



UNIVERSIDADE DO SUL DE SANTA CATARINA

GUSTAVO WEBER DE SIMAS

MARCIA PANCIERA FACCHIN

**IMPLEMENTAÇÃO DE UM MONITORAMENTO REMOTO COM
RADIO LORA E MODBUS TCP/IP**

Palhoça
2020

GUSTAVO WEBER DE SIMAS
MARCIA PANCIERA FACCHIN

**IMPLEMENTAÇÃO DE UM MONITORAMENTO REMOTO COM
RADIO LORA E MODBUS TCP/IP**

Trabalho de Conclusão de Curso apresentado
ao Curso de Engenharia Elétrica da
Universidade do Sul de Santa Catarina como
requisito parcial à obtenção do título de
Engenheiro Eletricista.

Orientador: Fábio Ignácio da Rosa, Eng. Esp.

Palhoça

2020

GUSTAVO WEBER DE SIMAS
MARCIA PANCIERA FACCHIN

**IMPLEMENTAÇÃO DE UM MONITORAMENTO REMOTO COM
RADIO LORA E MODBUS TCP/IP**

Este Trabalho de Conclusão de Curso foi julgado adequado à obtenção do título de Engenheiro Eletricista e aprovado em sua forma final pelo Curso de Engenharia Elétrica da Universidade do Sul de Santa Catarina.

Palhoça, 09 de julho de 2020.

Professor e orientador Fábio Ignácio da Rosa, Eng. Esp.
Universidade do Sul de Santa Catarina

Bruna Peixoto de Souza, Eng.
Instituto Federal de Santa Catarina

Edgar Marcelino Pinheiro, Eng.
Universidade do Sul de Santa Catarina

AGRADECIMENTOS

Gostaríamos de agradecer a todos os amigos e familiares que com muito carinho e apoio, não mediram esforços para que chegássemos até esta etapa de nossas vidas.

Ao Giulio Cesare Facchin por todo apoio e pela ajuda, que muito contribuiu para a realização deste trabalho.

Ao nosso orientador, Fábio Ignácio da Rosa, por todos os conselhos, pela ajuda e pela paciência com a qual guiou nosso aprendizado.

Agradecemos, também, à empresa Casan, pelas informações prestadas para que este trabalho se tornasse possível.

RESUMO

O conceito de Internet das Coisas está cada vez mais presente nas aplicações industriais com a expansão do uso dispositivos inteligentes para sensoriamento, processamento e comunicação. O monitoramento dos processos produtivos surge para promover a confiabilidade do sistema ao possibilitar o acompanhamento da operação remotamente. Neste contexto, este trabalho teve como objetivo utilizar a tecnologia LoRa de redes de comunicações móveis para monitoramento de dados do quadro elétrico no sistema de abastecimento de água. Foram utilizados um par de módulos com ESP32 e LoRa integrados, para comunicação por enlace de rádio, onde um deles opera como *endpoint* e o outro como *gateway* de borda com interface MODBUS IP. Uma interface gráfica foi desenvolvida no sistema supervisor para visualização dos dados coletados. Visando avaliar o alcance da tecnologia LoRa em um ambiente urbano, foram realizadas medições para avaliar a qualidade do sinal. As conclusões mostram a viabilidade desta tecnologia para este tipo de aplicação.

Palavras-chave: LoRa. Internet das Coisas. ESP32. Monitoramento Remoto.

ABSTRACT

The concept of Internet of Things is increasingly present in industrial applications with the expansion of its use in smart devices for sensing, processing and communication. The monitoring of Industrial processes can bring reliability to the system by enabling the monitoring of the operation remotely. This work used the LoRa technology of mobile communications networks to monitor data from an electrical panel of a water supply system. A couple of modules with integrated ESP32 and LoRa were used one of them operates as an endpoint and the other as an edge gateway with MODBUS IP interface; together they can communicate by radio link. A graphical interface was developed at the supervisory system to allow the visualization of the collected data. In order to assess the reach of LoRa technology in an urban environment, measurements were taken to assess signal quality. The conclusions show the viability of this technology for this type of application.

Keywords: LoRa. Internet of things. ESP32. Remote monitoring.

LISTA DE ILUSTRAÇÕES

Figura 1 - Sistema existente de comunicação sem fio	11
Figura 2 - Escopo de projeto	12
Figura 3 - Arduino UNO	16
Figura 4 - NodeMCU	17
Figura 5 - Módulo ESP32	18
Figura 6 - Módulo ESP32 WIFI LORA da HEALTEC	19
Figura 7 - Arquitetura da rede LORA.	24
Figura 8 - Modulação AM/DSB e AM/DSB-SC	27
Figura 9 - Modulação FM	28
Figura 10 - Tipos de Modulação Digital	29
Figura 11 - Demonstração do Filtro Gaussiano	30
Figura 12 - Arquitetura do Modelo OSI	33
Figura 13 - Estrutura da PDU	33
Figura 14 - Relacionamento de rede tipo mestre-escravo.	35
Figura 15 - Redes e Barramentos.	36
Figura 16 - Frame Modbus.	36
Figura 17 - Função Modbus	37
Figura 18 - Arquitetura de rede para SCADA	38
Figura 19 - Interface Gráfica	41
Figura 20 - Módulo A	43
Figura 21 - Módulo B <i>Gateway</i>	43
Figura 22 - Fluxograma Módulo B <i>Gateway</i>	45
Figura 23 - Fluxograma Módulo A <i>Endpoint</i>	46
Figura 24 - Display Módulo B <i>Gateway</i>	47
Figura 25 - Monitoramento do módulo A através da tela desenvolvida no Virtuino	47
Figura 26 – Comando para acionamento da bomba através do Virtuino	48
Figura 27 - Bomba acionada através da tela desenvolvida no Virtuino	49
Figura 28 - Monitoramento da chave 1 do módulo B através do Virtuino	49
Figura 29 - Monitoramento da chave 2 do módulo B através do Virtuino	50
Figura 30 - Acionamento do led do módulo B através da tela desenvolvida no Virtuino	50

SUMÁRIO

1	INTRODUÇÃO.....	9
1.1	JUSTIFICATIVA	10
1.2	OBJETIVOS	10
1.2.1	Objetivo Geral	10
1.2.2	Objetivos Específicos.....	10
1.3	SOLUÇÃO PROPOSTA	11
1.4	METODOLOGIA	12
1.5	ESTRUTURA DO TRABALHO	13
2	FUNDAMENTAÇÃO TEÓRICA.....	14
2.1	MICROCONTROLADORES.....	14
2.1.1	Arduino	15
2.1.2	NodeMCU	16
2.1.3	NodeMCU-32S.....	17
2.1.4	Wifi LoRa 32.....	18
2.2	COMUNICAÇÃO WIRELESS	19
2.2.2	Wifi	20
2.2.1	Rádio 900 Mhz.....	21
2.2.2	LoRa	22
2.2.3	LoRaWAN.....	22
2.2.4	Comparativo Rádio 900Mhz x LoRa.....	24
2.3	MODULAÇÃO.....	25
2.3.1	Modulação Analógica.....	26
2.3.2	Modulação Digital	28
2.4	COMUNICAÇÃO DE DADOS	31
2.4.1	Modelo OSI.....	32
2.4.2	Protocolos de Comunicação.....	34
2.5	SISTEMAS DE SUPERVISÃO	37
2.5.1	Virtuino Modbus	39
3	DESENVOLVIMENTO.....	40
3.1	ESCOPO DO PROJETO	40
3.2	DESCRIÇÃO DO HARDWARE	42
3.2.1	Módulo A - <i>Endpoint</i>	42

3.2.2 Módulo B - <i>Gateway</i>	43
3.3 DESCRIÇÃO DO CÓDIGO.....	44
3.4 TESTES DE CAMPO.....	46
4 CONCLUSÃO.....	51
ANEXOS	52
APÊNDICES	53

1 INTRODUÇÃO

Sistemas de aquisição de dados são amplamente utilizados por cientistas e engenheiros nos laboratórios de pesquisa para medições e testes. Atualmente, estes sistemas também estão presentes em aplicações industriais com a expansão do uso de dispositivos inteligentes para sensoriamento, processamento e comunicação, convergindo para o conceito de telemetria e *Internet of Things* - IoT. A telemetria pode ser definida como o processo de aquisição de dados de equipamentos remotos. (EMILIO, 2013)

Segundo (PARK, 2003, tradução nossa):

Aquisição de dados (*Data Acquisition System* – DAQ) é o processo pelo qual um fenômeno físico proveniente do mundo real é transformado em sinais elétricos que são medidos e convertidos em formato digital para ser processado, analisado e armazenado por um computador.

O sistema de aquisição de dados é constituído por três elementos, são eles: interface de aquisição de dados no meio físico (transdutores), forma de transmissão e recepção e a interface de usuário (*software*). (EMILIO, 2013)

A aquisição remota de dados é uma demanda crescente de empresas que possuem unidades operacionais distribuídas geograficamente e necessitam controlar e monitorar processos produtivos.

O sistema de telemetria da Companhia Catarinense de Águas e Saneamento - CASAN utiliza tecnologia de comunicação via rádio para monitoramento e controle remoto de poços de captação e da estação de tratamento de água do Campeche. Porém o crescimento urbano de Florianópolis tem sido o maior desafio a ser vencido tendo em vista que a construção de novos prédios obstrui a visada dos links de rádio existentes. Os equipamentos adotados nestes locais não possuem especificações adequadas para ultrapassar os obstáculos. Diante da problemática exposta, este trabalho visa estudar a tecnologia de rádio *Long Range* (LoRa) como alternativa para tentar solucionar o problema, tendo em vista suas características técnicas de funcionamento.

A tecnologia de rádio LoRa é bastante atual e tem recebido boas críticas por parte dos utilizadores, além de possuir um longo alcance mesmo com a presença de obstáculos de acordo com seus dados de fabricação.

1.1 JUSTIFICATIVA

A Companhia Catarinense de Águas e Saneamento – CASAN realiza o monitoramento e controle remoto de seus quadros de comando de poços de captação e da estação de tratamento de água no Bairro Campeche, na cidade de Florianópolis utilizando comunicação via rádio entre esses pontos. Entretanto com o crescimento urbano, novas construções estão impedindo o correto funcionamento desses equipamentos de rádio.

Em virtude disso, este trabalho irá estudar a viabilidade de implantação de um equipamento ESP32 com rádio LORA em um poço de captação e outro equipamento em uma estação de tratamento de água, afim de se monitorar e controlar o quadro de comando do poço pela estação e posteriormente utilizar o equipamento presente na estação como gateway para enviar as informações para um software supervisor, de modo a permitir, que esse quadro de comando possa ser monitorado via internet.

1.2 OBJETIVOS

Para validação do tema proposto, foram definidos o objetivo geral e três objetivos específicos, apresentados a seguir.

1.2.1 Objetivo Geral

Analisar a viabilidade de implantação de dois módulos ESP32 com rádio LoRa utilizando os protocolos de comunicação LoRaWAN e Modbus TCP/IP para realizar o monitoramento de um quadro de comando presente no Bairro Campeche na Cidade de Florianópolis remotamente por meio do software supervisor para dispositivos *mobile* chamado Virtuino.

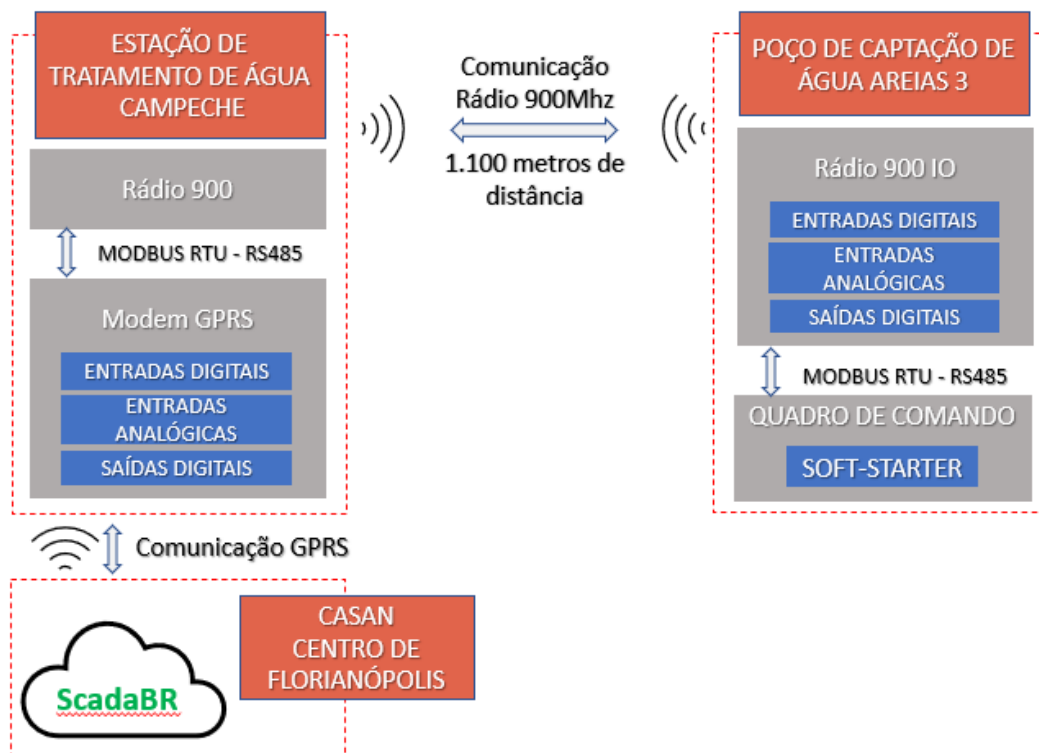
1.2.2 Objetivos Específicos

- a) Estudar a tecnologia LoRaWAN;
- b) Elaborar um protótipo funcional para testes práticos;
- c) Monitorar variáveis do quadro de comando do poço de captação de água bruta;

1.3 SOLUÇÃO PROPOSTA

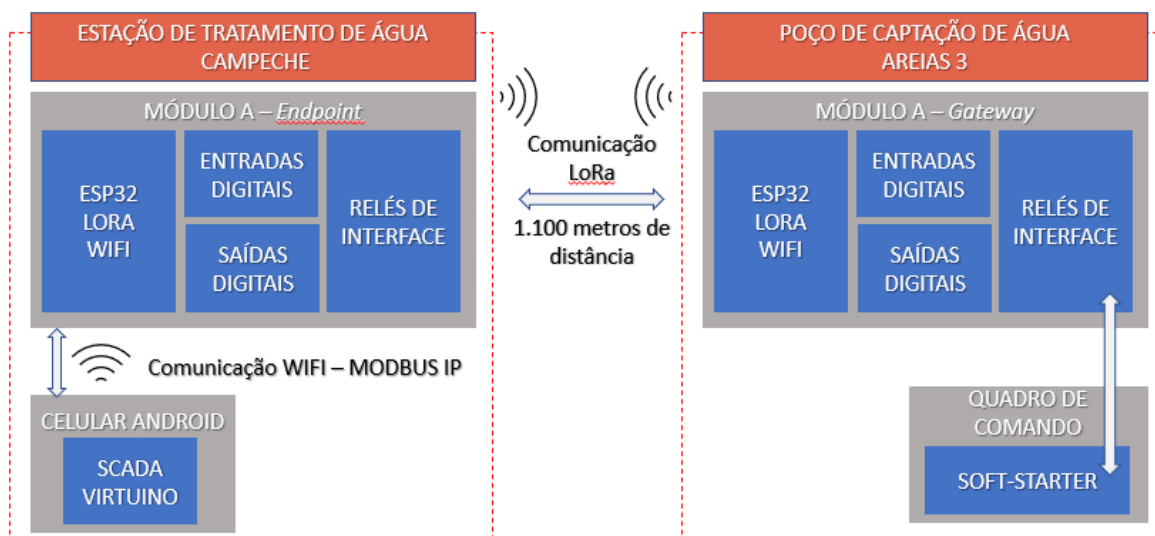
O sistema de comunicação existente entre a Estação de Tratamento de Água Campeche e o Poço de Captação Areias 3, ambos localizados no bairro Campeche em Florianópolis, contam com equipamentos de comunicação rádio em 900Mhz conforme apresentado pela Figura 1. A solução proposta para substituição da comunicação rádio em 900Mhz pode ser observada na Figura 2, onde foram utilizados módulos ESP32 e LoRa integrados para realizar a comunicação entre as duas unidades que estão 1.100 metros distantes.

Figura 1 - Sistema existente de comunicação sem fio



Fonte: Elaboração dos autores (2020).

Figura 2 - Escopo de projeto



Fonte: Elaboração dos autores (2020).

1.4 METODOLOGIA

Foi utilizada como metodologia para elaboração desse projeto, a pesquisa em referências bibliográficas através da leitura de livros, artigos e de outros trabalhos acadêmicos. Segundo (ALYRIO, 2009):

A pesquisa bibliográfica é o passo inicial na construção efetiva do processo de investigação, quer dizer, após a escolha de um assunto é necessário fazer uma revisão bibliográfica do tema apontado. Essa pesquisa auxilia na escolha em um método mais apropriado, assim como em um conhecimento das variáveis e na autenticidade da pesquisa.

Em conjunto, ao método descrito acima, foi aplicada a pesquisa experimental, que consiste segundo (FONTELLES *et al.*, 2009):

Neste tipo de pesquisa, o investigador seleciona as variáveis que serão estudadas, define a forma de controle sobre elas e observa os efeitos sobre o objeto de estudo, em condições pré-estabelecidas. Assim, pelo fato das variáveis, ou da variável, poderem ser manipuladas pelo pesquisador, equívocos e vieses praticamente desaparecem, sendo, por esta razão, considerada como o melhor tipo de pesquisa científica, pois proporciona maior confiabilidade em seus resultados.

A partir da aplicação dessas metodologias, elaborou-se esse trabalho acadêmico, integrando pesquisa bibliográfica e experimental, de forma a desenvolver uma solução que atendesse os objetivos propostos.

1.5 ESTRUTURA DO TRABALHO

Este trabalho está estruturado da seguinte forma: no capítulo 1 foi desenvolvida a introdução, sendo complementada pela metodologia de pesquisa, objetivos, justificativa e estrutura. O capítulo 2 contempla a revisão bibliográfica que aborda temas relevantes ao desenvolvimento deste projeto, como o conceito e tipos de microcontroladores, tecnologias de comunicação Wireless, tipos de modulação, protocolos de comunicação e sistemas de supervisão. Já o capítulo 3 discorre acerca do desenvolvimento, abordando desde o protótipo desenvolvido, hardware e firmware, os testes realizados e os resultados obtidos e em seguida o capítulo 4 traz a conclusão concebida a partir de todas as informações obtidas na aplicação do protótipo desenvolvido.

2 FUNDAMENTAÇÃO TEÓRICA

Neste tópico serão tratados temas essenciais para o desenvolvimento desse trabalho, como o hardware utilizado, as tecnologias empregadas, o aplicativo de monitoramento e controle aplicado.

2.1 MICROCONTROLADORES

Com o aprimoramento tecnológico, os componentes eletrônicos sofreram grandes avanços ao longo dos anos. As placas eletrônicas antes compostas por inúmeros circuitos digitais foram substituídas por placas com menos componentes, tornando-as menores e com custo menor, mediante os avanços obtidos nos microcontroladores. Os microcontroladores consistem em um único chip composto por um sistema computacional completo (processador, memória, periféricos de entrada e saída, temporizadores, conversores A/D, interface serial, módulos de temporização, entre outros). Este único *chip* pode ser programado de modo a executar inúmeras tarefas distintas, de acordo com a sua aplicação, dentre elas, pode-se citar a sua aplicação em fechaduras eletrônicas, centrais de alarme residencial, brinquedos, eletrodomésticos, sistemas embarcados em geral.

E são vastas as aplicações para esses componentes, estando presentes em grande parte dos dispositivos eletrônicos, como: fornos micro-ondas, geladeiras, televisores, elevadores, brinquedos, controles remotos, e diversos outros exemplos de sistemas embarcados. Para cada tipo de equipamento em que o microcontrolador será utilizado, deve ser selecionado o microcontrolador a ser aplicado de acordo com as características necessárias para cada projeto, como: memória, frequência de *clock*, quantidade de portas de entrada e saída, interfaces de comunicação, custo, tamanho. Além disso, cada microcontrolador terá uma programação distinta de acordo com a sua aplicação. (CARRO; WAGER, 2019)

A grande vantagem dos microcontroladores é a possibilidade de alterar o código gravado nesses *chip's* repetidas vezes. O projetista do produto tem a possibilidade de gravar e regravar o código sempre que necessário e poderá executar novas atualizações. De modo que esses *chip's* são considerados extremamente versáteis.

Com base nessas características, é importante selecionar o microcontrolador ideal de acordo com o projeto que está sendo desenvolvido. Atualmente existe uma grande variedade de microcontroladores de diferentes fabricantes, como: Microchip, Texas Instruments, Atmel, Espressif, dentre outros grandes fabricantes no mercado. Nos tópicos seguintes são

apresentados os hardwares mais conhecidos pela comunidade *maker* em eletrônica atualmente.

2.1.1 Arduino

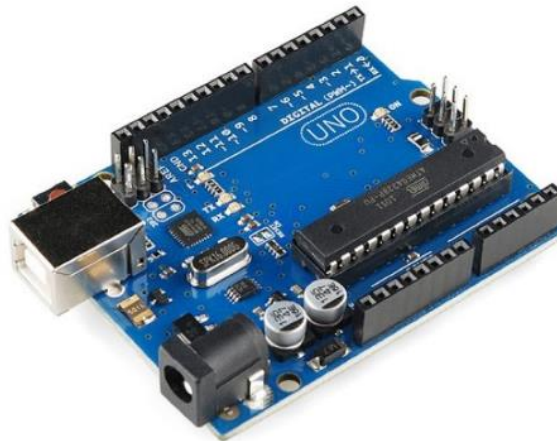
Os microcontroladores Atmel AVR se tornaram amplamente difundidos no meio da eletrônica, devido a estarem embutidos em placas Arduino, são alguns deles: ATmega8 (Arduino Extreme), ATmega168 (Arduino Mini), ATmega328 (Arduino Duemilanove, Arduino Uno, Arduino Nano), SAMD21 (Arduino Zero). Essas placas consistem em hardwares livres compostos pelo microcontrolador juntamente com os seus periféricos: cristal, conexão para alimentação, bateria, etc. É um hardware pronto e de fácil utilização. Sendo necessário apenas conectá-lo em um computador e começar a programar. Este hardware possui IDE de programação própria, entretanto pode ser programado também em outras IDEs, aceita linguagem C/C++ e possui uma grande comunidade com diversas bibliotecas prontas disponíveis para utilização. (MOTA, 2017)

Além disso, ele pode ser integrado com os chamados *Shields*, que correspondem a outras placas com o objetivo de aumentar suas funcionalidades. São placas de Wifi, *Bluetooth*, ZigBee, GPS, GPRS, Sensor Ultrassônico, Sensor Piezoelétrico, *Display* de LCD, controle *Joystick*, por exemplo. Essas placas são conectadas sobre a placa com o microcontrolador ampliando as funções do mesmo e tornando o seu uso mais simples e prático. Possibilitando até que pessoas com pouco conhecimento em eletrônica consigam desenvolver projetos mais versáteis.

São inúmeros modelos de placa Arduino, compostos por diferentes microcontroladores. Na Figura 3 - Arduino UNO, é possível observar uma das placas Arduino mais utilizadas, a qual é denominada Uno, que possui um microcontrolador modelo ATmega328. Ele possui as seguintes características:

- Microcontrolador: ATmega328
- Tensão de Operação: 5V
- Tensão de Entrada: 7-12 V
- Portas Digitais: 14 (6 podem ser usadas como PWM)
- Portas Analógicas: 6
- Corrente Pinos I/O: 40mA
- Corrente Pinos 3,3V: 50mA
- Memória *Flash*: 32KB (0,5KB usado no *bootloader*)
- SRAM: 2KB
- EEPROM: 1KB
- Velocidade do *Clock*: 16MHz. (SOUZA, 2020)

Figura 3 - Arduino UNO



Fonte: EMÍLIO (2013).

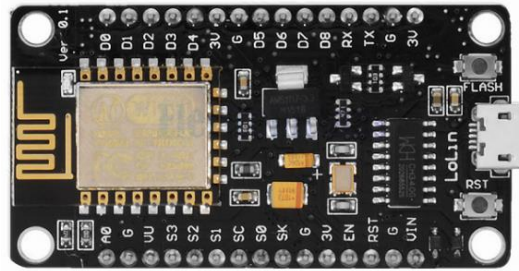
2.1.2 NodeMCU

O NodeMCU é um *hardware*, também *open source*, que possui o microcontrolador ESP8266, consiste em uma plataforma amplamente utilizada em projetos de IOT (Internet das Coisas). Ele possui conversor USB integrado para facilitar a troca de informações com o computador e possui Wifi Nativo. (OLIVEIRA, 2019)

Algumas das características principais desta placa, são:

- Processador ESP8266-12E
- Arquitetura RISC de 32 bits
- Processador pode operar em 80MHz/160MHz
- 4Mb de memória flash
- 64Kb para instruções
- 96Kb para dados
- WiFi nativo padrão 802.11b/g/n
- Alimentação de 5 Vdc através do conector micro USB
- Possui 11 pinos digitais
- Possui 1 pino analógico com resolução de 10 bits
- Possui interrupção, PWM, I2C e one wire
- Pinos operam em nível lógico de 3.3V
- Possui conversor USB Serial integrado
- Programável via USB ou WiFi (OTA)
- Compatível com a IDE e módulos e sensores da placa Arduino. (OLIVEIRA, 2019)

Figura 4 - NodeMCU



Fonte: OLIVEIRA (2019).

2.1.3 NodeMCU-32S

O hardware NodeMCU-32S é uma placa composta por um microcontrolador ESP32 fabricado pela Expressif. Este microcontrolador possui baixo custo e baixo consumo de energia, e é uma evolução do ESP8266, possui mais processamento, memória e recursos que seu antecessor. O ESP32 possui ainda Wifi e Bluetooth integrados. Ele pode ser programado em diversas IDEs, como a do Arduino, Codeblocks, Lua, Netbeans, entre outras. (MORAIS, 2017)

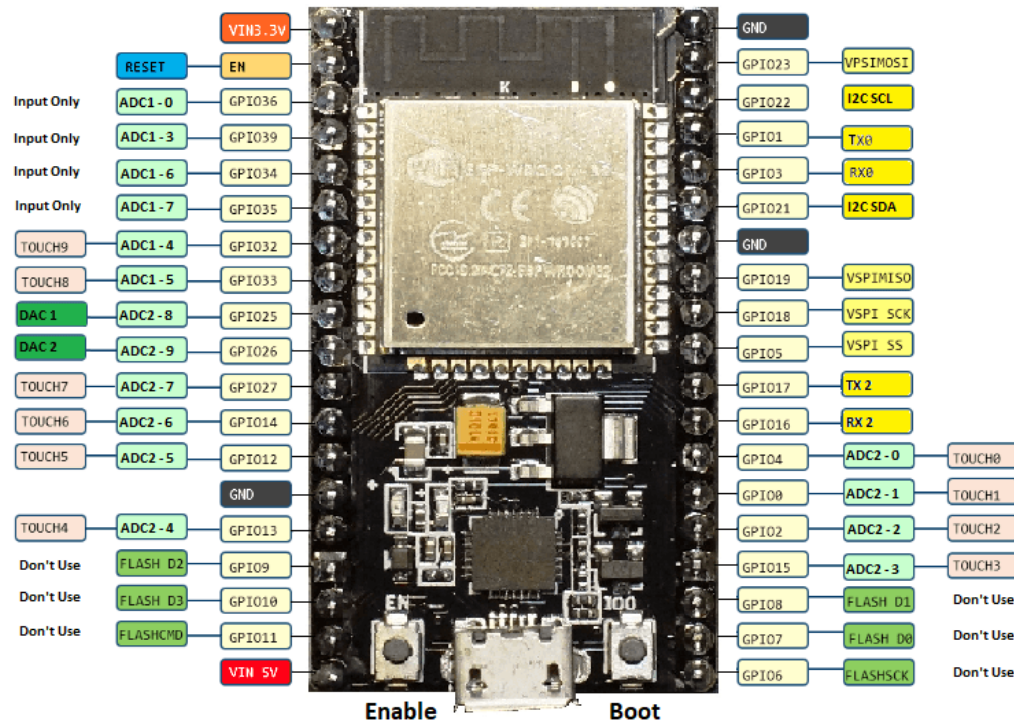
As características da placa NodeMCU-32S são as seguintes:

- CPU: Xtensa® Dual-Core 32-bit LX6;
- Memória ROM: 448 KBytes;
- Clock máximo: 240MHz;
- Memória RAM: 520 Kbytes;
- Memória Flash: 4 MB;
- Wireless padrão 802.11 b/g/n;
- Conexão Wifi de 2.4Ghz (máximo de 150 Mbps);
- Antena embutida na placa;
- Conector micro USB para comunicação e alimentação;
- Wi-Fi Direct (P2P), P2P Discovery, P2P Group Owner mode e P2P Power Management.
- Modos de operação: STA/AP/STA+AP;
- Bluetooth BLE 4.2;
- Portas GPIO: 11;
- GPIO com funções de PWM, I2C, SPI etc.;
- Tensão de operação: 4,5 ~ 9V;
- Conversor analógico digital (ADC). (MORAIS, 2017)

Além disso, o ESP32 possui 34 GPIOs programáveis, até 18 canais de ADC (12 bits de resolução), dois DAC de 8 bits, suporte a 10 sensores *touch*, 4 interfaces SPI, 2 interfaces I2S, 2 interfaces I2C, 3 interfaces UART, 1 sensor de efeito hall, 1 interface IR (RX/TX), 1

interface ethernet, entre outros. Na Figura 5 - Módulo ESP32, vê-se a imagem desse módulo com seus pinos e respectivas funções. (MURTA,2018)

Figura 5 - Módulo ESP32



Fonte: ESP32 (2019)

2.1.4 Wifi LoRa 32

Fabricado pela HELTEC, esse módulo contém além do ESP32 o chip de rádio LORA integrado e vem com um display OLED e conector JST na parte inferior para uma bateria (Li-Ion ou Li-Po) de 3.7V e até 1000mAh para alimentação remota.

O rádio LoRa presente neste módulo possui as seguintes características: sensibilidade de recepção de -139 dBm utilizando os chips SX1276 e SX1278, possui alcance máximo em região aberta de cerca de 3 km utilizando a antena oferecida no módulo, frequência de 868/915 MHz para o chip SX1276 e 433/470 MHz para chip SX1278. Na Figura 6 - Módulo ESP32 WIFI LORA da HEALTEC, têm-se a imagem do kit (ESP32 com LORA e antena), e no ANEXO A, é apresentada a pinagem do módulo ESP32 com LoRa. (BERTOLETI, 2019)

Em virtude das características apresentadas da placa Wifi LoRa 32, ela foi escolhida para o desenvolvimento desse projeto de TCC. Essa placa possui baixo custo, IDE de programação amigável e todas as funcionalidades necessárias já integradas (LoRa e Wifi) de modo a ter-se tornado o hardware utilizado neste projeto.

Figura 6 - Módulo ESP32 WIFI LORA da HEALTEC



Fonte: BERTOLETI (2019).

2.2 COMUNICAÇÃO WIRELESS

O meio físico onde transitam os sinais elétricos ou eletromagnéticos que transportam a informação entre um transmissor e um receptor é denominado canal ou meio de comunicação. (MEDEIROS, 2007)

Segundo (MEDEIROS, 2007):

Em telecomunicações o termo canal possui pelo menos três significados:

1. Usado para especificar um meio de comunicações: fio, rádio ou fibra óptica. Os materiais empregados dão nome aos meios de comunicação. [...]
3. Segmento do espectro de frequências com largura de banda (bandwidth) B ou faixa ocupada por uma transmissão de rádio.

Canais com pelo menos dois fios são classificados como um meio de comunicação guiada pois os sinais da informação transitam de forma conduzida no interior do fio.

Tendo em vista a ampla aplicação do meio de comunicação *wireless* e as vantagens que ela apresenta quando à mobilidade e flexibilidade, este projeto utiliza além do canal principal em LoRa, um segundo canal *WiFi* com o protocolo Modbus IP para a interação entre o módulo *gateway* e o telefone celular que roda a aplicação do sistema supervisorio Virtuino.

2.2.2 Wifi

A palavra do inglês *Wireless* significa sem fio, ou seja, corresponde a qualquer tipo de conexão para transmissão de informação sem a utilização de fios ou cabos. Ela permite o compartilhamento de informações entre dois ou mais dispositivos através de ondas de rádio. O surgimento das primeiras formas de comunicação sem fio, remontam o século passado, quando em 1897, Marchese Guglielmo Marconi (engenheiro, físico e inventor italiano) fez demonstrações do envio de informações a uma distância de 12 milhas utilizando sinais telegráficos. (MEDEIROS, 2007) Desde então essa forma de comunicação foi aprimorada, surgiram os primeiros equipamentos de rádio e posteriormente a tecnologia *Wi-Fi* (*Wireless Fidelity*) que foi responsável pela disseminação em massa dessa forma de comunicação.

Na atualidade as redes wireless já são amplamente aplicadas pois elas oferecem inúmeras vantagens em relação as redes cabeadas, como: mobilidade, flexibilidade, rapidez, facilidade e menor custo de implantação. Além disso, há locais em que a instalação de cabos é inviável, tornando-se a única opção de comunicação. A comunicação sem fio é utilizada tanto em sistemas mais simples, como em controles remotos de televisores, ar condicionado, até em sistemas mais complexos como a automação de equipamentos dentro de uma indústria.

Para que essa versatilidade de uso da tecnologia wireless seja possível, existem duas formas de transmissão da informação: infravermelho e ondas de rádio. O infravermelho corresponde a um tipo de radiação não ionizante e que não pertence ao espectro de luz visível. A transmissão se dá por meio de pulsos de luz infravermelha entre um emissor e um receptor, com capacidade de transmissão de no máximo 5 metros e que não consegue atravessar corpos sólidos. Essa forma de comunicação tem como vantagens o baixo custo dos componentes utilizados na transmissão e a resistência a interferência de aparelhos elétricos próximos. Já as desvantagens referem-se à interrupção da comunicação com o surgimento de um objeto entre emissor e receptor, ou com o desalinhamento destes e a sua baixa velocidade (mesma de uma porta USB). (JÚNIOR, 2020)

A comunicação via ondas de rádio trouxe melhorias nessa forma de comunicação. Uma vez que não dependia mais do alinhamento entre emissor e receptor e tornou possível a transmissão de informações em distancias muito maiores. O sinal da informação a ser transmitida via radiofrequência é incorporado em uma frequência portadora e então é enviado através de uma antena transmissora na forma de ondas eletromagnéticas. Por sua vez, a antena receptora capta essas ondas eletromagnéticas, envia ao receptor que recupera a informação transmitida.

Conforme (MACHADO, 2014):

Com o crescimento do uso de redes sem fio, surgiu a necessidade de se definir padrões para a comunicação entre equipamentos e dispositivos. Atualmente o mais conhecido é o padrão IEEE 802.11, que define especificações das funções de Controle de Acesso ao Meio (MAC – *Medium Access Control*) e Camada Física (PHY – *Physical Layer*) para comunicação em uma Rede Local sem Fio (WLAN – *Wireless Local Area Network*), onde a comunicação entre os dispositivos não ultrapasse a distância de 100m. Este padrão provê conectividade sem fio por estações fixas, portáteis e móveis dentro de uma rede local e estão bastante difundidas devido à facilidade de instalação e suporte à mobilidade.

A comunicação wireless possibilitou grande avanço tecnológico, e é imprescindível principalmente nesta era de internet das coisas (*Internet of Things* - IoT). O termo IoT foi criado na década de 1990, mas somente nos últimos anos tem se tornado expressivo na sociedade. A internet das coisas corresponde a interconexão de dispositivos inteligentes, capazes de controlar, trocar ou compartilhar informações e dados, sem a necessidade de interação física do ser humano. Essa forma de comunicação está fortemente integrada na sociedade atual e se tende a tornar-se cada vez mais presente. (CARRION; QUARESMA, 2019)

A tecnologia IoT juntamente com a comunicação via radiofrequência correspondem ao cerne desse projeto e, portanto, serão abordadas duas tecnologias de rádio utilizadas, sendo descrita suas características e peculiaridades, a fim de se verificar as principais diferenças entre elas.

2.2.1 Rádio 900 Mhz

O rádio 900 Mhz utilizado possui as seguintes características segundo o fabricante:

Tecnologia empregada	Frequência	Modulação	Potência de saída	Sensibilidade
FHSS	915 a 928 MHz	GFSK	20 dBm	melhor que -100 dBm

Atualmente a CASAN – Companhia Catarinense de Águas e Saneamento, utiliza o equipamento de rádio descrito acima, para realizar o monitoramento e controle do seu sistema de captação e tratamento de água do bairro Campeche em Florianópolis. Entretanto este sistema que outrora desempenhava bem o seu papel, está atualmente prejudicado devido ao crescimento urbano da região. Uma vez que, os novos prédios construídos estão servindo como obstáculos para o enlace da comunicação dessa tecnologia de rádio.

2.2.2 LoRa

À medida que ocorreu a evolução tecnológica, as técnicas de comunicação *wireless* também precisaram evoluir. Principalmente neste momento em que há uma grande integração dos equipamentos eletroeletrônicos, transmitindo e recebendo informações constantemente, é imprescindível o uso de tecnologias mais modernas.

Nesse contexto, têm-se aplicado uma nova forma de comunicação por radiofrequência conhecida como LoRa. A tecnologia de radiofrequência LoRa faz parte de um agrupamento de redes chamado LPWAN (*Low Power Wide Area Network*), que traduzindo significa “Rede de área ampla e baixa potência”, ou seja, corresponde a comunicação a longa distância e com baixo consumo de energia. (ALMEIDA, 2019) Essa tecnologia foi criada pela Semtech Corporation e promovida através da *LoRa Alliance*, que consiste em uma associação aberta e sem fins lucrativos de grandes companhias ao redor do mundo, que trabalham para o desenvolvimento e a aplicação da rede LoRa na Internet das Coisas. (TEIXEIRA; ALMEIDA, 2017)

A tecnologia LoRa é semelhante as tecnologias *Wifi* e *Bluetooth*, porém tem como principal diferença a possibilidade de comunicação em longas distâncias. O nome LoRa é uma abreviação de *Long Range*, que significa “longo alcance”.

É bastante aplicado nos sistemas de IoT para aquisição de informações e controle remoto de sensores (de pressão, luz, *on-off*, temperatura, entre outros), principalmente para aqueles operados a bateria, e que ficam em locais de difícil acesso. (SILVA JUNIOR, 2016) O LoRa tem como característica a capacidade de transmissão de informação em longas distâncias, entretanto, não é possível transmitir voz ou vídeo. O conceito do LoRa consiste em transmissão de poucos dados (até 240 bytes) usando pouca energia. Ele possibilita executar aplicações que não eram possíveis com o GSM, Wifi e Bluetooth, uma vez que o Wifi e Bluetooth funcionam com alcance reduzido e a rede celular consome muita energia. (ALMEIDA, 2019)

2.2.3 LoRaWAN

Enquanto o LoRa consiste em uma metodologia para modulação dos sinais de rádio, o LoRaWAN é o protocolo utilizado na comunicação dessa tecnologia. Ele é responsável pelos detalhes de funcionamento, qualidade do serviço, segurança, otimização dos equipamentos na

rede. Permitindo a interoperabilidade entre diferentes *end-points* e *gateways*, pois nem todos são produzidos pelo mesmo fabricante. (BERTOLETI, 2019)

O LoRaWAN é um protocolo aberto, com ele é possível criar uma rede completa com endereçamento de dispositivos, *gateways* e mecanismos anticisão de pacotes. (BERTOLETI, 2019)

Esse protocolo conforme pode ser visto na Figura 7 - Arquitetura da rede LORA, tem topologia estrela, onde cada dispositivo na rede se conecta a um ponto central e sua arquitetura é composta por 4 elementos básicos, são eles:

Os *endnodes* são os dispositivos conectados na rede através dos módulos LoRa, como os sensores. (ALMEIDA, 2019) Já os dispositivos *Gateways* funcionam como concentradores, eles recebem as informações dos *endnodes* e encaminham essas informações para a internet. Eles se conectam ao servidor de rede através de conexões IP e agem como uma ponte transparente, somente convertendo pacotes de rádio frequência em pacotes IP e vice-versa. Na rede LoRaWAN, os nós não estão associados a um *gateway* específico, os dados transmitidos por um nó podem ser recebidos por vários gateways e o *gateway* por sua vez, transmite a informação recebida para o servidor de rede. (LORA® ALLIANCE, 2005)

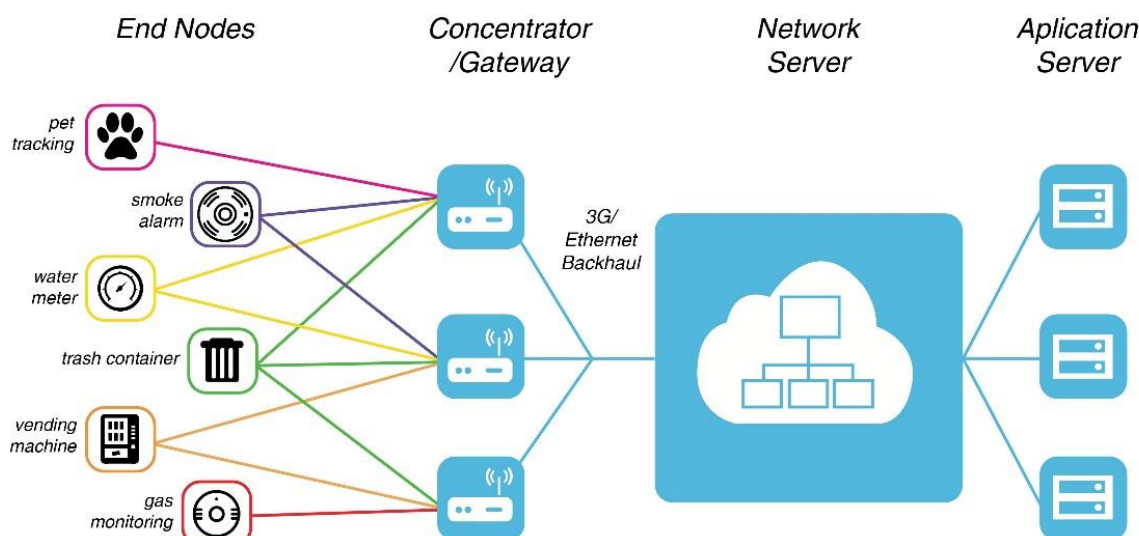
Os servidores de rede recebem, armazenam e gerenciam os dados recebidos dos gateways.

E os servidores de aplicações provêm o acesso aos dados para os clientes. (ALMEIDA, 2019)

Os nós da rede são assíncronos, se comunicando apenas quando estiverem prontos para enviar seus dados ou se forem programados, método chamado de Aloha. Diferentemente da rede síncrona, onde os nós precisam “acordar” para sincronizar com a rede e verificar se há mensagens, gerando um maior gasto de energia, reduzindo a vida útil da bateria dos dispositivos. Para que as redes em estrela de longo alcance sejam viáveis, o gateway precisa ter uma capacidade muito alta para receber uma grande quantidade de mensagens dos inúmeros nós. (LORA® ALLIANCE, 2005)

O LoRa opera em faixas de frequência não licenciadas, no Brasil ele opera na frequência de 915MHz. Uma das vantagens de se utilizar uma frequência não licenciada é a diminuição dos custos de implantação dessa rede, entretanto têm-se como desvantagem a possibilidade de interferência por outros sinais presentes também nessa frequência.

Figura 7 - Arquitetura da rede LORA



Fonte: ALMEIDA (2019).

Para amenizar esse problema, o LoRaWAN utiliza uma técnica de modulação chamada de *Chirp Spread Spectrum* (CSS), que codifica o dado a ser enviado, o divide em vários pedaços que são transmitidos separadamente por todo espectro de frequência da largura de banda disponível. Quanto menor a largura de banda e maior o fator de espalhamento do sinal, menor a taxa de perda de pacotes.

Além disso, outro mecanismo para correção de erros também é utilizado. Esse mecanismo é chamado de *Cyclic Redundancy Check* (CRC), que é aplicado em cada pedaço de mensagem.

Contudo, deve-se atentar para o fato de que o uso desses mecanismos de redução de erros e redução de perda de pacotes impactam no tempo total de transmissão da mensagem, de forma que quanto mais robusto for configurada a transmissão maior será o tempo para transmiti-la e consequentemente menor será a quantidade de dados enviados, mas em contrapartida, maior será a sensibilidade do receptor, permitindo que a distância do transmissor para o receptor seja maior ou que a comunicação se estabeleça mesmo com a presença de mais obstáculos entre os dois. (GRIESE; KLEINSCHMIDT, 2019)

2.2.4 Comparativo Rádio 900Mhz x LoRa

Abaixo destacam-se as principais características dos dois modelos de equipamento de rádio, o rádio utilizado atualmente na CASAN e o rádio utilizado nesse projeto:

Tabela 1 - Comparação entre duas tecnologias de rádio

Características	Rádio 900 Mhz	LoRa
Frequência de Operação	915 a 928 MHz	915Mhz
Modulação	GFSK	CSS
Potência de saída	20 dBm	20 dBm
Sensibilidade	melhor que -100 dBm	até - 139 dBm

Fonte: Elaboração dos autores (2020).

Através dessa tabela pode-se perceber que a principal diferença entre as duas tecnologias de rádio, corresponde ao tipo de modulação:

A modulação GFSK funciona de modo semelhante a modulação FSK, ou seja, varia a frequência da onda portadora de acordo com o bit a ser transmitido. Já a modulação CSS foi por muito tempo, utilizada exclusivamente pela área militar. Ela codifica o dado a ser enviado, o divide em vários pedaços que são transmitidos separadamente por todo espectro de frequência da largura de banda disponível. Possui robustez a degradações do canal e resistência contra efeito Doppler. (GRIESE; KLEINSCHMIDT, 2019) Esses conceitos sobre modulação serão discutidos mais a fundo no tópico seguinte.

2.3 MODULAÇÃO

Segundo (SBIZERA, 2003):

A modulação é o processo através do qual a informação a ser transmitida é convertida em uma forma conveniente para sua transmissão. Geralmente, esse processo envolve a translação da banda básica de informação em bandas de frequência muito mais altas, nas quais efetivamente ocorrerá a transmissão. O sinal original, ou seja, a informação propriamente dita, é chamado sinal modulador. O sinal resultante do processo de modulação é chamado sinal modulado. No receptor ocorre o processo inverso, no qual se extrai a informação do sinal modulado. Esse processo é conhecido por demodulação.

Ou seja, o sinal a ser enviado (normalmente um sinal de frequência mais baixa), é incorporado a um sinal de frequência mais alta (denominado portadora). Isso é necessário pois para que um sinal seja irradiado, a antena utilizada precisa ter comprimento igual ao comprimento da onda eletromagnética. Como o comprimento de onda corresponde a velocidade da onda, dividida pela frequência, quanto mais baixa a frequência do sinal, maior

o comprimento da antena. Dessa forma, utiliza-se a onda portadora em alta frequência para possibilitar o uso de antenas menores. Existem algumas formas de modulação, dentre as quais destacam-se a modulação analógica e a digital. (ROMACHO, 2012)

2.3.1 Modulação Analógica

Na modulação analógica, um sinal contínuo é responsável em realizar alterações em um sinal de maior frequência, denominado de onda portadora. Há três tipos de modulação analógica, sendo elas: modulação em amplitude (AM), modulação em frequência (FM) e modulação em fase (PM).

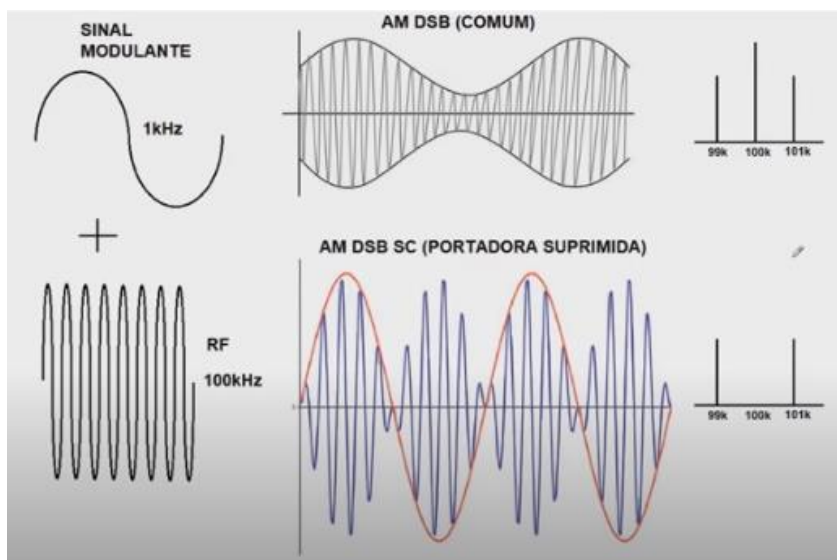
2.3.1.1 Modulação em Amplitude

A modulação em amplitude corresponde a variação da amplitude da onda portadora em função do sinal a ser transmitido, mantendo a frequência e fase constantes. Esse tipo de modulação pode ser dividida em categorias distintas:

- AM/DSB (*Amplitude Modulation / Double Side Band*) - Modulação em Amplitude com Banda Lateral Dupla: esse tipo de modulação transmite a portadora e as duas bandas laterais. (MEDEIROS, 2007)
- AM/DSB-SC (*Amplitude Modulation / Double Side Band - Supressed Carrier*) - Modulação em Amplitude com Banda Lateral Dupla e Portadora Suprimida: esse tipo de modulação surgiu, uma vez que, grande parte da potência era consumida com a transmissão da onda portadora na modulação AM/DSB, sendo que a informação está contida das bandas laterais. Dessa forma, a modulação AM/DSB-SC suprime a portadora afim de economizar potência. A Figura 8 mostra a diferença entre a modulação AM/DSB e AM/DSB-SC. (MEDEIROS, 2007)
- AM/SSB (*Amplitude Modulation / Single Side Band*) - Modulação em Amplitude com Banda Lateral Singela: Este é um tipo particular de modulação AM muito utilizada em comunicações por voz, marítimas, militares e de radioamadores. Além de suprimir a portadora, esse tipo de modulação suprime

também uma das bandas laterais. Essa diferença corresponde a um aumento significativo no rendimento. (MEDEIROS, 2007)

Figura 8 - Modulação AM/DSB e AM/DSB-SC



Fonte: LASKOSKI; MARCONDES; SZEREMETA (2019)

2.3.1.2 Modulação Angular

Como alternativa a modulação em amplitude, surgiu a modulação angular. Esse tipo de modulação pode ser dividido em modulação em frequência e modulação em fase.

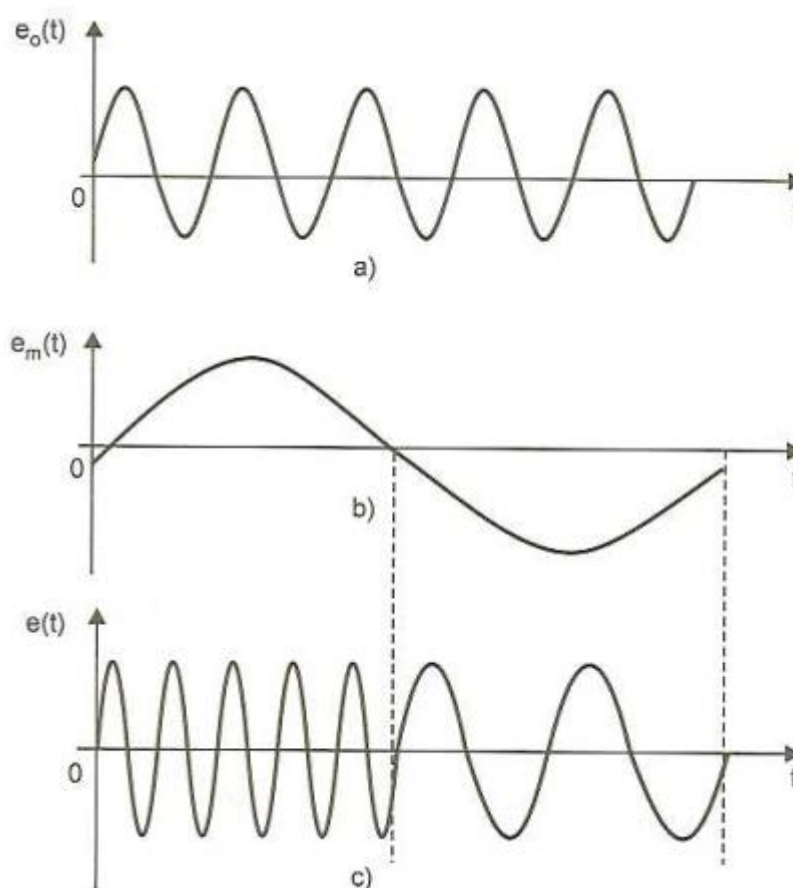
A modulação em fase modifica a fase da portadora, já a modulação em frequência caracteriza-se pela variação da frequência da portadora de acordo com a informação a ser transmitida, mantendo a sua amplitude. Essa forma de modulação surgiu como mecanismo para superar as interferências que afetam a recepção de rádio AM. Sendo menos susceptível a certos tipos de interferências como aquelas causadas por temporais e correntes elétricas parasitas de alguns equipamentos. (GOMES, 2020)

Gomes, Paulo, cita algumas vantagens da modulação FM em relação a AM:

Esses ruídos afetam a amplitude da onda de rádio, mas não sua frequência, assim um sinal de FM permanece virtualmente inalterado. FM é melhor que AM para transmissão de som estereofônico, sinais de áudio de televisão e retransmissão de telefonemas interurbanos por micro-ondas. O total de largura de banda necessário para transmitir um sinal de FM é maior que para AM, o que é um limite para alguns sistemas.

A Figura 9 exemplifica a modulação em FM, no item (a) pode ser vista a forma de onda da portadora, no item (b) o sinal que se deseja transmitir e no item (c) a forma de onda modulada.

Figura 9 - Modulação FM



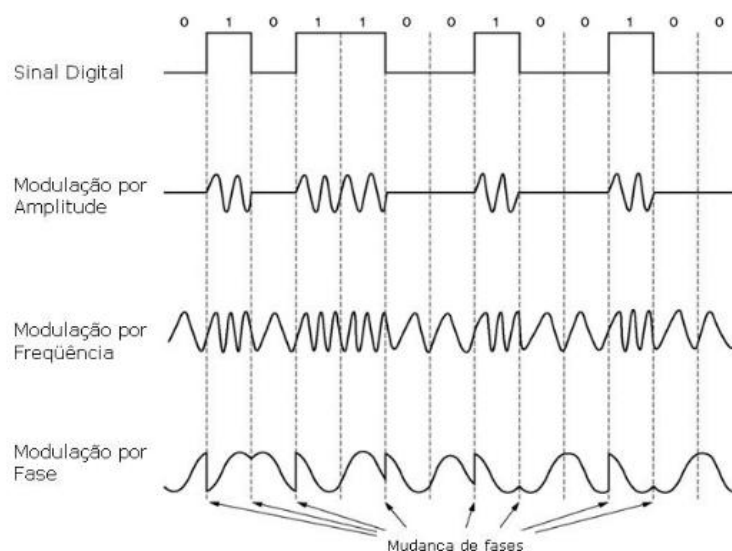
Fonte: MEDEIROS (2007).

2.3.2 Modulação Digital

A modulação digital é usada para transmissão de valores discretos. As principais técnicas de modulação para sinais digitais são: Modulação por chaveamento de amplitude (ASK), modulação por chaveamento de frequência (FSK), modulação por chaveamento de fase (PSK), modulação por chaveamento de fase e amplitude (APSK ou APK). De forma semelhante a modulação analógica, a modulação digital por chaveamento de amplitude altera a amplitude da onda portadora conforme o sinal, a modulação em frequência realiza a variação da frequência da portadora, e a modulação de fase altera a fase. Conforme pode ser visto na Figura 10 - Tipos de Modulação Digital. Cada técnica de modulação apresentada acima, possui suas particularidades.

Nesse trabalho será dada ênfase aos tipos de modulação em frequência, GFSK e CSS, por estarem presentes nas tecnologias de rádio utilizadas. (TELECO, 2020)

Figura 10 - Tipos de Modulação Digital



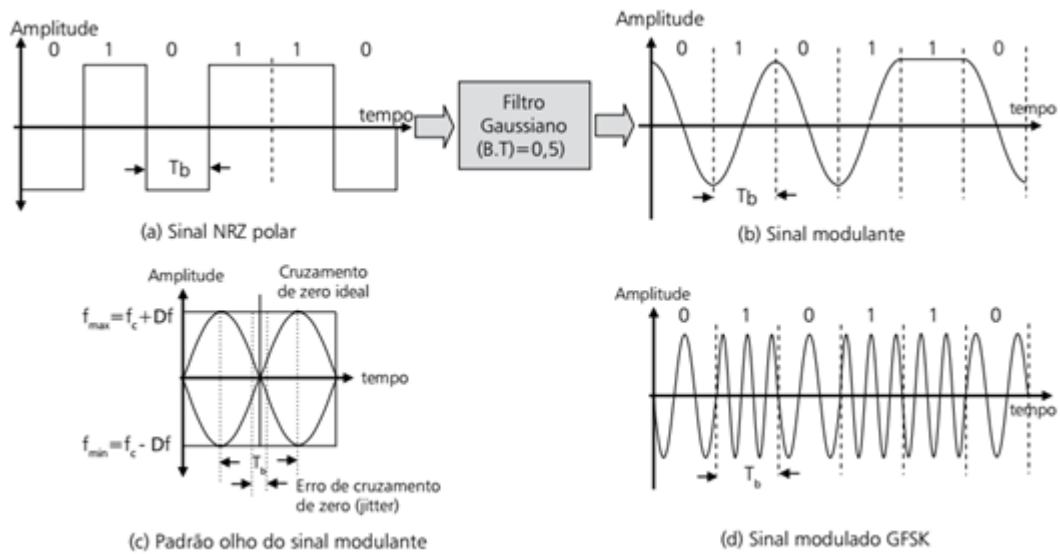
Fonte: FAGUNDES (2020).

2.3.2.1 GFSK

A modulação GFSK também conhecida como modulação Gaussiana FSK, consiste em uma variação da modulação do tipo FSK. A principal diferença entre esses dois tipos de modulação, advém da utilização de um filtro gaussiano, por onde os pulsos passam antes de irem para o modulador. A codificação dos dados ocorre de forma semelhante nos dois tipos de modulação, em virtude disso, pode ser utilizado o mesmo modulador.

O filtro gaussiano é utilizado para suavizar a transição entre os valores dos pulsos (0 e 1). Ele produz uma melhor eficiência espectral em relação a modulação FSK e é utilizado em sistemas Bluetooth. Na Figura 11 - Demonstração do Filtro Gaussiano, é possível visualizar um exemplo de forma de onda após passar pelo filtro gaussiano. (LASKOSKI; MARCONDES; SZEREMETA, 2006)

Figura 11 - Demonstração do Filtro Gaussiano



Fonte: ROCHOL (2018).

2.3.2.2 Spread Spectrum

A modulação SS ou Espalhamento Espectral foi desenvolvido durante a Segunda Guerra Mundial pelos militares com o objetivo de transformar o sinal enviado em algo semelhante a um ruído para diminuir as interceptações inimigas. Segundo (OLIVEIRA; FILHO, 2015):

A técnica de *spread spectrum* consiste em codificar e modificar o sinal de informação executando o seu espalhamento no espectro de frequências. O sinal espalhado ocupa uma banda maior que a informação original, porém possui baixa densidade de potência e, portanto, apresenta uma baixa relação sinal/ruído. Para os receptores convencionais esta comunicação pode até ser imperceptível.

A técnica de modulação *Spread Spectrum* possui a seguintes vantagens:

- Alta resolução devido a faixa espectral utilizada: Com o espalhamento espectral utilizam-se várias faixas de frequência e aumenta-se a resolução do sinal;
- Potência espectral de baixa densidade: A técnica SS utiliza baixa potência de transmissão, diminuindo interferências e consumo de energia;
- Maior duração das baterias e menor impacto ambiental: Como a técnica SS gera menor consumo de energia, as baterias têm maior durabilidade, trazendo benefícios tanto econômica quanto ambientalmente;

- Capacidade *anti-jamming* (interferência gerada): Para interferir em uma comunicação de rádio, é necessário identificar a frequência de transmissão e enviar um sinal mais forte nessa frequência, porém, como no espalhamento espectral o sinal está espalhado, codificado e ainda pode ser utilizada a técnica *Frequency Hopping*, que troca a faixa de frequência utilizada, tornando essa interferência mais difícil de ocorrer. (FRANÇA, 1997)

Existem alguns subtipos de modulações do tipo *Spread Spectrum*, como a *Frequency Hoping Spread Spectrum* - FHSS (espalhamento espectral de salto em frequência), *Direct Sequence Spread Spectrum* – DSSS (espalhamento espectral por sequência direta) e o *Chirp Spread Spectrum* – CSS (espalhamento espectral de *chirp*) utilizado na tecnologia LoRa. (DELLALIBERA, 2018)

A modulação CSS aplica o espalhamento do sinal no espectro de frequência gerando um sinal de portadora de *chirp*. *Chirp* em inglês, *Compressed High Intensity Radar Pulse* é traduzido como Pulso de Radar de Alta Intensidade Comprimido, corresponde a sinais que possuem amplitude constante e varrem toda a largura de banda, alterando a frequência de forma linear em um determinado espaço de tempo. Quando a frequência varia da menor para maior (ocorre o chamado *Up-chirp*), quando a variação é da frequência maior para menor (chama-se *Down-chirp*). (NERY, 2018) Essa modulação tem como características confiabilidade e baixo consumo de potência. (DELLALIBERA, 2018)

2.4 COMUNICAÇÃO DE DADOS

O conceito básico de comunicação de dados define que a existência de uma rede permite que dados possam ser enviados de um lugar a outro. O processo de comunicação de dados entre sistemas remotos pode ser denominada rede e envolve a conexão de computadores, mídia e dispositivos de rede. As redes são divididas em duas categorias que tem diferentes funcionalidades, são elas: redes locais (LANs) e redes remotas (WANs). (FOROUZAN, 2010).

Os modelos de redes servem para organizar, unificar e controlar os componentes de hardware e software das comunicações de dados e das redes. Globalmente existem modelos que estabelecem alguns padrões de comunicação, dentre eles podemos citar o modelo OSI.

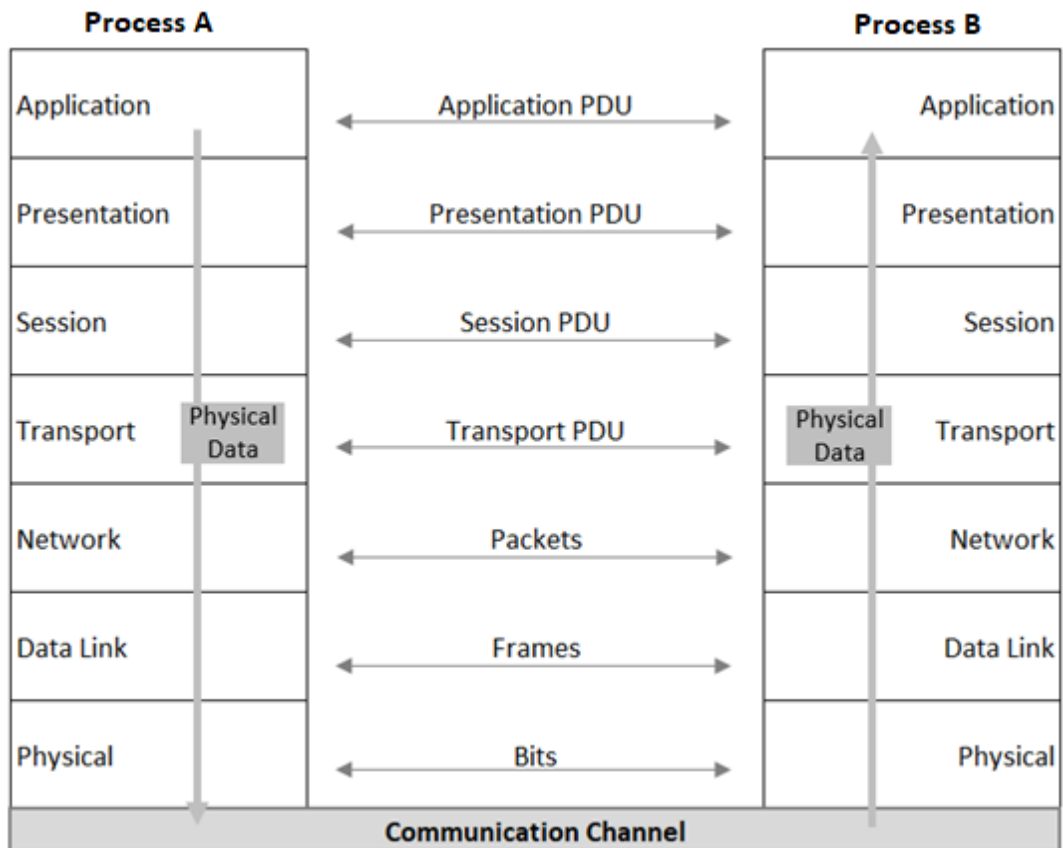
2.4.1 Modelo OSI

Em 1978, diante do crescimento de sistemas de redes fechadas, a Organização Internacional de Normatização (ISO) definiu um modelo de referência para comunicação entre sistemas abertos denominado modelo OSI (ISO7498). O modelo OSI é fundamentalmente uma estrutura de gerenciamento de comunicações de dados, que divide as comunicações de dados em uma hierarquia gerenciável de sete camadas. Cada camada tem o objetivo definido e faz interface com as camadas acima e abaixo dela. A flexibilidade da divisão por camadas possibilita o desenvolvimento de protocolos para cada camada, independentes um do outro. (REYNDERS; MACHAY; WRIGHT, 2005)

O modelo OSI é uma estrutura geral que direciona a escrita de um protocolo ou conjunto de regras para a comunicação de dados. A estrutura defini especificamente as funções e serviços que precisam ser entregues por cada uma das sete camadas ou níveis. O modelo OSI especifica as funções e os recursos esperados em cada camada, porém não descreve como essa funcionalidade deve ser implementada. O foco do modelo está na interconexão entre as camadas e dessa forma foram definidos 7 níveis que são apresentados na Figura 12 - Arquitetura do Modelo OSI. (REYNDERS; MACHAY; WRIGHT, 2005)

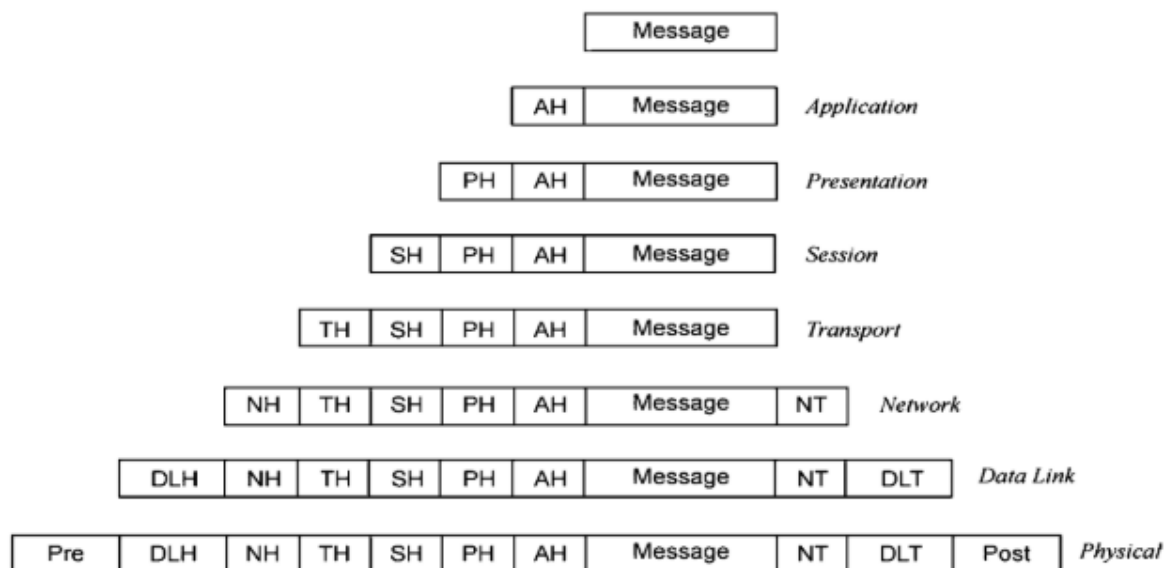
A comunicação inicia-se no transmissor, onde o usuário passa a mensagem e as informações de controle para a camada mais alta da pilha de protocolos através do sistema. O sistema por sua vez, passa a mensagem pelas 7 camadas, adicionando cabeçalhos e solicita funções de acordo com as regras do protocolo. Os cabeçalhos adicionam as informações de controle de protocolo pertinentes à cada camada. Essa combinação de mensagem e cabeçalhos é denominada como unidade de dados de protocolo (PDU). O processo inverso é realizado quando a PDU chega no destino. A PDU passa por cada camada que retira o cabeçalho correspondente ao nível. A Figura 13 - Estrutura da PDU mostra os efeitos da adição de cabeçalhos, para a PDU em cada camada.

Figura 12 - Arquitetura do Modelo OSI



Fonte: REYNDERS; MACHAY; WRIGHT (2005).

Figura 13 - Estrutura da PDU



Fonte: REYNDERS; MACHAY; WRIGHT (2005).

2.4.2 Protocolos de Comunicação

Atualmente a Automação Industrial está muito presente nas indústrias de grande porte, mas não é fácil indicar o momento exato do seu surgimento. Entretanto, é possível marcar o início da Automação Industrial com o aumento de produção produto da invenção inglesa da máquina à vapor. As décadas seguintes foram marcadas pelo surgimento de novas fontes de energia, a substituição do ferro pelo aço e a criação de dispositivos mecânicos chamados relés. Essa sequência de acontecimentos recebeu o nome de II Revolução Industrial.

O conceito de indústria já estava bem difundido no início do século XX, mas a automação estava presente de forma muito rudimentar. As indústrias buscavam novamente o aumento da produtividade e novos conceitos de produção que são conhecidos até os dias atuais. Henri Ford em 1909, idealizou o sistema por Linha de Montagem e com ela a Ford produzia automóveis em larga escala, porém a alta complexidade mecânica dos sistemas a relés exigia muita energia.

O primeiro Controlador Lógico Programável foi desenvolvido em 1968 pela MODICON (*Modular Digital Controller*) com o objetivo de garantir flexibilidade e economia à produção e surge a necessidade de comunicação entre os equipamentos. (SILEVIRA; LIMA, 2003)

2.4.2.1 Modbus RTU

O protocolo Modbus foi criado na década de 70 e tornou-se o protocolo mais difundido para comunicação entre dispositivos de controle e automação industrial.

Os motivos principais para o uso do Modbus em ambiente industrial são:

- Foi desenvolvido especialmente para aplicações industriais;
- Domínio público e sem cobrança de direitos autorais;
- Fácil de utilizar e manter;

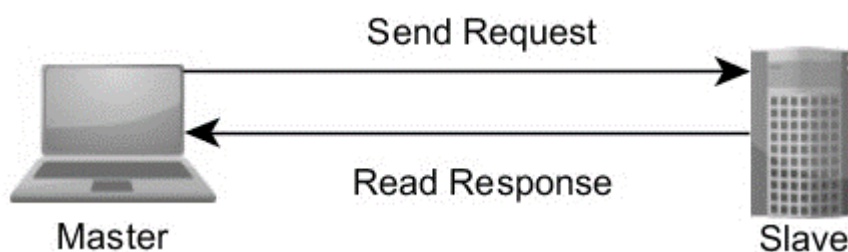
Comunicação de *bits* e *words* entre dispositivos de diferentes fabricantes sem restrições.

O Modbus é um protocolo de requisição-resposta que utiliza um relacionamento mestre-escravo. A comunicação sempre ocorre em pares no relacionamento mestre-escravo. Um dispositivo mestre deve iniciar a comunicação com uma requisição e aguardar por uma resposta do escravo. Usualmente, o mestre é um sistema SCADA (*Supervisory Control and Data Acquisition*) ou uma interface homem-máquina (IHM) e o escravo é um controlador

lógico programável (CLP), controlador programável para automação (CPA) ou um sensor. As mensagens que trafegam seguem uma estrutura lógica definida pelo protocolo que possui diversos comandos, para solicitar apenas o dado desejado. (MODBUS, 2020)

O mestre pode solicitar dados para os escravos por meio de mensagens *broadcast* e *unicast*. Mensagens *broadcast* possuem o endereço 0 e são destinadas a todos os escravos da rede. Mensagens *unicast* são endereçadas à um escravo específico com ID único da rede que pode variar de 1 a 247. (ALFACOMP, 2019)

Figura 14 - Relacionamento de rede tipo mestre-escravo.

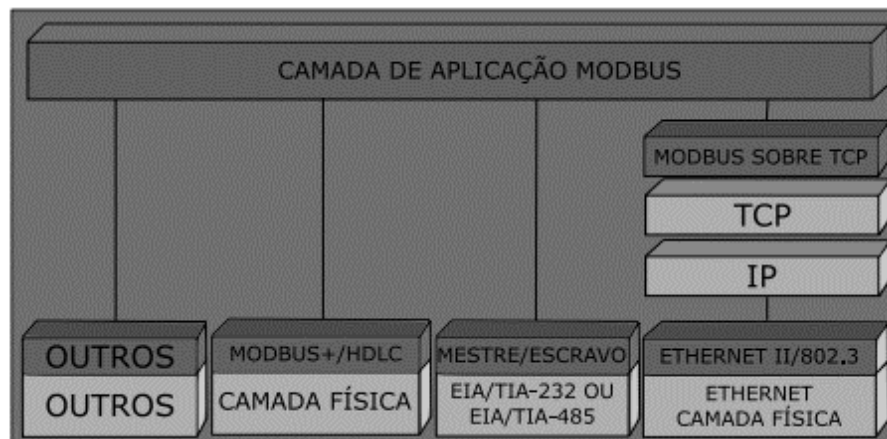


Fonte: Alfacom (2019).

O protocolo Modbus foi implementado inicialmente para ser usado na camada física baseada no padrão RS232, RS422 e RS485. Entretanto, atualmente o protocolo está presente em diferentes meios físicos, como mostra a Figura 15.

O protocolo determina dois modos de envio serial que definem como o dado é encapsulado e decodificado dentro dos pacotes transmitidos. O modo RTU (*Remote Terminal Unit*) e o modo ASCII (*American Standard Code for Information Interchange*). (NATIONAL INSTRUMENTS, 2019)

Figura 15 - Redes e Barramentos.



Fonte: National Instruments (2019).

No modo RTU os dados são transmitidos em conjuntos de 8 bits, contendo 2 caracteres no formato hexadecimal, o padrão prevê que todos os dispositivos da rede (mestres e escravos) devem suportar este formato. O frame do protocolo contém o endereço, o código da função e os dados de verificação (CRC - *Cyclic redundancy check*), como mostra a Figura 16:

Figura 16 - Frame Modbus.

Endereço do Escravo	Código da Função	Dados	CRC
1 Byte	1 Byte	0 a 255 Byte(s)	2 Bytes

Fonte: Elaboração dos autores (2020).

O código de função é composto de 1 byte que descreve a ação ou dados que o mestre está solicitando. A Figura 17 apresenta as principais funções descritas pelo protocolo na classe 2:

Figura 17 - Função Modbus

Código	Descrição
1	<i>Read Coils</i>
2	<i>Read Discrete Inputs</i>
4	<i>Read Input Registers</i>
5	<i>Write Single Coil</i>
6	<i>Write Single Register</i>
7	<i>Read Exception Status (somente serial)</i>

Fonte: Elaboração dos autores (2020).

2.5 SISTEMAS DE SUPERVISÃO

O software de supervisão ou software SCADA, permite monitorar e operar processos. Na indústria de manufatura inúmeros processos podem ser monitorados e operados através de um sistema supervisório, tais como: processo contínuo, batelada, elétrico, automação residencial e predial, entre outros. Além disto, esses sistemas estão presentes em sistemas de serviços públicos, como de tratamento de água e esgoto, e em sistemas de Internet das Coisas (IoT).

Segundo STRAUSS (2003), nossa tradução:

O termo SCADA é a abreviação para Sistema de Controle e Aquisição de Dados. Isto engloba a coleta de informações (aquisição de dados), transferência dos dados através de meios físicos (o campo da telemetria / comunicação de dados) e o processamento e exibição dos dados na estação principal. A estação principal também é usada para centralização controle sobre a rede de comunicação e / ou para iniciar comandos externos (controle de supervisão).

A estação mestre da rede de comunicação, que inclui o hardware e software, geralmente é denominado SCADA. Os primeiros sistemas de supervisão foram desenvolvidos de forma proprietária pelas fabricantes de hardware, porém o acelerado desenvolvimento e popularidade atingida pelos computadores pessoais (PC) exigiu dos fornecedores de software SCADA soluções compatíveis com PC. Atualmente, os SCADA's possuem ampla

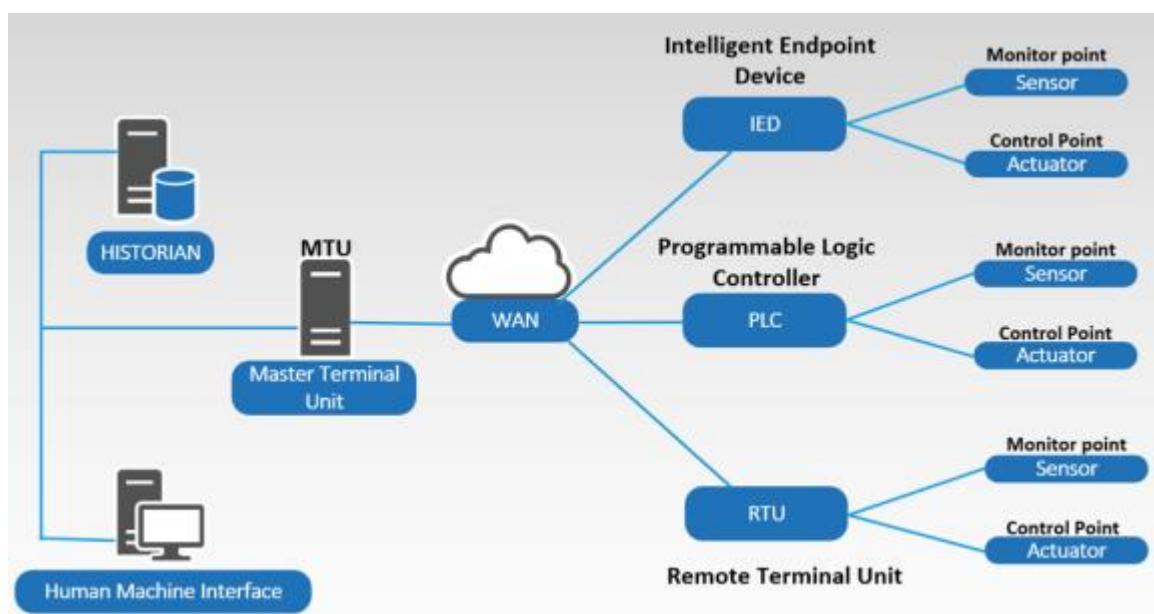
interoperabilidade e podem se comunicar com praticamente todos os dispositivos capazes de comunicações avançadas. (STRAUSS, 2003)

O sucesso da operação do SCADA depende da integração de vários componentes do sistema, são eles: hardware, software e comunicação.

Sistemas supervisório são vastamente empregados em diversas áreas da Engenharia. As informações coletadas de um processo produtivo ou instalação física são coletadas e concentradas em bancos de dados dos sistemas SCADA. Estes sistemas permitem que seja monitorada e rastreada a informação de um processo produtivo. Equipamentos de aquisição de dados são usados para coletar as informações que são manipuladas, analisadas, armazenadas e apresentadas ao usuário. Estas características tornam o SCADA parte fundamental para operação de um processo produtivo. (CONSTAIN, 2011)

A Figura 18 - Arquitetura de rede para SCADA, apresenta um exemplo de arquitetura onde o SCADA é chamado de MTU (Master Terminal Unit).

Figura 18 - Arquitetura de rede para SCADA



Fonte: LogRhythm (2019).

2.5.1 Virtuino Modbus

O Vituino é um software SCADA gratuito desenvolvido para sistemas operacionais Android com possibilidade de integração com hardware e software utilizados pela comunidade *maker*, como Arduino e ESP32. Sua interface pode ser totalmente customizada de acordo com os elementos necessários para o projeto. A versão Vituino Modbus possui compatibilidade com o protocolo de comunicação Modbus exercendo a função de master na rede de comunicação. Apesar de ser considerado um SCADA, o objetivo final deste software é monitorar as unidades remotas e comandá-las, sem o armazenamento dos dados históricos. O funcionamento desse SCADA será detalhado nos próximos capítulos.

3 DESENVOLVIMENTO

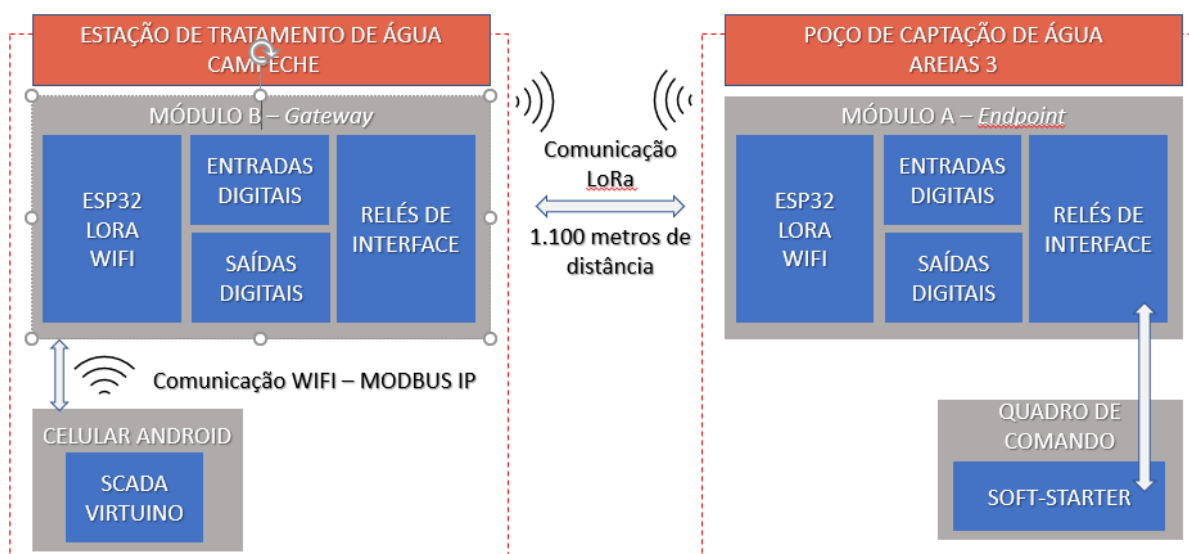
A fim de se verificar a aplicabilidade da solução proposta para o problema de comunicação entre dois pontos no Bairro Campeche em Florianópolis, foram realizados alguns testes de campo.

Para a realização desses testes que consistiam no enlace do rádio LoRa e no controle e monitoramento do quadro de comando presente no poço de captação, foi necessária a aquisição dos módulos ESP 32 com rádio LoRa integrado. Para a programação desses módulos, foram desenvolvidos códigos em linguagem C através do software de programação Arduino. Além disto, uma tela gráfica foi desenvolvida no aplicativo SCADA Virtuino a fim de possibilitar o acesso remoto ao sistema. Todas as etapas necessárias para a validação do projeto serão descritas nos tópicos seguintes.

3.1 ESCOPO DO PROJETO

O projeto tem como ponto de partida a criação de um enlace de modulação LoRa para comunicação entre as unidades operacionais da empresa, sendo elas a Estação de Tratamento de Água Campeche e o Poço de Captação de Água Areias 3. A Figura 2, apresentada novamente, ilustra as características globais do projeto.

Figura 2 - Escopo do projeto



Fonte: Elaboração dos autores (2020).

A comunicação entre os dispositivos do projeto foi baseada no protocolo Modbus IP e viabiliza o monitoramento de variáveis discretas e controle de ambas as unidades através do SCADA Virtuino. A tela de interface gráfica criada no SCADA Virtuino pode ser observada na Figura 19. Encontram-se na região superior da tela as variáveis correspondentes ao módulo A. As chaves 1 e 2 indicam o estado das entradas digitais do módulo e o led 1 fornece o feedback da saída digital. Já o botão *on/off* foi utilizado para alternar o nível lógico da saída digital do módulo B, que por sua vez, aciona um relé de interface para ligar a motobomba. As variáveis correspondentes ao módulo B encontram-se na metade inferior da tela. Analogamente ao módulo A, o módulo B conta com duas entradas digitais e uma saída digital. Indicando respectivamente, o *status* da bomba, a falta de uma ou mais das três fases do quadro de comando e o *status* do botão que liga/desliga a bomba.

Figura 19 - Interface Gráfica



Fonte: Elaboração dos autores (2020).

3.2 DESCRIÇÃO DO HARDWARE

Os protótipos desenvolvidos realizam um conjunto de operações, dentre as quais, executam a coleta de dados obtida através das entradas digitais e enviam comandos para o acionamento de um LED e de um relé, ambos conectados nas saídas digitais. Todo o gerenciamento dessas atividades é realizado através de um sistema microcontrolado. Sendo assim, para esta aplicação foi definido o módulo ESP 32 LoRa, por ser uma ferramenta de fácil utilização para prototipagens rápidas, além de apresentar todos os recursos necessários para implementação do projeto.

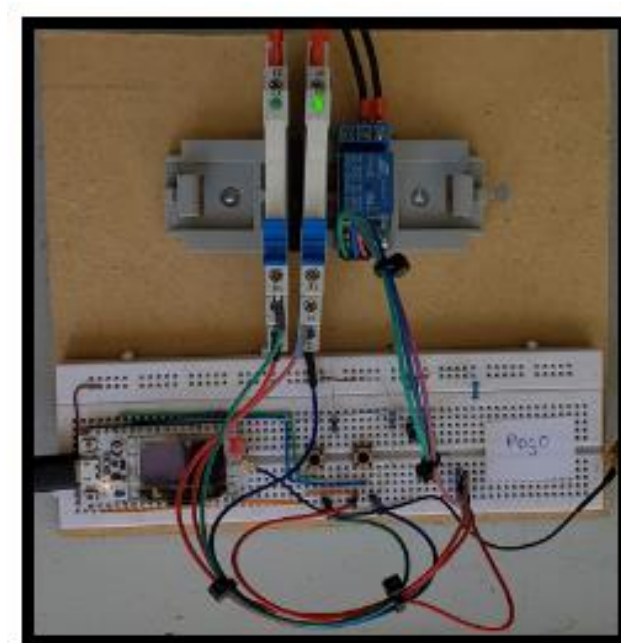
Para complementar essa solução, utilizou-se um par de módulos com ESP 32 e LoRa integrado, para comunicação por enlace de rádio, onde um deles opera como *endpoint* e o outro como *gateway* de borda com interface MODBUS IP. Uma interface gráfica foi criada para monitoramento e comando no sistema de supervisão e aquisição de dados (SCADA) para dispositivos mobile com sistema operacional Android chamado Virtuino.

3.2.1 Módulo A - *Endpoint*

A concepção da integração de hardware do módulo A considerou a necessidade de utilizar uma saída digital do módulo microcontrolado para o acionamento de um relé responsável pela interface com o circuito de acionamento do quadro de comando em 24Vcc. O quadro de comando existente conta com uma soft-starter modelo Altistar22 da marca Schneider utilizada para realizar a partida da motobomba de captação.

Também foram utilizadas duas entradas digitais do microcontrolador, uma para monitoramento do estado da bomba (ligada ou desligada) e outra para simulação do monitoramento do relé de falta de fase do quadro elétrico (com energia ou sem energia), para monitoramento dessas variáveis foram integrados ao módulo dois relés que fazem a interface de 220 Vac para 3,3 Vcc (tensão utilizada pelo ESP 32). Em paralelo com os dois relés de monitoramento das informações do quadro elétrico, foram instaladas duas chaves *push-button* para exemplificar o acionamento dessas variáveis, a chave 1 está em paralelo com a informação do estado da bomba e a chave 2 simula a informação de falta de fase. Na Figura 20, pode ser visto o módulo descrito.

Figura 20 - Módulo A

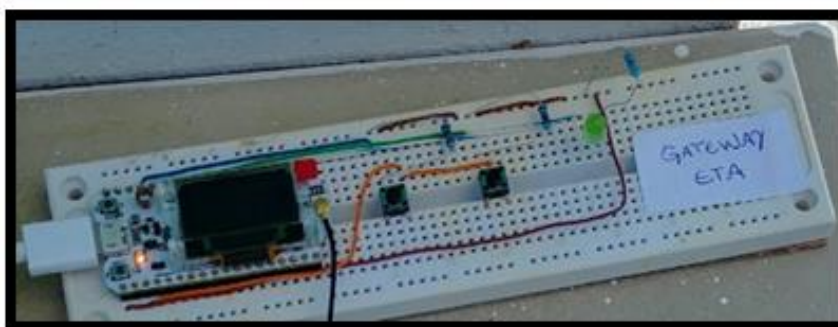


Fonte: Elaboração dos autores (2020).

3.2.2 Módulo B - Gateway

O módulo B, por sua vez, é formado por um sistema microcontrolado, duas chaves modelo *push-button* e um LED. Essas chaves e o LED foram incluídos, a fim de se ampliar os testes realizados. Essas variáveis, foram utilizadas para testar o protocolo Modbus TCP/IP entre o módulo e o Virtuino. Na Figura 21 é possível visualizar a composição desse módulo.

Figura 21 - Módulo B Gateway



Fonte: Elaboração dos autores (2020).

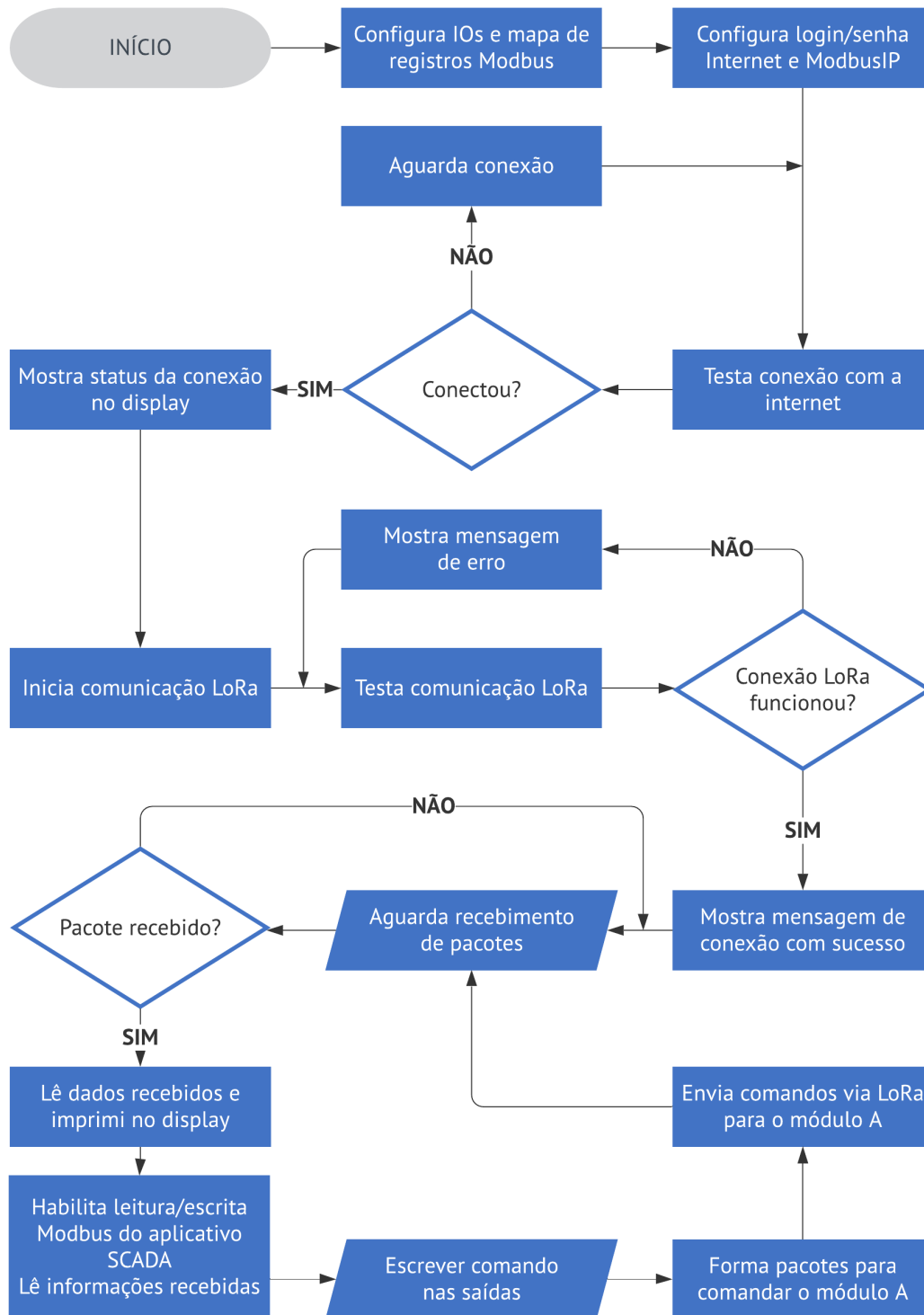
3.3 DESCRIÇÃO DO CÓDIGO

Para validar o sistema proposto neste projeto, foram desenvolvidos dois códigos em linguagem C, um para o módulo B instalado na Estação de Tratamento de Água do Campeche (*gateway*) e outro para o módulo A instalado no Poço de Captação do Campeche (*endpoint*).

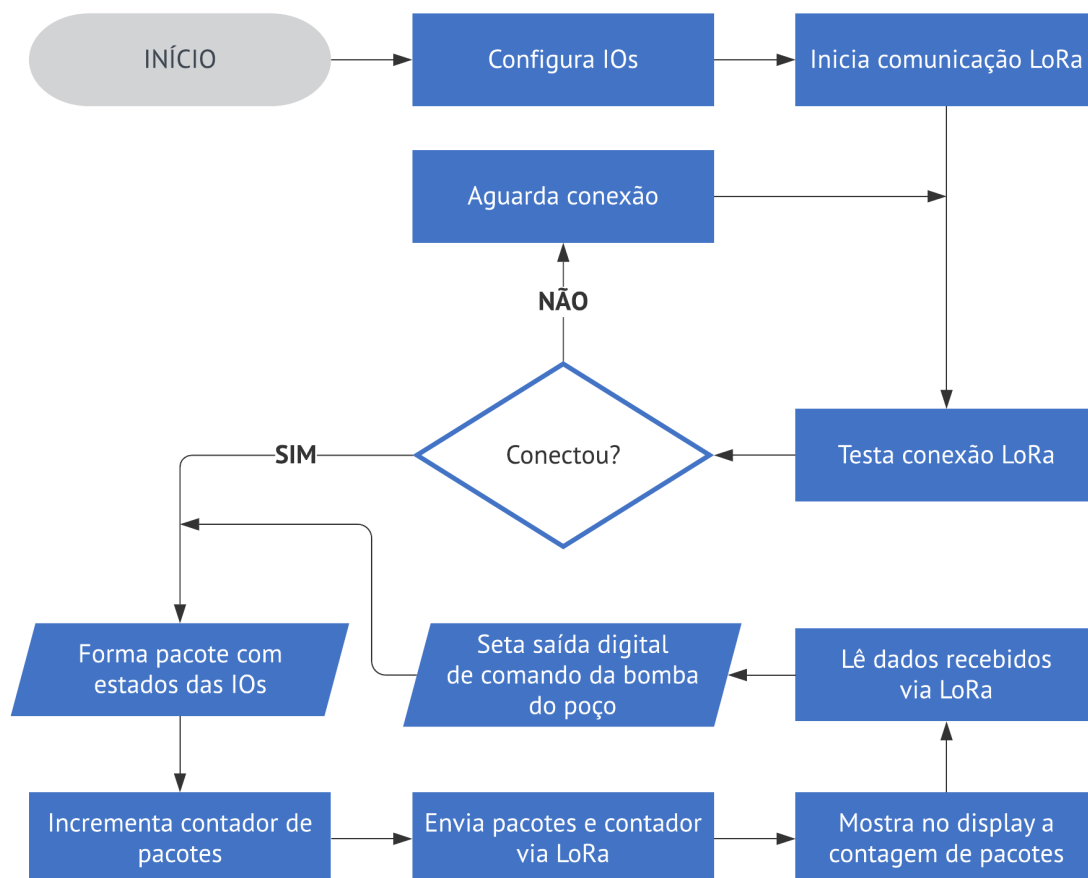
A programação desenvolvida para o módulo *gateway*, consiste na criação de um mapa de registro Modbus, que corresponde aos registros para acesso as variáveis que serão monitoradas (entradas digitais) e as variáveis de comando (saídas digitais) de ambos os módulos. Foi realizada a conexão TCP/IP com o roteador *Wifi*, em seguida realizou-se a comunicação LoRa com o módulo *endpoint* (recebendo status das entradas e saídas desse módulo e enviando os comandos a ele, recebidos pelo software SCADA). O display *oled* foi habilitado para mostrar os status de conexões e recebimento de pacotes via LoRa. O fluxograma que exemplifica a programação desse módulo pode ser visto na Figura 22 - Fluxograma Módulo B *Gateway*.

Já a programação desenvolvida para o módulo *endpoint* corresponde ao estabelecimento da comunicação via LoRa com o módulo anterior, o envio dos status de entradas e saídas digitais para esse módulo e o recebimento de comandos para a sua saída digital responsável por acionar a bomba de captação do poço. O fluxograma que exemplifica a programação desse módulo corresponde a Figura 23 - Fluxograma Módulo A *Endpoint*.

Figura 22 - Fluxograma Módulo B Gateway



Fonte: Elaboração dos autores (2020).

Figura 23 - Fluxograma Módulo A *Endpoint*

Fonte: Elaboração dos autores (2020).

3.4 TESTES DE CAMPO

Os protótipos desenvolvidos foram testados em campo. O módulo A foi instalado no quadro de comando do Poço de Captação denominado Areais 3 localizado na Servidão Olindina Maria Lopes, bairro Campeche, cidade Florianópolis – SC. O módulo B foi instalado na Estação de Tratamento de Água, também presente no bairro Campeche localizada na Rua Tereza Lopes.

Inicialmente foi estabelecida a conexão com o roteador Wifi. É possível verificar na Figura 24 - Display Módulo B *Gateway* que a segunda linha do display *oled* informa o IP adquirido, possibilitando assim que esse IP seja também utilizado no software SCADA Virtuino, para que ambos estejam na mesma rede LAN e seja possível realizar o monitoramento e controle remoto. Após a conexão da rede, foi estabelecida a comunicação

LoRa entre os dois módulos. Na primeira linha do display vê-se o nível de sinal da comunicação e na terceira linha os pacotes recebidos via LoRa.

O pacote recebido foi dividido da seguinte forma: o primeiro bit corresponde ao estado da bomba de captação, o bit 2 corresponde ao estado da chave 2 (que representa a informação de falta de energia no quadro de comando) e o bit 3 o estado da bomba de captação do poço, todos referentes ao módulo A. Em seguida é apresentado no display um contador de pacotes que para conferir o recebimento contínuo dos pacotes.

Figura 24 - Display Módulo B *Gateway*



Fonte: Elaboração dos autores (2020).

Na Figura 25 - Monitoramento do módulo A através da tela desenvolvida no Virtuino, é possível verificar o acionamento do LED chamado F. Fase (Falta de Fase). Este LED foi acionado através da chave 2 presente no conjunto que compõe o Módulo A e serve representativamente para verificar, caso haja alguma falta de energia no local ou o disjuntor geral do quadro elétrico esteja desarmado.

Figura 25 - Monitoramento do módulo A através da tela desenvolvida no Virtuino

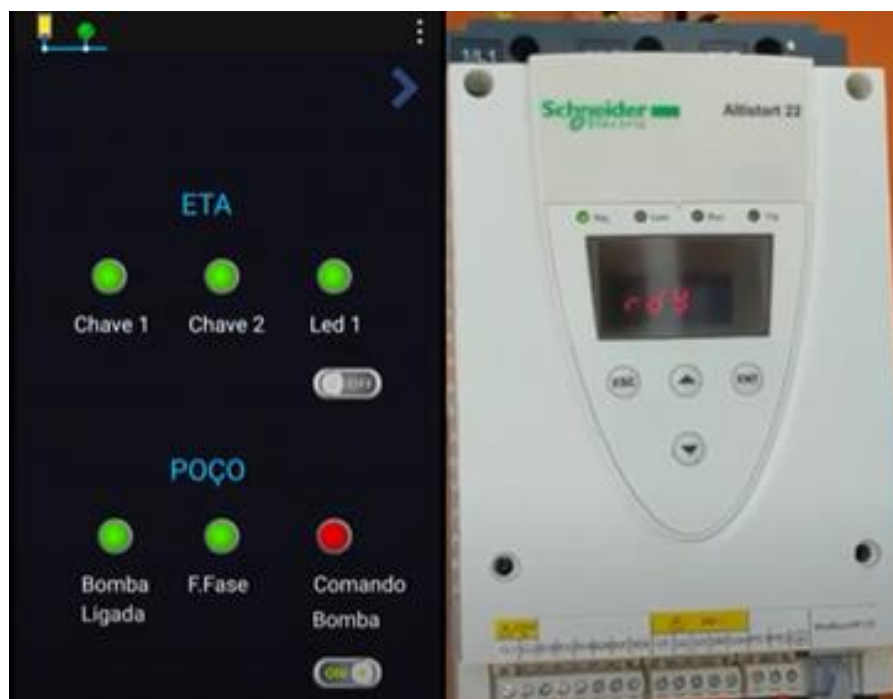


Fonte: Elaboração dos autores (2020).

Em seguida, foram realizados os testes de acionamento da bomba de captação do poço. A Figura 26 indica que o botão virtual criado na tela desenvolvida no Virtuino está acionado e o led que representa a saída digital que corresponde ao comando da bomba está aceso. Neste momento a bomba ainda encontrasse parada e por isso, é apresentada a informação de *Ready* no display da soft-starter, indicando que ela está pronta e aguardando o comando e o LED indicativo chamado Bomba Ligada no Virtuino também encontrasse apagado. Ao receber o comando via LoRa, a soft-starter parte a bomba de captação e apresenta no display da *soft-starter* a informação de *Run*, conforme é mostrado na Figura 27, onde verifica-se na tela do Virtuino que o LED que representa a Bomba Ligada acendeu.

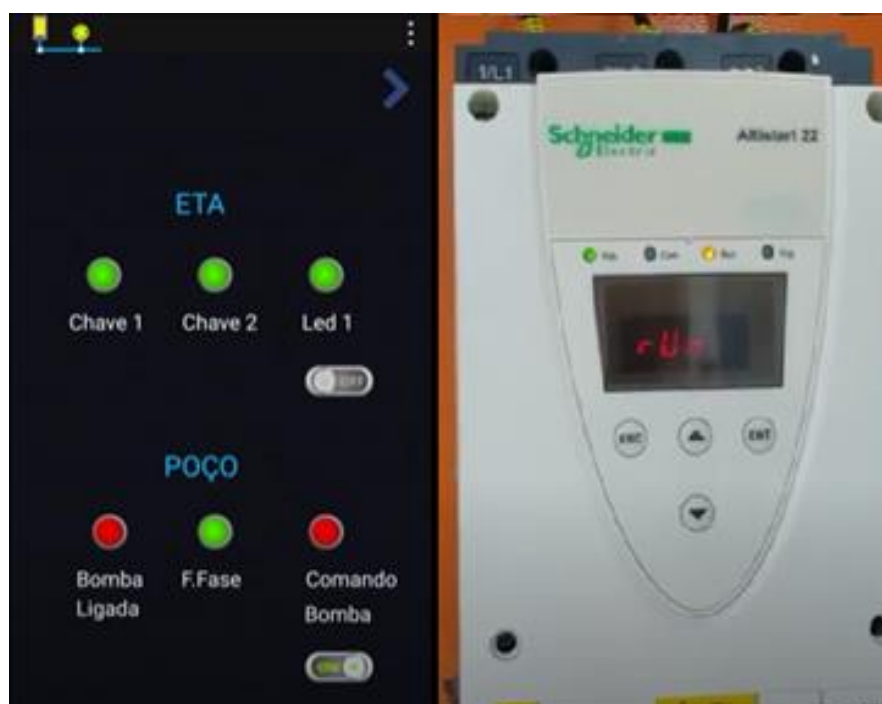
Foram realizados também, os testes de variação do estado das chaves 1 e 2 do módulo A, monitorando essas variáveis a partir da tela desenvolvida do Virtuino. Pode se visualizar nas Figuras 28 e 29, o acionamento dessas chaves, através dos LEDs na cor vermelha. Já na Figura 30, é exibido o botão virtual criado no Virtuino acionado e o estado do LED 1 comandado por este botão.

Figura 26 – Comando para acionamento da bomba através da tela desenvolvida no Virtuino



Fonte: Elaboração dos autores (2020).

Figura 27 - Bomba acionada através da tela desenvolvida no Virtuino



Fonte: Elaboração dos autores (2020).

Figura 28 - Monitoramento da chave 1 do módulo B através da tela desenvolvida no Virtuino



Fonte: Elaboração dos autores (2020).

Figura 29 - Monitoramento da chave 2 do módulo B através da tela do Virtuino



Fonte: Elaboração dos autores (2020).

Figura 30 - Acionamento do led do módulo B através da tela do Virtuino



Fonte: Elaboração dos autores (2020).

4 CONCLUSÃO

Este trabalho foi baseado na integração de tecnologias distintas presentes em redes de automação e novas tecnologias desenvolvidas no campo da Internet das Coisas, como o protocolo de comunicação Modbus TCP/IP e a comunicação rádio LoRa.

O presente projeto buscou validar a comunicação rádio LoRa entre duas unidades operacionais da CASAN em Florianópolis integrando o protótipo desenvolvido a estrutura operacional existente.

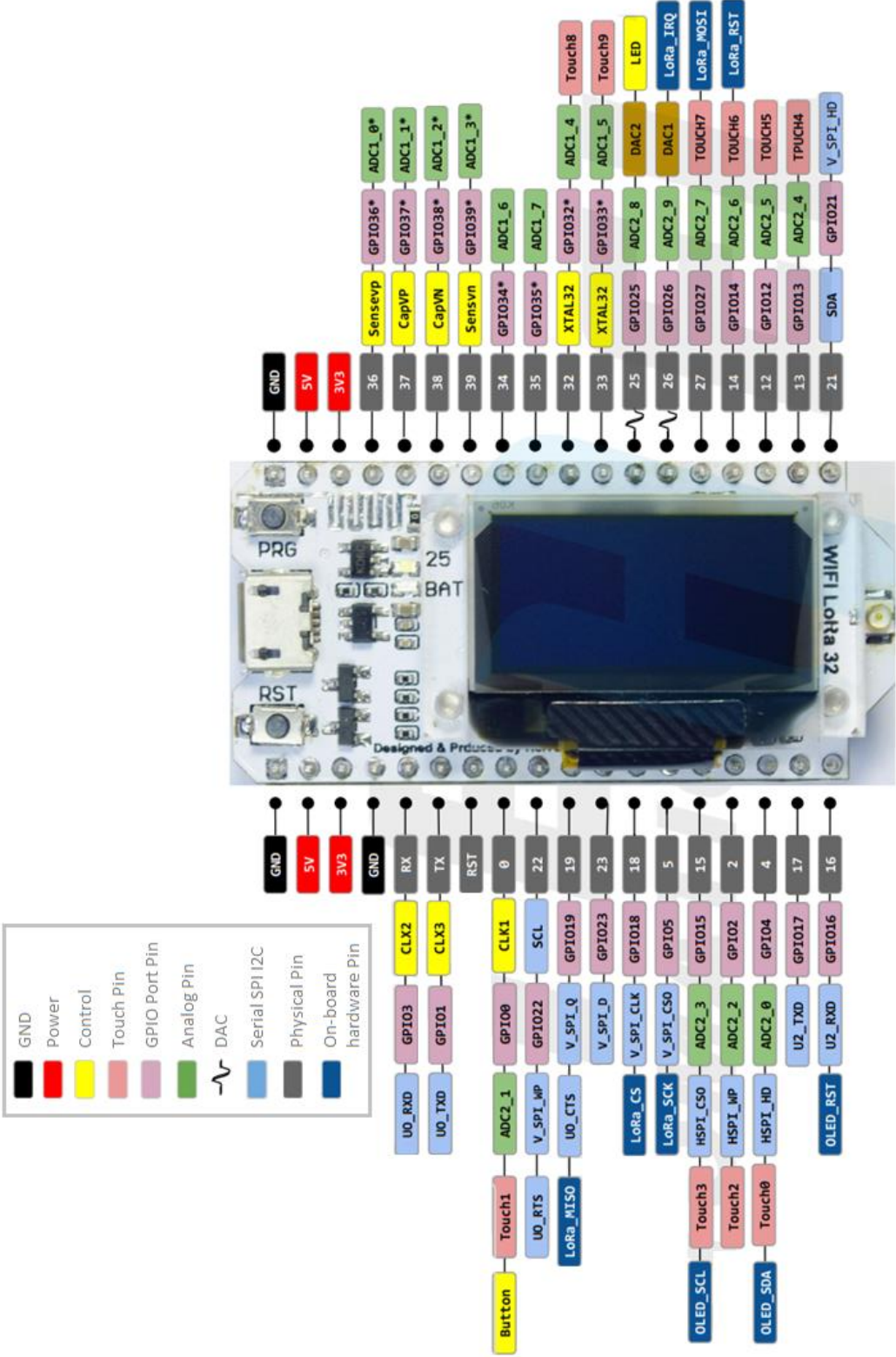
O sistema completo elaborado consiste em dois módulos instalados em pontos distintos e programados de modo a realizar a troca de informações entre si, com a possibilidade de monitoramento e controle de um poço de captação de água a partir de qualquer local, mediante o uso de uma conexão de rede TCP/IP através de uma interface desenvolvida em um software supervisor.

No decorrer do desenvolvimento deste projeto, foram atingidos todos os objetivos definidos, desde o estudo das tecnologias empregadas, até a elaboração e implementação dos protótipos em campo e a partir disso o monitoramento e controle do poço de captação de água Areias 3 utilizando a comunicação rádio LoRa com a Estação de Tratamento de Água do Campeche.

Mediante a implantação do sistema proposto foi possível concluir que este atende aos requisitos impostos no início deste projeto de forma satisfatória. Tendo contribuído para demonstrar a viabilidade da tecnologia LoRa neste cenário.

ANEXOS

Anexo A- Pinagem Módulo ESP32 WIFI LORA da HELTEC



Fonte: WIFI (2019)

APÊNDICES

Código Módulo *Gateway*

```
//Para Lora e Display
#include <SPI.h> //responsável pela comunicação serial
#include <LoRa.h> //responsável pela comunicação com o WIFI Lora
#include <Wire.h> //responsável pela comunicação i2c
#include "SSD1306.h" //responsável pela comunicação com o display

// Definição dos pinos
#define SCK 5 // GPIO5 -- SX127x's SCK
#define MISO 19 // GPIO19 -- SX127x's MISO
#define MOSI 27 // GPIO27 -- SX127x's MOSI
#define SS 18 // GPIO18 -- SX127x's CS
#define RST 14 // GPIO14 -- SX127x's RESET
#define DI00 26 // GPIO26 -- SX127x's IRQ(Interrupt Request)
#define BAND 915E6 //Frequencia do radio - podemos utilizar ainda : 433E6,
868E6, 915E6

//parametros: address,SDA,SCL
SSD1306 display(0x3c, 4, 15); //construtor do objeto que controlaremos o display

String rssi = "RSSI --"; //indicador de sinal recebido
String packSize = "--";
String packet ;

//TESTE
String pacote[4];
String pacote0;
int posicaoDaBarra0=0;
int posicaoDaBarra[4];
int posicaoDaBarraAnt=0;

unsigned long tempoAux=millis(); //para substituir o delay no loop do Lora
unsigned long tempoAux2=millis();

long lastSendTime = 0; // last send time
int interval = 500; // interval between sends

//Para Modbus
#include <Modbus.h>
#include <ModbusIP_ESP32.h>
```

```

IPAddress ip;
String sIP="";

//ModbusIP object
ModbusIP mb;

//Entradas e Saídas Utilizadas
const int chave1Pin_ETA = 36; //GPIO36-esp32
const int chave2Pin_ETA = 37; //GPIO37-esp32
const int led1Pin_ETA = 12; //GPIO38-esp32

//ETA - Modbus Registers Offsets (0-9999)
const int TEST_HREG = 100;
const int CHAVE1_ISTS_ETA= 101; //mb=modbus
const int CHAVE2_ISTS_ETA = 102;
const int LED1_COIL_ETA = 103;

//POÇO - Modbus Registers Offsets
const int CHAVE1_ISTS_POCO= 104;
const int CHAVE2_ISTS_POCO = 105;
const int LED1_COIL_POCO = 106;
const int LED1_ISTS_POCO = 107;

//Variavel auxiliar comando Poco
int comando_Led1_Poco;

void setup() {

//Para Modbus
  Serial.begin(115200);

  //Conf. Roteador
  mb.config("TP-LINK_D9E4D6", "38065965");

  //Conf. Celular
  // mb.config("cel_marcia", "senhamarcia");

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());

```



```

ip = WiFi.localIP();
for (int i=0; i<4; i++) {
  sIP += i ? "." + String(ip[i]) : String(ip[i]);
}

//mb.addHreg(TEST_HREG, 0xABCD);
mb.addHreg(TEST_HREG);

//ETA - Config pinos e registros
pinMode(chave1Pin_ETA, INPUT);
mb.addIsts(CHAVE1_ISTS_ETA);

pinMode(chave2Pin_ETA, INPUT);
mb.addIsts(CHAVE2_ISTS_ETA);

pinMode(led1Pin_ETA, OUTPUT);
mb.addCoil(LED1_COIL_ETA);

//POCO - Config pinos e registros
mb.addIsts(CHAVE1_ISTS_POCO);
mb.addIsts(CHAVE2_ISTS_POCO);
mb.addCoil(LED1_COIL_POCO);
mb.addIsts(LED1_ISTS_POCO);

//Para Lora e Display
//configura os pinos como saida
pinMode(16,OUTPUT); //RST do oled
digitalWrite(16, LOW); // reseta o OLED
delay(50);
digitalWrite(16, HIGH); // enquanto o OLED estiver ligado, GPIO16 deve estar
HIGH
display.init();
display.flipScreenVertically();
display.setFont(ArialMT_Plain_10);
delay(1500);
display.clear();

SPI.begin(SCK,MISO,MOSI,SS); //inicia a comunicação serial com o Lora
LoRa.setPins(SS,RST,DI00); /*configura os pinos que serão
utilizados pela biblioteca (deve ser chamado antes do LoRa.begin)*/

//inicializa o Lora com a frequencia específica.
if (!LoRa.begin(BAND)) {
  display.drawString(0, 0, "Starting LoRa failed!");
  display.display();
  while (1);
}

//indica no display que inicilizou corretamente.

```

```

display.drawString(0, 0, "LoRa Initial success!");
display.drawString(0, 10, "Wait for incomm data...");
display.display();
delay(1000);

LoRa.receive(); //habilita o Lora para receber dados
}

void loop() {

  mb.task();

  onReceive();

  leituraEscritaModbus();

  if (millis() - lastSendTime > interval) {
    sendMessage();
    lastSendTime = millis();    // timestamp the message
    interval = random(500) + 500; // 0,5 a 1 seconds
  }
}

void onReceive(){
  //Para Lora e Display
  if( tempoAux<=millis() ){ //para substituir o delay
    tempoAux=millis()+10;
    //parsePacket: checa se um pacote foi recebido
    //retorno: tamanho do pacote em bytes. Se retornar 0 (ZERO) nenhum pacote foi
recebido
    int packetSize = LoRa.parsePacket();
    /*caso tenha recebido pacote chama a função para configurar
os dados que serão mostrados em tela*/
    if (packetSize) {
      cbk(packetSize);
    }
  }
}

void sendMessage(){
  LoRa.beginPacket();          // start packet
  LoRa.write(comando_Led1_Poco);
  LoRa.endPacket();            // finish packet and send it
}

void leituraEscritaModbus(){
  if( tempoAux2<=millis() ){ //para substituir o delay
    tempoAux2=millis()+30;
    //mb.Hreg(TEST_HREG, 12345);
  }
}

```

```

//Leitura escrita ETA
mb.Ists(CHAVE1_ISTS_ETA, digitalRead(chave1Pin_ETA)); //le o pino e
escreve no modbus
mb.Ists(CHAVE2_ISTS_ETA, digitalRead(chave2Pin_ETA));
digitalWrite(led1Pin_ETA, mb.Coil(LED1_COIL_ETA));

//Leitura escrita POCO
mb.Ists(CHAVE1_ISTS_POCO, pacote0.toInt() );
mb.Ists(CHAVE2_ISTS_POCO, pacote[1].toInt() );
mb.Ists(LED1_ISTS_POCO, pacote[2].toInt() );
comando_Led1_Poco = mb.Coil(LED1_COIL_POCO);
}
}

//função responsável por recuperar o conteúdo do pacote recebido
//parametro: tamanho do pacote (bytes)
void cbk(int packetSize) {
    packet = "";
    packSize = String(packetSize, DEC); //transforma o tamanho do pacote em String
    para imprimirmos
    for (int i = 0; i < packetSize; i++) {
        packet += (char) LoRa.read(); //recupera o dado recebido e concatena na variável
        "packet"
    }
    rssi = "RSSI= " + String(LoRa.packetRssi(), DEC) + "dB"; //configura a String de
    Intensidade de Sinal (RSSI)

    //TESTE
    posicaoDaBarra0=packet.indexOf("|");
    pacote0 = packet.substring(0 , posicaoDaBarra0);
    posicaoDaBarraAnt=posicaoDaBarra0;

    for(int i=1; i<4; i++){
        posicaoDaBarra[i]=packet.indexOf("|", posicaoDaBarraAnt+1);
        pacote[i] = packet.substring(posicaoDaBarraAnt+1, posicaoDaBarra[i]);
        posicaoDaBarraAnt=posicaoDaBarra[i];
    }

    Serial.print(pacote0);
    Serial.print(" - ");
    Serial.print(pacote[1]);
    Serial.print(" - ");
    Serial.print(pacote[2]);
    Serial.print(" - ");
    Serial.println(pacote[3]);

    loraData();
}

```

```
void loraData(){
  display.clear();
  display.setTextAlignment(TEXT_ALIGN_LEFT);
  display.setFont(ArialMT_Plain_16);
  // display.drawString(0 , 18 , "Rx " + packSize + " bytes");
  display.drawString(0, 18, sIP);

  display.drawStringMaxWidth(0 , 39 , 128, packet);
  display.drawString(0, 0, rssi);
  display.display();
}
```

Código Módulo *Endpoint*

```

#include <SPI.h> //responsável pela comunicação serial
#include <LoRa.h> //responsável pela comunicação com o WIFI Lora
#include <Wire.h> //responsável pela comunicação i2c
#include "SSD1306.h" //responsável pela comunicação com o display
//#include "images.h" //contém o logo para usarmos ao iniciar o display

// Definição dos pinos
#define SCK    5  // GPIO5  -- SX127x's SCK
#define MISO   19 // GPIO19 -- SX127x's MISO
#define MOSI   27 // GPIO27 -- SX127x's MOSI
#define SS     18 // GPIO18 -- SX127x's CS
#define RST    14 // GPIO14 -- SX127x's RESET
#define DI00   26 // GPIO26 -- SX127x's IRQ(Interrupt Request)

#define BAND   915E6 //Frequencia do radio

//variável responsável por armazenar o valor do contador (enviaremos esse valor para
o outro Lora)
unsigned int counter = 0;

//parametros: address,SDA,SCL
SSD1306 display(0x3c, 4, 15); //construtor do objeto que controlaremos o display

String rssi = "RSSI --";
String packSize = "--";
String packet ;

//Entradas e Saídas Utilizadas
const int chave1_pin = 36; //GPIO36-esp32
const int chave2_pin = 37; //GPIO37-esp32
const int led1_pin = 12; //GPIO38-esp32

```

```

long lastSendTime = 0;    // last send time
int interval = 500;      // interval between sends

void setup()
{
    //Configura pinos para chave e led
    pinMode(chave1_pin, INPUT);
    pinMode(chave2_pin, INPUT);
    pinMode(led1_pin, OUTPUT);

    //configura os pinos como saida
    pinMode(16,OUTPUT); //RST do oled
    pinMode(25,OUTPUT);

    digitalWrite(16, LOW); // reseta o OLED
    delay(50);
    digitalWrite(16, HIGH); // enquanto o OLED estiver ligado, GPIO16 deve estar
HIGH

    display.init(); //inicializa o display
    display.flipScreenVertically();
    display.setFont(ArialMT_Plain_10); //configura a fonte para um tamanho maior

    delay(1500);
    display.clear(); //apaga todo o conteúdo da tela do display

    SPI.begin(SCK,MISO,MOSI,SS); //inicia a comunicação serial com o Lora
    LoRa.setPins(SS,RST,DI00); //configura os pinos que serão utilizados pela biblioteca
(deve ser chamado antes do LoRa.begin)

    //inicializa o Lora com a frequencia específica.

```

```

if (!LoRa.begin(BAND))
{
    display.drawString(0, 0, "Starting LoRa failed!");
    display.display(); //mostra o conteúdo na tela
    while (1);
}
//indica no display que inicilizou corretamente.
display.drawString(0, 0, "LoRa Initial success!");
display.display(); //mostra o conteúdo na tela
delay(1000);
}

void loop()
{
    //apaga o conteúdo do display
    display.clear();
    display.setTextAlignment(TEXT_ALIGN_LEFT);
    display.setFont(ArialMT_Plain_16);

    display.drawString(0, 0, "Sending packet: ");
    display.drawString(40, 26, String(counter));
    display.display(); //mostra o conteúdo na tela

    if (millis() - lastSendTime > interval) {
        sendMessage();
        lastSendTime = millis();      // timestamp the message
        interval = random(500) + 500; // 1-2 seconds
    }

    onReceive(LoRa.parsePacket());
}

void sendMessage(){

```

```

//beginPacket : abre um pacote para adicionarmos os dados para envio
LoRa.beginPacket();
//print: adiciona os dados no pacote
LoRa.print( digitalRead(chave1_pin) );
LoRa.print("|");
LoRa.print( digitalRead(chave2_pin) );
//LoRa.print(counter);
LoRa.print("|");
LoRa.print( digitalRead(led1_pin) );
LoRa.print("|");
LoRa.print(counter);
LoRa.print("|");
LoRa.endPacket(); //retorno= 1:sucesso | 0: falha
counter++;
}

void onReceive(int packetSize) {
  if (packetSize == 0) return;      // if there's no packet, return
  // read packet header bytes:
  int comandoLed1 = LoRa.read();
  digitalWrite(led1_pin, comandoLed1);
}

```


REFERÊNCIAS

FOROUZAN, Behrouz A. **Comunicação de Dados e Redes de Computadores**. 4. ed. Porto Alegre: Amgh, 2010.

REYNDERS, Deon; MACHAY, Steve; WRIGHT, Edwin. **Practical Industrial Data Communications: best practice techniques**. London: Elsevier, 2005

STRAUSS, Cobus. **Electrical Network Automation and Communication Systems**. London: Elsevier, 2003.

SILEVIRA. Leonardo, LIMA, Weldson Q. **Um breve histórico conceitual da Automação Industrial e Redes para Automação Industrial**. UFRN, Natal, 2003. Disponível em: <https://www.dca.ufrn.br/~affonso/FTP/DCA447/trabalho1/trabalho1_13.pdf> Acesso em: 07 set. 2019.

PARK, John et al. **Practical Data Acquisition for Instrumentation and Control Systems**. Londres: Elsevier, 2003.

DURÃO, Fábio Prudente. **Uma Proposta de Desenvolvimento de Laboratório Remoto de Automação e Acionamento de Máquinas Elétricas para Uso Didático, Utilizando software Open Source**. UNESP. Disponível em: <<https://repositorio.unesp.br/handle/11449/87087?show=full>> Acesso em: 17 set. 2019.

EMILIO, Maurizio di Paolo. **Data Acquisition Systems: from fundamentals to applied design**. Nova York: Springer, 2013.

MORAES, José. **Conhecendo o ESP32**. 2017. Disponível em: <<https://portal.vidadesilicio.com.br/conhecendo-o-esp32/>>. Acesso em: 13 out. 2009.

CARRO, Luigi; WAGNER, Flávio Rech. **Sistemas Computacionais Embarcados**. Disponível em: <http://www.maxpezzin.com.br/aulas/6_EAC_Sistemas_Embarcados/8_SE_Visao_Sistemic_a.pdf>. Acesso em: 20 ago. 2019.

MURTA, Gustavo. **Conhecendo o ESP32**. 2018. Disponível em:
<<https://blog.eletrogate.com/conhecendo-o-esp32-introducao-1/>>. Acesso em: 13 out. 2009

ESP32 – Especificações e projetos. Disponível em: <https://athoselectronics.com/esp32/>.
Acesso em: 13 out. 2019.

WIFI LoRa 32 (V2). Disponível em: <https://heltec.org/project/wifi-lora-32/#weglot_switcher>. Acesso em: 13 out. 2019.

ALMEIDA, Orlan. Tecnologia LoRa: **O que é, distância e teste prático**. 2019. Disponível em: <<https://easyiot.com.br/blog/tecnologia-lora/>>. Acesso em: 13 out. 2019.

TEIXEIRA, Grazielle Bonaldi; ALMEIDA, João Vítor Peroni de. **Rede LoRa® e protocolo LoraWan® aplicados na agricultura de precisão no brasil**. 2017. 75 f. TCC (Graduação) - Curso de Bacharelado em Engenharia Eletrônica, Coordenação de Engenharia Eletrônica, Universidade Tecnológica Federal do Paraná, Ponta Grossa, 2017. Disponível em:
<http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/8436/1/PG_COELE_2017_2_01.pdf>
Acesso em: 14/10/2019

BERTOLETI, Pedro. **Projetos com ESP32 e LoRa**. São Paulo: NCB, 2019. E-book. Acesso restrito via Kindle.

SILVA JUNIOR, Vidal Pereira da. **Conheça a tecnologia LoRa® e o protocolo LoRaWAN™**. 2016. Disponível em: <<https://www.embarcados.com.br/conheca-tecnologia-lora-e-o-protocolo-lorawan/>>. Acesso em: 14 out. 2019.

BERTOLETI, Pedro. **Comunicação Simples Ponto a Ponto entre Dois Módulos WiFi LoRa 32(V2) (MIC217)**. Disponível em:
<<http://www.newtoncbraga.com.br/index.php/microcontrolador/143-tecnologia/16333-comunicacao-simples-ponto-a-ponto-entre-dois-modulos-wifi-lora-32-v2-mic217>>
Acesso em: 14 out. 2019.

WHAT is LoRaWAN®. 2005. LoRa® Alliance Technical Marketing Workgroup. Disponível em: <<https://loro-alliance.org/resource-hub/what-lorawanr>>. Acesso em: 14 out. 2019.

GRIESE, Marcelo George; KLEINSCHMIDT, João Henrique. **Aplicação da tecnologia LoRa em arquitetura para fiscalização eletrônica de veículos.** Disponível em: <<http://nuvem.ufabc.edu.br/images/workshop-nuvem-2017/TER-P-06.pdf>> Acesso em: 10 jul. 2019.

OLIVEIRA, Greici. **NodeMCU:** uma plataforma com características singulares para o seu projeto iot. Uma plataforma com características singulares para o seu projeto IoT. Disponível em: <<https://blogmasterwalkershop.com.br/embarcados/nodemcu/nodemcu-uma-plataforma-com-caracteristicas-singulares-para-o-seu-projeto-iot/>>. Acesso em: 13 ago. 2019.

MEDEIROS, Júlio César de Oliveira. **Princípios de Telecomunicações:** teoria e prática. 2. ed. São Paulo: Érica, 2007. 316 p

SILVA JÚNIOR, Joab Silas da. **Bluetooth e infravermelho.** Disponível em: <<https://brasilecola.uol.com.br/fisica/bluetooth-infravermelho.html>>. Acesso em: 20 abr. 2020.

ROMACHO, Plínio Augusto. **COMUNICAÇÃO POR RÁDIO FREQUÊNCIA PARA CASOS DE EMERGÊNCIAS.** 2012. 46 f. Monografia (Especialização) - Curso de Engenharia Elétrica, Universidade São Francisco, Itatiba, 2012. Disponível em: <<http://lyceumononline.usf.edu.br/salavirtual/documentos/2330.pdf>> Acesso em: 25 abr. 2020.

SBIZERA, Daniela de Oliveira. **Um sistema de comunicação para transmissão de dados a longa distância em aeronaves do projeto ARARA.** 2003. 143 f. Dissertação (Mestrado) - Curso de Ciências de Computação e Matemática Computacional, Instituto de Ciências Matemáticas e de Computação, São Carlos, 2003. Disponível em: <<https://teses.usp.br/teses/disponiveis/55/55134/tde-20122017-083316/publico/DanieladeOliveiraSbizera.pdf>> Acesso em: 25 maio 2020.

GOMES, Paulo. **Modulação.**

Disponível em:

<<http://pgomes.com.br/arquivos/7e55280848b003093bdeb621d54bda02.pdf>>

Acesso em: 28 maio 2020.

FAGUNDES, Eduardo. **Tipos de modulação (I).**

Disponível em: <<https://efagundes.com/networking/sistema-telefonico/tipos-de-modulacao-i/>> Acesso em: 25 abr. 2020.

LASKOSKI, Gustavo Theodoro; MARCONDES, Maicon; SZEREMETA, Oscar. **MODULAÇÃO DIGITAL: tópicos de comunicações. TÓPICOS DE COMUNICAÇÕES.** 2006.

Disponível em: <www.docsity.com> Acesso em: 20 jun. 2020.

Spread Spectrum: O que é.

Disponível em: <https://www.teleco.com.br/tutoriais/tutorialss/pagina_1.asp>

Acesso em: 19 maio 2020.

FRANÇA, Guilherme Augusto Toledo. **Estudo e especificação de um sistema de telemetria usando a tecnologia spread spectrum.** 1997. 82 f. Dissertação (Mestrado) - Curso de Faculdade de Engenharia Elétrica e Computação, Departamento de Eletrônica e Microeletrônica, Universidade Estadual de Campinas, Campinas, 1997.

DELLALIBERA, Leonardo Arten. **Técnicas de Múltiplo Acesso para Redes LORAWAN.** 2018. 50 f. TCC (Graduação) - Curso de Engenharia de Telecomunicações, Universidade Estadual Paulista "Júlio de Mesquita Filho", São João da Boa Vista, 2018.

NERY, Frederico Henrique. **DESENVOLVIMENTO DE UM PROTÓTIPO COM UTILIZAÇÃO DE LORA COMO ALTERNATIVA DE COMUNICAÇÃO NO CARREGAMENTO FERROVIÁRIO.** 2018. 62 f. TCC (Graduação) - Curso de Bacharelado em Engenharia Elétrica, Instituto Federal de Minas Gerais, Formiga, 2018.
Disponível em:

<https://www.formiga.ifmg.edu.br/documents/2018/Biblioteca/TCCs_e_Artigos/Frederico-Henrique-Nery_EE.pdf>. Acesso em: 23 maio 2020.

MACHADO, Sérgio Ricardo da Silva. **SISTEMA DE COMUNICAÇÕES SEM FIO EM INSTALAÇÕES PORTUÁRIAS::** redes em malha. 2014. 62 f. Monografia (Especialização) - Curso de Gerência de Redes de Computadores e Tecnologia Internet, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2014. Disponível em: <<https://pantheon.ufrj.br/bitstream/11422/3341/1/SMachado.pdf>>. Acesso em: 15 abr. 2020.

ROCHOL, Juergen. Sistemas de comunicação sem fio: conceito e aplicações. Porto Alegre: Bookman, 2018 485 p. (Livros didáticos informática UFRGS; v. 24).

ALYRIO, Danilo Rovigati. **MÉTODOS E TÉCNICAS DE PESQUISA**. 2009. Disponível em: <<http://www.faculdaderaizes.edu.br/files/images//M%C3%89TODOS%20E%20T%C3%89CNICAS%20DE%20PESQUISA.pdf>>. Acesso em: 20 jun. 2020.

FONTELLES, Mauro José *et al.* **METODOLOGIA DA PESQUISA CIENTÍFICA: DIRETRIZES PARA A ELABORAÇÃO DE UM PROTOCOLO DE PESQUISA**. 2009. Disponível em: <https://files.cercomp.ufg.br/weby/up/150/o/Anexo_C8_NONAME.pdf>. Acesso em: 10 mar. 2020.

Modulação em AM DSB SC. 2019. P&B. Disponível em: <https://www.youtube.com/watch?v=_5BmJry2E-U>. Acesso em: 22 jun. 2020.

MODBUS. **Modbus Protocol**. Disponível em: < <http://www.modbus.org/specs.php>>. Acesso em: 28 set. 2019.

NATIONAL INSTRUMENTS. **O Protocolo Modbus em Detalhes**. Disponível em: <<https://www.ni.com/pt-br/innovations/white-papers/14/the-modbus-protocol-in-depth.html#section-2027456811>>. Acesso em: 03 out. 2019.

ALFACOMP. **Modbus – Como funciona o protocolo de comunicação mais utilizado na automação industrial.** Disponível em: <<https://alfacompbrazil.com/2019/02/27/protocolo-modbus/>>. Acesso em: 01 out. 2019.

CONSTAIN, Nicole Beatriz Portilla. **Integração de sistemas SCADA com a implementação de controle supervisão em CLP para sistemas de manufatura.** UFSC. 2011. Disponível em: <<https://repositorio.ufsc.br/xmlui/handle/123456789/95357>> Acesso em: 01/10/2019.

PROFIBUS. **Tutorial SCADA.** Disponível em: <<http://www.profibus.org.br/news/fevereiro2012/tutorial>> Acesso em: 01 out. 2019.

LOGRHYTHM. **Scada network security monitoring.** Publicado em 01 de novembro de 2017. Disponível em: <<https://logrhythm.com/blog/scada-network-security-monitoring/>> Acesso em: 21 jun. 2020.

MOTA, Allan. **O que é Arduino e como funciona?** 2017. Disponível em: <<https://portal.vidadesilicio.com.br/o-que-e-arduino-e-como-funciona/>> Acesso em: 25 maio 2020.

SOUZA, Fábio. **Arduino Uno.** Disponível em: <<https://www.embarcados.com.br/arduino-uno/>> Acesso em: 20 maio 2020.

CARRION, Patrícia; QUARESMA, Manuela. **Internet da Coisas (IoT): Definições e aplicabilidade aos usuários finais.** 2019. Disponível em: <<http://www.revistas.udesc.br/index.php/hfd/article/viewFile/2316796308152019049/9858>> Acesso em: 20 maio 2020

TVDIGITAL II: Modos de Transmissão e Modulação. Disponível em: <https://www.teleco.com.br/tutoriais/tutorialtvdconsis2/pagina_4.asp. Acesso em: 20 maio 2020>