

Sistemas Inteligentes de Escoamento Urbano: Um estudo de caso sobre as enchentes em São Paulo

MARCOS ALEXANDRE PIELLUSCH
Setembro de 2024

POLITÉCNICO DO PORTO
INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

Sistemas Inteligentes de Escoamento Urbano: Um estudo de caso sobre as enchentes em São Paulo

Marcos Alexandre Zioni Ré Piellusch

Mestrado em Engenharia Electrotécnica e de Computadores
Área de Especialização em Automação e Sistemas



DEPARTAMENTO DE ENGENHARIA ELETROTÉCNICA
Instituto Superior de Engenharia do Porto

Setembro, 2024

*Esta dissertação satisfaz, parcialmente, os requisitos que constam da Ficha de
Unidade Curricular de Tese/Dissertação, do 2º ano, do Mestrado em
Engenharia Electrotécnica e de Computadores, Área de Especialização em
Automação e Sistemas.*

Candidato: Marcos Alexandre Zioni Ré Piellusch, N.º 1210375,
1210375@isep.ipp.pt

Orientação Científica: Prof. Dr. Lino Figueiredo, lbf@isep.ipp.pt



DEPARTAMENTO DE ENGENHARIA ELETROTÉCNICA
Instituto Superior de Engenharia do Porto
Rua Dr. António Bernardino de Almeida, 431, 4200-072 Porto

Setembro, 2024

Dedico esse trabalho à minha família e minha namorada Rebeca, que sempre me apoiou e incentivou, e aos amigos e professores que contribuíram para minha jornada académica. . . .

Agradecimentos

Agradeço, primeiramente, à minha família, pelo apoio incondicional e incentivo ao longo de toda a minha trajetória acadêmica. Aos amigos, por sua companhia e palavras de encorajamento nos momentos mais desafiadores.

Aos meus orientadores, professores e colegas, cujos ensinamentos e orientações foram essenciais para a realização deste trabalho, expresso minha profunda gratidão.

Um agradecimento especial à oportunidade de realizar o intercâmbio em Portugal, que não só me proporcionou uma experiência acadêmica enriquecedora, mas também a possibilidade de obter uma dupla titulação, culminando nesta tese de mestrado...

Resumo

Este trabalho aborda a implementação de sistemas de monitoramento inteligente para combater inundações urbanas, com foco no desenvolvimento de um sistema de drenagem inteligente utilizando tecnologia IoT. O estudo detalha a criação de uma rede de sensores ultrassônicos integrados aos sistemas de drenagem da cidade para monitoramento em tempo real das variações de nível de água. É apresentado um protótipo que utiliza microcontroladores ESP32 integrados com LoRa, permitindo comunicação sem fio de longo alcance para transmissão eficiente de dados através de vastas áreas urbanas. A pesquisa explora também o papel vital da análise de dados em tempo real na gestão de infraestruturas urbanas, oferecendo insights sobre a arquitetura do sistema, a tecnologia envolvida e seu potencial para aplicações mais amplas.

Palavras-Chave: Inundações Urbanas, IoT, Sensores Ultrassônicos, ESP32, LoRa.

Abstract

This work addresses the implementation of intelligent monitoring systems to combat urban flooding, focusing on the development of a smart drainage system using IoT technology. The study details the creation of a network of ultrasonic sensors integrated into the city's drainage systems for real-time monitoring of water level variations. A prototype is presented that uses ESP32 microcontrollers integrated with LoRa, enabling long-range wireless communication for efficient data transmission across vast urban areas. The research also explores the vital role of real-time data analysis in managing urban infrastructure, offering insights into the system architecture, the technology involved, and its potential for broader applications.

Keywords: Urban Floodings, IoT, Ultrasonic Sensors, ESP32, LoRa.

Índice

Lista de Figuras	ix
Lista de Tabelas	xi
Listagens	xiii
Lista de Acrónimos	xv
Lista de Símbolos	xvii
1 Introdução	1
1.1 Contextualização	1
1.2 Objetivos	2
1.3 Cronograma do Projeto	3
1.4 Organização do Relatório	3
2 Estado da Arte	5
2.1 Microcontroladores	5
2.1.1 Estrutura	5
2.1.2 Arquiteturas	6
2.1.3 Consumo de Energia	7
2.1.4 Interfaces de Comunicação	7
2.1.5 Modelos de Microcontroladores	9
Modulo de Desenvolvimento ESP32	9
Arduino	10
STM32	12
2.2 Sensores Ultrassónicos	13
2.2.1 Classificação dos Sensores Ultrassónicos	13
Sensores de Medição de Distância	14
Sensores de Detecção de Objetos	14
2.2.2 Princípios de Funcionamento	14
Emissão e Receção de Ondas Sonoras	14
Medição do Tempo de Ida e Volta	14
Cálculo da Distância	15

	Considerações sobre o Ambiente	15
	Calibração e Precisão	15
2.2.3	Aplicações e Casos de Uso	16
	Sensor de Distância Ultrassônico HC-SR04	16
	Módulo Sensor Ultrassônico Impermeável JSN-SR04M	16
2.2.4	Desafios e Limitações	17
	Operação em Ambientes Hostis	17
	Resistência à Água e Humidade	17
	Durabilidade e Vida Útil	17
	Consumo de Energia	17
	Limitações de Detecção	17
2.3	Comunicação Sem Fio	17
2.3.1	Protocolos e Padrões de Comunicação Sem Fio	18
	Bluetooth Low Energy (BLE)	18
	Zigbee	18
	LoRa e LoRaWAN	19
	Narrowband IoT (NB-IoT)	20
	SigFox	21
2.3.2	Alcance Estendido e Tecnologias <i>Long Range</i>	22
	Comparação de Alcance e Eficiência	22
2.3.3	Custo e Disponibilidade em São Paulo	22
	LoRa em São Paulo	22
2.3.4	Modelos de Módulos de Comunicação Sem Fio	23
	Modelos Comuns e Suas Aplicações	23
2.4	Gateways	24
2.4.1	Classificação de <i>Gateways</i> IoT	24
2.4.2	Protocolos de Comunicação para <i>Gateways</i>	25
2.4.3	Arquitetura e Funcionamento dos <i>Gateways</i>	25
2.5	Soluções Avançadas de <i>Cloud</i> para IoT	25
2.5.1	Arquiteturas de Nuvem Adaptadas para IoT	26
2.5.2	Plataformas e Serviços de Nuvem Especializados para IoT	26
2.5.3	Segurança e Privacidade de Dados em Ambientes IoT de Nuvem	27
2.5.4	Escalabilidade e Análise de Custos em Soluções de Cloud IoT	27
2.5.5	ThingSpeak	27
	Arquitetura e Funcionamento	28
	Recursos e Funcionalidades	28
	Modelos de Uso e Aplicações	28
	Vantagens e Desvantagens	29
	Conclusão	29

3	Arquitetura da Aplicação	31
3.1	Hardware Utilizado	31
3.1.1	Módulo ESP32/LoRa	31
	Facilidade de Programação e Recursos Avançados	32
	Baixo Consumo e Alta Eficiência	32
	Integração de Micro-controlador e Módulo LoRa com Display OLED	32
	Vantagens da Escolha de um Kit Integrado	32
3.1.2	Sensor Ultrassônico JSN-SR04T	32
	Características e Especificações Técnicas	33
	Vantagens na Implementação do Projeto	33
	Integração com o Sistema e Calibração	34
3.2	Software IoT	34
3.2.1	Escolha da Plataforma	35
	Integração e Compatibilidade com Hardware IoT	35
	Capacidades de Trigger Automatizado para Resposta Rápida	35
	Suporte à Análise de Dados em Tempo Real	35
	Custos	35
3.2.2	Implementação no ThingSpeak	35
3.3	Topologia do Projeto	36
3.3.1	Descrição da Topologia	36
3.3.2	Diagrama de Blocos	36
3.4	Análise dos Custos do Projeto	37
3.4.1	Custos do Módulo ESP32/Lora	37
3.4.2	Custos do Sensor Ultrassônico	38
4	Implementação do Sistema	41
4.1	Montagem do Hardware	41
4.1.1	Montagem do Módulo de Medição - Nó	41
	Sensor Ultrassônico	41
	Esquema Elétrico	42
4.1.2	Montagem do Gateway	43
4.2	Programação do Nó - ESP32	44
4.2.1	Configuração Inicial e Bibliotecas	44
4.2.2	Funções de Medição de Distância e Transmissão de Dados	44
4.2.3	Loop Principal	45
4.3	Programação do Gateway - ESP32	46
4.3.1	Configuração Inicial e Bibliotecas	46
4.3.2	Configurações de Conectividade e Variáveis de Canal ThingS- peak	46

4.3.3	Recepção de Dados LoRa e Transmissão para ThingSpeak . .	47
4.3.4	Loop Principal	48
4.4	Configuração do Dashboard ThingSpeak	48
4.4.1	Criação do Canal e Configuração dos Campos	48
4.4.2	API Keys e Conexão com o Gateway	49
4.4.3	Visualização e Análise de Dados	49
4.4.4	Gerenciamento e Compartilhamento de Dados	50
5	Resultados	51
5.1	Monitoramento da Distância	51
5.1.1	Configuração do Alerta	51
5.1.2	Demonstração do Protótipo em Funcionamento	52
5.1.3	Análise dos Dados Coletados	54
5.2	Desempenho do Alerta Automático	55
5.3	Conclusão dos Resultados	55
5.4	Limitações e Trabalhos Futuros	55
6	Conclusões	57
6.1	Objetivos Alcançados	57
6.2	Perspectivas Futuras	58
6.3	Considerações Finais	58
	Referências	59

Lista de Figuras

1.1	Cronograma de Desenvolvimento do Projeto	3
2.1	Kit de Desenvolvimento ESP32-DevKitC contendo o Microcontrolador ESP32	10
2.2	Kit de Desenvolvimento Arduino Uno R3	11
2.3	Placa de desenvolvimento ARM STM32 STM32F103C8T6	13
2.4	Módulo Bluetooth Low Energy - HC-08	18
2.5	Módulo XBee S2 S2C Zigbee	19
2.6	Módulo Lora SX1276	20
2.7	Módulo NB-IOT 4 CLICK	21
2.8	Módulo SIGFOX-MOD1-C	21
2.9	Placa ESP32 LoRa SX1276 868/915Mhz com OLED	24
3.1	Arquitetura Inicial do Projeto	37
3.2	Sensor de Distância TeraRanger Evo 15m	38
4.1	Ligações Sensor Ultrassônico JSN-SR04T	42
4.2	Esquema Elétrico de ligação Sensor Ultrassônico - ESP32	43
4.3	Módulo de Medição - Nó	43
4.4	Módulo Gateway	44
4.5	Dashboard Thingspeak	49
5.1	Configuração do alerta no Dashboard ThingSpeak	52
5.2	Protótipo em estado normal, com alerta inativo.	53
5.3	Protótipo em situação de alerta, com nível de água elevado e alerta ativo.	54
5.4	Análise dos Dados de Distância Coletados pelo Sensor Ultrassônico	54

Lista de Tabelas

2.1	Comparação entre LoRa, NB-IoT e SigFox	22
3.1	Comparação de custos dos Módulos ESP32/Lora	37
3.2	Comparação de custos do Sensor Ultrassónico	38
3.3	Comparação de custos do Sensor Ultrassónico	39

Listagens

Lista de Acrônimos

ADC	<i>Analog to Digital Converter</i>
CAN	<i>Controller Area Network</i>
CGE	Centro de Gerência de Emergência
CISC	<i>Complex Instruction Set Computer</i>
CoAP	<i>Constrained Application Protocol</i>
CPU	<i>Central Process Unit</i>
DAC	<i>Digital to Analog Converter</i>
DSP	<i>Digital Signal Processor</i>
GDPR	<i>General Data Protection Regulation</i>
GPIO	<i>General Purpose Input/Output</i>
HTTP	<i>Hypertext Transfer Protocol</i>
I2C	<i>Inter-Integrated Circuit</i>
IOT	<i>Internet Of Things</i>
LAN	<i>Local Area Network</i>
MQTT	<i>Message Queuing Telemetry Transport</i>
MQTT-SN	<i>Message Queuing Telemetry Transport for Sensor Networks</i>
PCB	<i>Printed Circuit Board</i>
PWM	<i>Pulse Width Modulation</i>
RAM	<i>Random Access Memory</i>
RISC	<i>Reduced Instruction Set Computer</i>
ROM	<i>Read Only Memory</i>
SCL	<i>Serial Clock Line</i>

SDA	<i>Serial Data Line</i>
SPI	<i>Serial Peripheral Interface</i>
UART	<i>Universal Asynchronous Receiver/Transmitter</i>
USB	<i>Universal Serial Bus</i>

Lista de Símbolos

Símbolo	Descrição	Unidades
f	força	N
i	corrente	A
M	massa	kg
P	potência	W
x	deslocamento	m
θ	deslocamento angular	rad
ω	velocidade angular	rad s ⁻¹

Capítulo 1

Introdução

A intensificação de eventos climáticos extremos, como chuvas torrenciais e inundações, nos últimos anos, impôs desafios crescentes às cidades em diversas regiões do mundo. As inundações urbanas, em particular, representam uma séria ameaça, gerando danos materiais consideráveis, interrupção de serviços essenciais e impactos ambientais deletérios. Diante dessa realidade premente, a necessidade de desenvolver soluções inteligentes para prevenir e mitigar os efeitos dessas inundações se torna cada vez mais urgente.

Este trabalho se propõe a explorar, de forma abrangente e crítica, o desenvolvimento, os desafios e os benefícios do uso de sistemas inteligentes de monitoria como solução para o problema citado. Através de uma análise aprofundada da literatura e da investigação de estudos de caso relevantes, este estudo busca contribuir para o avanço do conhecimento sobre essa temática crucial e auxiliar na formulação de soluções tecnológicas eficazes para a prevenção de inundações em áreas urbanas.

1.1 Contextualização

A cidade de São Paulo, Brasil, com sua densa malha urbana e crescente população, enfrenta um cenário preocupante no que diz respeito às inundações. A metrópole paulista vivenciou um cenário preocupante durante a última temporada de chuvas, entre novembro de 2022 e abril de 2023. De acordo com dados do Centro de Gerência de Emergência (CGE), a cidade registou um aumento alarmante de 57% no número de alagamentos, totalizando 771 ocorrências em seis meses. Essa situação

contrasta com o período chuvoso anterior (2021-2022), quando foram registadas 491 inundações.

Essa situação resultou em diversos transtornos, incluindo alagamentos, deslizamentos de terra e interrupções de serviços essenciais, afetando milhares de pessoas e causando prejuízos consideráveis à infraestrutura urbana.

A Prefeitura de São Paulo reconhece a urgência em combater as enchentes que assolam a cidade e investiram R\$ 3,4 bilhões entre 2022 e 2023 para mitigar os riscos e garantir a segurança da população. Essa iniciativa robusta demonstra o compromisso da administração em transformar a relação da cidade com as chuvas, abrindo portas para o desenvolvimento de novas soluções tecnológicas.

Este trabalho tem como objetivo explorar o desenvolvimento de sistemas de drenagem inteligentes como uma solução inovadora para prevenir inundações em áreas urbanas. Serão abordados aspectos técnicos, funcionais e operacionais desses sistemas, bem como os benefícios que podem trazer para a infraestrutura urbana e para a qualidade de vida da população.

1.2 Objetivos

Para combater os alagamentos em São Paulo, é proposto o desenvolvimento de um sistema de monitoramento inteligente de bueiros. Através de uma rede de sensores nos bueiros, a solução permitirá a coleta de dados em tempo real sobre o nível da água e a presença de obstruções.

Esses dados serão enviados para uma plataforma centralizada, que fornecerá uma visão completa da situação das redes de bueiros, permitindo a identificação de pontos críticos com antecedência e a agilização das ações de limpeza e manutenção. A solução trará diversos benefícios, como redução do tempo de resposta a alagamentos, otimização da limpeza e maior segurança para a população.

Os objetivos parciais do projeto são:

- **Estudo aprofundado das tecnologias IoT:** Realizar uma análise detalhada das tecnologias envolvidas em uma aplicação IoT, incluindo microcontroladores, sensores, comunicação sem fio e computação em nuvem.
- **Desenvolvimento de um protótipo IoT funcional:** Criar um protótipo que permita a medição de distância e a emissão de alertas, demonstrando a viabilidade do sistema proposto.
- **Configuração de rede sem fio:** Estabelecer uma rede sem fio que suporte múltiplos nós de medição e permita a conexão com uma nuvem IoT via WiFi, assegurando a transmissão eficiente dos dados coletados.

- **Testes e validação do sistema:** Implementar testes rigorosos para validar a precisão e a confiabilidade do sistema, garantindo que ele possa ser utilizado de maneira eficaz em cenários reais.
- **Análise de impacto e benefícios:** Avaliar o impacto potencial do sistema na redução de alagamentos, na otimização dos processos de manutenção e na melhoria da segurança pública.
- **Documentação e disseminação dos resultados:** Documentar todo o processo de desenvolvimento e os resultados obtidos, e disseminar as conclusões por meio de publicações e apresentações para incentivar a adoção de soluções semelhantes em outras áreas urbanas.

1.3 Cronograma do Projeto

A figura 1.1 ilustra o cronograma planejado para cumprimento dos objetivos determinados.

Objetivo	jan/24	fev/24	mar/24	abr/24	mai/24	jun/24	jul/24	ago/24	set/24
Estudo aprofundado das tecnologias IoT									
Desenvolvimento de um protótipo IoT funcional									
Configuração de rede sem fio									
Testes e validação do sistema									
Análise de impacto e benefícios									
Documentação e disseminação dos resultados									

Figura 1.1: Cronograma de Desenvolvimento do Projeto

1.4 Organização do Relatório

Este relatório está estruturado em seis capítulos, organizados da seguinte forma:

- O **Capítulo 1** apresenta a introdução ao tema, com a contextualização, os objetivos do trabalho, o cronograma do projeto e a organização do relatório.
- O **Capítulo 2** descreve o estado da arte, abordando os conceitos teóricos que embasam este estudo. Este capítulo inclui seções detalhadas sobre microcontroladores, sensores ultrassônicos, comunicação sem fio, gateways IoT e soluções avançadas em nuvem para IoT, com destaque para a plataforma ThingSpeak.
- O **Capítulo 3** discute a arquitetura da aplicação desenvolvida, descrevendo o hardware e software utilizados, bem como a topologia do projeto e uma análise dos custos envolvidos.

- O **Capítulo 4** foca na implementação do sistema, detalhando a montagem do hardware, a programação do nó e do gateway, além da configuração do dashboard na plataforma ThingSpeak.
- O **Capítulo 5** apresenta os resultados obtidos, incluindo o monitoramento de distâncias, desempenho do alerta automático, análise dos dados coletados, e discute as limitações encontradas, bem como as sugestões para trabalhos futuros.
- O **Capítulo 6** contém as conclusões finais, revisando os objetivos alcançados, apresentando perspectivas para futuros desenvolvimentos e finalizando com considerações gerais sobre o trabalho.

Capítulo 2

Estado da Arte

Nesta seção, será explorado o panorama atual das tecnologias que sustentam os sistemas de monitoramento, processamento e transmissão de dados para projetos de *Internet Of Things* (IOT). Cada subseção se concentrará em analisar as principais opções disponíveis, suas características e especificações, fornecendo uma visão abrangente das ferramentas que viabilizam a implementação de soluções eficazes para o monitoramento de redes de drenagem urbana.

2.1 Microcontroladores

Na escolha do microprocessador para um sistema embarcado, diversos fatores são cruciais para garantir o bom funcionamento e atender às necessidades específicas da aplicação. Nesta seção, será explorado os principais aspectos a serem considerados como Arquitetura, Consumo de energia, etc.

2.1.1 Estrutura

Os microcontroladores são dispositivos integrados que possuem uma arquitetura complexa, mas altamente eficiente para a execução de tarefas específicas em sistemas embarcados. Sua estrutura pode ser dividida em várias partes fundamentais: [1]

- *Central Process Unit* (CPU): É o núcleo do microcontrolador responsável pela execução das instruções do programa. Pode variar em termos de arquitetura

(como *Reduced Instruction Set Computer* (RISC), *Complex Instruction Set Computer* (CISC), *Digital Signal Processor* (DSP), etc) e velocidade de clock.

- Memória de Programa (Flash ou *Read Only Memory* (ROM)): Armazena o código executável do programa. A capacidade de memória de programa pode variar significativamente entre diferentes microcontroladores e é crucial para determinar a complexidade e o tamanho do software que pode ser executado.
- *Random Access Memory* (RAM): Utilizada para armazenar variáveis, resultados intermediários e pilha de execução. A quantidade de RAM disponível afeta diretamente a capacidade do microcontrolador de lidar com grandes conjuntos de dados e realizar operações complexas.
- Periféricos Integrados: Incluem uma variedade de componentes como portas de entrada/saída *General Purpose Input/Output* (GPIO), conversores analógico-digitais *Analog to Digital Converter* (ADC), conversores digitais-analógicos *Digital to Analog Converter* (DAC), interfaces de comunicação (*Universal Asynchronous Receiver/Transmitter* (UART), *Serial Peripheral Interface* (SPI), *Inter-Integrated Circuit* (I2C)), temporizadores, *Pulse Width Modulation* (PWM), entre outros. Esses periféricos permitem a interação do microcontrolador com o ambiente externo e a realização de diversas funções.
- Barramentos de Comunicação: Permitem a comunicação entre os diferentes componentes do microcontrolador, como CPU, memória, periféricos e outros dispositivos externos conectados.
- Controladores de Interrupção: Gerenciam interrupções externas, permitindo que o microcontrolador responda a eventos em tempo real sem interromper a execução do programa principal.
- Clock e Osciladores: Fornecem a base de tempo para as operações do microcontrolador, determinando a velocidade de execução das instruções e a taxa de comunicação com dispositivos externos.

2.1.2 Arquiteturas

A arquitetura de um micro-controlador desempenha um papel crucial na sua performance, eficiência energética e flexibilidade. As arquiteturas mais comuns incluem:

- RISC: Arquitetura com foco em instruções simples e de rápida execução, ideal para sistemas com restrições de energia ou tempo real. [2]
- CISC: Arquitetura com instruções mais complexas e multi-funcionais, oferecendo maior flexibilidade, mas com maior consumo de energia. [2]

- DSP: Arquitetura otimizada para processamento de sinais digitais, ideal para aplicações que exigem alta performance em cálculos matemáticos. [3]

2.1.3 Consumo de Energia

O consumo de energia em microcontroladores é um aspecto crítico, especialmente em aplicações móveis, portáteis ou que demandam baixo consumo para prolongar a vida útil da bateria.[1] Para reduzir o consumo de energia E , diversas técnicas são aplicáveis:

- Redução da Frequência de Clock: Ao diminuir a frequência de operação, o consumo de energia diminui proporcionalmente

$$E \propto f \quad (2.1)$$

Essa redução demonstrada na equação 2.1 pode ser estática, ajustando a frequência mínima necessária, ou dinâmica, variando conforme as exigências temporais da aplicação.

- Redução da Tensão de Operação: A energia consumida é proporcional ao quadrado da tensão, conforme a equação 2.2:

$$E \propto U^2 \quad (2.2)$$

Reduzir a tensão impacta significativamente o consumo, mas dentro da faixa especificada para o controlador, evitando comportamentos indesejados.

- Desligamento de Módulos Não Utilizados: Desativar módulos não usados durante operações internas ou períodos de espera reduz o consumo de energia. Alguns controladores oferecem modos de suspensão que desligam diferentes componentes para economizar energia.
- Design Otimizado: Controladores como a família MSP430 da Texas Instruments são projetados com foco na eficiência energética desde o início, reduzindo drasticamente o consumo durante a operação normal e em modo de espera.

2.1.4 Interfaces de Comunicação

As interfaces de comunicação em microcontroladores desempenham um papel crucial ao permitir a comunicação com outros dispositivos, sejam eles outros microcontroladores, periféricos ou um PC hospedeiro. Elas podem ser categorizadas com base em propriedades como serial ou paralela, síncrona ou assíncrona, comunicação ponto a ponto ou por barramento, duplex completo ou meio-duplex, e princípio mestre-escravo ou de parceiros iguais.

As interfaces seriais transmitem dados sequencialmente em um único bit por vez, enquanto as paralelas usam várias linhas de dados para transferir mais de um bit simultaneamente. A sincronicidade da comunicação refere-se à relação entre os clocks de envio e recebimento, sendo que interfaces síncronas estão ligadas aos clocks de envio, enquanto as assíncronas não.

A topologia de barramento permite a conexão de múltiplos dispositivos, enquanto as conexões ponto a ponto são para apenas dois dispositivos. A comunicação bidirecional é comum entre controlador e periféricos, com conexões full-duplex permitindo transmissão simultânea e half-duplex permitindo apenas um nó transmitir por vez. Além disso, as interfaces podem ser diferenciadas como single-ended ou differential, sendo que as differential oferecem vantagens em termos de ruído e distâncias de transmissão. [1]

- **UART:** Interface serial assíncrona comumente usada para comunicação entre dispositivos. Ela transmite dados de forma sequencial, um bit por vez, utilizando linhas de transmissão e recepção separadas. A comunicação UART não requer um clock de sincronização compartilhado entre os dispositivos.
- **SPI:** O SPI é uma interface serial síncrona utilizada para transferir dados entre um microcontrolador e dispositivos periféricos, como sensores, memórias e displays. Ele usa linhas dedicadas para dados de entrada e saída, um clock de sincronização e linhas de controle para seleção de dispositivos.
- **I2C:** O I2C é uma interface serial síncrona usada para comunicação entre dispositivos em um barramento de dois fios (*Serial Data Line* (SDA) e *Serial Clock Line* (SCL)). Ele permite a conexão de múltiplos dispositivos em um mesmo barramento, utilizando endereços únicos para cada dispositivo e protocolo de comunicação mestre-escravo.
- **Controller Area Network (CAN):** É um protocolo de comunicação serial diferencial amplamente utilizado em aplicações automotivas e industriais. Ele oferece alta imunidade a ruídos, transmissão confiável e suporte para comunicação multi-mestre em redes de controle e monitoramento de sistemas.
- **Ethernet:** Interface de comunicação de rede padrão que utiliza cabos de par trançado ou fibras ópticas para transmitir dados em alta velocidade. Ela é amplamente empregada em redes locais (*Local Area Network* (LAN)) para conectar dispositivos como computadores, roteadores e dispositivos IoT.
- **Universal Serial Bus (USB):** É uma interface serial de propósito geral usada para conectar dispositivos periféricos a computadores e outros dispositivos hospedeiros. Ele suporta transferência de dados em alta velocidade, alimentação elétrica para dispositivos e protocolos de comunicação complexos.

2.1.5 Modelos de Microcontroladores

Os microcontroladores são peças-chave na engenharia de sistemas eletrônicos, oferecendo uma gama de opções com características específicas. Destacam-se três exemplos notáveis: o ESP32, com sua conectividade avançada e recursos de segurança; a plataforma Arduino, reconhecida pela facilidade de programação e ampla comunidade, que utiliza microcontroladores como o ATmega328P; e o STM32, valorizado por sua alta performance e eficiência energética. Esta seção explorará em detalhes as características e aplicações desses modelos e plataformas, fornecendo informações cruciais para a escolha adequada em diferentes projetos.

Modulo de Desenvolvimento ESP32

O microcontrolador ESP32, desenvolvido pela Espressif Systems, representa um marco na computação embarcada e sistemas IoT. Sua arquitetura de 32 bits o capacita a lidar com tarefas complexas e processamento em tempo real, crucial em aplicações críticas e sistemas embarcados avançados.

Um dos seus grandes trunfos é a conectividade avançada, suportando Wi-Fi 802.11 b/g/n/e/i (com taxas de transferência de até 150 Mbps) e Bluetooth Classic/LE (com suporte a BLE 5.0 para maior alcance e menor consumo de energia). Essa versatilidade o torna a escolha ideal para aplicações IoT, permitindo interação sem fio com outros dispositivos e integração em redes locais e globais. Além disso, sua compatibilidade com diversos periféricos, como UART (até 5 canais), SPI (até 4 canais), I2C (até 2 canais), GPIOs (até 34 pinos configuráveis), ADC (até 18 canais com resolução de até 12 bits), DAC (2 canais com resolução de 8 bits), PWM (até 16 canais) e interfaces para sensores de toque capacitivo, amplia suas possibilidades de interação com o mundo exterior e facilita a conexão de sensores, atuadores e dispositivos periféricos.

Em termos de memória e armazenamento, o ESP32 oferece uma combinação robusta: até 4 MB de memória flash para armazenar o firmware e dados da aplicação, até 520 KB de SRAM para execução de código e armazenamento de variáveis, e 8 KB de RTC SRAM para retenção de dados durante o modo de baixo consumo. Além disso, conta com uma memória EEPROM interna de até 4 KB para armazenamento não volátil de pequenos conjuntos de dados, como configurações e calibrações, mesmo quando o dispositivo é desligado.

No quesito segurança, o ESP32 incorpora recursos como criptografia AES, RSA e ECC, para proteger dados sensíveis e garantir a integridade das comunicações. Mecanismos como Secure Boot e Flash Encryption são empregados para prevenir manipulações maliciosas e assegurar a autenticidade do firmware.

O ecossistema de desenvolvimento em torno do ESP32 é robusto, com suporte eficiente em ambientes como Arduino IDE, PlatformIO e Espressif IDF (framework

de desenvolvimento oficial da Espressif). A disponibilidade de uma ampla variedade de bibliotecas e exemplos simplifica o desenvolvimento de aplicações, acelerando a implementação de soluções inovadoras.

Em suma, o ESP32 (figura 2.1) se destaca como uma plataforma de alto desempenho para uma vasta gama de aplicações em sistemas embarcados e IoT. Sua combinação de poder de processamento, conectividade avançada, recursos de segurança e facilidade de desenvolvimento o consolida como uma escolha estratégica para projetos de engenharia e inovação tecnológica.

[4]

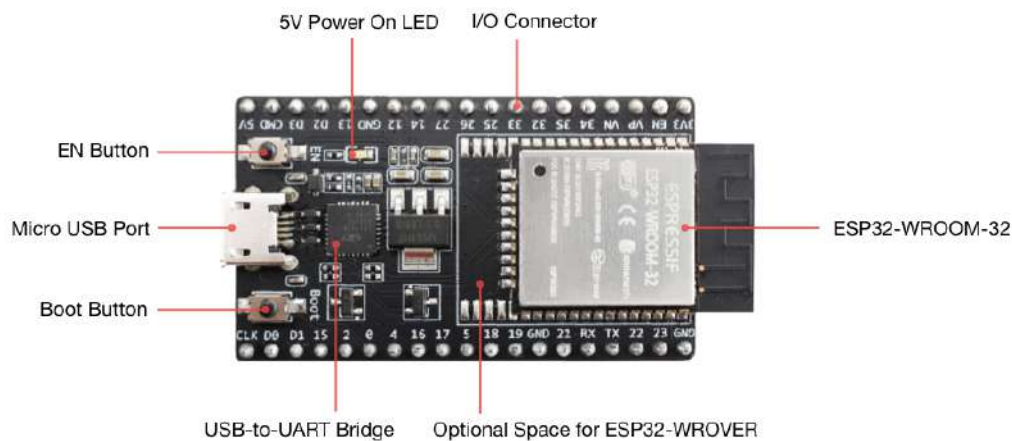


Figura 2.1: Kit de Desenvolvimento ESP32-DevKitC contendo o Microcontrolador ESP32

Arduino

A plataforma Arduino é reconhecida mundialmente por sua acessibilidade, flexibilidade e facilidade de uso no desenvolvimento de projetos de eletrônica e sistemas embarcados. Com uma ampla gama de placas e módulos disponíveis, o Arduino oferece uma abordagem simplificada para prototipagem rápida e implementação de soluções inovadoras.

As placas Arduino utilizam microcontroladores como o ATmega328P (presente no popular Arduino Uno), ATmega2560 (no Arduino Mega) e outros da família AVR, além de microcontroladores baseados em ARM, como o SAMD21 (no Arduino Zero). Essa variedade de microcontroladores proporciona um ambiente de desenvolvimento amigável para usuários de diversos níveis de experiência, atendendo a diferentes necessidades de processamento e recursos.

A conectividade é um aspecto fundamental da plataforma Arduino, com opções que variam desde USB para comunicação com o computador até módulos Wi-Fi e Bluetooth para conectividade sem fio. Essa versatilidade torna o Arduino adequado para uma ampla gama de aplicações, desde projetos simples de automação residencial até sistemas de monitoramento e controle complexos.

Em termos de desenvolvimento de software, o Arduino IDE oferece uma interface intuitiva para escrever e carregar programas nos microcontroladores das placas Arduino. Além disso, a extensa biblioteca padrão e a vasta comunidade de desenvolvedores contribuem para a disponibilidade de recursos e exemplos que facilitam o processo de desenvolvimento.

A plataforma Arduino também se destaca pela sua adaptabilidade e escalabilidade. Com diferentes modelos de placas, como Arduino Uno, Mega, Nano e mais, os usuários têm a flexibilidade de escolher a placa que melhor atende às necessidades de seus projetos. Além disso, a capacidade de expansão por meio de shields e módulos adicionais amplia ainda mais as possibilidades de aplicação.

No contexto da educação e aprendizado, o Arduino desempenha um papel significativo, proporcionando uma plataforma acessível para estudantes e entusiastas explorarem conceitos de eletrônica, programação e automação. Sua abordagem hands-on e a vasta quantidade de recursos educacionais disponíveis contribuem para a disseminação do conhecimento técnico e estimulam a criatividade e inovação.

Em resumo, a plataforma Arduino (figura 2.2) representa uma ferramenta valiosa para a prototipagem e implementação de projetos de eletrônica e sistemas embarcados. Sua combinação de acessibilidade, conectividade, facilidade de uso e vasta comunidade de suporte a torna uma escolha popular entre estudantes, hobbistas e profissionais da área de tecnologia.

[5]

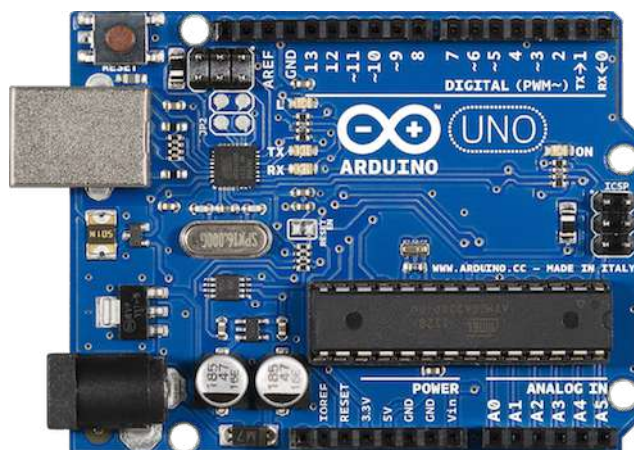


Figura 2.2: Kit de Desenvolvimento Arduino Uno R3

STM32

O microcontrolador STM32, desenvolvido pela STMicroelectronics, é uma poderosa plataforma de computação embarcada amplamente utilizada em uma variedade de aplicações, desde sistemas industriais até dispositivos IoT avançados. Com sua arquitetura baseada em núcleos ARM Cortex-M, o STM32 oferece um equilíbrio excepcional entre desempenho, eficiência energética e recursos integrados.

A arquitetura ARM Cortex-M presente nos microcontroladores STM32 proporciona uma alta capacidade de processamento, permitindo a execução eficiente de tarefas complexas e a gestão de sistemas em tempo real. A variedade de núcleos disponíveis, como Cortex-M0, M3, M4 e M7, oferece opções adaptadas às necessidades específicas de cada aplicação, desde projetos de baixo consumo até sistemas de alto desempenho.

A conectividade é um ponto forte dos microcontroladores STM32, com suporte a interfaces como UART, SPI, I2C, USB e Ethernet, além de opções de conectividade sem fio, como Wi-Fi e Bluetooth, por meio de módulos externos ou integrados em determinados modelos. Essa capacidade de comunicação ampla e flexível é fundamental para a integração em redes locais e globais, bem como para a interação com dispositivos externos.

Os periféricos integrados nos STM32 são abrangentes e incluem GPIOs, ADCs, DACs, timers, PWMs, e muitos outros, proporcionando uma ampla gama de funcionalidades para controle de dispositivos e sensores. Além disso, a presença de memória flash e RAM em diversos tamanhos permite o armazenamento de código, dados e configurações de forma eficiente.

A segurança é uma preocupação central nos STM32, com recursos como criptografia de *hardware*, mecanismos de proteção de memória e opções avançadas de autenticação e gerência de chaves. Esses recursos são essenciais para garantir a integridade, confidencialidade e autenticidade dos dados em aplicações sensíveis.

A plataforma de desenvolvimento para STM32 oferece suporte em diversos ambientes, incluindo o STM32CubeIDE, Keil µVision, e outros, facilitando o desenvolvimento de *firmware* e aplicativos. Além disso, a disponibilidade de bibliotecas e exemplos de código acelera o processo de desenvolvimento e depuração de sistemas baseados em STM32.

Em resumo, os microcontroladores STM32 representam uma escolha sólida e versátil para uma ampla variedade de aplicações embarcadas e IoT. Sua combinação de desempenho, conectividade, segurança e facilidade de desenvolvimento os torna uma plataforma preferida por desenvolvedores e engenheiros em todo o mundo. A figura 2.3 ilustra uma placa de desenvolvimento com microcontrolador STM32

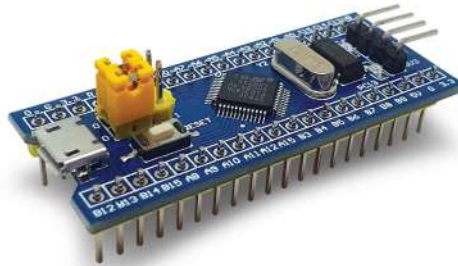


Figura 2.3: Placa de desenvolvimento ARM STM32F103C8T6

2.2 Sensores Ultrassônicos

A presente secção, dedicada aos sensores ultrassônicos oferece uma análise detalhada sobre esses dispositivos essenciais em aplicações de medição de distância, detecção de objetos e controle de nível de líquidos. A classificação dos sensores, incluindo tipos e funcionalidades, é abordada inicialmente para estabelecer uma compreensão abrangente de suas capacidades e aplicações. Em seguida, explora-se os princípios de funcionamento fundamentais por trás dos principais modelos, como o Sensor de Distância Ultrassônico HC-SR04 e o Módulo Sensor Ultrassônico Impermeável JSN-SR04M, destacando suas características distintas e usos típicos. Adicionalmente, são discutidas as aplicações específicas para cada modelo, os desafios enfrentados em ambientes hostis e a importância da escolha do sensor adequado para garantir desempenho confiável em diferentes cenários operacionais.

2.2.1 Classificação dos Sensores Ultrassônicos

Os sensores ultrassônicos são classificados de acordo com sua função e capacidades, sendo amplamente utilizados em diversas aplicações. Uma das classificações mais comuns é baseada no tipo de sensor e em sua funcionalidade específica.

Sensores de Medição de Distância

Existem sensores ultrassônicos projetados para medição de distância, que utilizam ondas sonoras de alta frequência para calcular a distância até um objeto com base no tempo de ida e volta do sinal. Esses sensores são frequentemente empregados em sistemas de navegação de robôs, detecção de obstáculos em veículos autônomos e controle de acesso em ambientes industriais.

Sensores de Detecção de Objetos

Outra categoria de sensores ultrassônicos é voltada para a detecção de objetos ou também chamados sensores de presença. Esses sensores são capazes de identificar a presença ou ausência de objetos em determinada área, sendo utilizados em sistemas de segurança, alarmes de proximidade e controle de presença em ambientes comerciais e residenciais.

2.2.2 Princípios de Funcionamento

Os sensores ultrassônicos operam com base nos princípios da eco-localização, também conhecidos como sonar, onde ondas sonoras são emitidas e o tempo de retorno do eco é utilizado para calcular a distância até o objeto. Esse processo envolve várias etapas fundamentais que explicam o funcionamento dos sensores de forma detalhada [7].

Emissão e Recepção de Ondas Sonoras

O funcionamento de um sensor ultrassônico inicia com a emissão de ondas sonoras de alta frequência, geralmente acima do limiar de audição humana, tipicamente na faixa de 40 kHz a 1 MHz. A escolha da frequência de operação influencia tanto a resolução da medição quanto a capacidade de penetração em diferentes materiais. Frequências mais altas proporcionam maior resolução, mas podem ser mais facilmente atenuadas em meios densos. Essas ondas são emitidas através de um transdutor piezoelétrico, que converte sinais elétricos em ondas sonoras e vice-versa. Após a emissão, o sensor entra em modo de espera para captar o eco das ondas sonoras que são refletidas de volta ao sensor por objetos presentes no seu campo de detecção. O ângulo de abertura do feixe sonoro emitido é um fator crucial a ser considerado, pois determina a área de cobertura do sensor e influencia a detecção de objetos pequenos ou fora do eixo central.

Medição do Tempo de Ida e Volta

A característica central do sensor ultrassônico é calcular a distância até um objeto baseando-se no tempo de ida e volta do sinal sonoro. Ao receber o eco, o sensor mede

o intervalo de tempo entre o momento da emissão do sinal e o momento da recepção do eco utilizando um temporizador de alta resolução. Este intervalo é diretamente proporcional à distância percorrida pelo sinal, considerando que a velocidade do som no ar é aproximadamente 343 metros por segundo à temperatura ambiente [8]. No entanto, é importante ressaltar que a velocidade do som pode variar com a temperatura e a umidade do ar, o que pode afetar a precisão das medições.

Cálculo da Distância

A distância até o objeto é calculada utilizando a seguinte fórmula (2.3):

$$\text{Distância} = \frac{\text{Velocidade do Som} \times \text{Tempo de Ida e Volta}}{2} \quad (2.3)$$

onde a *Velocidade do Som* no ar é aproximadamente 343 metros por segundo à temperatura ambiente, e o *Tempo de Ida e Volta* é o tempo que leva para as ondas sonoras viajarem até o objeto e retornarem ao sensor. A precisão da medição de distância depende da resolução do temporizador interno do sensor e da estabilidade da velocidade do som no ambiente. Sensores de alta qualidade podem alcançar precisões da ordem de milímetros.

Considerações sobre o Ambiente

A precisão dos sensores ultrassônicos pode ser afetada por várias condições ambientais, como temperatura, humidade e propriedades do material do objeto. Variações na temperatura do ar, por exemplo, podem alterar a velocidade do som, afetando a precisão das medições de distância. A presença de obstáculos ou superfícies irregulares no caminho do feixe sonoro também pode causar reflexões indesejadas e interferir na detecção do objeto. Os desenvolvedores desses sistemas precisam considerar esses fatores ao implementar e calibrar sensores ultrassônicos em aplicações práticas [9].

Calibração e Precisão

A calibração dos sensores ultrassônicos é crucial para garantir a precisão das medições, especialmente em ambientes com variações de temperatura ou humidade. Geralmente, isso envolve ajustar o sensor para um objeto de referência a uma distância conhecida e comparar a leitura do sensor com a distância real. O processo pode ser repetido em diferentes distâncias para mapear a resposta do sensor e compensar qualquer desvio. Em alguns casos, pode ser necessário ajustar parâmetros internos do sensor, como o ganho do recetor ou a compensação de temperatura.

A precisão dos sensores ultrassônicos é influenciada por diversos fatores, incluindo a qualidade do transdutor, a frequência de operação, a resolução do temporizador interno e as condições ambientais. Em geral, sensores de alta qualidade

podem alcançar precisões da ordem de milímetros. No entanto, é importante lembrar que a precisão pode ser afetada por fatores externos, como a presença de ruído eletromagnético, vibrações mecânicas e variações na temperatura e umidade do ar.

2.2.3 Aplicações e Casos de Uso

Nesta seção, exploramos as aplicações práticas de dois modelos específicos de sensores ultrassônicos, detalhando seus contextos de uso e vantagens operacionais que os distinguem em cenários específicos. Essa abordagem destaca a utilidade prática do Sensor de Distância Ultrassônico HC-SR04 e do Módulo Sensor Ultrassônico Impermeável JSN-SR04M.

Sensor de Distância Ultrassônico HC-SR04

Características Técnicas:

- **Faixa de medição:** tipicamente de 2 cm até 400 cm.
- **Precisão:** pode variar de acordo com condições ambientais, mas geralmente é adequada para a maioria das aplicações de hobbyismo e educação.

Contextos de Uso Recomendados:

- **Projetos Educacionais:** Devido ao seu custo-benefício e facilidade de uso, é ideal para introduzir conceitos de física e engenharia em salas de aula.
- **Prototipagem Rápida:** Utilizado em *makerspaces* para desenvolver protótipos que requerem medição de distância ou detecção de proximidade.

[7]

Módulo Sensor Ultrassônico Impermeável JSN-SR04M

Características Técnicas:

- **Impermeabilidade:** Projetado para resistir a condições de umidade e submersão temporária.
- **Faixa de medição ampliada:** Funciona efetivamente em distâncias de 25 cm até 450 cm.

Contextos de Uso Recomendados:

- **Monitoria Ambiental:** Eficiente em aplicações externas como o monitoria de rios ou reservatórios para controle de enchentes.
- **Indústria Naval:** Adequado para integrar sistemas de medição de profundidade em embarcações.

[9]

2.2.4 Desafios e Limitações

Os sensores ultrassônicos enfrentam vários desafios e limitações que podem comprometer sua eficácia em determinadas condições e aplicações. Abaixo, detalhamos esses desafios em subsecções, discutindo aspectos específicos como ambientes hostis, resistência à água, durabilidade e consumo de energia.

Operação em Ambientes Hostis

A operação em ambientes hostis, que incluem altas temperaturas, pressões extremas e ruídos acústicos intensos, pode afetar adversamente a integridade do sinal ultrassônico, comprometendo a precisão das medições [10].

Resistência à Água e Humidade

Os sensores ultrassônicos são frequentemente suscetíveis a danos causados por água e humidade, que podem corroer componentes eletrônicos e degradar o transdutor, resultando em falhas de funcionamento e redução da vida útil [11].

Durabilidade e Vida Útil

A durabilidade é um fator crítico, afetada por fatores como vibrações mecânicas e exposição a ambientes corrosivos. Isso pode exigir manutenções frequentes ou substituições prematuras dos sensores [12].

Consumo de Energia

O consumo de energia é uma preocupação em aplicações que exigem operação contínua dos sensores, especialmente em locais remotos onde a troca frequente de baterias não é viável, aumentando os custos operacionais [13].

Limitações de Detecção

Sensores ultrassônicos podem falhar ao detetar objetos feitos de materiais que absorvem som ou que são muito pequenos, além de serem sensíveis à posição e angulação do objeto detetado, o que pode resultar em medições imprecisas [14].

2.3 Comunicação Sem Fio

A comunicação sem fio é um pilar fundamental para a IoT, permitindo a interconexão e interação entre dispositivos sem a necessidade de infraestruturas físicas tradicionais. Esta secção detalha os principais protocolos e padrões, explorando suas características teóricas, aplicações práticas e desafios específicos.

2.3.1 Protocolos e Padrões de Comunicação Sem Fio

Diversos protocolos foram desenvolvidos para atender às necessidades variadas da IoT, cada um com características específicas voltadas para eficiência energética, alcance de comunicação ou capacidade de transmissão de dados.

Bluetooth Low Energy (BLE)

O BLE é uma tecnologia de comunicação de curto alcance projetada para aplicações que requerem trocas de dados com baixo consumo de energia. BLE utiliza a banda de 2.4 GHz e emprega técnicas de salto de frequência adaptativo (AFH) para minimizar interferências. É eficaz para dispositivos de saúde, acessórios vestíveis e sensores de ambiente, operando com baixo custo e eficiência energética. A figura 2.4 ilustra um módulo BLE. [15]

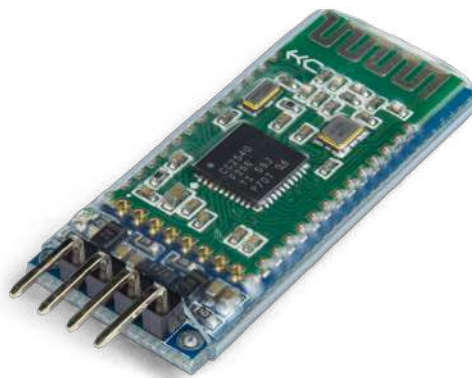


Figura 2.4: Módulo Bluetooth Low Energy - HC-08

Zigbee

Utilizando o padrão IEEE 802.15.4 na camada física e MAC, Zigbee suporta redes *mesh* com baixo consumo de energia, ideal para automação residencial e industrial. Sua estrutura permite que dispositivos se comuniquem em distâncias consideráveis sem necessidade de um ponto central, alcançando taxas de dados de até 250 kbps. A Figura 2.5 apresenta um módulo Zigbee. [16]



Figura 2.5: Módulo XBee S2 S2C Zigbee

LoRa e LoRaWAN

LoRa é uma tecnologia de modulação de rádio de longo alcance que utiliza espectro espalhado chirp para alcançar alta sensibilidade e robustez a interferências. Essa técnica permite que os dispositivos LoRa transmitam dados a longas distâncias com baixo consumo de energia, mesmo em ambientes com ruído ou obstruções.

LoRaWAN, por sua vez, é um protocolo de rede de área ampla de baixa potência (LPWAN) que se baseia na tecnologia LoRa para estabelecer uma arquitetura de comunicação eficiente e escalável para dispositivos IoT. Ele define a camada de rede, gerenciando o endereçamento, a segurança e a otimização do consumo de energia dos dispositivos. LoRaWAN opera em uma topologia estrela-de-gateways, onde os dispositivos (nós) se comunicam com um ou mais gateways que, por sua vez, se conectam a um servidor de rede central. Essa arquitetura permite a cobertura de grandes áreas com um número relativamente pequeno de gateways, tornando-a ideal para aplicações em larga escala, como cidades inteligentes, agricultura e monitoramento ambiental.

A combinação da tecnologia LoRa com o protocolo LoRaWAN oferece uma solução completa para a conectividade de dispositivos IoT, proporcionando longo alcance (até 15 km em áreas rurais), baixo consumo de energia, alta capacidade de penetração em obstáculos e escalabilidade para suportar um grande número de dispositivos. A Figura 2.6 mostra um módulo LoRa, que pode ser utilizado tanto em redes LoRaWAN quanto em outras aplicações que utilizem a tecnologia LoRa.



Figura 2.6: Módulo Lora SX1276

Narrowband IoT (NB-IoT)

Focado em aplicações que requerem transmissões ocasionais de pequenas quantidades de dados, o NB-IoT opera dentro das bandas de frequência LTE existentes, proporcionando uma integração eficiente e custo-benefício em ambientes urbanos densos. Ele utiliza uma largura de banda estreita de 180 kHz e suporta baixas taxas de dados, o que o torna ideal para aplicações como medidores inteligentes e rastreamento de ativos [17]. A Figura 2.7 apresenta um módulo NB-IoT.



Figura 2.7: Módulo NB-IOT 4 CLICK

SigFox

Operando na banda ultra-estreita (UNB), SigFox permite comunicações de longo alcance com mensagens curtas e baixo consumo de energia. Ele utiliza uma técnica de modulação diferencial binária de fase mínima (DBPSK) e opera em bandas de frequência ISM licenciadas. É usado predominantemente em aplicações que necessitam de mínima manutenção e comunicação infrequente, como sensores remotos [18]. A Figura 2.8 ilustra um módulo SigFox.



Figura 2.8: Módulo SIGFOX-MOD1-C

2.3.2 Alcance Estendido e Tecnologias *Long Range*

Em cenários onde a comunicação precisa ser eficiente e robusta em grandes distâncias, como em aplicações IoT de larga escala ou em ambientes desafiadores, as tecnologias de longo alcance se destacam como soluções promissoras.

Comparação de Alcance e Eficiência

LoRa, NB-IoT e Sigfox oferecem diferentes capacidades de alcance e eficiência energética, adaptadas para cenários de uso específicos. LoRa e Sigfox são mais adequados para ambientes rurais ou suburbanos devido ao seu alcance superior, enquanto o NB-IoT é ideal para uso urbano devido à sua integração com infraestrutura celular existente [19]. A tabela 2.1 apresenta uma comparação detalhada entre essas tecnologias.

Tabela 2.1: Comparação entre LoRa, NB-IoT e SigFox

Característica	LoRa	NB-IoT	SigFox
Alcance (típico)	Até 15 km (rural) / 5 km (urbano)	Até 10 km (rural) / 1 km (urbano)	Até 10 km (rural) / 3 km (urbano)
Taxa de dados	0.3 kbps a 50 kbps	100 bps a 100 kbps	100 bps a 600 bps
Consumo de energia	Muito baixo	Baixo	Muito baixo
Latência	Alta (segundos a minutos)	Média (centenas de milissegundos)	Alta (segundos a minutos)
Custo de implementação	Baixo a moderado	Moderado a alto	Baixo

2.3.3 Custo e Disponibilidade em São Paulo

A escolha do protocolo de comunicação em São Paulo deve considerar tanto a cobertura de rede quanto o custo associado à implementação e operação.

LoRa em São Paulo

LoRa é particularmente vantajoso em São Paulo devido ao seu longo alcance e baixo consumo de energia. A tecnologia LoRa permite a implementação de redes privadas ou públicas utilizando *gateways* que podem comunicar-se com múltiplos dispositivos em distâncias significativas, o que é ideal para a vasta metrópole de São Paulo.

Implementação de Redes LoRa A implementação de redes LoRa em São Paulo pode ser realizada através de *gateways* instalados em pontos estratégicos, que cobrem áreas extensas, desde centros urbanos até regiões periféricas. Estes *gateways* são capazes de receber e enviar dados de e para centenas de dispositivos simultaneamente, tornando a tecnologia ideal para aplicações de cidade inteligente, monitoria ambiental e gestão de serviços públicos.

Vantagens da LoRa As principais vantagens da utilização da LoRa em São Paulo incluem:

Alcance Extensivo: A tecnologia LoRa pode alcançar até 15 km em áreas rurais e cerca de 5 km em ambientes urbanos densos, como São Paulo, dependendo da topologia e da elevação dos *gateways*.

Redução de Custos: Ao utilizar LoRa, as organizações podem reduzir significativamente os custos operacionais, pois a tecnologia requer menos infraestrutura de rede em comparação com outras tecnologias celulares.

Flexibilidade de Implementação: LoRa oferece flexibilidade para configurar redes privadas que podem operar independentemente das redes celulares tradicionais, proporcionando controle total sobre a infraestrutura de comunicação.

A escolha de LoRa como uma alternativa viável para projetos de IoT em São Paulo pode ser fundamentada pelas suas características únicas que oferecem uma solução robusta e econômica para a conectividade em larga escala [20].

Desafios No entanto, a implementação de LoRa enfrenta desafios como a necessidade de planejamento cuidadoso da localização dos *gateways* para maximizar a cobertura e a eficiência, e a gestão de espectro, uma vez que LoRa opera em bandas de frequência que podem estar sujeitas a interferências em ambientes urbanos densamente povoados.

2.3.4 Modelos de Módulos de Comunicação Sem Fio

Os módulos de comunicação são essenciais para a implementação prática de soluções IoT, com vários modelos disponíveis que atendem a diferentes necessidades tecnológicas e de aplicação.

Modelos Comuns e Suas Aplicações

O SX1276 é amplamente utilizado para aplicações LoRa devido à sua eficiência energética e longo alcance, como pode ser visto na Figura 2.6. Placas como o ESP32 com módulo LoRa SX1276 integrado, ilustrado na Figura 2.9, oferecem funcionalidades adicionais como Wi-Fi e Bluetooth, ideal para desenvolvimento rápido de protótipos e produtos IoT [21].



Figura 2.9: Placa ESP32 LoRa SX1276 868/915Mhz com OLED

2.4 Gateways

Os *gateways* IoT são componentes críticos em sistemas de Internet das Coisas, atuando como intermediários que facilitam a comunicação entre dispositivos IoT e a nuvem ou outros sistemas de processamento de dados. Eles não apenas gerenciam o tráfego de dados, mas também realizam funções essenciais de segurança, análise de dados e tradução de protocolos.

2.4.1 Classificação de *Gateways* IoT

Gateways podem ser classificados em várias categorias, dependendo de suas funcionalidades e características:

- ***Gateways* de Agregação:** Agregam dados de diversos sensores e dispositivos antes de enviar para processamento posterior, reduzindo a carga de transmissão e melhorando a eficiência [22].
- ***Gateways* de Protocolo:** Realizam a tradução entre diferentes protocolos de comunicação IoT, como entre Zigbee e Wi-Fi, ou entre Bluetooth e LTE.
- ***Gateways* Inteligentes:** Equipados com capacidades avançadas de processamento, esses *gateways* podem executar análises preliminares e tomar decisões autônomas com base nos dados recebidos [23].

2.4.2 Protocolos de Comunicação para Gateways

Os *gateways* utilizam diversos protocolos para garantir uma comunicação eficaz e segura:

- **Message Queuing Telemetry Transport (MQTT):** Um protocolo leve de mensageria baseado em publicação/assinatura, ideal para ambientes com largura de banda limitada.
- **Constrained Application Protocol (CoAP):** Projetado para redes simples e dispositivos com capacidades limitadas, oferecendo uma alternativa eficiente ao *Hypertext Transfer Protocol* (HTTP).
- **HTTP e Message Queuing Telemetry Transport for Sensor Networks (MQTT-SN):** Enquanto o HTTP é amplamente usado para integração com a web, o MQTT-SN é adaptado para redes de sensores sem fio.

2.4.3 Arquitetura e Funcionamento dos Gateways

A arquitetura de um *gateway* IoT é complexa e desenhada para suportar múltiplas funções:

- **Interfaces de Comunicação:** Suportam uma variedade de padrões de comunicação e conectividade, de Wi-Fi e Ethernet a LTE e Zigbee.
- **Processamento e Armazenamento:** Capacidades integradas permitem a análise de dados em tempo real e o armazenamento temporário de informações críticas [24].
- **Módulos de Segurança:** Implementam funcionalidades de segurança avançadas para proteger os dados contra acessos não autorizados e ataques cibernéticos [25].

Essa estrutura multi-funcional permite que os *gateways* desempenhem um papel vital na orquestração da comunicação entre o ambiente IoT e sistemas de maior escala, facilitando uma rede mais inteligente e responsiva.

2.5 Soluções Avançadas de Cloud para IoT

As soluções de *cloud* são essenciais para uma boa gestão do volume massivo de dados gerados pelos dispositivos IoT. Elas não só proporcionam o armazenamento de dados em larga escala, mas também viabilizam processamento avançado, análise detalhada e visualização em tempo real dos dados e alertas, otimizando a tomada de decisão e a automação em ambientes IoT.

2.5.1 Arquiteturas de Nuvem Adaptadas para IoT

Diversas arquiteturas de nuvem são projetadas para suportar aplicações IoT, cada uma oferecendo funcionalidades específicas para atender requisitos de escalabilidade, desempenho e segurança distintos. Estas incluem:

- **Nuvem Pública:** Plataformas como AWS IoT Core, Google Cloud IoT e Microsoft Azure IoT Hub são exemplos proeminentes de serviços gerenciados que escalonam automaticamente para acomodar milhões de dispositivos. Eles fornecem infraestruturas robustas que suportam comunicações massivas e simultâneas, essenciais para operações IoT em grande escala [26].
- **Nuvem Privada:** Configurações personalizadas oferecem controle exclusivo sobre os recursos de computação, resultando em maior segurança e conformidade, ideal para organizações que lidam com dados sensíveis ou têm requisitos regulatórios específicos.
- **Nuvem Híbrida:** Esta arquitetura combina os elementos de nuvens públicas e privadas, permitindo um balanceamento eficaz entre controle, flexibilidade e otimização de custos, tornando-a adequada para empresas que necessitam de adaptabilidade em suas operações de IoT.

2.5.2 Plataformas e Serviços de Nuvem Especializados para IoT

Principais plataformas de nuvem projetadas especificamente para IoT incluem:

- **AWS IoT Core:** Facilita a conexão segura de dispositivos IoT à nuvem, suportando protocolos como MQTT, WebSockets e HTTP, o que permite uma vasta gama de opções de comunicação e integração [27].
- **Microsoft Azure IoT Hub:** Destaca-se pela sua capacidade de comunicação bidirecional, permitindo não apenas o envio de dados para a nuvem, mas também o envio de comandos e políticas de atualizações para os dispositivos.
- **Google Cloud IoT Core:** Oferece ferramentas robustas para conexão, gestão e análise de dados em larga escala, integrando-se perfeitamente com outras soluções do Google Cloud para processamento de dados e *Machine Learning*.
- **IBM Watson IoT:** Proporciona poderosas ferramentas analíticas e de aprendizado de máquina, ideal para transformar grandes volumes de dados em *insights* acionáveis e melhorar continuamente os processos IoT através de inteligência artificial.
- **ThingSpeak:** Permite a integração e análise de dados IoT com suporte para *scripts* MATLAB, facilitando a realização de análises complexas e visualização de dados em tempo real.

2.5.3 Segurança e Privacidade de Dados em Ambientes IoT de Nuvem

A segurança e a privacidade são críticas para soluções de *cloud* para IoT devido ao acesso e controle remoto sobre dispositivos e dados:

- **Criptografia:** Implementação de protocolos de criptografia de última geração tanto para dados em trânsito como em repouso, garantindo que os dados sensíveis estejam protegidos contra acessos não autorizados.
- **Autenticação:** Utilização de sistemas de autenticação multi-fator e gerência de identidade para assegurar que apenas usuários e dispositivos autenticados tenham acesso às redes e dados IoT.
- **Privacidade:** Adoção de políticas rigorosas e mecanismos de conformidade que estão em linha com as normas globais de proteção de dados, como o *General Data Protection Regulation* (GDPR), para garantir a privacidade dos dados do usuário [28].

2.5.4 Escalabilidade e Análise de Custos em Soluções de Cloud IoT

A escalabilidade e a eficiência de custos são fundamentais para o sucesso e a sustentabilidade de projetos IoT:

- **Escalabilidade:** As plataformas devem ser capazes de se adaptar dinamicamente ao aumento no número de dispositivos conectados e ao volume crescente de dados, garantindo desempenho e estabilidade sem comprometer a qualidade do serviço.
- **Custos:** Análises de custo-benefício são essenciais para entender o investimento inicial necessário e os custos operacionais contínuos, permitindo uma gestão financeira eficiente e planejamento estratégico.

2.5.5 ThingSpeak

ThingSpeak é uma plataforma de computação em nuvem de código aberto, projetada para aplicações de Internet das Coisas (IoT). Ela oferece ferramentas para coleta, armazenamento, análise e visualização de dados em tempo real, com recursos de automação baseados em eventos. A integração nativa com MATLAB permite realizar análises complexas e criar visualizações interativas, tornando-a uma ferramenta poderosa para projetos de IoT.

Arquitetura e Funcionamento

A arquitetura do ThingSpeak segue o modelo cliente-servidor, onde dispositivos IoT enviam dados para a plataforma em nuvem via protocolos como HTTP, MQTT ou CoAP. Esses dados são armazenados em um banco de dados relacional e podem ser acessados por uma interface web ou APIs RESTful. A plataforma suporta vários hardwares, como Arduino, Raspberry Pi e ESP32, facilitando a integração com diferentes dispositivos.

Recursos e Funcionalidades

ThingSpeak oferece diversos recursos para simplificar o desenvolvimento de aplicações IoT:

- **Canais:** Mecanismo principal de armazenamento de dados, com até 8 campos por canal. A versão gratuita oferece até 4MB, enquanto a paga permite até 16GB.
- **Análise com MATLAB:** Permite análises avançadas diretamente na plataforma, possibilitando a criação de modelos preditivos e a detecção de anomalias em tempo real.
- **Visualizações:** Diversas opções de gráficos e mapas, personalizáveis e incorporáveis em páginas web ou painéis de controle.
- **Apps:** Aplicativos pré-construídos para tarefas como alertas, integração com redes sociais e acionamento de dispositivos.
- **TimeControl:** Agendamento de ações e controle de dispositivos com base em horários ou eventos.
- **TalkBack:** Comunicação bidirecional para enviar comandos e receber respostas dos dispositivos IoT.

Modelos de Uso e Aplicações

ThingSpeak é versátil e aplicável em diversos cenários, como:

- **Monitoramento ambiental:** Coleta e análise de dados de sensores para monitoramento ambiental.
- **Automação residencial e industrial:** Controle de sistemas em residências e indústrias.
- **Agricultura de precisão:** Monitoramento de condições agrícolas para otimizar a produção.

- **Saúde e bem-estar:** Análise de dados de dispositivos vestíveis e sensores de saúde.
- **Cidades inteligentes:** Gestão de infraestrutura urbana, como tráfego e segurança.

Vantagens e Desvantagens

As principais vantagens do ThingSpeak incluem facilidade de uso, integração com MATLAB, suporte a múltiplos protocolos e recursos de análise e visualização. As limitações incluem o armazenamento restrito na versão gratuita e a dependência da nuvem, o que pode levantar preocupações com privacidade e segurança.

Conclusão

ThingSpeak é uma plataforma robusta para o desenvolvimento de aplicações IoT, oferecendo ferramentas abrangentes para coleta, análise e visualização de dados em tempo real. Sua integração com MATLAB e usabilidade a tornam uma opção atraente para projetos de IoT em diversos setores, desde soluções simples até complexas e escaláveis.

Capítulo 3

Arquitetura da Aplicação

Este capítulo detalha a arquitetura do sistema desenvolvido como solução para monitorar o nível dos bueiros por meio de sensores ultrassônicos, com o objetivo de prevenir alagamentos urbanos causados pelo escoamento inadequado de água da chuva. Utilizando a tecnologia IoT, o sistema proposto é capaz de capturar dados em tempo real, processá-los e transmiti-los para uma plataforma de visualização e alerta. A arquitetura é projetada para ser escalável, permitindo a expansão do número de bueiros monitorados através de uma rede de comunicação LoRa. A descrição inclui os dispositivos de hardware e software utilizados, justificativas e topologia do projeto.

3.1 Hardware Utilizado

O hardware escolhido desempenha um papel fundamental no desenvolvimento do sistema de monitoria inteligente para bueiros. Esta seção detalha o módulo ESP32/LoRa e o sensor ultrassônico, destacando as características que os tornam adequados para este projeto.

3.1.1 Módulo ESP32/LoRa

O módulo utilizado é a Placa WiFi LoRa 32 da Heltec Automation, que integra múltiplas funcionalidades essenciais para aplicações de IoT. Este módulo é baseado no chip ESP32-D0WDQ6, combinando capacidades de Wi-Fi e Bluetooth Low Energy

com a tecnologia de comunicação LoRa do chip SX1276 operando na frequência de 915MHz.

Facilidade de Programação e Recursos Avançados

O ESP32 é notável pela sua facilidade de programação, suportando uma ampla variedade de ambientes de desenvolvimento. O processador Xtensa 32-bit LX6 dual-core permite operações de 80 a 240 MHz, oferecendo um excelente equilíbrio entre desempenho e consumo de energia.

Baixo Consumo e Alta Eficiência

O baixo consumo energético do ESP32 é ideal para a longa duração de operações autônomas necessárias em ambientes urbanos. O design inclui um circuito de carga e descarga para baterias de lítio, facilitando a manutenção e a operacionalidade prolongada do sistema.

Integração de Micro-controlador e Módulo LoRa com Display OLED

A integração do *display* OLED de 0,96 polegadas permite a visualização de dados diretamente no dispositivo, uma vantagem significativa durante a fase de desenvolvimento e testes, proporcionando uma interface homem-máquina eficiente e direta.

Vantagens da Escolha de um Kit Integrado

A decisão de escolher um kit que já integra o micro-controlador, o módulo LoRa e o *display* OLED, ao invés de componentes separados, foi motivada principalmente pela economia de tempo e pelo desejo de criar um protótipo mais organizado. Esta abordagem elimina a necessidade de soldagem complexa e gerência de múltiplos componentes e conexões, reduzindo significativamente o risco de erros e problemas técnicos. Além disso, a utilização de um kit integrado assegura um *layout* mais limpo e compacto, facilitando tanto o desenvolvimento quanto os testes subsequentes.

3.1.2 Sensor Ultrassônico JSN-SR04T

Para a detecção de obstruções e monitoramento dos níveis de água nos bueiros, foi escolhido o sensor de distância ultrassônico JSN-SR04T. Este sensor, encapsulado em uma carcaça à prova d'água com classificação IP67, é especialmente adequado para o projeto devido às suas características que favorecem a utilização em ambientes urbanos desafiadores, como os encontrados em sistemas de drenagem.

Características e Especificações Técnicas

O JSN-SR04T opera com base no princípio da ecolocalização, emitindo pulsos ultrassônicos com frequência central de 40 kHz e medindo o tempo de voo do eco refletido por um objeto para determinar a distância. A faixa de detecção do sensor se estende de 25 cm a 2,50 metros, com uma precisão típica de ± 3 mm. Essa ampla faixa de detecção, aliada à sua alta resolução, permite monitorar com precisão os níveis de água e detectar obstruções em uma variedade de cenários, desde bueiros rasos até galerias profundas.

O sensor possui um ângulo de abertura do feixe de 75° , o que significa que ele cobre uma área cônica à medida que o sinal ultrassônico se propaga. Essa característica é vantajosa para aplicações em bueiros, pois permite uma cobertura adequada da área de interesse sem a necessidade de múltiplos sensores. No entanto, é importante considerar o ângulo de abertura ao posicionar o sensor para garantir que o feixe sonoro incida perpendicularmente à superfície da água ou do objeto a ser detectado, maximizando a intensidade do eco e a precisão da medição.

O JSN-SR04T é alimentado por uma tensão de 3,5 V a 5 V e consome uma corrente de operação de 40 mA durante a emissão do pulso ultrassônico. Em modo de espera, o consumo de corrente é significativamente menor, contribuindo para a eficiência energética do sistema. O sensor possui quatro pinos de conexão: VCC (alimentação), GND (terra), Trig (gatilho) e Echo (eco). O pino Trig é utilizado para iniciar a emissão do pulso ultrassônico, enquanto o pino Echo fornece um sinal de saída cuja duração é proporcional ao tempo de voo do eco.

Vantagens na Implementação do Projeto

A escolha do JSN-SR04T oferece diversas vantagens para o projeto de monitoramento de bueiros:

- **Resistência à Água (IP67):** A capacidade do sensor de operar em ambientes úmidos e em contato direto com a água é crucial para garantir sua durabilidade e confiabilidade em condições adversas, como as encontradas em bueiros durante períodos de chuva intensa.
- **Alcance Adequado:** A faixa de detecção de até 2,50 metros é adequada para as condições esperadas no ambiente urbano, onde os bueiros podem variar consideravelmente em profundidade e condições de enchimento. A distância mínima de detecção de 25 cm foi considerada suficiente para identificar situações de alerta, como o acúmulo de detritos ou o aumento do nível de água.
- **Instalação Flexível:** O sensor vem com um cabo de 2,5 metros, permitindo maior flexibilidade na instalação e no posicionamento do microcontrolador em

um local seguro e de fácil acesso, enquanto o sensor pode ser posicionado de forma otimizada para a medição.

- **Ângulo de Leitura Amplo:** Com um ângulo de leitura de 75° , o sensor oferece uma cobertura adequada para capturar dados precisos sobre o nível de água e a presença de obstruções, minimizando a necessidade de múltiplos sensores e reduzindo a complexidade do sistema.
- **Baixo Custo:** O JSN-SR04T é um sensor de baixo custo, o que o torna uma opção atraente para projetos de grande escala, como o monitoramento de múltiplos bueiros em uma cidade.
- **Facilidade de Integração:** A interface simples de quatro pinos do sensor facilita sua integração com micro-controladores e outras plataformas de hardware, agilizando o desenvolvimento e a implantação do sistema.

Integração com o Sistema e Calibração

A pinagem da placa de conversão do sensor (5 V / GND / Trigger / Echo) permite uma conexão direta e simplificada com o módulo ESP32, otimizando a montagem e a manutenção do sistema. A compatibilidade com a faixa de tensão de alimentação do ESP32 (3,3 V) e o baixo consumo de corrente do sensor garantem que ele possa ser alimentado diretamente pelo micro-controlador sem a necessidade de componentes adicionais.

A calibração do sensor é essencial para garantir a precisão das medições, especialmente em ambientes com variações de temperatura e umidade. O processo de calibração envolve a medição da distância a um objeto de referência em diferentes condições ambientais e o ajuste dos parâmetros do sensor, como o tempo de atraso entre a emissão do pulso e a leitura do eco, para compensar qualquer desvio. Essa calibração pode ser realizada utilizando um programa simples no ESP32 que compara as leituras do sensor com as distâncias reais e ajusta os parâmetros de acordo.

Com sua combinação de robustez, alcance adequado, facilidade de integração e baixo custo, o sensor ultrassônico JSN-SR04T se apresenta como uma escolha promissora para o desenvolvimento de um sistema de monitoramento de bueiros eficiente e confiável.

3.2 Software IoT

A escolha de um software adequado é crucial para o sucesso de qualquer sistema de monitoria baseado em IoT, especialmente quando se trata de aplicativos urbanos complexos como o monitoria de bueiros. Neste projeto, optou-se pela utilização da plataforma ThingSpeak, descrita na seção 2.5.5.

3.2.1 Escolha da Plataforma

Integração e Compatibilidade com Hardware IoT

ThingSpeak apresenta uma compatibilidade notável com uma ampla gama de dispositivos IoT, incluindo o módulo ESP32 utilizado neste projeto. Sua capacidade de receber e processar dados de múltiplos sensores simultaneamente, sem a necessidade de intervenção manual ou ajustes complexos, oferece uma integração praticamente nativa. Esta característica reduz significativamente o tempo de desenvolvimento e implementação, permitindo uma maior concentração em aspectos analíticos e de resposta do sistema. Além disso, foi encontrada uma diversa documentação detalhando as instruções para a conexão do ESP32 com o ThingSpeak.

Capacidades de Trigger Automatizado para Resposta Rápida

A plataforma ThingSpeak oferece funcionalidades avançadas de *triggers* automáticos que são essenciais para um sistema de monitoria de bueiros. A capacidade de configurar ações automáticas baseadas em condições específicas dos dados coletados permite uma resposta rápida e eficaz a situações críticas, como o rápido aumento do nível da água, que é uma condição frequentemente observada durante as chuvas intensas em áreas urbanas.

Suporte à Análise de Dados em Tempo Real

ThingSpeak fornece ferramentas robustas de análise e visualização de dados, fundamentais para monitorizar e prever tendências que possam indicar riscos iminentes de enchente. O suporte integrado para *scripts* MATLAB é particularmente valioso, permitindo a implementação de algoritmos complexos de processamento de dados diretamente na plataforma. Isso habilita o projeto a não apenas coletar dados, mas também interpretá-los de maneira inteligente para informar decisões de manejo e intervenção.

Custos

A plataforma se destacou significativamente como a melhor opção na fase de desenvolvimento devido a possibilidade de uso gratuito. Para uso em larga escala existe a possibilidade de adquirir licenças de acordo com o tamanho da aplicação.

3.2.2 Implementação no ThingSpeak

A implementação prática no ThingSpeak envolveu a configuração de canais específicos para cada tipo de dado coletado pelos sensores ultrassônicos. Foram estabelecidos *triggers* específicos para cada cenário de alerta, garantindo que o sistema possa

responder automaticamente e de maneira eficaz conforme as condições detetadas em campo.

Este enfoque na escolha do ThingSpeak como plataforma de IoT para o projeto foi fundamentado não apenas em suas capacidades técnicas, mas também na necessidade de um sistema robusto, escalável e capaz de oferecer respostas rápidas e baseadas em dados confiáveis para a prevenção e gestão de enchentes urbanas.

3.3 Topologia do Projeto

Esta secção descreve a arquitetura geral do sistema e apresenta um diagrama de blocos que ilustra a interconexão entre os elementos do sistema.

3.3.1 Descrição da Topologia

O sistema é projetado para maximizar a eficiência na coleta e transmissão de dados, garantindo ao mesmo tempo que as operações se mantenham robustas e confiáveis sob as condições urbanas variadas. A arquitetura é composta pelos seguintes elementos principais:

- **Sensores Ultrassônicos:** Localizados estrategicamente nos bueiros para medir o nível da água e detetar possíveis obstruções. Cada sensor está conectado a um módulo ESP32 com módulo LoRa SX1276.
- **Kit de desenvolvimento ESP32/Lora:** Responsáveis pela coleta de dados dos sensores ultrassônicos e transmissão via tecnologia LoRa na frequência de 915MHz.
- **Gateway LoRa:** Recebe dados transmitidos pelos módulos ESP32 e os encaminha por meio do módulo Wi-Fi para a plataforma ThingSpeak na nuvem.
- **Plataforma ThingSpeak:** Serve como o núcleo analítico do sistema, onde os dados são processados, analisados e visualizados. Podendo receber alertas quando a distância detetada pelo sensor ultrassônico for alarmante.

3.3.2 Diagrama de Blocos

O diagrama de blocos a seguir ilustra a topologia descrita, destacando as conexões e fluxos de comunicação entre os diversos componentes do sistema.

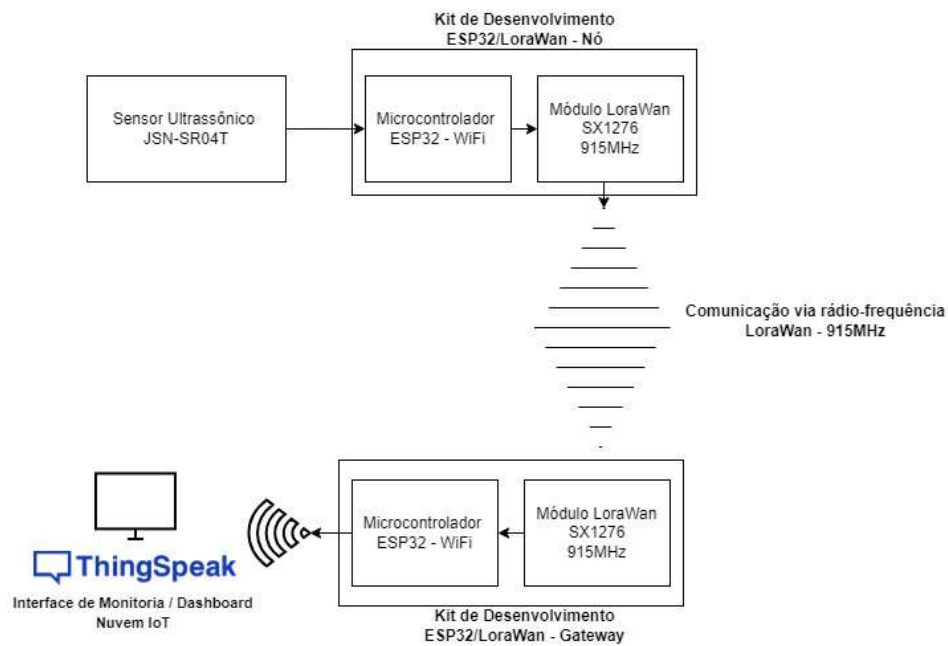


Figura 3.1: Arquitetura Inicial do Projeto

3.4 Análise dos Custos do Projeto

A análise de custos é essencial para avaliar a viabilidade económica do sistema de monitoria de bueiros utilizando tecnologia IoT. Esta secção compara os custos dos componentes principais escolhidos para o projeto com os de seus principais concorrentes no mercado. As tabelas a seguir estão preparadas para incluir informações sobre os preços e lojas onde os componentes podem ser adquiridos.

3.4.1 Custos do Módulo ESP32/Lora

A tabela abaixo lista o kit ESP32/Lora utilizado no projeto, juntamente com o preço individual de compra do ESP32 e do módulo LoRa.

Fornecedor	Kit ESP32/LoRa	Placa ESP32	LoRa SX1276
Saravati	R\$211,41	R\$62,91	R\$65,61
Aliexpress	R\$139,60	R\$32,62	R\$33,35
Curto Circuito	R\$225,25	R\$57,00	R\$78,00
SmartKits	R\$269,90	R\$49,90	R\$72,90

Tabela 3.1: Comparação de custos dos Módulos ESP32/Lora

Tendo em vista os valores da Tabela 3.1 pode-se concluir que economicamente é mais vantajoso a compra dos módulos separados principalmente ao escalar o projeto para uso de vários nós e gateways. Apesar disso, para o presente projeto o uso de

um kit completo e pronto facilitou bastante, evitando a necessidade de produção de *Printed Circuit Board* (PCB), soldagem e montagens mais complexas.

3.4.2 Custos do Sensor Ultrassônico

A tabela a seguir apresenta uma comparação de custos para o sensor ultrassônico JSN-SR04T e dois principais concorrentes.

Fornecedor	JSN-SR04T	HC-SR04	TeraRanger Evo 15m
Mouser	não disponível	não disponível	R\$606,10
Digikey	não disponível	não disponível	R\$608,00
Aliexpress	R\$18,16	R\$7,46	não disponível
Saravati	R\$89,91	R\$9,81	não disponível

Tabela 3.2: Comparação de custos do Sensor Ultrassônico



Figura 3.2: Sensor de Distância TeraRanger Evo 15m

Para melhores conclusões abaixo estão descritas as principais características de cada um dos sensores.

Característica	JSN-SR04T	HC-SR04	TeraRanger Evo 15m
Tecnologia	Ultrassônico	Ultrassônico	Tempo de Vôo Infravermelho
Características	Impermeável	Não Impermeável	Uso Industrial
Alcance	23 cm a 6 m	2 cm a 4 m	0,5 cm a 15 m
Resolução	1 cm	0,3 cm	0,5 cm
Taxa de Atualização	20Hz	40Hz	240Hz
Ângulo de Medida	37, 5°	30°	2°

Tabela 3.3: Comparação de custos do Sensor Ultrassônico

As tabelas 3.2 e 3.3 demonstram que o modelo JSN-SR04T é um dos modelos com o melhor custo benefício financeiro e o mais apropriado para a aplicação em áreas molhadas e agressivas. Para maior precisão e fiabilidade o modelo TeraRanger Evo 15m seria uma excelente opção pela seu amplo alcance e robustez. Apesar dos itens citados acima, para o atual projeto o modelo JSN-SR04T será suficiente fornecendo as características necessárias.

Capítulo 4

Implementação do Sistema

Este capítulo é dedicado à exploração detalhada do desenvolvimento do software, bem como à montagem e implementação do sistema proposto para monitoria inteligente de bueiros. Aqui estão descritos os processos técnicos e metodológicos adotados, desde a escolha das ferramentas e tecnologias até a integração final.

4.1 Montagem do Hardware

Nesta secção serão demonstradas todas as conexões e etapas para a montagem do circuito do Nó e Gateway.

4.1.1 Montagem do Módulo de Medição - Nó

Após separar os elementos, foi feito primeiro a soldagem do barramentos de pinos no kit de desenvolvimento ESP32 LoRa que seria utilizado como Nó, enviando os dados do sensor ultrassónico. Em seguida a conexão da antena LoRaWAN inclusa no Kit de desenvolvimento. Depois dessa etapa as seguintes conexões foram efetuadas:

Sensor Ultrassónico

O Sensor Ultrassónico JSN-SR04T escolhido para o projeto utilizou as seguintes conexões por meio de *jumpers* para comunicar-se com o micro-controlador.

- **Alimentação:** Foi conectado a saída de 5V do micro-controlador para energizar o sensor.

- **GND (Terra):** O pino de terra do sensor foi conectado ao pino GND do ESP32 para completar o circuito de alimentação.
- **Echo:** O pino Echo do sensor, que envia o sinal de retorno após detectar um objeto, foi conectado ao pino 16 do ESP32. Este pino é configurado como entrada no microcontrolador para receber o sinal de eco.
- **Trigger:** O pino Trigger do sensor, que emite o pulso ultrassônico, foi conectado ao pino 13 do ESP32. Este pino é configurado como saída para enviar o comando de disparo ao sensor.

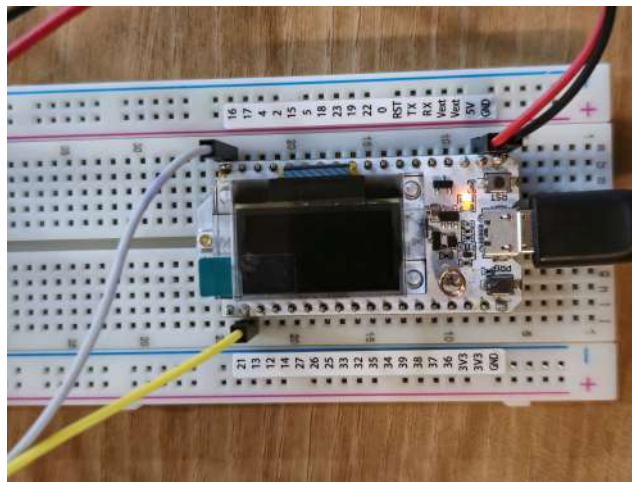


Figura 4.1: Ligações Sensor Ultrassônico JSN-SR04T

Esquema Elétrico

Conforme ilustrado no esquema elétrico da figura 4.2, as conexões permitem que o ESP32 controle o sensor ultrassônico e receba dados sobre a distância até os objetos detetados, que são essenciais para as funcionalidades de medição do projeto.

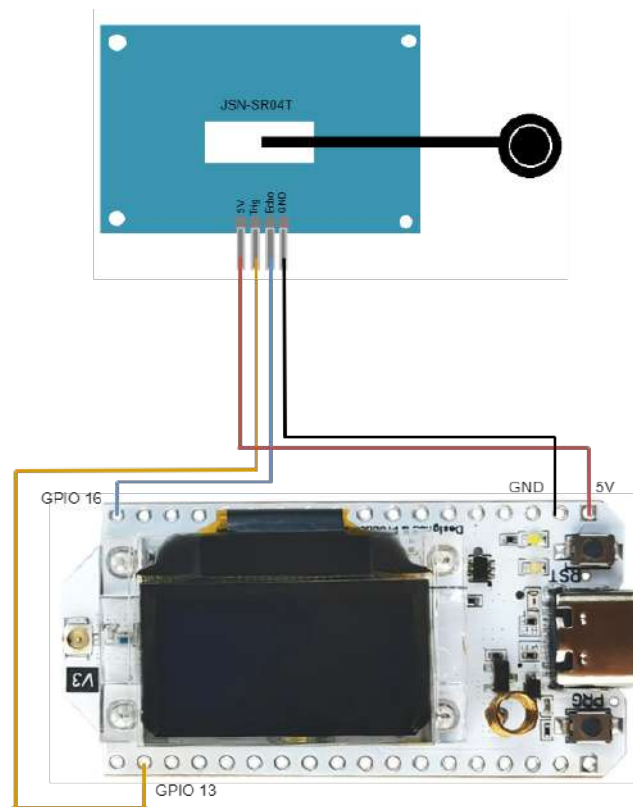


Figura 4.2: Esquema Elétrico de ligação Sensor Ultrassónico - ESP32

A figura 4.3 ilustra o resultado após todas as conexões com o auxílio de uma *protoboard*.

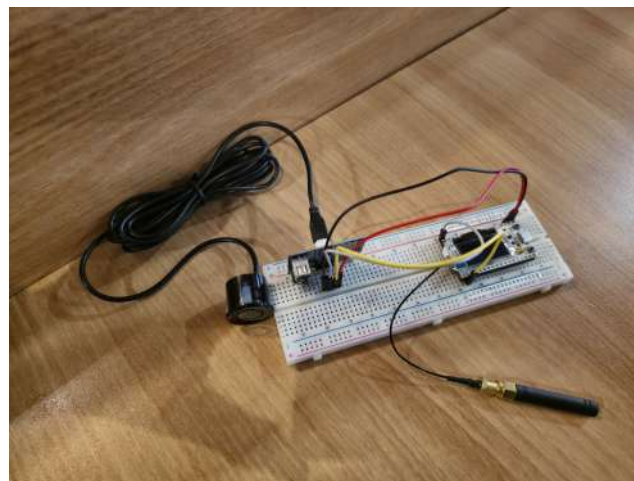


Figura 4.3: Módulo de Medição - Nó

4.1.2 Montagem do Gateway

Para montagem do gateway não foi necessária conexões nem a solda do barramento de pinos. Para receber os dados via Lora e enviar via WiFi para o ThingSpeak foi

necessário apenas a ligação da antena LoRaWAN inclusa no kit de desenvolvimento.

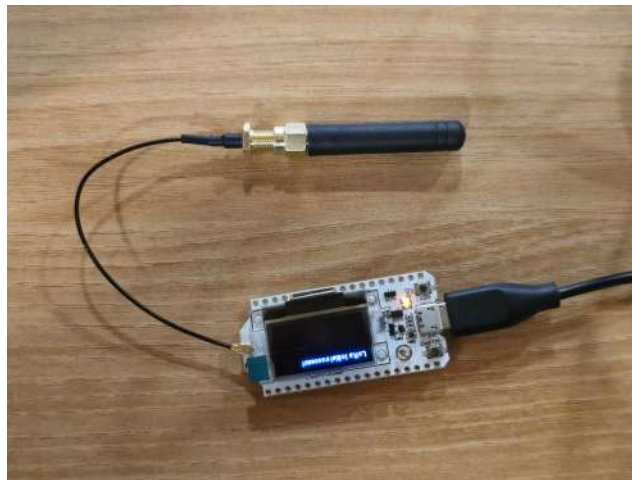


Figura 4.4: Módulo Gateway

4.2 Programação do Nó - ESP32

Esta secção condensa as informações técnicas e práticas necessárias para compreender a programação e operação do nó.

4.2.1 Configuração Inicial e Bibliotecas

A programação do nó sensorial é feita utilizando o kit Heltec ESP32 LoRa, que requer a inclusão da biblioteca Heltec para acessar funcionalidades de comunicação LoRa e operação do display OLED. A biblioteca é incluída no início do script:

```
#include "heltec.h" // Biblioteca para o Heltec ESP32
```

A biblioteca `heltec.h` para o kit Heltec ESP32 LoRa oferece um conjunto abrangente de funções e classes que simplificam o desenvolvimento de aplicações LoRa e Wi-Fi. Por exemplo, a função `Heltec.begin()` inicializa tanto o módulo LoRa quanto o Wi-Fi, enquanto `LoRa.beginPacket()` e `LoRa.endPacket()` permitem o envio de pacotes LoRa de forma estruturada. Para aplicações que exigem o uso do display OLED, a biblioteca fornece funções como `Heltec.display->drawString()` para exibir texto e `Heltec.display->display()` para atualizar a tela.

4.2.2 Funções de Medição de Distância e Transmissão de Dados

A função `setup()` configura os pinos do sensor ultrassônico e inicializa o kit Heltec ESP32 para comunicação via LoRa. A função `Heltec.begin()` configura o rádio LoRa, o display OLED, e outros componentes necessários.

```
void setup() {  
    pinMode(trigPin, OUTPUT); // Define o pino do gatilho como saída  
    pinMode(echoPin, INPUT);  // Define o pino do eco como entrada  
    Heltec.begin(true, true, true, true, BAND, PABOOST); // Inicializa o kit  
    Heltec ESP32 LoRaWAN  
}
```

A função `getDistance()` é responsável por medir a distância. Ela gera um pulso no `trigPin`, mede a duração do pulso recebido no `echoPin` e calcula a distância baseada na velocidade do som.

```
int getDistance() {  
    digitalWrite(trigPin, LOW); // Garante que o pino de gatilho está baixo  
    delayMicroseconds(10);      // Aguarda 10 microssegundos  
    digitalWrite(trigPin, HIGH); // Define o pino de gatilho como alto  
    delayMicroseconds(50);      // Aguarda 50 microssegundos  
    digitalWrite(trigPin, LOW); // Define o pino de gatilho como baixo  
    duration = pulseIn(echoPin, HIGH); // Calcula a duração do pulso no pino  
    de eco  
    return duration * 0.034 / 2; // Calcula a distância com base na duração  
    do pulso  
}
```

A função `sendPacket()` envia a distância medida via LoRa. Ela cria um pacote LoRa, adiciona os dados de distância e envia o pacote. Além disso, exibe uma mensagem no display OLED indicando que o pacote está sendo enviado.

```
void sendPacket(int distance) {  
    LoRa.beginPacket(); // Inicia a criação do pacote LoRa  
    LoRa.print(distance); // Adiciona a distância ao pacote  
    LoRa.endPacket(); // Finaliza o envio do pacote  
    Heltec.display->drawString(0, 30, "Enviando pacote via LoRa");  
    //Define a mensagem a ser exibida no display OLED  
    Heltec.display->display(); //Inicializa a exibição no display  
}
```

4.2.3 Loop Principal

O `loop()` principal mede a distância utilizando a função `getDistance()`, envia os dados usando `sendPacket()` e aguarda um segundo antes de repetir o processo.

```
void loop() {  
    int currentDistance = getDistance(); // Mede a distância usando o
```

```
sensor ultrassônico
sendPacket(currentDistance); // Envia a distância medida via LoRa
delay(1000); // Aguarda 1 segundo antes de realizar a próxima medição
}
```

Este trecho do código demonstra a integração entre o sensor ultrassônico e o módulo LoRa do ESP32, realizando medições periódicas e transmitindo os dados para um gateway.

4.3 Programação do Gateway - ESP32

Esta seção condensa as informações técnicas e práticas necessárias para compreender a programação e operação do gateway. O gateway ESP32 é programado para receber os dados enviados via LoRa e enviá-los para a plataforma ThingSpeak.

4.3.1 Configuração Inicial e Bibliotecas

A programação do ESP32 como um gateway requer a utilização de bibliotecas especializadas para comunicação LoRa e Wi-Fi:

```
#include "heltec.h" // Para funcionalidades LoRa e OLED
#include <WiFi.h>    // Biblioteca padrão para conexão Wi-Fi
#include <ThingSpeak.h> // Biblioteca para conectar com ThingSpeak
```

Estas bibliotecas facilitam a interação com o hardware do kit Heltec ESP32, bem como a gestão da conexão Wi-Fi e a integração com a plataforma ThingSpeak. A biblioteca `WiFi.h` simplifica a conexão Wi-Fi, com funções como `WiFi.begin()` para iniciar a conexão e `WiFi.status()` para verificar o status. Já a `ThingSpeak.h` facilita a interação com a plataforma ThingSpeak, oferecendo funções como `ThingSpeak.writeField()` para enviar dados e `ThingSpeak.readField()` para ler dados dos canais.

4.3.2 Configurações de Conectividade e Variáveis de Canal ThingSpeak

Para a transmissão de dados para a nuvem através do ThingSpeak, é necessário configurar o ESP32 com as credenciais da rede Wi-Fi e os detalhes do canal ThingSpeak:

```
const char* ssid = "SSID_REDE"; // Nome da rede Wi-Fi
const char* password = "SENHA_REDE"; // Senha da rede Wi-Fi
unsigned long channelID = CHANNEL_ID; // ID do canal no ThingSpeak
const char* writeAPIKey = "WRITE_API_KEY"; // Chave de escrita do canal
WiFiClient client; // Cria um cliente Wi-Fi
```

Aqui, foram definidas as credenciais da rede Wi-Fi e as configurações do ThingSpeak, incluindo o ID do canal e a chave de escrita.

Nota: As variáveis *channelID* e *writeAPIKey* são obtidas durante o processo de criação do canal no ThingSpeak. Estas são essenciais para a autenticação e o envio de dados ao canal correto na plataforma.

4.3.3 Recepção de Dados LoRa e Transmissão para ThingSpeak

O loop principal do código é responsável pela recepção de pacotes LoRa e pelo envio desses dados ao ThingSpeak, utilizando as configurações de rede previamente estabelecidas:

```
void setup() {  
  WiFi.begin(ssid, password); // Conecta o ESP32 à rede Wi-Fi  
  while (WiFi.status() != WL_CONNECTED) { // Espera até que a conexão Wi-Fi  
    seja estabelecida  
    delay(1000);  
    Serial.println("Conectando ao Wi-Fi...");  
  }  
  ThingSpeak.begin(client); // Inicializa o ThingSpeak  
  Heltec.begin(true, true, true, true, BAND, PABOOST); // Inicializa o kit  
  Heltec ESP32 LoRaWAN  
}
```

A função `setup()` conecta o ESP32 à rede Wi-Fi e inicializa o ThingSpeak e o kit Heltec ESP32 para comunicação via LoRa.

Envio dos Dados para o ThingSpeak: Após a recepção dos dados via LoRa, o ESP32 entra em ação para transmiti-los à plataforma ThingSpeak. Essa etapa crucial garante que as informações coletadas sejam armazenadas e disponibilizadas para análise e visualização. A biblioteca ThingSpeak desempenha um papel fundamental nesse processo, simplificando a comunicação com a plataforma na nuvem.

Protocolo HTTP e a Biblioteca ThingSpeak O protocolo utilizado para essa comunicação é o HTTP, através de requisições do tipo POST. A biblioteca ThingSpeak, integrada ao código, abstrai a complexidade de construir e enviar essas requisições. Ela se encarrega de formatar os dados recebidos via LoRa em parâmetros adequados para a URL da requisição, além de incluir a chave de API do canal ThingSpeak para autenticação. A biblioteca também lida com a resposta do servidor, verificando se os dados foram enviados com sucesso ou se houve algum erro, garantindo a confiabilidade da transmissão.

4.3.4 Loop Principal

O `loop()` principal verifica se há um pacote LoRa disponível. Se um pacote for recebido, os dados são lidos e enviados para o ThingSpeak. O `Serial.println()` é utilizado para depuração, exibindo os dados recebidos e confirmando o envio para o ThingSpeak.

```
void loop() {  
  int packetSize = LoRa.parsePacket(); // Verifica se há um pacote LoRa  
  disponível  
  if (packetSize) {  
    String receivedData = LoRa.readString(); // Lê os dados recebidos via  
    LoRa  
    Serial.println("Dados recebidos via LoRa: " + receivedData);  
    ThingSpeak.setField(1, receivedData); // Configura o campo de dados do  
    ThingSpeak com os dados recebidos  
    ThingSpeak.writeFields(channelID, writeAPIKey); // Envia os dados para  
    o ThingSpeak  
    Serial.println("Dados enviados para ThingSpeak.");  
  }  
}
```

4.4 Configuração do Dashboard ThingSpeak

Este capítulo descreve a configuração do dashboard no ThingSpeak para a visualização e análise de dados recebidos do sistema de monitoramento LoRa. ThingSpeak é uma plataforma de IoT que permite coletar, visualizar e analisar dados de sensores em tempo real na nuvem.

4.4.1 Criação do Canal e Configuração dos Campos

A configuração inicial no ThingSpeak envolve a criação de um canal, que serve como um repositório para os dados enviados pelo gateway. Cada canal pode ter até oito campos de dados, que são configurados conforme as especificidades dos dados recebidos.

1. **Acessar ThingSpeak:** Primeiramente, é necessário criar uma conta no ThingSpeak e acessar o painel de controle.
2. **Criar um Novo Canal:** Dentro do painel, selecionar "Channels", depois "My Channels", e clicar em "New Channel".

3. **Configurar os Campos:** Nomear o canal, por exemplo, "Monitoramento Sistema de Escoamento - LoRaWAN", e ativar os campos necessários. No caso, o "Field 1" pode ser nomeado como "Distância" para representar a medição de distância em centímetros.

4.4.2 API Keys e Conexão com o Gateway

Para que o gateway possa enviar dados ao ThingSpeak, é necessário configurar a *Write API Key*, que autentica as requisições de envio ao canal:

1. **Obter a Write API Key:** Na configuração do canal, navegar até a aba "API Keys" e copiar a chave de escrita (*Write API Key*).
2. **Substituição no Código do Gateway:** Substituir o valor da variável `writeAPIKey` no código do gateway pela chave obtida.

4.4.3 Visualização e Análise de Dados

O ThingSpeak oferece ferramentas para visualização em tempo real dos dados:

1. **Configurar Widgets:** Pode-se adicionar widgets no dashboard do canal para visualizar os dados de forma gráfica. Por exemplo, um gráfico de linha pode ser usado para visualizar as mudanças na distância ao longo do tempo, conforme mostrado na imagem.
2. **MATLAB Analysis and Visualization:** ThingSpeak integra-se com MATLAB, permitindo realizar análises mais complexas e criar visualizações avançadas dos dados coletados.

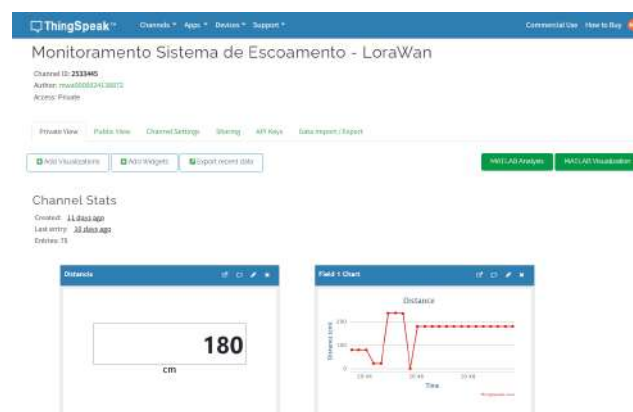


Figura 4.5: Dashboard Thingspeak

4.4.4 Gerenciamento e Compartilhamento de Dados

Finalmente, o ThingSpeak permite configurar o compartilhamento dos dados e gerenciar quem pode visualizar ou acessar as informações:

1. **Configurações de Privacidade:** É possível configurar o canal como privado ou público, dependendo da necessidade de compartilhar os dados com outros usuários ou mantê-los confidenciais.
2. **Exportação de Dados:** Os dados podem ser exportados em vários formatos para análise offline ou backup.

Este capítulo oferece uma visão detalhada sobre como configurar e utilizar o ThingSpeak para monitoramento e análise de dados em projetos de IoT utilizando LoRaWAN, destacando a flexibilidade e a eficácia da plataforma para a gestão de dados sensoriais em tempo real.

Capítulo 5

Resultados

Nesta seção, são apresentados os resultados obtidos a partir da implementação do sistema de monitoramento de bueiros utilizando sensores ultrassônicos e a plataforma IoT ThingSpeak. O objetivo principal foi verificar a eficácia do sistema em medir a distância dentro dos bueiros e gerar alertas automáticos em caso de possíveis obstruções.

5.1 Monitoramento da Distância

O sistema foi capaz de monitorar continuamente a distância medida pelo sensor ultrassônico JSN-SR04T instalado nos bueiros. As medições de distância foram transmitidas em tempo real para a plataforma ThingSpeak através do módulo ESP32/LoRaWAN. Os dados coletados foram armazenados e visualizados em um painel de controle (*dashboard*) no ThingSpeak, permitindo uma análise detalhada das variações de distância ao longo do tempo.

5.1.1 Configuração do Alerta

Para exemplificar o funcionamento do alerta, foi definido um *trigger* no ThingSpeak que acenderia uma luz vermelha no dashboard caso a distância medida fosse menor que 50 centímetros, conforme ilustrado na Figura 8. Esta medida foi estimada para representar uma situação de alerta, indicando um possível entupimento de um dos bueiros por objetos, lixo ou líquido acumulado.

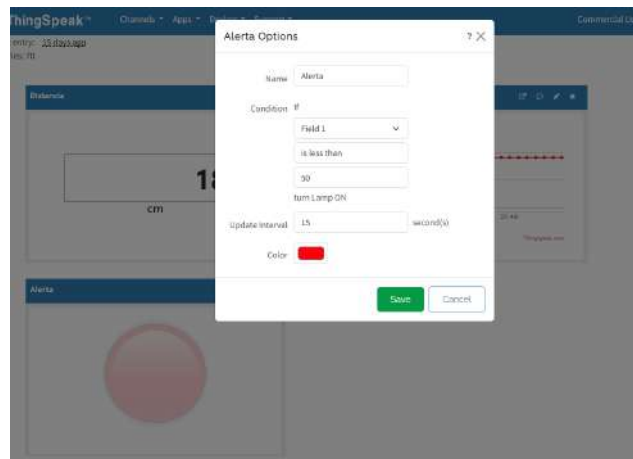


Figura 5.1: Configuração do alerta no Dashboard ThingSpeak

5.1.2 Demonstração do Protótipo em Funcionamento

Para ilustrar o funcionamento do sistema na prática, foram capturadas imagens do protótipo em diferentes estados ao lado de uma fita métrica. A Figura 5.2 mostra o sistema em seu estado normal, detetando uma distancia de aproximadamente 117 centímetros. Essa situação representa o bueiro desobstruído e o alerta inativo. Já a Figura 5.3 exhibe o sistema em uma situação de alerta, detetando uma distância de aproximadamente 30 centímetros. Esse caso seria a representação do sistema com o nível de água elevado e o alerta ativo, indicando um possível risco de alagamento.

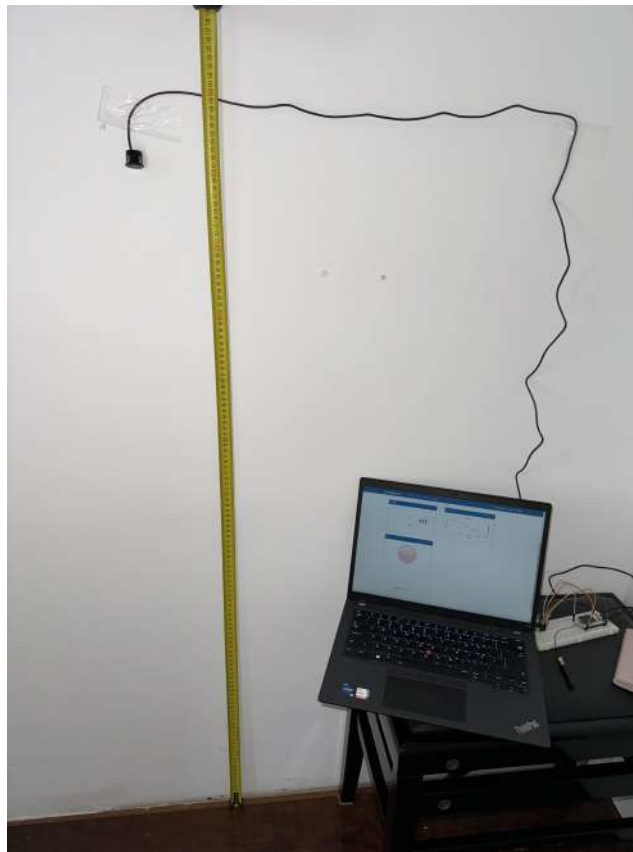


Figura 5.2: Protótipo em estado normal, com alerta inativo.

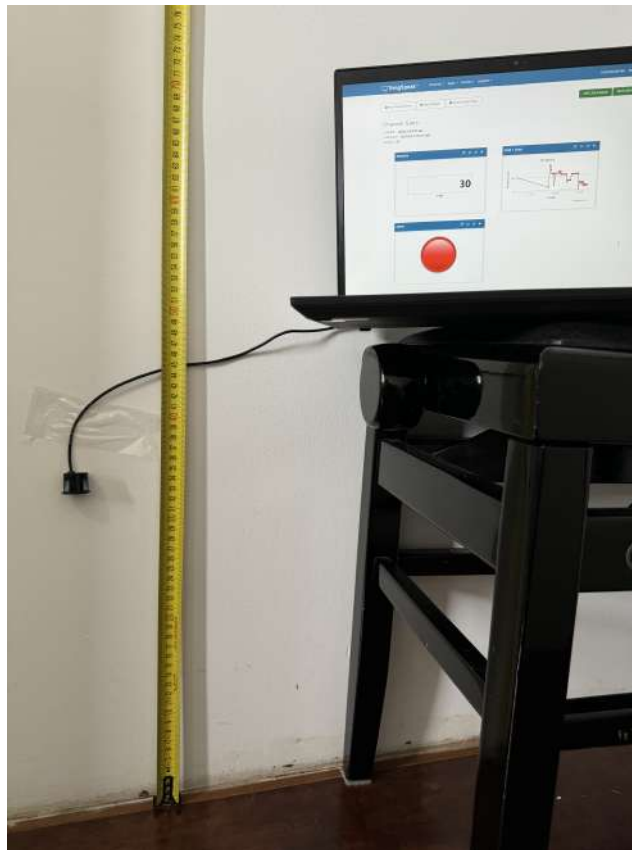


Figura 5.3: Protótipo em situação de alerta, com nível de água elevado e alerta ativo.

5.1.3 Análise dos Dados Coletados

Os dados coletados pelo sensor ultrassônico foram analisados para verificar a precisão e confiabilidade das medições. O gráfico a seguir mostra a variação da distância medida pelo sensor ao longo de um período de 24 horas.



Figura 5.4: Análise dos Dados de Distância Coletados pelo Sensor Ultrassônico

Os resultados da Figura 9 mostraram que o sistema foi eficaz em detectar mudanças na distância, fornecendo leituras consistentes e precisas.

5.2 Desempenho do Alerta Automático

O alerta automático configurado no ThingSpeak demonstrou ser uma ferramenta eficaz para identificar situações de risco em tempo real. Sempre que a distância medida caiu abaixo do limite definido, o sistema gerou um alerta imediato. Esta funcionalidade é crucial para a prevenção de alagamentos urbanos, permitindo uma resposta rápida a obstruções nos bueiros.

5.3 Conclusão dos Resultados

Os resultados obtidos confirmam que o sistema de monitoramento de bueiros utilizando sensores ultrassônicos e a plataforma ThingSpeak é capaz de medir a distância de forma precisa e confiável. O uso de *triggers* automáticos para gerar alertas mostrou-se eficaz na detecção de possíveis obstruções, proporcionando uma ferramenta valiosa para a gestão de infraestruturas urbanas e a prevenção de alagamentos.

O sucesso desta implementação demonstra o potencial das tecnologias IoT para aplicações em monitoramento e gestão de sistemas urbanos, destacando a importância da inovação tecnológica na solução de problemas ambientais e de infraestrutura.

5.4 Limitações e Trabalhos Futuros

Apesar dos resultados promissores, o sistema desenvolvido possui algumas limitações que podem ser abordadas em trabalhos futuros. Atualmente, o protótipo depende de uma fonte de alimentação externa, o que restringe sua instalação em locais sem acesso à energia elétrica. Além disso, o sistema foi testado em um ambiente controlado, e sua performance em condições reais, como em dias de chuva intensa, ainda precisa ser avaliada.

Capítulo 6

Conclusões

Neste capítulo final são apresentadas as conclusões gerais do trabalho desenvolvido, revisão dos objetivos traçados e perspectivas para trabalhos futuros.

6.1 Objetivos Alcançados

Tendo em vista os objetivos iniciais traçados na seção 1.2 podemos tirar as seguintes conclusões:

- Estudo aprofundado das tecnologias IoT: Foram estudadas diversas tecnologias relacionadas a Internet das Coisas, sendo possível analisar as características de cada uma, suas vantagens e desvantagens. Além disso, por meio do estudo foi possível escolher tecnologias que se encaixariam melhor no projeto.
- Desenvolvimento de um protótipo IoT funcional: No presente trabalho, foi desenvolvido com sucesso um protótipo completo e funcional com todas as características do IoT.
- Configuração de rede sem fio: Por meio do estudo das redes sem fios e conexão entre nós e *gateways*, foi possível estabelecer uma ligação com delay pequeno e que fornecesse informações próximo de tempo real.
- Testes e validação do sistema: Foram executados diversos testes que confirmaram o funcionamento do sistema e dos gatilhos configurados para alertar os usuários no dashboard.

- Documentação e disseminação dos resultados: O presente trabalho escrito é resultado da coleta de informações de todas as etapas pesquisas, desenvolvimento e conclusões.

6.2 Perspectivas Futuras

As possibilidades de aprimoramento e expansão deste projeto são vastas. Algumas das principais ideias para trabalhos futuros incluem:

- Implementação de Baterias de Lítio: A inclusão de baterias de lítio proporcionaria autonomia energética ao sistema, permitindo sua instalação em qualquer bueiro, independentemente da disponibilidade de energia elétrica.
- Expansão da Rede de Sensores: A adição de múltiplos nós sensores permitiria o monitoramento de uma área mais ampla, aumentando a cobertura e a eficácia do sistema na prevenção de alagamentos. Essa alteração poderia ser feita de forma simples, necessitando pequenas alterações no código.
- Desenvolvimento de um Produto Mínimo Viável (MVP): A criação de um MVP, possivelmente integrado à tampa do bueiro, tornaria o sistema mais compacto, discreto e de fácil instalação.
- Apresentação para Prefeituras e Empresas: A demonstração do sistema para potenciais usuários, como prefeituras e empresas de saneamento, poderia viabilizar sua implementação em larga escala, gerando impacto social e econômico positivo.
- Atualizações OTA (Over-the-Air): A implementação de atualizações OTA permitiria a manutenção e o aprimoramento contínuo do sistema de forma remota, sem a necessidade de intervenção física nos dispositivos instalados.
- Inclusão de Outros Sensores: A adição de sensores de fluxo de água, umidade e outros parâmetros relevantes poderia enriquecer os dados coletados, fornecendo informações mais completas sobre as condições dos bueiros e do sistema de drenagem.

6.3 Considerações Finais

O sistema de monitoramento de bueiros desenvolvido neste trabalho representa um passo importante na aplicação da Internet das Coisas para a resolução de problemas urbanos. Os resultados positivos alcançados demonstram o potencial dessa tecnologia para melhorar a gestão de infraestruturas, prevenir desastres naturais e promover cidades mais inteligentes e resilientes.

Referências

- [1] G. Gridling and B. Weiss, “Introduction to microcontrollers,” *Vienna University of Technology Institute of Computer Engineering Embedded Computing Systems Group*, p. 25, 2007. [Citado nas páginas 5, 7 e 8]
- [2] L. R. M. da Silva, *Circuitos digitais: fundamentos, aplicações e inovações*. Rio de Janeiro: Freitas Bastos, 2023. E-book. [Citado na página 6]
- [3] W. Alexander and C. M. Williams, *Digital Signal Processing: Principles, Algorithms and System Design*. USA: Academic Press, Inc., 1st ed., 2016. [Citado na página 7]
- [4] M. Babiuch, P. Foltýnek, and P. Smutný, “Using the esp32 microcontroller for data processing,” in *2019 20th International Carpathian Control Conference (ICCC)*, pp. 1–6, IEEE, 2019. [Citado na página 10]
- [5] M. Schmidt, “Arduino: a quick-start guide,” *Arduino: A Quick-Start Guide*, pp. 1–324, 2015. [Citado na página 11]
- [6] C. Noviello, “Mastering stm32,” *Leadpub*. Obtenido de <http://www2.keil.com/mdk5/uvision>, 2017. [Citado na página 12]
- [7] A. Carullo and M. Parvis, “An ultrasonic sensor for distance measurement in automotive applications,” *IEEE Sensors Journal*, vol. 1, pp. 143–147, 2001. [Citado nas páginas 14 e 16]
- [8] J. Borenstein and Y. Koren, “Obstacle avoidance with ultrasonic sensors,” *IEEE Journal on Robotics and Automation*, vol. 4, no. 2, pp. 213–218, 1988. [Citado na página 15]
- [9] P. Hauptmann, N. Hoppe, and A. Püttmer, “Application of ultrasonic sensors in the process industry,” *Measurement Science and Technology*, vol. 13, no. 8, p. R139, 2002. [Citado nas páginas 15 e 16]
- [10] S. Olisa, M. Khan, and A. Starr, “Review of current guided wave ultrasonic testing (gwut) limitations and future directions,” *Sensors*, vol. 21, no. 3, p. 811, 2021. [Citado na página 17]

-
- [11] J. Rhee and J. Seo, “Low-cost curb detection and localization system using multiple ultrasonic sensors,” *Sensors*, vol. 19, no. 6, p. 1389, 2019. [Citado na página 17]
 - [12] S. Hassani and U. Dackermann, “A systematic review of advanced sensor technologies for non-destructive testing and structural health monitoring,” *Sensors*, vol. 23, no. 4, p. 2204, 2023. [Citado na página 17]
 - [13] R. Soman, J. Wee, and K. Peters, “Optical fiber sensors for ultrasonic structural health monitoring: A review,” *Sensors*, vol. 21, no. 21, p. 7345, 2021. [Citado na página 17]
 - [14] N. Godin, P. Reynaud, and G. Fantozzi, “Challenges and limitations in the identification of acoustic emission signature of damage mechanisms in composites materials,” *Applied Sciences*, vol. 8, no. 8, p. 1267, 2018. [Citado na página 17]
 - [15] D. Rani and N. Gill, “Review of various iot standards and communication protocols,” *International Journal of Engineering Research and Technology*, vol. 12, no. 5, pp. 10–15, 2019. [Citado na página 18]
 - [16] C. Bayılmış, M. Ebleme, Çavuşoğlu, and K. Küçük, “A survey on communication protocols and performance evaluations for internet of things,” *Digital Communications and Networks*, 2022. [Citado na página 18]
 - [17] S. Al-Sarawi, M. Anbar, K. Alieyan, and M. Alzubaidi, “Internet of things (iot) communication protocols,” *IEEE Access*, vol. 5, pp. 22946–22957, 2017. [Citado na página 20]
 - [18] T. Moraes, B. Nogueira, V. Lira, *et al.*, “Performance comparison of iot communication protocols,” *IEEE International Conference on Systems, Man, and Cybernetics*, 2019. [Citado na página 21]
 - [19] M. Diwan and M. D’Souza, “A framework for modeling and verifying iot communication protocols,” in *Symposium on Reliable Distributed Systems*, Springer, 2017. [Citado na página 22]
 - [20] L. Tightiz and H. Yang, “A comprehensive review on iot protocols’ features in smart grid communication,” *Energies*, vol. 13, no. 11, p. 2762, 2020. [Citado na página 23]
 - [21] J. Sidna, B. Amine, N. Abdallah, and H. El Alami, “Analysis and evaluation of communication protocols for iot applications,” *Proceedings of the 13th International Conference on Theory and Practice of Electronic Governance*, 2020. [Citado na página 23]

-
- [22] M. Lombardi, F. Pascale, and D. Santaniello, “Internet of things: A general overview between architectures, protocols and applications,” *Information*, vol. 12, no. 2, p. 87, 2021. [Citado na página 24]
 - [23] B. Gupta and M. Quamara, “An overview of internet of things (iot): Architectural aspects, challenges, and protocols,” *Concurrency and Computation: Practice and Experience*, vol. 32, no. e4946, pp. 1–25, 2020. [Citado na página 24]
 - [24] P. Desai, A. Sheth, and P. Anantharam, “Semantic gateway as a service architecture for iot interoperability,” pp. 313–319, IEEE, 2015. [Citado na página 25]
 - [25] Q. Zhu, R. Wang, Q. Chen, Y. Liu, and W. Qin, “Iot gateway: Bridging wireless sensor networks into internet of things,” in *2010 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*, pp. 347–352, IEEE, 2010. [Citado na página 25]
 - [26] D. Bastos, “Cloud for iot—a survey of technologies and security features of public cloud iot solutions,” *IEEE*, 2019. [Citado na página 26]
 - [27] P. Pierleoni, R. Concetti, A. Belli, and L. Palma, “Amazon, google and microsoft solutions for iot: Architectures and a performance comparison,” *IEEE Access*, 2019. [Citado na página 26]
 - [28] D. Saini, K. Kumar, and P. Gupta, “Security issues in iot and cloud computing service models with suggested solutions,” *Security and Communication Networks*, 2022. [Citado na página 27]