regulador de tensão para 3,3V[18]. Na figura 3.3 tem-se a imagem de uma ESP32.



**Figura 3.3** ESP32.

### 3.1.1.3 Sensor Ultrassônico JSN-SR04T

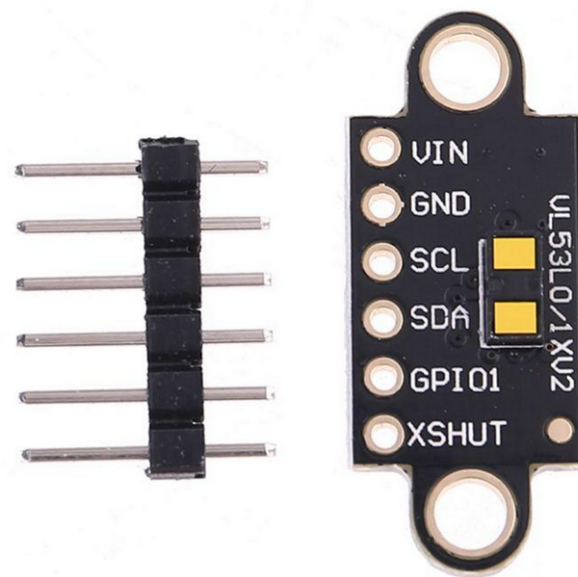
O Sensor Ultrassônico de Distância JSN-SR04T foi desenvolvido para aperfeiçoar projetos de robótica e microeletrônica, mostrando-se ideal para calcular a distância com precisão de objetos entre 20 centímetros a 6 metros. Seu principal diferencial é sua ampla e eficiente resistência à umidade, sendo principalmente utilizado em ambientes úmidos, permitindo manter ampla distância do MCU já que possui um fio com 2,5 metros de comprimento. De forma a facilitar a ligação entre o Sensor e o MCU, o sensor é acompanhado por um módulo especialmente desenvolvido para atuar em conjunto com MCU's, possuindo 4 portas de conexão: 5V (VCC), Trig (RX), Echo (TX) e GND. O funcionamento de sensor é simples, ele emite sinais ultrassônicos que refletem no objeto a ser atingido e retornam ao sensor, precisando a distância do alvo, tomando o tempo de trânsito do sinal, podendo ser utilizado para detecção de objetos, controle de tráfego, segurança, etc[19]. A figura 3.4 ilustra o módulo do sensor ultrassônico utilizado neste projeto.



**Figura 3.4** Módulo sensor ultrassônico JSN-SR04T.

#### 3.1.1.4 Sensor Laser VL5310/1x v2

O VL5310 possui comunicação  $I^2C$  e é usado para medir com precisão quanto tempo leva para que os pulsos de luz infravermelha emitidos por um laser alcancem o objeto mais próximo e sejam refletidos de volta para um detector. Essa medida de tempo permite determinar com precisão a distância absoluta para um alvo sem que a refletância do objeto influencie muito na medição. O sensor pode reportar distâncias de até 2 metros, com resolução de 1 mm, mas seu alcance efetivo e precisão dependem muito das condições do ambiente e das características do alvo, como refletância e tamanho. A precisão do sensor é especificada para variar de 3% na melhor das hipóteses a  $\pm 10\%$  em condições menos ideais. A figura 3.5 ilustra o módulo do sensor laser utilizado neste projeto[20].



**Figura 3.5** Sensor Laser VL5310/1x v2.

### 3.1.1.5 Antena 2.4Ghz 3dBi

A Antena 2.4Ghz 3dBi pode ser utilizada em modems, placas de rede e dispositivos que possuam conector no padrão SMA. Possui adaptador para conexão em módulos wifi e pode ser utilizada nas mais diversas aplicações que necessitem de uma antena externa para aumentar a distância de atuação do módulo. A figura 3.6 ilustra a antena utilizada neste projeto.



**Figura 3.6** Antena 2.4Ghz 3dBi.

## 3.2 STM Cube IDE

STM32 Cube IDE é uma plataforma avançada de desenvolvimento C/C++ com recursos de configuração periférica, geração de código, compilação de código e depuração para microcontroladores e microprocessadores STM32. Essa IDE é baseada na estrutura ECLIPSE/CDT e na cadeia de ferramentas GCC para o desenvolvimento, e no GDB para depuração. Ela permite a integração de centenas de plugins existentes que completam os recursos do ECLIPSE IDE.

O STM32 Cube IDE integra todas as funcionalidades do STM32 Cube MX para oferecer uma experiência total do uso da ferramenta e economizar tempo de instalação e desenvolvimento. Após a seleção de um STM32 MCU ou MPU vazio ou de um microcontrolador ou microprocessador pré-configurado a partir da seleção de uma placa, o projeto é criado e o có-

digo de inicialização é gerado. A qualquer momento durante o desenvolvimento, o usuário pode retornar à inicialização e configuração dos periféricos e regenerar o código de inicialização sem afetar o código do usuário. O STM32 Cube IDE inclui analisadores de compilação e empilhamento que fornecem ao usuário informações úteis sobre o status do projeto e os requisitos de memória. O STM32 Cube IDE também inclui recursos de depuração avançados, incluindo visualizações dos registros principais da CPU, memórias e registros periféricos, além de um analisador de falhas.

### **3.3 Rede Sigfox**

Sigfox é uma tecnologia de banda ultra-estreita com infraestrutura parecida com as redes de operadoras do celular. Ela conta com torres, antenas e servidores de rede necessários para que uma aplicação possa enviar os dados para a nuvem. A principal vantagem da Sigfox é a conexão confiável entre dispositivos a longa distância, mesmo com ruídos e interferências. Isso porque, na hora de se comunicar, os dispositivos conectados através da rede Sigfox contam com um plano B e plano C, ou seja, são enviados duas réplicas da mensagem em diferentes frequências e tempo. Esse “backup de mensagem” aumenta a tolerância a interferências e permite que os dados cheguem onde precisam por diferentes caminhos. A rede Sigfox tem restrições com relação à frequência de envio de dados, o tamanho das mensagens que podem ser enviadas e a área de cobertura necessária. Como o tamanho da rede permite tráfego de no máximo 12 bytes por pulso, o protocolo não terá sucesso em aplicações que necessitem de grande tráfego de dados, como por exemplo monitoramentos em tempo real. Por outro lado, para aplicações onde seja necessário apenas envio de mensagens como um alerta ou envio de mensagens de forma esporádica, a rede Sigfox é uma ótima escolha[21].



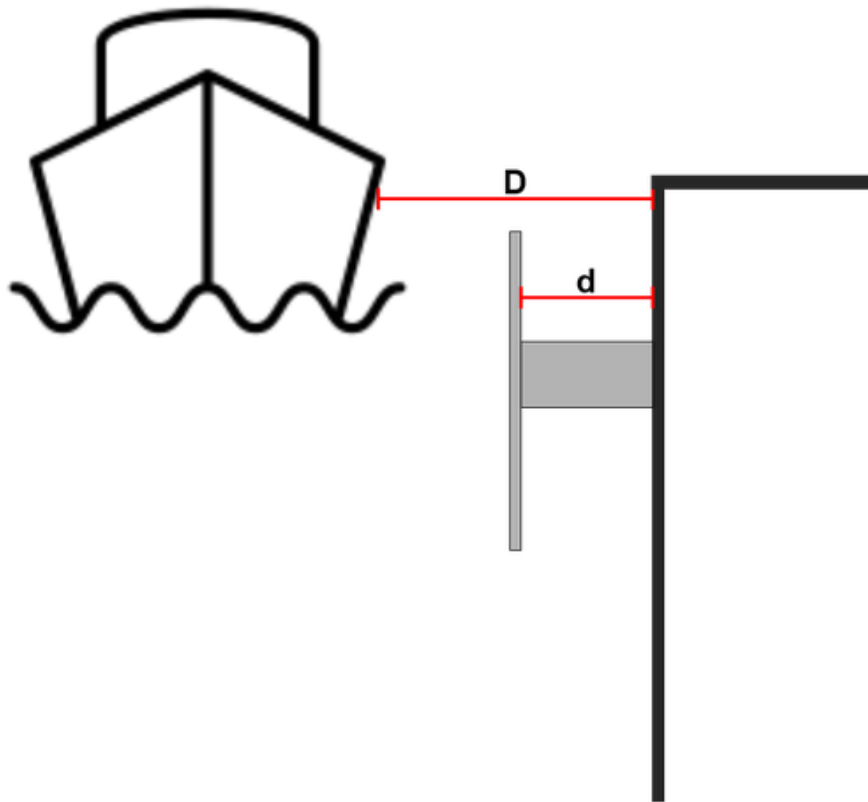
### 3.4 TagoIO

TagoIO é uma plataforma web 100% na cloud e de alto nível para monitoramento de ambientes via dispositivos IoT conectados à sua rede. A Tago hoje possui dois diferenciais: a velocidade com a qual pode ser implementada uma solução de IoT no mercado, e também o fato de ser uma ferramenta poderosa que oferece diversas funcionalidades necessárias para tal implementação. Por meio da plataforma, o usuário pode observar os dados coletados em um *dashboard* prático para construir suas aplicações. Tendo um painel simples e dinâmico, é possível compartilhar e rastrear o uso da aplicação, além de criar níveis de acesso para diversos usuários, definindo o que cada um poderá visualizar e editar. A plataforma é indicada para quem precisa de uma solução rápida, de configurações simples e pronta para começar seu sistema de monitoramento[22].

## CAPÍTULO 4

# Implementação

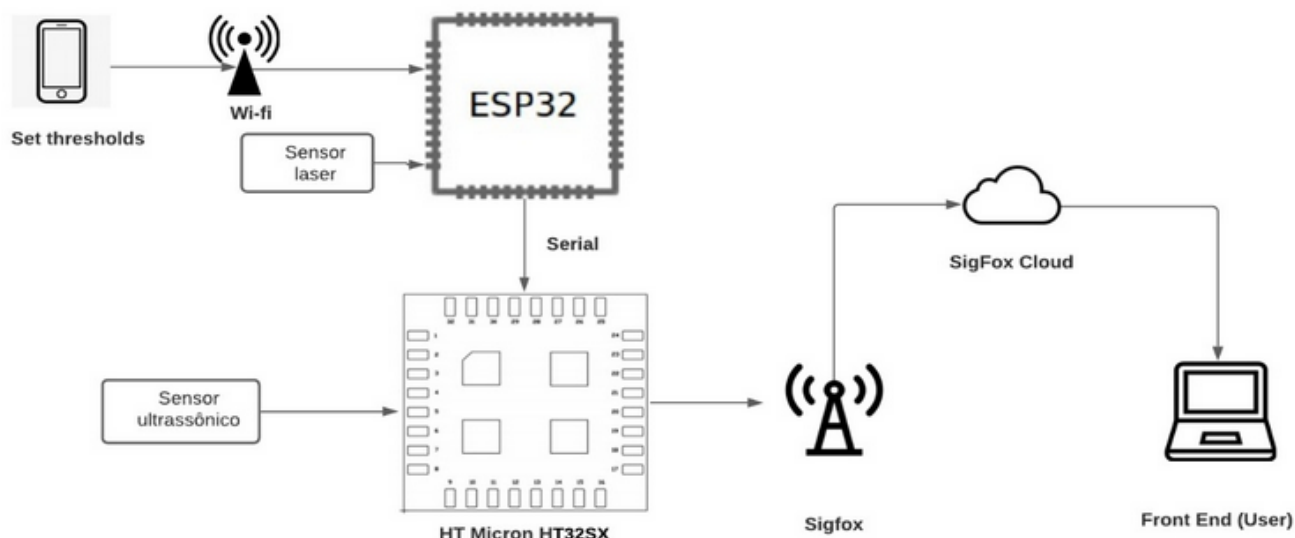
Nesta seção, mostra-se a implementação em HW do protótipo de monitoramento dos berços de atracação. É mostrado como são realizados a aquisição, o processamento e a visualização dos dados, mostrando como cada elemento de HW citado anteriormente contribui com o sistema. Para facilitar a visualização do posicionamento do navio em um terminal marítimo, a figura 4.1 ilustra a posição da embarcação em relação à defesa e em relação ao cais, deixando claro as distâncias que serão medidas pelo protótipo.



**Figura 4.1** Ilustração da embarcação atracada no cais.

## 4.1 Aquisição de dados

Utilizou-se o sensor ultrassônico e o sensor a laser para medir as distâncias  $D$  e  $d$ , respectivamente. O sensor ultrassônico mede a distância  $D$  devido ao fato que esse sensor é capaz de medir distâncias de até 6 metros, o que é mais que suficiente para detectar se o navio está se afastando demais do cais. Já o sensor laser é utilizado para medir a distância  $d$  pois a sua resolução é na ordem de milímetros. Sabe-se que a defesa é utilizada como uma mola para frear o movimento do navio caso este se dirija na direção do cais, e essas defensas variam muito pouco a sua distância em relação ao cais, visto que a deformação é muito pequena mesmo com um choque muito grande. O fato do sensor laser possuir uma resolução na ordem de milímetros faz com que essa pequena variação de compressão possa ser reconhecida pelo sensor. O fato do sensor laser medir distâncias de no máximo 2 metros não é um problema, tendo em vista que o anteparo da defesa localiza-se a uma distância fixa de menos de 2 metros em relação ao cais. Dessa forma, através desse sistema fixado às margens do cais, é possível identificar o quanto a defesa está sendo comprimida ou o quanto o navio está se afastando do cais, e a partir dessa informação enviar um alerta às autoridades portuárias competentes para tomar as medidas cabíveis de segurança. Na figura 4.2 é possível visualizar um esquemático de toda a arquitetura do protótipo.



**Figura 4.2** Arquitetura do sistema.


Nota-se que no diagrama da arquitetura visto na figura 4.2 o sensor laser é utilizado na ESP32, ao passo que o sensor ultrassônico é conectado ao iMCP-HT32SX. Isso acontece devido a uma dificuldade que foi enfrentada ao tentar realizar a comunicação  $I^2C$  do sensor laser com a placa da HT Micron. Dada essa dificuldade, resolveu-se utilizar o sensor laser com a ESP32 pela facilidade de implementação do protocolo  $I^2C$  com essa placa, e aproveitou-se o uso da ESP32 para utilizá-la no modo AP (Access Point), de maneira que os valores de limites fossem passados através de uma conexão com um smartphone. A distância medida pelo sensor laser e lida pela ESP32 é enviada para o iMCP-HT32SX via comunicação serial, e essa placa fica responsável por fazer o processamento dos dados e verificar se os valores encontram-se dentro dos intervalos de limites definidos pelo usuário, como mostrado anteriormente no fluxograma da figura 3.1.

























## 4.2 Processamento de dados

O processamento dos dados na placa da HT Micron é realizado de maneira bastante simples. Após receber os valores da ESP32 que foram lidos pelo sensor laser e receber também as medidas obtidas pelo sensor ultrassônico, o iMCP-HT32SX faz uma média de todas as distâncias medidas em certo intervalo de tempo, por cada sensor, no intuito de amenizar possíveis erros de leitura que fogem completamente da realidade, o que é chamado de eliminação de *outliers*. Essas médias ficam sendo calculadas a medida que esses valores são recebidos, e esses valores são comparados com os valores de intervalo de limiar definido pelo usuário. Caso a média das distâncias medidas por cada sensor não pertença a esse intervalo, é sinal que o navio está se aproximando ou se afastando demais do cais, o que faz com que uma mensagem de notificação seja enviada às autoridades via Sigfox, e dessa maneira seja recebida na nuvem e mostrada em um *dashboard* no browser através da plataforma do TagoIO.

## 4.3 Visualização de dados

Os dados enviados podem ser visualizados diretamente no sistema de *back-end* da rede Sigfox. Como a placa da HT Micron já possui o protocolo da rede Sigfox, é necessário apenas criar um login no site e depois utilizar o número de referência da placa para estabelecer a comunicação. Dessa maneira a rede sabe exatamente de qual dispositivo os dados estão sendo enviados, e assim é possível começar a receber e visualizar os dados no sistema. O *back-end* Sigfox pode ser acessado através do endereço <https://backend.sigfox.com/auth/login>. No *back-end* Sigfox, os dados são recebidos exatamente como são enviados, em uma mensagem de 12 bytes com valores em hexadecimal. Do ponto de vista prático isso é um problema, pois o usuário final não tem discernimento para saber o que aquela mensagem tem a dizer. Por isso é necessário integrar o TagoIO ao sistema do *back-end* Sigfox para que dessa maneira os 12 bytes possam ser tratados e apresentados em formato de *dashboards*. Abaixo, as figuras 4.3 e 4.4 ilustram os dados recebidos no *back-end* Sigfox e apresentados no TagoIO, respectivamente.

page 1 

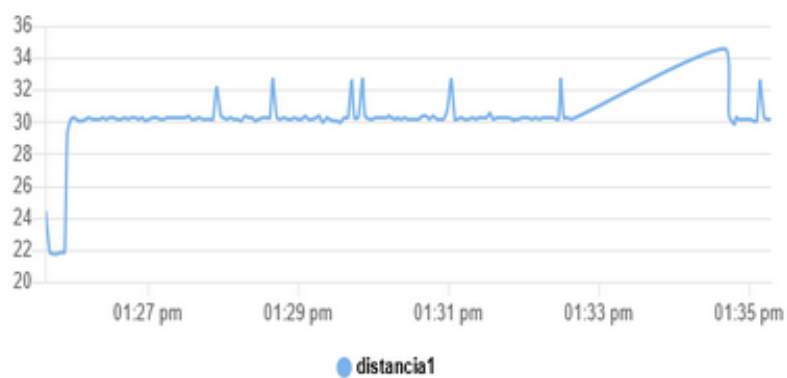
Time	Seq Num	Data / Decoding	LQI	Callbacks	Location
2021-08-09 13:35:16	423	0302030709dd000000000000			
2021-08-09 13:35:13	422	0302030709dd000000000000			
2021-08-09 13:35:10	421	0303030709dd000000000000			
2021-08-09 13:35:07	420	031a030803dd000000000000			
2021-08-09 13:35:05	419	0301030803dd000000000000			
2021-08-09 13:35:01	418	0301030802dd000000000000			
2021-08-09 13:34:59	417	0302030802dd000000000000			
2021-08-09 13:34:56	416	0302030802dd000000000000			

**Figura 4.3** Mensagens recebidas no *Back-end* Sigfox.

Distância Naval



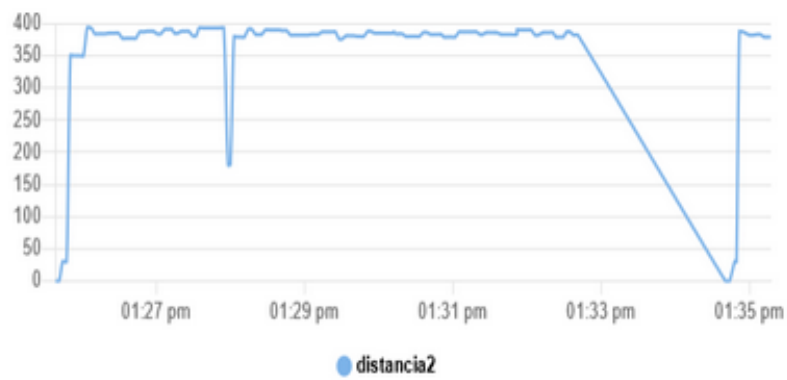
Line Chart #1



Distância defesa



Dist2 chart



**Figura 4.4** Mensagens recebidas no TagoIO.

## CAPÍTULO 5

# Resultados

Os resultados obtidos foram bastante satisfatórios. Conseguiu-se montar o protótipo implementando todo o HW em uma placa padrão, para que fosse possível realizar os testes.

Os testes foram realizados em escala, utilizando uma piscina e um navio controlado por rádio. Fixou-se o protótipo na borda da piscina e aproximou-se o navio até verificar-se que a distância da embarcação estava muito próxima da borda da piscina, simulando dessa forma uma compressão excessiva da defesa.

Depois disso, afastou-se o navio da borda da piscina, simulando um afastamento excessivo do navio em relação ao cais. Em ambos os casos, de aproximação excessiva e afastamento excessivo, as notificações foram enviadas via rede Sigfox e os dados foram recebidos no TagoIO, como foi ilustrado pelas figuras 4.3 e 4.4.

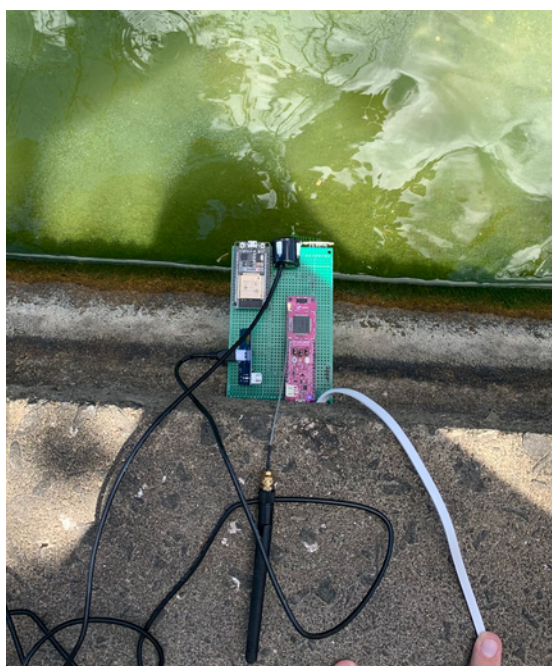
Esse projeto foi um dos sete finalistas na competição *IoT Student Contest - 2021*, uma competição destinada a alunos de graduação e ensino técnico que tem como objetivo propiciar a oportunidade de desenvolvimento de sistemas inteligentes e inovadores, contribuindo para a evolução deste segmento no Brasil, promovido pela empresa HT MICRON, IEEE CASS e SBMicro, e aconteceu dentro do *Chip in the Fields*, conferência virtual que foi realizada pela Sociedade Brasileira de Microeletrônica (SBMicro) e pela Sociedade Brasileira de Computação (SBC), entre os dias 24 e 28 de agosto de 2021[23].

Nas figuras 5.1, 5.2 e 5.3 é possível verificar o protótipo implementado, a maneira com a qual ele foi posicionado às margens da piscina, e a imagem do navio em escala reduzida utilizado para os testes.





**Figura 5.1** Protótipo do sistema.



**Figura 5.2** Posicionamento do protótipo às margens da piscina.



**Figura 5.3** Testes do protótipo com o navio em escala.

## CAPÍTULO 6

# Conclusão

Este trabalho serviu para mostrar como é possível resolver um problema que está sujeito a acontecer em todo o mundo, utilizando ferramentas de eletrônica, sistemas embarcados e internet das coisas. Hoje, a solução que os portos têm para o problema apresentado é utilizar células de carga que são instaladas nas amarras para medir a tração em que esta se encontra. Através da tração medida, é possível determinar se o navio está se afastando demais do cais. No entanto, essas células de carga têm um custo muito elevado, e levando em consideração que devem ser instaladas em diversas amarras de atracação, o custo para o porto se torna ainda maior. Outro problema é que essas células de carga possuem especificidades de aplicação, são dimensionadas para certo tipo de tração, o que pode fazer com que sejam utilizadas para embarcações de portes específicos. Para se ter uma ideia, a figura 6.1 ilustra duas células de carga, e seus respectivos custos, que são utilizadas hoje no contexto portuário.



**Figura 6.1** Células de carga e seus respectivos preços.

Em comparação com os valores das células de carga ilustrados na figura 6.1, a tabela 6.1 ilustra o custo dos componentes de hardware necessários para o desenvolvimento do protótipo apresentado nesse trabalho.

**Tabela 6.1** Componentes necessários para o protótipo e seus respectivos custos.

Componente	Custo
Esp32	40,00
Sensor Ultrassônico JSN-SR04T	49,90
Sensor Laser VL5310/1x v2	115,00
Antena 2.4Ghz 3dBi	23,90
Total	227,90

É fácil notar que o preço do iMCP-HT32SX não se encontra na tabela 6.1. Isso se dá porque até o momento em que esse trabalho foi realizado, não foi encontrado a comercialização desse chip no mercado. A placa de desenvolvimento utilizada na concepção desse protótipo foi fornecida pela própria HT Micron durante o *IoT Student Contest - 2021*, então não há informação sobre o custo desse produto.

O protótipo desenvolvido nesse trabalho pode substituir as células de carga utilizada nos portos, trazendo menos custos para os portos e garantindo uma segurança de operação que pode preservar vidas e evitar prejuízos financeiros. Para trabalhos futuros, pode ser feito o projeto e desenvolvimento de um sistema mecânico que inclua um case para proteção do sistema eletrônico contra as intempéries da natureza, e uma estrutura de fixação para manter o sistema fixado ao cais em segurança.

# Referências

- [1] pt. URL: <https://www.fazcomex.com.br/blog/portos-brasileiros-quais-os-principais/>.
- [2] pt. URL: <http://www.suape.pe.gov.br/pt/porto/infraestrutura-portuaria/guia-portuario>.
- [3] *Internet das coisas*. pt. URL: [https://pt.wikipedia.org/wiki/Internet\\_das\\_coisas](https://pt.wikipedia.org/wiki/Internet_das_coisas).
- [4] pt. URL: <https://www.oracle.com/br/internet-of-things/what-is-iot/>.
- [5] Jack G.; Michael Barr. “Embedded Systems Dictionary”. Em: *Taylor Francis* (2003). ISBN 1-578-20120-9.
- [6] Elecia White. “Making Embedded Systems: Design Patterns for Great Software”. Em: *O’Reilly Media, Inc.* (2011). ISBN 1-449-30214-9.
- [7] Jack Ganssle. “The Art of Programming Embedded Systems”. Em: *Elsevier* (2012). ISBN 0-080-49942-2.
- [8] *Microcontrolador - Parte III (MIC094)*. pt. URL: <https://www.newtoncbraga.com.br/index.php/microcontroladores/142-texas-instruments/8217-microcontrolador-msp430-parte-iii-mic094>.
- [9] *Comunicação serial*. pt. URL: [https://pt.wikipedia.org/wiki/Comunica%C3%A7%C3%A3o\\_serial](https://pt.wikipedia.org/wiki/Comunica%C3%A7%C3%A3o_serial).
- [10] *Protocolo I2C*. pt. URL: <http://www.univasf.edu.br/~romulo.camara/novo/wp-content/uploads/2013/11/Barramento-e-Protocolo-I2C.pdf>.

- [11] *Comunicação via RF*. pt. URL: [http://www.univasf.edu.br/~gari/futvasf/paginas/download/Apresenta%C3%A7%C3%A3oRF\\_Manoel%2009-04-2010.pdf](http://www.univasf.edu.br/~gari/futvasf/paginas/download/Apresenta%C3%A7%C3%A3oRF_Manoel%2009-04-2010.pdf).
- [12] André Schneider de Oliveira; Fernando Souza de Andrade. “Sistemas embarcados: hardware e firmware na prática”. Em: *Editora Érica Ltda* (2006). ISBN 8-536-50105-7.
- [13] Gary Stringham. “Hardware/Firmware Interface Design: Best Practices for Improving Embedded Systems Development”. Em: *Newnes* (2009). ISBN 0-080-88019-3.
- [14] Miloš Dragutin Ercegovic; Tomás Lang. “Digital systems and hardware/firmware algorithms”. Em: *Wiley* (1985). ISBN 0-471-88393-X.
- [15] *Site plataforma TagoIO*. pt. URL: <https://tago.io/>.
- [16] *Stm Cube IDE*. pt. URL: <https://www.st.com/en/development-tools/stm32cubeide.html>.
- [17] *HT32SX - HT Micron's Sigfox™ Monarch SiP*. en. URL: <https://github.com/htmicron/ht32sx>.
- [18] *ESP32 Datasheet*. pt. URL: [https://img.filipeflop.com/files/download/Datasheet\\_ESP8266\\_esp32\\_en.pdf](https://img.filipeflop.com/files/download/Datasheet_ESP8266_esp32_en.pdf).
- [19] *JSNSR04T*. pt. URL: <https://storage.googleapis.com/audaeletronicadatasheet/SN-SR04T.pdf>.
- [20] *VL53L0X Datasheet - STMicroelectronics*. en. URL: <https://pdf1.alldatasheet.com/datasheet-pdf/view/948120/STMICROELECTRONICS/VL53L0X.html>.
- [21] *Tecnologia Sub-GHz Sigfox: Conectividade de rádio frequência de longo alcance*. pt. URL: <https://www.macnicadhw.com.br/blog/2019-05-13-tecnologia-sub-ghz-sigfox-conectividade-de-radio-frequencia-de-longo-alcance>.
- [22] *TagoIO soluções para IoT*. pt. URL: <https://www.khomp.com/pt/tagoio-solucao-iot/>.

[23] *Iot Student Contest - 2021*. pt. URL: <https://www.ufrgs.br/emicro/iotstudentcontest>