Introdução à Internet das Coisas

Tecnologias de Rede

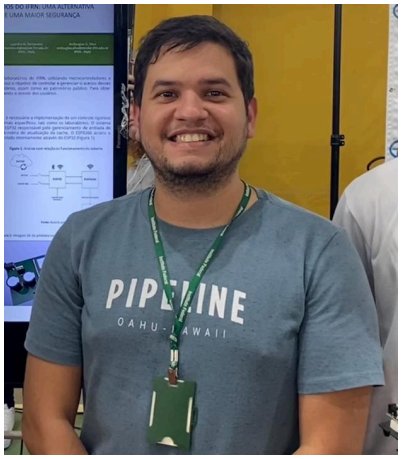
Apostila do aluno

Ph.D. Andouglas Gonçalves da Silva Júnior
Ph.D. Manoel do Bonfim Lins de Aquino
Marcos Fábio Carneiro e Silva
Autor da apostila

Ph.D. Andouglas Gonçalves da Silva Júnior
Ph.D. Manoel do Bonfim Lins de Aquino
Instrutor do curso

Larissa Jéssica Alves – Analista de Suporte Pedagógico
Revisão da apostila

Autor



Andouglas Gonçalves da Silva Júnior

Doutor em Engenharia Elétrica e da Computação - UFRN. Mestre em Engenharia Mecatrônica na área de Sistemas Mecatrônicos. Bacharel em Ciências e Tecnologia pela EC&T - Escola de Ciências e Tecnologia - UFRN. Engenheiro Mecatrônico - UFRN. Professor de Ensino Básico, Técnico e Tecnológico no Instituto Federal de Educação Tecnológica do Rio Grande do Norte (IFRN). Integrante da Rede de Laboratórios NatalNet, LAICA e colaborador ISASI-CNR-Itália. Desenvolve projetos na área de Machine Learning, Internet das Coisas e Holografia Digital. Colabora no projeto do N-Boat (Veleiro Robótico Autônomo), principalmente no desenvolvimento de sistemas para monitoramento da qualidade da água e identificação de micropartículas usando holografia digital e IA.

Manoel do Bonfim Lins de Aquino

Possui graduação (2006), mestrado (2008) e doutorado (2022) em Engenharia Elétrica e da Computação, pela Universidade Federal do Rio Grande do Norte - UFRN. Tem experiência na área de projetos de Telecomunicações, atuando na Siemens como engenheiro de Telecomunicações (2008-2010). Sou Professor (2010 - atual) do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte - IFRN, membro do NADIC, Núcleo de Análise de Dados e Inteligência Computacional, onde venho desenvolvendo projetos de P&D nas áreas de desenvolvimento de sistemas, IoT e Inteligência Artificial.

Marcos Fábio Carneiro e Silva

Estudante de informática no Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte. Participou, como bolsista, de projeto de pesquisa voltado à automação em ambiente escolar com IoT (2022 - 2023), Possui experiência em redes de computadores, eletrônica e IoT, com ênfase na utilização de microcontroladores e desenvolvimento de Hardware. Além disso, possui conhecimentos básicos em Python e C++. Atualmente é membro do NADIC (Núcleo de Análise de Dados e Inteligência Computacional) do IFRN.

APRESENTAÇÃO

Bem-vindo ao curso de **Introdução à Internet das Coisas** do *CEPEDI!*

A Internet das Coisas, ou IoT, é uma revolução tecnológica que está transformando a maneira como interagimos com o mundo ao nosso redor. Essa inovadora e crescente rede de dispositivos interconectados, que variam desde sensores e aparelhos domésticos até veículos e equipamentos industriais, está desencadeando uma mudança fundamental em como coletamos, compartilhamos e utilizamos informações.

Neste curso, introduziremos o fascinante mundo da IoT, definindo conceitos básicos, seus principais componentes e aplicações. Além disso, introduziremos os protocolos de comunicação mais usados em aplicações de Internet das Coisas, além dos principais dispositivos utilizados hoje, como ESP32, Arduino, Raspberry Pi Pico, entre outros.

Além disso, pretendemos oferecer um curso que mescle a teoria com a prática. Para isso, utilizaremos aplicações livres como Wokwi e Bipes para desenvolvimento de projetos que nos auxiliarão no aprendizado dos conceitos teóricos.

Recomendamos ao aluno que, ao final da leitura de cada seção, realize os exercícios propostos e acesse os materiais indicados nas referências bibliográficas, para aprofundar a leitura desse material e complementar o que foi lido aqui.

Desejo a você, prezado aluno, que tenha um excelente curso!!

Boa Leitura !!

Sumário

1 Introdução

Os protocolos de comunicação de rede são conjuntos de regras e convenções que definem como os dispositivos, em uma rede, devem se comunicar. Eles desempenham um papel crucial na garantia de uma comunicação eficiente, segura e padronizada. Neste capítulo, trataremos principalmente das diferentes interfaces de rede que podemos usar em aplicações IoT. Para isso, precisamos entender alguns conceitos básicos sobre redes de computadores.

1.1 Introdução à Redes de Computadores

As redes de computadores desempenham um papel central na revolução digital, permitindo a interconexão de dispositivos e a troca eficiente de informações em escala global. Uma rede de computadores é um conjunto de dispositivos eletrônicos interconectados, como computadores, servidores, roteadores e outros dispositivos, que compartilham recursos e informações. Essas redes desempenham um papel fundamental na comunicação e colaboração, proporcionando a base para a expansão da conectividade em praticamente todos os setores da sociedade.

No cerne das redes de computadores está o desejo de superar as limitações da comunicação ponto a ponto, permitindo que dispositivos em diferentes locais geográficos se comuniquem de maneira eficaz. Essa interconexão é realizada por meio de protocolos e padrões que definem como os dados são transmitidos, roteados e recebidos. Uma variedade de tecnologias, como cabos de fibra óptica, redes sem fio e satélites, é empregada para criar essa teia de comunicação que abrange o globo.

As redes de computadores podem ser classificadas em diferentes escalas, desde redes locais (LANs), que conectam dispositivos em uma área geográfica limitada, até redes globais, como a Internet, que interligam continentes inteiros. Cada tipo de rede apresenta desafios específicos e exige soluções adaptadas para garantir eficiência, segurança e confiabilidade.

Ao longo das últimas décadas, as redes de computadores evoluíram significativamente, acompanhando o avanço tecnológico e a crescente demanda por conectividade. Elas não apenas possibilitam a comunicação entre pessoas, mas também suportam a infraestrutura crítica de setores como saúde, educação, negócios e pesquisa científica.

As redes de computadores podem ser classificadas com base em vários critérios, como a escala geográfica, a relação entre os dispositivos, a tecnologia de transmissão, entre outros. A classificação com base em alguns desses critérios são apresentados a seguir.

Escala Geográfica

- Redes Locais (LAN): Cobrem uma área geográfica limitada, como um escritório, uma casa ou um campus.

- Redes Metropolitanas (MAN): Abrangem uma área geográfica maior do que as LANs, geralmente uma cidade ou região metropolitana.
- Redes de Longa Distância (WAN): Englobam uma área geográfica extensa, como um país, um continente ou globalmente.

Topologia

- Redes de Topologia em Estrela: Cada dispositivo é conectado a um ponto central (hub ou switch).
- Redes de Topologia em Anel: Os dispositivos são conectados em um formato de anel fechado.
- Redes de Topologia em Barramento: Todos os dispositivos compartilham o mesmo meio de comunicação.
- Redes de Topologia em Malha: Cada dispositivo está conectado a todos os outros dispositivos na rede.

Relação entre os dispositivos

- Redes Cliente-Servidor: Os dispositivos desempenham papéis distintos de cliente e servidor.
- Redes Ponto a Ponto (P2P): Todos os dispositivos têm funções semelhantes e podem agir tanto como clientes quanto como servidores.

Tecnologia de Transmissão

- Redes com Fio: Usam cabos físicos para a transmissão de dados.
- Redes Sem Fio (Wi-Fi): Utilizam comunicação sem a necessidade de cabos físicos.

Finalidade

- Redes de Armazenamento (SAN): Focadas no compartilhamento eficiente de recursos de armazenamento.
- Redes de Área Pessoal (PAN): Destinadas a conectar dispositivos pessoais em uma área restrita.
- Redes de Satélite: Usam satélites para comunicação, úteis em áreas remotas.

Um conceito importante na perspectiva de redes de computadores é o modelo de camadas OSI (Open Systems Interconnection) que consiste em um framework conceitual que define e padroniza as funções de comunicação de rede em sete camadas. Esse modelo foi desenvolvido pela ISO (International Organization for Standardization) para proporcionar uma abordagem modular e hierárquica para o design de redes. Cada camada no modelo OSI desempenha um papel específico na comunicação de dados, e a comunicação entre camadas adjacentes é realizada por meio de interfaces bem definidas. O modelo OSI ajuda a entender e organizar os diversos aspectos envolvidos em sistemas de comunicação de rede.

Falamos um pouco das camadas do modelo OSI quando definimos, no capítulo 1 do nosso curso, as camadas no contexto de aplicação IoT. O modelo OSI é baseado em 7 camadas, listadas abaixo, desde a camada mais baixa até a camada mais alta.

Camada Física (Physical Layer): Trata da transmissão física de bits sobre um meio de comunicação. Ela define as características elétricas, mecânicas e funcionais do hardware de rede.

Camada de Enlace de Dados (Data Link Layer): Fornece um serviço de entrega de quadros confiável entre dispositivos diretamente conectados. Ela trata do endereçamento físico (MAC), controle de acesso ao meio e detecção de erros.

Camada de Rede (Network Layer): Responsável pelo roteamento de dados entre diferentes redes. Ela fornece a capacidade de encaminhar pacotes de dados da origem para o destino através de redes intermediárias.

Camada de Transporte (Transport Layer): Oferece serviços de comunicação fim-a-fim, garantindo que os dados sejam entregues de maneira ordenada, sem erros e eficientemente. Ela controla o fluxo, a correção de erros e a retransmissão de dados.

Camada de Sessão (Session Layer): Esta camada estabelece, gerencia e encerra sessões de comunicação entre aplicações em diferentes dispositivos. Ela fornece serviços de diálogo, sincronização e controle de tokens.

Camada de Apresentação (Presentation Layer): A camada de apresentação lida com a tradução, compressão e criptografia dos dados, garantindo que os dados estejam em um formato legível e compreensível entre as aplicações.
Camada de Aplicação (Application Layer): Fornece interfaces para que as aplicações de software se comuniquem com a rede. Ela inclui protocolos de aplicação, como HTTP, SMTP e FTP.

Na Figura 4.1 temos uma tabela que relaciona as camadas do modelo OSI, suas respectivas unidades de dados de protocolo (do inglês, *Protocol Data Unit* - **PDU**), e os protocolos de comunicação.

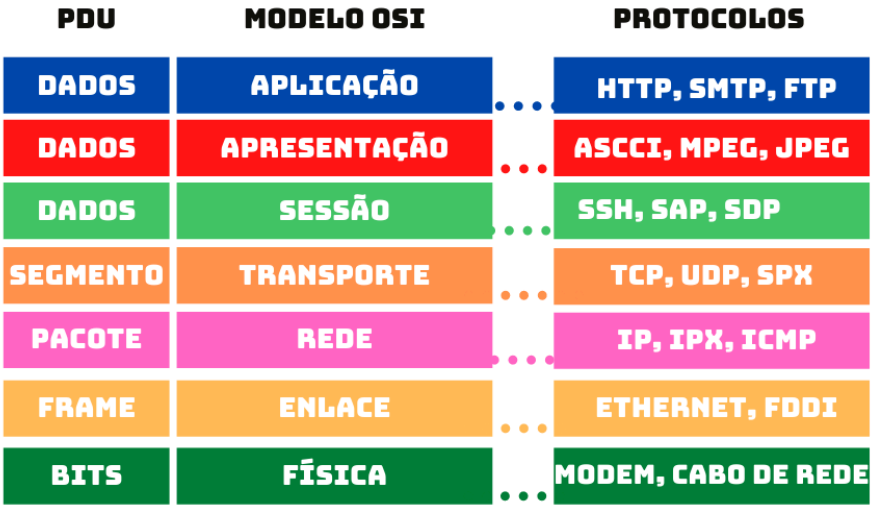


Fig. 4.1 - Camadas do modelo OSI (Fonte: https://edca.com.br/blog/modelo-osi).

Não entraremos em detalhes mais específicos sobre as camadas, já que queremos oferecer apenas uma introdução a essa temática, mas encorajamos o aluno a buscar mais informações sobre essa abordagem já que é bem importante dentro do contexto de comunicação de dados.

Para finalizar essa subseção, gostaríamos ainda de tratar do protocolo TCP/IP (Transmission Control Protocol/Internet Protocol) que consiste em um conjunto de protocolos de comunicação que define como os dispositivos em uma rede devem se comunicar. Ele serve como o modelo padrão para a comunicação em redes de computadores e é fundamental para o funcionamento da Internet. O TCP/IP é composto por dois protocolos principais, o TCP e o IP, além de outros protocolos relacionados. Listamos abaixo alguns conceitos importantes do TCP/IP.

IP (Internet Protocol): O IP é responsável pela atribuição de endereços únicos a cada dispositivo conectado à rede. Esses endereços IP são usados para identificar e localizar dispositivos em uma rede, permitindo a rotação eficiente dos dados entre eles. Existem duas versões principais do IP em uso: IPv4 (Internet Protocol version 4) e IPv6 (Internet Protocol version 6), sendo o IPv6 desenvolvido para resolver o esgotamento de endereços IPv4.

Máscara de Rede: A máscara de rede é um componente do endereço IP que indica quais partes do endereço são destinadas à identificação da rede e quais partes são destinadas à identificação do host (dispositivo específico dentro da rede). A máscara de rede é usada em conjunto com o endereço IP para definir a sub-rede à qual um dispositivo pertence. Ela é geralmente expressa como uma sequência de números binários ou em notação decimal, como "255.255.255.0". A máscara de rede ajuda a dividir o espaço de endereçamento IP em sub-redes, permitindo uma melhor organização e gerenciamento das redes.

Gateway: Falamos sobre o gateway no primeiro capítulo do nosso curso. Apenas para relembrar, o gateway é um dispositivo que atua como ponto de entrada ou saída entre duas redes distintas, permitindo a comunicação entre elas. Em redes domésticas, o gateway geralmente é o roteador, que conecta a rede local à Internet. O gateway gerencia o tráfego de dados entre a rede local e a rede externa, encaminhando os dados corretamente.

Porta: Refere-se a um número de identificação atribuído a um processo específico ou serviço que está sendo executado em um dispositivo. A porta destina-se a permitir que vários serviços ou processos diferentes em um mesmo dispositivo se comuniquem de maneira eficiente, garantindo que os dados sejam entregues à aplicação correta.

MAC: (Media Access Control) é um identificador exclusivo atribuído a uma interface de rede para comunicação em uma rede de computadores. Cada dispositivo de rede, como placas de rede, adaptadores Wi-Fi e dispositivos de rede embutidos, possui um endereço MAC único. Esses endereços são atribuídos pelos fabricantes de hardware e são utilizados para identificar de forma única cada dispositivo em uma rede. O endereço MAC é geralmente uma sequência alfanumérica única, composta por 12 caracteres hexadecimais (0-9,

A-F), organizados em pares. Por exemplo, um endereço MAC pode ser algo como "00:1A:2B:3C:4D:5E".

TCP (Transmission Control Protocol): O TCP é um protocolo de transporte confiável que fornece uma comunicação orientada à conexão. Ele garante que os dados sejam transmitidos de forma ordenada e sem erros entre dispositivos. O TCP divide os dados em pacotes, reorganiza-os na extremidade de recepção e solicita retransmissões de pacotes perdidos.

UDP (User Datagram Protocol): Assim como o TCP, o UDP é um protocolo de transporte, mas não oferece a mesma confiabilidade. Ele é usado quando a velocidade e a eficiência na transmissão de dados são mais importantes do que a garantia de entrega. O UDP é comumente usado em aplicações de transmissão ao vivo, videoconferência e jogos online.

ICMP (Internet Control Message Protocol): O ICMP é usado para enviar mensagens de controle e mensagens de erro. Por exemplo, é utilizado para testar a conectividade entre dispositivos em uma rede por meio do comando "ping".

HTTP (Hypertext Transfer Protocol): O HTTP é um protocolo de aplicação usado para transferir informações na World Wide Web. Ele define como os documentos, como páginas da web, são formatados e transmitidos pela Internet. Falaremos mais sobre este protocolo no próximo capítulo.

FTP (File Transfer Protocol): O FTP é um protocolo que permite a transferência de arquivos entre computadores em uma rede. Ele é frequentemente usado para o upload e download de arquivos em servidores.

Agora que já entendemos um pouco sobre alguns conceitos básicos de redes de computadores, podemos explorar algumas das tecnologias mais utilizadas em aplicações de internet das coisas.

1.2 Ethernet

O protocolo Ethernet representa um dos pilares fundamentais das redes de computadores, delineando como os dados fluem entre dispositivos conectados em uma rede local (LAN). Ele é adotado em diversos cenários, desde ambientes domésticos até redes empresariais e industriais.

Uma característica marcante do protocolo Ethernet é sua natureza de acesso ao meio compartilhado. Neste modelo, todos os dispositivos na rede dividem o mesmo canal de comunicação. Para evitar possíveis colisões dos dados que trafegam nesse canal, o protocolo incorpora um mecanismo de controle de acesso chamado CSMA/CD (*Carrier Sense Multiple Access with Collision Detection*). Esse mecanismo garante eficiência na transmissão dos dados, conforme a seguinte lógica:

- Um dispositivo que pretende transmitir dados verifica se o canal está livre.
- Se o canal estiver livre, o dispositivo inicia a transmissão dos dados.
- Se o canal estiver ocupado, o dispositivo espera até que esteja livre.
- Caso dois dispositivos tentem transmitir simultaneamente, ocorre uma colisão.
- Em caso de colisão, os dispositivos envolvidos tentam transmitir novamente após um intervalo aleatório.

Esse fluxo é apresentado na Figura 4.2.

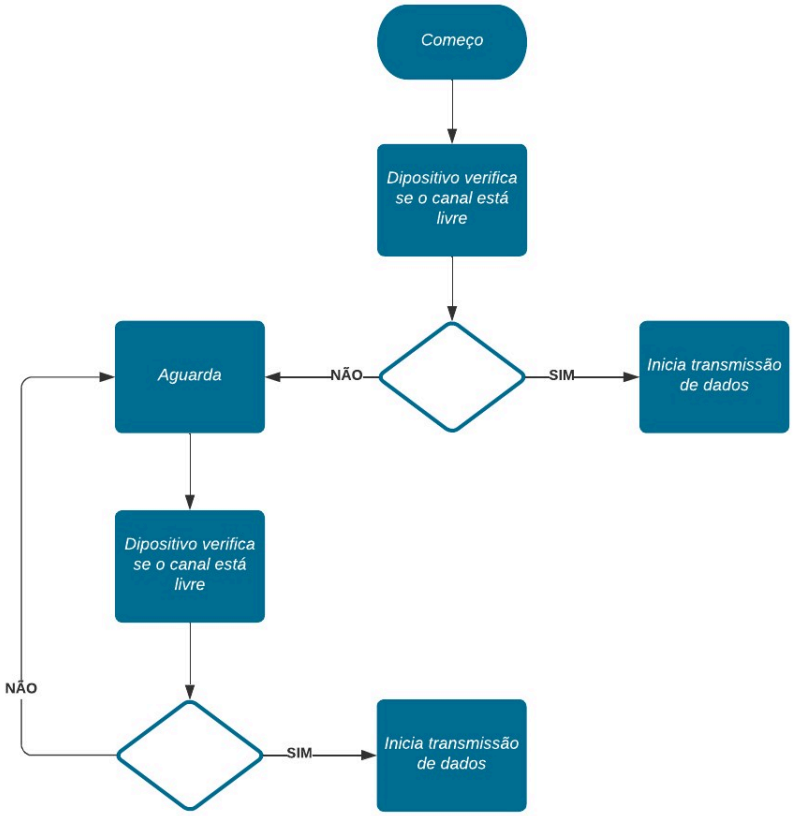


Fig. 4.2 - Fluxograma de transmissão de dados na Ethernet. Autoria Própria.

A arquitetura de camadas é uma característica marcante do Ethernet, onde a camada física, especificada pelo padrão IEEE 802.3, é responsável pela transmissão dos dados na rede. Além disso, o ethernet se destaca por sua robustez e confiabilidade, tornando-o uma escolha ideal para aplicações de Internet das Coisas (IoT) que demandam alta disponibilidade e confiabilidade. Por outro lado, tem-se a necessidade de conexões cabeadas, que podem ser inviáveis dependendo da aplicação.

A utilização da Ethernet em sistemas baseados em ESP ou Pi Pico não é muito comum, já que para sua utilização é necessário a utilização de *shields* ou *hats*. Além disso, essas placas já apresentam outras formas de comunicação de rede, como WiFi e Bluetooth, o que as tornam ótimas opções para redes sem fio.

O Exemplo 4.1 mostra um código da utilização do Shield Ethernet em um Arduino UNO que aciona um relé quando recebe a mensagem

“releParam=1” via protocolo HTTP (falaremos mais sobre este protocolo no próximo capítulo). É importante salientar que a comunicação entre os dois componentes acontecem usando o protocolo SPI. Basicamente, abrimos uma conexão usando o método *begin* da biblioteca Ethernet, onde passamos o mac, o ip, o gateway e a máscara de rede. Então, um servidor é iniciado na porta padrão (80) que aguarda a comunicação de um cliente. Quando essa comunicação é realizada, é feita a leitura dos parâmetros da URL que indica se o pino ligado ao relé é para ser colocado em nível alto ou baixo.

Exemplo 4.1 - Exemplo de utilização da biblioteca Ethernet para acionamento de um relé no Arduino

```
#include <SPI.h>
#include <Ethernet.h>
//Atribuindo um endereço MAC ao Shield Ethernet
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
//IP baseado na configuração de rede do seu roteador
byte ip[] = { 192, 168, 0, 175 };
//Gateway de conexão
byte gateway[] = { 192, 168, 0, 1 };
//Máscara de Rede
byte subnet[] = { 255, 255, 255, 0 };
//Inicialização do server na porta padrão 80
EthernetServer server(80);
//Pino onde o relé está conectado a placa
const int relay = 3;
//Variável para leitura da url
String readString = String(30);

void setup() {
 //Inicia a biblioteca eo servidor
 Ethernet.begin(mac, ip, gateway, subnet);
 server.begin();
 //Inicia o pino do relé em nível baixo
 digitalWrite(relay, LOW);
}

void loop() {
 //Cria uma conexão com o cliente
 EthernetClient client = server.available();
 // Se o cliente existe, enquanto ele estiver conectado e
disponível
 if (client) {
   while (client.connected()) {
     if (client.available()) {
```

```
        //Lê o caracter da requisição HTTP
        char c = client.read();
        if (readString.length() < 100) {
          // "readstring" recebe o valor da URL
          readString += c;
        }
        //Quando encontra o "\n" significa que é o fim do
cabeçalho da requisição
        if (c == '\n') {
          //Procura o caracter ? para encontrar os parâmetros
          if (readString.indexOf("?") < 0) {
          }
          //Se relayParam=1 aciona o relé
          else if (readString.indexOf("relayParam=1") > 0) {
            digitalWrite(relay, HIGH);  //Aciona o relé
          } else {
            //Desativa o relé
            digitalWrite(relay, LOW);
          }
          //Variável é reinicializada
          readString = "";         ]
      //Finaliza a conexão com o cliente
      client.stop();
        }
      }
    }
  }
}
```

A Figura 4.3 mostra o shield Ethernet e o módulo relé ligado ao Arduino Uno.

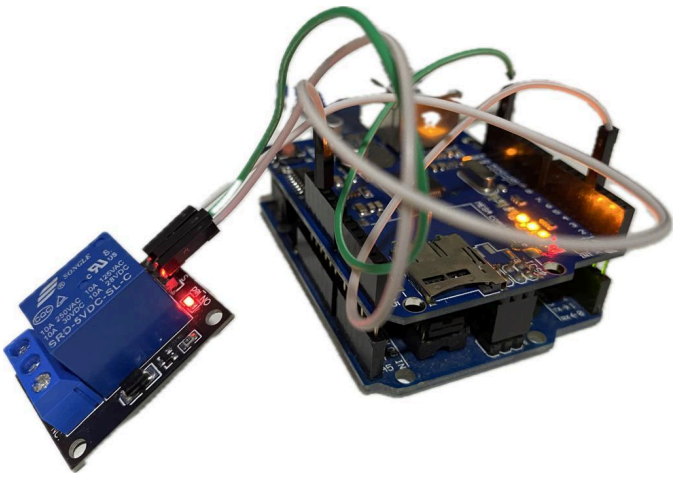


Fig. 4.3 -Shield Ethernet e módulo relé ligado ao Arduino UNO. Autoria Própria.

1.3 Wi-Fi

O Wi-Fi, abreviação de "Wireless Fidelity," é uma tecnologia de comunicação sem fio que permite a conectividade de dispositivos em redes locais (LANs) e à internet. Wi-Fi é baseado em uma variedade de padrões definidos pelo IEEE (Institute of Electrical and Electronics Engineers), sendo os mais comuns aqueles que pertencem à família IEEE 802.11.

O Wi-Fi opera em várias camadas do modelo OSI. A camada física (camada 1) lida com a transmissão de sinais sem fio, enquanto a camada de enlace (camada 2) trata de questões como o controle de acesso ao meio e o endereçamento MAC (Media Access Control). As camadas superiores do modelo OSI (camadas de rede, transporte, sessão, apresentação e aplicação) não são especificamente ligadas ao Wi-Fi, mas aplicam-se da mesma forma a redes com e sem fio.

Existem vários padrões Wi-Fi, como 802.11a, 802.11b, 802.11g, 802.11n, 802.11ac, 802.11ax (Wi-Fi 6), e assim por diante. Cada padrão define especificações para a frequência, largura de banda, velocidade de transmissão, segurança e outras características da rede sem fio. Novos padrões são desenvolvidos periodicamente para melhorar o desempenho e a eficiência das redes Wi-Fi.

O exemplo 4.X apresenta a utilização da rede WiFi no ESP32 usando a biblioteca *network* do micropython. Uma das vantagens do simulador do wokwi é justamente a capacidade que ele tem de se comunicar com a rede em que o computador está conectado.

Exemplo 4.2 - Exemplo de Conexão com a Internet usando o WiFI do ESP32 - Micropython

```
import network
import time

print("Conectando no WiFi", end="")
#Criamos uma conexão do tipo WLAN para comunicação com a
internet
sta_if = network.WLAN(network.STA_IF)
#Ativamos o módulo WiFi da placa
sta_if.active(True)
#Usamos o 'Wokwi-GUEST' quando queremos que o wokwi conecte
com
```

```
#a rede em que o computador está conectado
sta_if.connect('Wokwi-GUEST', '')
#Tenta realizar a conexão
while not sta_if.isconnected():
 print(".", end="")
 time.sleep(0.1)
#Se conectado mostra
print(" Conectado!")
```

Saiba mais! Podemos verificar se a placa está de fato conectada à nossa rede local usando comandos no terminal do linux, como *nmap* e *ping*. O *nmap* permite que encontremos todos os dispositivos conectados ao roteador. Por exemplo, o comando a seguir busca os hosts conectado a sub-rede 192.168.0.0/24

```
sudo nmap -sn 192.168.0.0/24
```

Já o comando *ping* permite que mandemos um "sinal" para a placa e recebamos um retorno, quando conhecemos o endereço IP do dispositivo dentro da rede. Por exemplo, supondo que o dispositivo esteja conectado a rede com o ip 192.168.0.175, usamos o seguinte comando para realizar a comunicação com o dispositivo de outra máquina.

```
ping 192.168.0.175
```

1.4 Bluetooth

O Bluetooth, protocolo de rádio que revolucionou a conectividade de dispositivos, surgiu em 1994 como resposta à crescente necessidade de interligar gadgets móveis de forma eficiente. Essa inovação, impulsionada por empresas pioneiras como Ericsson, IBM, Nokia, Intel e Toshiba, proporcionou uma reviravolta na maneira como fones de ouvido, celulares, impressoras e outros dispositivos se comunicam.

O nome "Bluetooth" tem suas raízes na história, homenageando o rei Harald Bluetooth, que governou à distância os reinos da Dinamarca e Noruega. Assim como o monarca conectava terras distantes, o protocolo conecta dispositivos em curtos alcances, variando de 10 a 100 metros.

Desde sua estreia em 1999 até a versão mais recente, o Bluetooth 4.0, que alcança velocidades de 24 Mbit/s, a tecnologia evoluiu constantemente. Seu objetivo é complementar ou até mesmo substituir redes cabeadas convencionais, oferecendo uma alternativa sem fio mais flexível e conveniente.

Além disso, sua aplicação abrange áreas como comunicação, entretenimento e Internet das Coisas (IoT), consolidando sua posição como uma ferramenta indispensável na era da conectividade.

1.4.1 Principais Características do Bluetooth

A principal característica do Bluetooth é sua natureza sem fio, eliminando a necessidade de cabos para conectar dispositivos. Essa ausência de fios não apenas simplifica a configuração, mas também amplia as possibilidades de interconexão em ambientes diversos.

Outro ponto a favor do Bluetooth é seu custo acessível, tornando-o uma opção viável para uma ampla gama de usuários. Essa acessibilidade contribui para sua popularidade e adoção generalizada em diversos setores. Dentre algumas características mais importantes, podemos destacar a segurança, a taxa de transferência e o tempo de detecção.

Segurança

A segurança do Bluetooth é uma prioridade, respaldada por uma série de mecanismos e protocolos que garantem uma comunicação protegida entre dispositivos. A base dessa segurança está na modulação FHSS (*Frequency Hopping Spread Spectrum*), um método que constantemente altera as frequências de operação, dificultando interferências de outros dispositivos na mesma faixa.

Esse protocolo exemplifica seu comprometimento com a segurança por meio de quatro níveis distintos. O primeiro, **a autenticação**, assegura que apenas dispositivos autorizados possam estabelecer comunicação. Este nível é altamente recomendado para situações críticas, como transferência de dados sensíveis. Em contrapartida, o **Modo Inseguro** é a opção para situações em que a segurança não é prioritária, sendo ideal para transferência de dados não confidenciais. Já o ***Service Level Security* (SLS)** proporciona um nível de segurança dinâmico, adaptando-se às demandas específicas de cada aplicação. Por fim, o ***Link Level Security* (LLS)** oferece uma camada de segurança comum a todas as aplicações, ideal para cenários em que a autenticação não é viável.

A segurança da comunicação Bluetooth é solidificada pelo uso de criptografia de 128 bits, conferindo um alto padrão de proteção aos dados transmitidos. A modulação FHSS, ao dividir a faixa de frequência em 79 canais, permutados 1600 vezes por segundo, reduz significativamente a probabilidade de dois dispositivos operarem na mesma frequência simultaneamente.

Taxa de transferência

O Bluetooth, ao longo de sua evolução em quatro versões distintas – 1.0, 2.0, 3.0 e 4.0 –, testemunhou melhorias substanciais, especialmente no que diz respeito às taxas de transmissão de dados. A versão inaugural, 1.0, estabeleceu um patamar com uma taxa de 723,2 kbit/s e um alcance de até 10 metros. Contudo, o avanço se tornou notório com o tempo, culminando na versão 4.0, que, enquanto priorizava a economia de energia, mantinha uma taxa de transmissão padrão de 1 Mbit/s.

Ao se posicionar em relação a outras tecnologias, o Bluetooth destaca-se por seu equilíbrio entre velocidade, alcance e eficiência energética. Em comparação com o ZigBee (veremos esse protocolo um pouco mais a frente no nosso curso), conhecido por sua eficiência energética, o Bluetooth oferece taxas de transmissão mais elevadas, tornando-se uma escolha atraente para aplicações que demandam um equilíbrio entre desempenho e economia de energia.

Tempo de detecção

O tempo de detecção de dispositivos emerge como um aspecto crucial na avaliação das tecnologias sem fio, sendo esse um fator determinante tanto para a eficiência energética quanto para a agilidade operacional dos dispositivos móveis. Nesse contexto, o Bluetooth sobressai-se como a tecnologia líder, apresentando o menor tempo de detecção, com um intervalo notável de 2,5 milissegundos a 10,24 segundos.

Comparativamente, o Zigbee, outra tecnologia sem fio, demanda aproximadamente 30 milissegundos para realizar a detecção de dispositivos. Já o Wi-Fi, embora seja amplamente utilizado para conectividade de alta velocidade, requer um tempo mais substancial, levando 30 segundos para identificar dispositivos.

A relevância do tempo de detecção vai além da mera eficiência operacional, sendo crucial para a gestão energética dos dispositivos móveis. Um tempo prolongado de detecção pode resultar em um aumento significativo no consumo de energia, impactando diretamente a vida útil da bateria dos dispositivos.

Além disso, o tempo de detecção também desempenha um papel vital na agilidade de visualização dos dispositivos disponíveis no sistema. Um tempo prolongado pode retardar o processo de seleção de dispositivos, afetando a experiência do usuário e a eficácia das interações.

1.4.2 Comunicação entre dispositivos

A estrutura de uma rede Bluetooth revela-se como um sistema flexível e dinâmico, possibilitando a comunicação simétrica entre dispositivos, que podem alternar entre as posições de cliente e servidor conforme a necessidade. Essa comunicação é organizada na unidade fundamental chamada *piconet*, composta por um dispositivo **mestre** e até sete dispositivos **escravos**. Dentro dessa estrutura, a interação ocorre exclusivamente entre o mestre e os escravos, nunca diretamente entre estes últimos.

A criação de uma *scatternet*, que permite a conexão de várias *piconets*, amplia ainda mais as possibilidades de interconexão, conferindo flexibilidade ao sistema Bluetooth. O estabelecimento de conexões Bluetooth é conduzido por um processo em três etapas: *scan, inquiry e page*, garantindo uma configuração eficiente A Figura 4.4 exemplifica a organização da *Piconet* e da *Scatternet*.

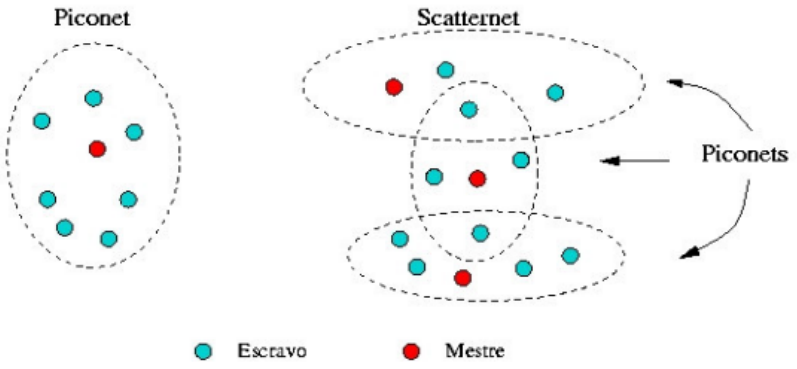


Fig. 4.4. *Piconet* e *Scatternet* (FERREIRA et. al., 2005).

A anatomia dos dispositivos Bluetooth compreende seis componentes essenciais, apresentados na Figura 4.5: *host controller*, *link control processor*, *baseband controller*, *transceiver RF*, *RF front-end* e antena. Essa complexidade é gerenciada por uma pilha de protocolos Bluetooth, que inclui protocolos como RFCoom (Protocolo de Substituição de Cabo) e TCS-BIN (Protocolo de Controle de Telefonia), além de adotar protocolos bem conhecidos como PPP, TCP/UDP/IP, OBEX e WAP.

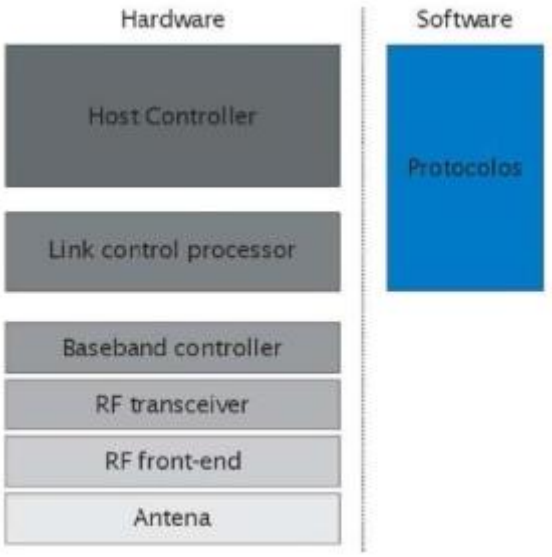


Fig. 4.5. Pilhas de protocolo e hardware do Bluetooth (fonte: https://www.gta.ufrj.br/grad/10_1/rssf/protocolos.html).

A *piconet*, caracterizada por ser uma rede ad hoc não gerenciada centralmente, opera na faixa de frequência de 2,4 GHz. Essa operação é facilitada pela modulação FHSS (*Frequency Hopping Spread Spectrum*), uma técnica que altera rapidamente a frequência de transmissão, reduzindo a interferência de outros dispositivos na mesma faixa.

O Bluetooth destaca-se ainda por ser uma tecnologia de baixo consumo de energia, tornando-a especialmente adequada para aplicações móveis. Essa característica reforça sua posição como uma opção eficaz para conectar dispositivos em uma variedade de contextos.

1.4.3 Aplicações

O Bluetooth desempenha um papel crucial no cenário da Internet das Coisas (IoT), oferecendo uma variedade de aplicações que impulsionam a

conectividade e a interação entre dispositivos inteligentes. Algumas das principais aplicações do Bluetooth no contexto do IoT incluem:

- Dispositivos vestíveis (*Wearables*): Os dispositivos Bluetooth, como *smartwatches, fitness trackers* e outros dispositivos vestíveis, são amplamente utilizados para monitorar a saúde, rastrear atividades físicas e interagir com outros dispositivos conectados.
- Conectividade de veículos: O Bluetooth é integrado a sistemas de entretenimento, comunicação e segurança em veículos. Além disso, é usado para conectar *smartphones* a sistemas de infoentretenimento, permitindo chamadas mãos-livres, *streaming* de música e integração com assistentes virtuais.

Um exemplo de utilização do Bluetooth no ESP32 é mostrado no exemplo 4.3. Usaremos a biblioteca *BluetoothSerial* para criar uma ponte entre o Serial da placa e o Bluetooth.

Exemplo 4.3 - Exemplo de uitlização do Bluetooth em um ESP32 - C++

```
//Inclusão da biblioteca
#include "BluetoothSerial.h"
//O "if" a seguir verifica se o o Bluetooth está habilitado
#if !defined(CONFIG_BT_ENABLED) ||
!defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled! Please run `make
menuconfig` to and enable it
#endif
//Cria uma instância da biblioteca
BluetoothSerial SerialBT;

void setup() {
  //Inicializa a Serial
  Serial.begin(115200);

  //Inicializa o Bluetooth com o nome ESP32Test
  SerialBT.begin("ESP32test");
  Serial.println("Dispositivo iniciado! Pode realizar o
pareamento...");
}

void loop() {
  if (Serial.available()) {
```

```
    //Se disponível na serial, escreve no Bluetooth
    SerialBT.write(Serial.read());
  }
  //Se disponível na Bluetooth, escreve na Serial
  if (SerialBT.available()) {
    Serial.write(SerialBT.read());
  }
  delay(20);
}
```

1.5 Bluetooth Low Energy (BLE)

A versão clássica do Bluetooth, por si só, tem se mostrado ser uma tecnologia muito bem estruturada e útil para diversas aplicações. Dentre essas aplicações, podemos destacar seu uso em soluções de Internet das Coisas, que possibilita conexões em redes pessoais, conhecidas como PAN (Personal Area Network). O grande problema dessa versão é a grande quantidade de energia que é requerida para manter as conexões Bluetooth.

Em dispositivos como notebooks e celulares, que possuem uma bateria de alta capacidade, isso não é um problema, mas para dispositivos de IoT que possuem limitadas reservas de energia, acaba sendo um grande limitador.

Neste cenário surge o *Bluetooth Low Energy* (BLE) no campo das comunicações sem fio. Especialmente projetada para dispositivos que demandam eficiência energética e transmissão de dados descontínua. Seu principal diferencial reside na capacidade de manter dispositivos em modo sleep durante a maior parte do tempo, ativando-se brevemente para conexões rápidas.

O BLE é baseado no Bluetooth clássico e fornece várias de suas características, mantendo um baixo consumo de energia. É claro que isso diminui as possibilidades de aplicação, principalmente quando envolve o envio de uma grande quantidade de dados. Desta forma, o BLE é uma ótima tecnologia quando se pretende transferir (enviar e receber) pequenas quantidades de dados em uma rede PAN, cobrindo áreas menores. Entender como o BLE funciona é fundamental para saber empregá-lo em aplicações IoT.

1.5.1 Funcionamento

O Bluetooth LE, também conhecido como Bluetooth Smart, usa a mesma tecnologia de saltos que o Bluetooth clássico usa. Ele também transmite dados e conecta com vários dispositivos eletrônicos usando a banda de 2.4 GHz. Porém, ele é projetado para ser mais devagar, enviando no máximo aproximadamente 1Mbps. Isso permite um consumo mais baixo de energia, ficando entre 0.01 e 0.5 watts. Para se ter um parâmetro de comparação, a versão clássica consome 1 watt de energia.

Não é somente a redução de velocidade de transmissão que possibilita ao BLE ser uma tecnologia com consumo de energia baixo. Por exemplo, quando dois dispositivos estão conectados, eles se comunicam por apenas alguns poucos segundos. Além disso, os dispositivos podem "dormir" ou se desligarem entre cada conexão. Ou seja, ao invés de transmitirem por horas, os dispositivos BLE transmitem dados de forma efetiva, quando necessários, usando assim, menos energia.

A comunicação entre dispositivos BLE pode acontecer de dois modos diferentes: orientado à conexão ou *broadcasting*.

Orientado à Conexão

Neste modo, um dispositivo BLE pode atuar como dispositivo periférico ou servidor. Neste caso, ele recebe uma requisição de pareamento depois de ter sido encontrado no radar de um dispositivo central. Toda a comunicação é composta por 4 passos - *advertise* (anunciar), *initiate* (iniciar), *connect* (conectar), e *exchange* (troca de informação). O processo acontece da seguinte forma:

1) O dispositivo periférico envia pacotes de **anúncios** temporizados;
2) O dispositivo central escaneia e usa os pacotes de anúncio para encontrar os dispositivos periféricos.
3) Começa então o estágio de **inicialização**, onde o dispositivo periférico inicia uma requisição de inicialização e a **conexão** é estabelecida.
4) Por fim, a **transmissão** de dados inicia.

Broadcasting

Esse modo de comunicação é também chamado de *bluecasting*. Neste caso não existe um processo de comunicação, o dispositivo BLE simplesmente envia dados em um formato unidirecional e qualquer dispositivo ao redor pode receber essa informação.

Perceba que neste método não é fornecido uma camada de segurança de dados, já que qualquer dispositivo pode acessar o dado compartilhado sem segurança. Ou seja, este modo é usado em aplicações em que a segurança da informação compartilhada não é um problema.

1.5.2 Arquitetura

A base da arquitetura do BLE consiste no modelo em que um dispositivo se comporta como servidor e outro como cliente. A Figura 4.6 mostra uma estrutura em camadas dessa arquitetura.

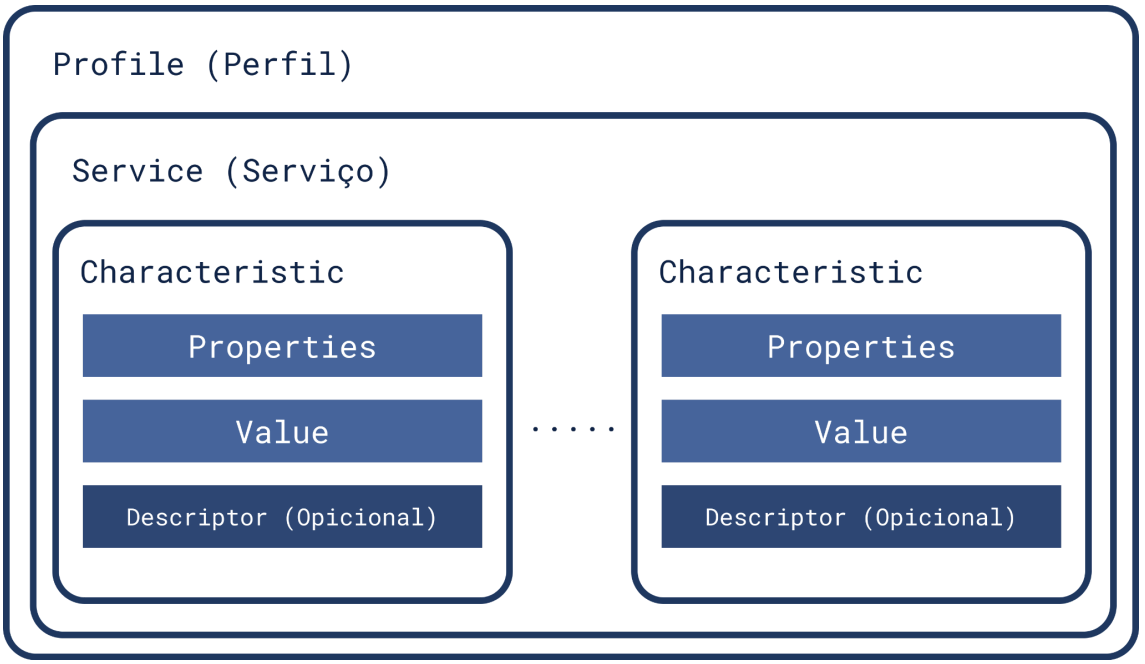


Fig. 4.6 - Arquitetura do BLE

Os conceitos dos termos são apresentados a seguir.

Profile (Perfil) - Um perfil Bluetooth define a funcionalidade específica que um dispositivo pode oferecer. Ele descreve como os serviços, características e

atributos (como propriedades e valores) estão organizados para suportar uma aplicação ou caso de uso específico. Alguns *profiles* já definidos incluem:

- BLP – Blood Pressure Profile (Pressão Sanguínea)
- HTP – Health Thermometer Profile (Termômetro)
- GLP – Glucose Profile (Glicose)
- HRP – Heart Rate Profile (Frequência cardíaca)
- CPP – Cycling Power Profile (Ciclismo)
- WSP – Weight Scale Profile (Peso)
- LNP – Location and Navigation Profile (Localização e Navegação)
- ESP – Environmental Sensing Profile (Sensoriamento Ambiental)

Service (Serviço): Um serviço é um conjunto de características que juntas oferecem uma funcionalidade específica. Cada serviço tem um identificador único chamado UUID (Universally Unique Identifier)

Characteristic (Característica): Uma característica é uma unidade de dados dentro de um serviço que representa uma propriedade ou valor específico. Cada característica também tem um UUID exclusivo.

Properties (Propriedades): As propriedades de uma característica descrevem a natureza da operação que pode ser realizada na característica, como leitura (READ), gravação (WRITE) ou notificação (NOTIFY).

Value (Valor): O valor é a informação real contida em uma característica. Por exemplo, pode ser um dado de temperatura, um texto ou qualquer outro tipo de informação que a característica está projetada para transportar.

Descriptor (Descritor): Um descritor fornece informações adicionais sobre uma característica, como configurações ou metadados. Ele descreve características específicas da característica, como limites, unidades de medida, entre outros.

Saiba mais! UUID (Universally Unique Identifier) é um identificador único universal. É uma cadeia alfanumérica de 128 bits (16 bytes) que é gerada de maneira que seja única em um espaço e tempo específicos. A intenção por trás dos UUIDs é garantir uma identificação única mesmo em ambientes distribuídos, onde a chance de colisão (duas entidades gerando o mesmo UUID) é extremamente baixa. Os UUIDs são geralmente representados como uma sequência de 32 caracteres hexadecimais divididos em grupos, por exemplo:

550e8400-e29b-41d4-a716-446655440000

Para ficar mais claro, vamos criar um exemplo para ilustrar essa arquitetura.

- **Profile**
  - Nome: Monitor de Saúde
  - UUID: 123e4567-e89b-12d3-a456-426614174000
- **Service**
  - Nome: Monitor de Batimentos Cardíacos
  - UUID: 5678abcd-1234-5678-5678-abcdef123456
- **Characteristic**
  - Nome: Batimentos Cardíacos Atuais
  - UUID: 9abcdef0-5678-9abc-1234-56789abcdef0
- **Properties**
  - READ - Pode ser lida para obter o valor atual dos batimentos cardíacos.
  - NOTIFY - Pode notificar assincronamente sobre mudanças nos batimentos cardíacos.
- **Value**
  - Exemplo: 75 bpm (batimentos por minuto)
- **Descriptor**
  - Nome: Limite Superior de Batimentos Cardíacos
  - UUID: def01234-5678-9abc-def0-123456789abc
  - Valor do descritor: 120 bpm (definindo o limite superior para notificações)

Vamos agora estudar um exemplo de um código usando micropthyon que usa o BLE do ESP32 para acionar um LED baseado no recebimento de um dado de texto de um outro dispositivo, como uma comunicação UART. Esse exemplo é completo e analisaremos ele por partes. Primeiramente, importamos as bibliotecas que iremos utilizar.

Exemplo 4.4 - Parte I

```
from machine import Pin, Timer
from time import sleep_ms
import ubluetooth #biblioteca para utilização do BLE
```

Depois, vamos criar uma classe chamada *BLE* para controlar a sua utilização baseado nos conceitos de sua arquitetura que estudamos nesta seção. Iniciamos a classe instânciando a classe BLE da biblioteca *bluetooth.* Além disso, definimos o led que fará a indicação luminosa dos diversos estágios da aplicação e os *timers* que controlarão o tempo de on/off do LED.

Exemplo 4.4 - Parte II

```
class BLE():
    def __init__(self, name):
        self.name = name
        #Instanciando a classe BLE do ubluetooth
        self.ble = ubluetooth.BLE()
        #Ativando o BLE
        self.ble.active(True)

        #Definição do LED e dos Timers
        self.led = Pin(2, Pin.OUT)
        self.timer1 = Timer(0)
        self.timer2 = Timer(1)

        #Atribuição das possíveis estágios da app
        self.disconnected()
        self.ble.irq(self.ble_irq)
        self.register()
        self.advertiser()
```

Agora, definiremos os possíveis estágios que a aplicação, dentro do contexto de utilização do BLE, poderá estar definida. O método *register* corresponde às definições do **serviço** da nossa aplicação. Usaremos um serviço Bluetooth GATT (Generic Attribute Profile) desenvolvido pela Nordic Semiconductors, chamado Nordic UART Service (NUS). Ou seja, o **perfil** da nossa aplicação é genérico e o **serviço** é o NUS. Primeiro, definidos os UUID do serviço e de cada **característica**, que no caso da nossa aplicação é o envio

(TX) e o recebimento (RX) de dados. As **propriedades** dessas características são, respectivamente, *NOTIFY* e *WRITE*.

Exemplo 4.4 - Parte III

```
def register(self):
        # Nordic UART Service (NUS)
        NUS_UUID = '6E400001-B5A3-F393-E0A9-E50E24DCCA9E'
        RX_UUID = '6E400002-B5A3-F393-E0A9-E50E24DCCA9E'
        TX_UUID = '6E400003-B5A3-F393-E0A9-E50E24DCCA9E'

        # Definição do UUID do serviço
        BLE_NUS = ubluetooth.UUID(NUS_UUID)

        # Definição da característica RX
        BLE_RX = (ubluetooth.UUID(RX_UUID),
ubluetooth.FLAG_WRITE)

        # Definição da característica TX
        BLE_TX = (ubluetooth.UUID(TX_UUID),
ubluetooth.FLAG_NOTIFY)

        # Definição do serviço
        BLE_UART = (BLE_NUS, (BLE_TX, BLE_RX,))

        # Lista de serviços, no nosso caso apenas um
        SERVICES = (BLE_UART, )

        # Registro dos serviços no perfil
        ((self.tx, self.rx,), ) =
self.ble.gatts_register_services(SERVICES)
```

Os métodos *send* e *advertiser*, como os nomes sugerem, servem para enviar dados (no caso da propriedade TX) e entrar no modo de anúncio (advertise). No caso do método *gatts_notify*, indicamos o cliente que receberá a notificação, o serviço (no caso, *self.tx*) e o dado que queremos enviar. Já no *gap_advertise,* indicamos o intervalo de tempo em microssegundos que o anúncio será disponibilizado e o dado que queremos enviar.

Exemplo 4.4 - Parte IV

```
    #Método para envio de dados
    def send(self, data):
        self.ble.gatts_notify(0, self.tx, data + '\n')

    #Método para entra no modo de 'advertise'
    def advertiser(self):
        name = bytes(self.name, 'UTF-8')
        self.ble.gap_advertise(100,
bytearray('\x02\x01\x02') + bytearray((len(name) + 1, 0x09))
+ name)
```

Ok! Mas o que significa o dado que estamos enviando no *gap_advertise?* Aqui precisamos entender que estamos enviando um dado dentro de um protocolo e, para isso, fazemos algumas indicações para servir de parâmetro para o dispositivo que vai receber. No nosso caso específico, iniciamos o dado com o array de bytes '\x02\x01\x02'. Esse cabeçalho indica que, da esquerda para a direita, em seguida enviaremos \x02 = 2 bytes no campo de Flags; onde o primeiro byte corresponde ao tipo da FLAG, no caso \x01=1; e o segundo byte corresponde as FLAGs específicas que estamos enviando, no nosso caso \x02=00000010. Em outras palavras, a sequência **\x02\x01\x02** nos dados de anúncio significa que o campo de flags tem um comprimento de 2 bytes, o tipo de dado é "Flags" e o dispositivo está em modo de descoberta geral (General Discoverable Mode).

Saiba mais! O segundo byte do campo de flags contém os bits específicos das flags. Cada bit neste byte representa uma característica ou configuração específica do dispositivo.

- Bit 0 (LSB): Se este bit estiver definido (1), o dispositivo está em modo de descoberta limitada (Limited Discoverable Mode).
- Bit 1: Se este bit estiver definido (1), o dispositivo está em modo de descoberta geral (General Discoverable Mode).
- Bits 2-7: Reservados.

Finalmente, os métodos *connected* e *disconnected* são usados para controlar o funcionamento dos times. O método *ble_irq* funciona como uma interrupção, controlando os estados da aplicação baseado nos eventos. Por exemplo, se o ESP estiver desconectado de outros dispositivos, então ele entra no modo de *advertise* para que outros dispositivos possam se conectar com ele.

Exemplo 4.4 - Parte V

```
#Método para resetar os timers (temporizadores)
    def connected(self):
        self.timer1.deinit()
        self.timer2.deinit()

    #Método que inicializa os temporizadores para indicação
luminosa dos LEDs
    def disconnected(self):
        self.timer1.init(period=1000, mode=Timer.PERIODIC,
callback=lambda t: self.led(1))
        sleep_ms(200)
        self.timer2.init(period=1000, mode=Timer.PERIODIC,
callback=lambda t: self.led(0))

    #Método de interrupção para verificação dos estágios do
BLE
    def ble_irq(self, event, data):
        # Dispositivo conectado
        if event == 1:
            self.connected()
            self.led(1)

        # Dispositivo desconectado
        elif event == 2:
            '''Central disconnected'''
            self.advertiser()
            self.disconnected()

        # Nova mensagem recebida
        elif event == 3:
            buffer = self.ble.gatts_read(self.rx)
            message = buffer.decode('UTF-8').strip()
            print(message)
            if message == 'red_led':
                red_led.value(not red_led.value())
                print('red_led', red_led.value())
                ble.send('red_led' + str(red_led.value()))
```

1.6 LoRaWan

O LoRaWAN (Long Range Wide Area Network) é um protocolo de comunicação de baixo consumo de energia, projetado para redes de área ampla e longo alcance, que suporta comunicação de dispositivos de Internet das Coisas (IoT) em ambientes de baixa potência, como sensores, medidores e

rastreadores. Este protocolo funciona sobre a tecnologia LoRa (Long Range) que utiliza modulação por espalhamento espectral para transmitir dados a longas distâncias (várias dezenas de quilômetros em áreas rurais) com consumo de energia muito baixo. Vamos entender como o LoRa funciona.

1.6.1 LoRa

LoRa é uma tecnologia de transmissão de dados sem fio desenvolvida e mantida pela Semtech™, projetada para resolver desafios específicos em comunicações sem fio. Ela se destaca por suas capacidades de longo alcance, baixo consumo de energia em sistemas de borda e transmissão segura de dados. Importante diferenciar, LoRa refere-se especificamente à técnica de modulação proprietária da Semtech™ e não deve ser confundida com a descrição geral do sistema de comunicação LPWAN (Low Power Wide Area Network).

A rede LPWAN é frequentemente utilizada em IoT quando há necessidade de enviar poucos dados, em distâncias relativamente longas, garantindo maior vida útil para as baterias a serem implementadas durante os processos de comunicação e aplicação. Essa tecnologia utiliza topologia em estrela e para garantir melhorias de comunicações entre dispositivos e o ponto de acesso podem ser aplicados repetidores de sinais, que têm como papel suprir as necessidades de latência, área de cobertura e confiabilidade.

A tecnologia LoRa opera dentro da rede LoRaWAN, que é um padrão aberto crucial para a definição do protocolo MAC (Medium Access Control) em redes LPWAN. Este aspecto é essencial, pois permite que outras empresas implementem camadas MAC proprietárias sobre o chip LoRa, possibilitando a criação de soluções híbridas e com performances variadas. Assim, enquanto LoRa descreve a técnica de modulação, LoRaWAN se refere ao padrão que regula todo o sistema de comunicação LPWAN.

Esta tecnologia foi desenvolvida pra tentar resolver alguns problemas comuns em redes IoT, como por exemplo, consumo de energia e alcance da rede. Geralmente as aplicações possuem dispositivos sensores que enviam dados periodicamente por um longo período de tempo, o que faz com que haja um gasto considerável de energia. Outro ponto é que, do ponto de vista de integridade dos dados e disponibilidade, os dispositivos devem,

obrigatoriamente, estar localizados ao alcance da rede para que a comunicação seja confiável.

Desta forma, o LoRa resolve esses problemas, fornecendo a possiblidade de conexão entre dispositivos que operem em longas distâncias com um baixo consumo energético. Essas características também são obtidas em detrimento da largura de banda, que se torna bem limitada nesse tipo de tecnologia.

Em termos de segurança, o LoRaWAN oferece várias camadas de segurança para proteger as comunicações, incluindo criptografia de ponta a ponta, autenticação de dispositivos e chaves de sessão.

1.6.2 Topologia de Rede

O LoRa opera em frequências diferentes em diferentes regiões do planeta e utiliza a frequência de uso não restrito ISM (Industrial Scientific Medical). Por exemplo, nas Américas do norte e do sul se utiliza a frequência de 915 MHz; na Europa, 868 MHz; na Índia, 867 MHz; no Pacífico, 434 MHz; e na Ásia, se utiliza tanto a frequência de 434 MHz como a de 867 MHz.

Basicamente, a tecnologia projetada para redes de estrela ou de árvore, onde os dispositivos finais (nós) se comunicam com gateways. Os gateways, por sua vez, encaminham os dados para uma rede central. Com relação a arquitetura, a Figura 4.X mostra a topologia básica de um rede LoRa baseada em camadas.

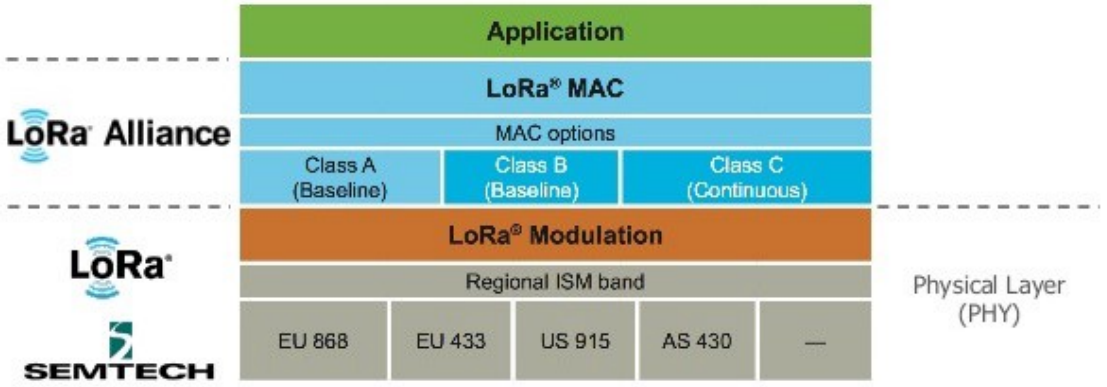


Fig. 4.7 - Topologia básica do LoRa. Fonte: IOP Institute of Physics

1.6.3 Classes de Dispositivos

O LoRaWAN suporta três classes de dispositivos:

Classe A: Dispositivos com comunicação bidirecional, mas com janelas de recepção definidas. Após a transmissão, eles só ouvem as respostas em janelas de tempo específicas.

Classe B: Além da comunicação bidirecional, esses dispositivos têm janelas de recepção programadas e adicionam sincronização ao ciclo.

Classe C: Esses dispositivos estão sempre prontos para receber dados.

1.6.4 Aplicativos IoT Diversos

O LoRaWAN é adequado para uma ampla variedade de aplicações IoT, incluindo monitoramento ambiental, agricultura inteligente, rastreamento de ativos, gerenciamento de estacionamento, monitoramento de qualidade de água, entre outros.

As placas usadas em aplicações IoT possuem chips que fazem com que a configuração para utilização da tecnologia LoRaWan seja simplificada. Na Figura 4.8, por exemplo, temos um módulo RF Wireless LoRa de 433MHz que possui um chip controlador SX1278 que realiza a comunicação com o microcontrolador usando interface serial (UART). Ou seja, uma vez conectado às portas TX e RX, a transmissão do dado acontece da mesma forma como é programado a comunicação Serial padrão.

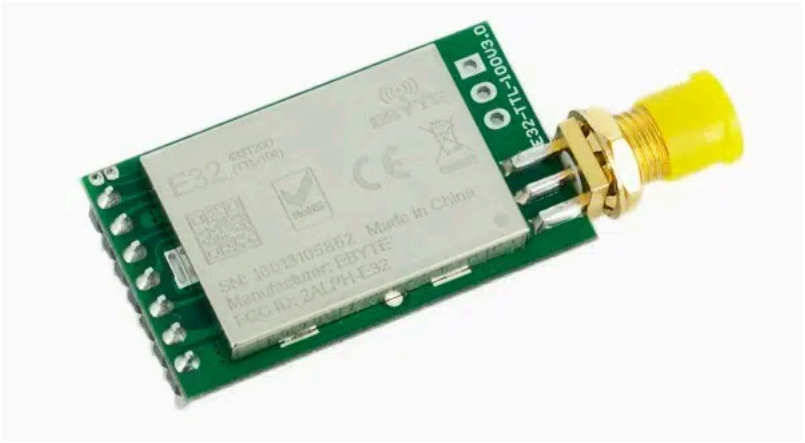


Fig. 4.8 - Módulo RF Wireless LoRa 433MHz. Fonte: MakerHero

Algumas características importantes deste módulo podem ser encotradas no seu *datasheet.* como taxa de transferência de 2.4 Kbps, distância de transissão de 3 km em zona urbana, podendo chegar até 12 km na zona rual com conexão de antena SMA.

Vamos ver um exemplo da utilização desse módulo para fazer um buzzer acionar a partir de um botão pressionado em um outro dispositivo. Dividindo nosso exemplo em duas partes, iremos explicar a parte elétrica e a programação de cada dispositivo.

Começando pelo transmissor, a Figura 4.9 mostra como fica as ligações do Arduino com o módulo e um botão *push-button* com trava, que mantém o botão acionado, sem a necessidade de segura-lo com o dêdo. Para desativar, é só pressiona-lo novamente.

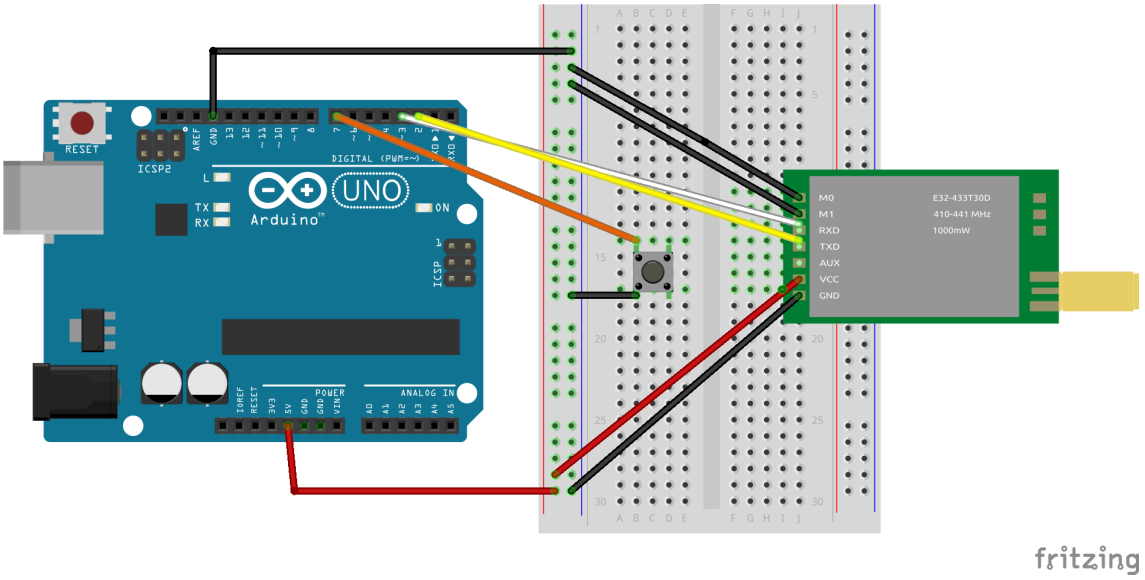


Fig. 4.9 - Protótipo do dispositivo transmissor. Fonte: fritzing

Agora, vamos programar o dispositivo trasmissor.

Exemplo 4.5 - Transmissor (C++)

```
#include <SoftwareSerial.h>

#define BTN1  4

SoftwareSerial loraSerial(2, 3); // TX, RX

String turnOn = "on";
```

```
String turnOff = "off";


void setup() {
 pinMode(BTN1, INPUT_PULLUP);
 Serial.begin(9600);
 Serial.print("Lora-Test");
 loraSerial.begin(9600);

}
void loop() {
Serial.println(digitalRead(BTN1));
 if(digitalRead(BTN1) == 0) {
   loraSerial.print(turnOn);
   while(digitalRead(BTN1) == 0);
   delay(50);
 }else{
   loraSerial.print(turnOff);
   while(digitalRead(BTN1) == 1);
 }
 delay(1000);
}
```

Para o dispositivo receptor, a parte da eletrônica é apresentada na Figura 4.10.

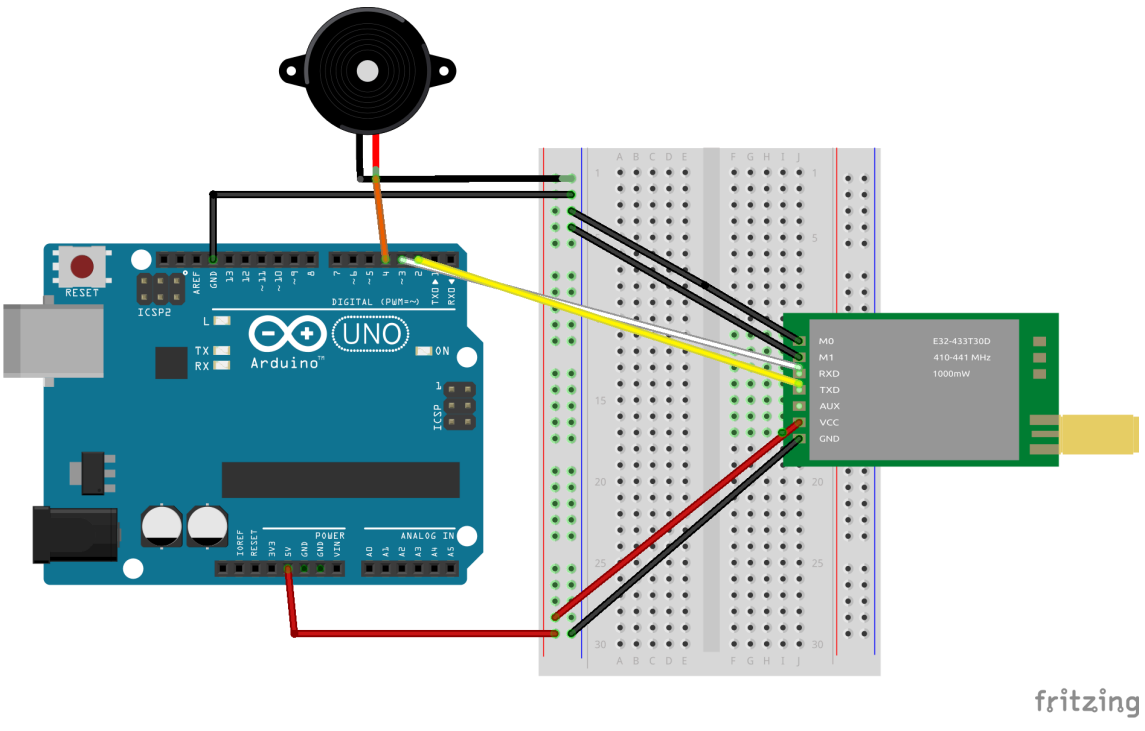


Fig. 4.10 - Protótipo do dispositivo receptor. Fonte: fritzing

Agora, vamos ver como fica o código do disposito receptor, que recebe a informação para acionar ou não o buzzer.

Exemplo 4.5 - Receptor (C++)

```
#include <SoftwareSerial.h>

#define buzzer  4

SoftwareSerial loraSerial(2, 3); // TX, RX

void setup() {
  pinMode(buzzer, OUTPUT);
  Serial.begin(9600);
  loraSerial.begin(9600);
}

void loop() {
  if(loraSerial.available() > 1){
    String input = loraSerial.readString();
    Serial.println(input);
    if(input == "on") {
      digitalWrite(buzzer, HIGH);
    }
    if(input == "off") {
      digitalWrite(buzzer, LOW);
    }
  }
  delay(20);
}
```

Importante! Para o correto funcionamento do módulo é necessário utilizar as antenas SMA. Esse tipo de antena é projetada para trabalhar com uma impedância de 50 ohms, pardão para a maioria das aplicações de RF (Rádio Frequência).

1.7 Redes Móveis (3G, 4G e 5G)

Os protocolos de redes móveis 3G, 4G e 5G são padrões de comunicação sem fio que definem como os dispositivos móveis se conectam às

redes de telefonia celular e à internet. Cada geração representa um avanço tecnológico em termos de velocidade, capacidade, latência e recursos.

Para cada geração, novas funcionalidades, aumento de capacidade e desenvolvimento da tecnologia vem sendo desenvolvida. As primeiras duas gerações (1G e 2G) foram responsáveis por introduzir a comunicação móvel analógica (limitada a chamadas de voz) e transição para comunicação digital, possibilitando SMS e serviços de dados básicos, respectivamente.

Já o 3G foi a terceira geração de tecnologia de telefonia móvel e representou uma grande melhoria em relação às redes 2G anteriores, como o GSM. As redes 3G oferecem velocidades de dados mais rápidas em comparação com o 2G, com taxas de transferência típicas variando de 384 Kbps a 2 Mbps. Além disso, permite comunicação de voz e dados simultânea, o que possibilitou serviços como vídeo chamadas e acesso à internet móvel. O Universal Mobile Telecommunications System (UMTS) é um dos padrões 3G mais amplamente utilizados.

Na quarta geração (4G) houve um grande salto na capacidade de dados e velocidade em comparação com o 3G. Ela permitiu uma experiência de internet móvel mais rápida e confiável. As redes 4G oferecem velocidades de dados muito mais rápidas, com taxas de transferência típicas variando de 10 Mbps a vários Gbps. Além disso, o 4G reduziu significativamente a latência, o que tornou a comunicação mais responsiva. O Long-Term Evolution (LTE) é um dos padrões 4G mais amplamente utilizados, mas existem várias implementações e variações regionais.

Agora estamos vivenciando o desenvolvimento da tecnologia 5G que promete revolucionar a forma como nos comunicamos e consumindo dados. Por isso, iremos focar nossos estudos nesta tecnologia, tentando compreender o seu funcionamento, características, arquitetura e principais aplicações.

1.7.1 Quinta Geração (5G)

O 5G é a mais recente geração de redes móveis, projetada para fornecer velocidades ainda mais rápidas, menor latência e suportar uma variedade de aplicativos e dispositivos inteligentes. Além disso, oferece velocidades que variam de centenas de Mbps a vários Gbps, tornando-se

substancialmente mais rápido do que o 4G, tendo como objetivo uma latência muito baixa, permitindo assim aplicações críticas em tempo real, como carros autônomos e telesaúde.

Além disso, foi projetada para suportar um grande número de dispositivos conectados simultaneamente, o que é essencial para a Internet das Coisas (IoT). Uma outra característica importante desta tecnologia é que ele utiliza uma variedade de frequências, incluindo bandas de ondas milimétricas (mmWave) para alcançar altas velocidades e frequências mais baixas para maior alcance.

A tecnologia 5G possui as seguintes características:

- Conexões de 1-10 Gbps para ponto fim da rede, ou seja, a velocidade efetiva para o usuário e não velocidade máxima teórica;
- Latência de 1ms em rotas fim-a-fim;
- Incremento de 1000x na largura de banda por unidade de área;
- Incremento de 10-100x no número de usuários conectados;
- 99,999% de disponibilidade da rede;
- 100% de cobertura geográfica;
- 90% de redução no uso de energia de rede;
- 10 anos de vida útil da bateria para dispositivos de baixo uso energético.

A tecnologia 5G distingue-se significativamente das anteriores, sendo concebida para atender às exigências de um mundo novo e interconectado, onde a comunicação via internet sem fio será uma realidade constante para todos. Esta visão requer o cumprimento desses requisitos específicos, cruciais para possibilitar a interconexão simultânea de bilhões de dispositivos.

Algumas aplicações dependem de uma estrutura de rede como essas especificadas pela tecnologia 5G para funcionar de forma satisfatória. Entre essas, podemos destacar as seguintes soluções.

Direção autônoma e Veículos Conectados - A comunicação de veículos com o ambiente externo pode levar a um uso mais eficiente e seguro das infraestruturas rodoviárias existentes. Na teoria, se veículos estiverem conectados a uma rede comum que inclua um sistema de gerenciamento de tráfego, eles poderiam se deslocar em velocidades significativamente mais altas e manter uma proximidade muito menor entre si, minimizando quase

completamente o risco de acidentes. Este cenário se torna especialmente promissor no contexto de veículos autônomos.

Ambientes colaborativos complexos - A baixa latência do 5G permite interações em tempo real sem atrasos perceptíveis. Isso é crucial em ambientes colaborativos onde equipes podem estar trabalhando juntas de locais remotos, necessitando de comunicações instantâneas e confiáveis. Projetar grandes construções ou arquitetar ambientes, como softwares de modelagem 3D, podem se tornar aplicações cada vez mais comuns, até dentro do contexto do Metaverso.

Colaboração Humano-Máquina - Em ambientes industriais ou de pesquisa, o 5G pode permitir a colaboração entre robôs e humanos, com controle e monitoramento em tempo real, melhorando a segurança e a eficiência. Esse tipo de aplicação permitira uma melhor estrutura das linhas de produção, muito presente no emprego de tecnologias na indústria 4.0.

Telemedicina avançada - No setor de saúde, o 5G pode permitir colaborações mais eficientes entre especialistas localizados em diferentes geografias, possibilitando procedimentos como cirurgias remotas e consultas em tempo real com especialistas.

Em um contexto prático, a utilização de redes móveis dependem de módulos que permitam a conexão por meio de um chip. A Figura 4.11 mostra o módulo SIM800L, muito usado em projetos com ESP ou Arduino, para conexão GPRS (2G). É importante lembrar que para sua utilização é necessário ter um chip de um operadora com um plano de dados ativo.

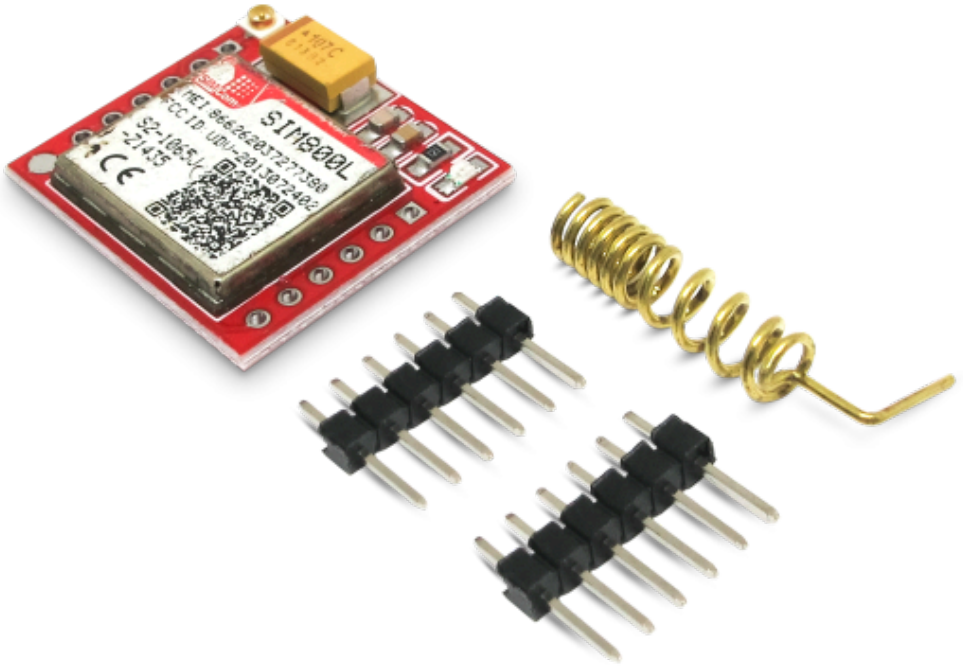


Fig. 4.11 - Módulo GSM GPRS SIM80L. Fonte: Robocore

1.8 Sigfox

A rede SigFox é uma inovação notável no cenário das comunicações sem fio, especialmente no contexto da Internet das Coisas (IoT). Seus principais diferenciais residem na combinação de alcance global, eficiência energética e protocolos avançados.

Essa rede se destaca por ser uma rede LPWAN (*Low Power Wide Area Network*) com alcance global, permitindo a conectividade de dispositivos IoT remotos em todo o mundo. Sua eficiência energética é notável devido à utilização de uma banda ISM livre de licenças, operando com baixa taxa de bits (100-600 bps) e mensagens curtas (12 bytes). Essa abordagem otimizada resulta em um **baixo consumo de energia**, tornando-a uma escolha ideal para dispositivos alimentados por baterias.

Apesar de ser definida como de “alcance global”, essa tecnologia ainda está em expansão e não está presente em muitos lugares. Para se ter uma idéia, no site oficial da tecnologia (sigfox.com) podemos encontrar em que lugares do mundo a sigfox já pode ser utilizada. A Figura 4.12 mostra os pontos (em azul) onde já é possível encontrar essa tecnologia no Brasil.

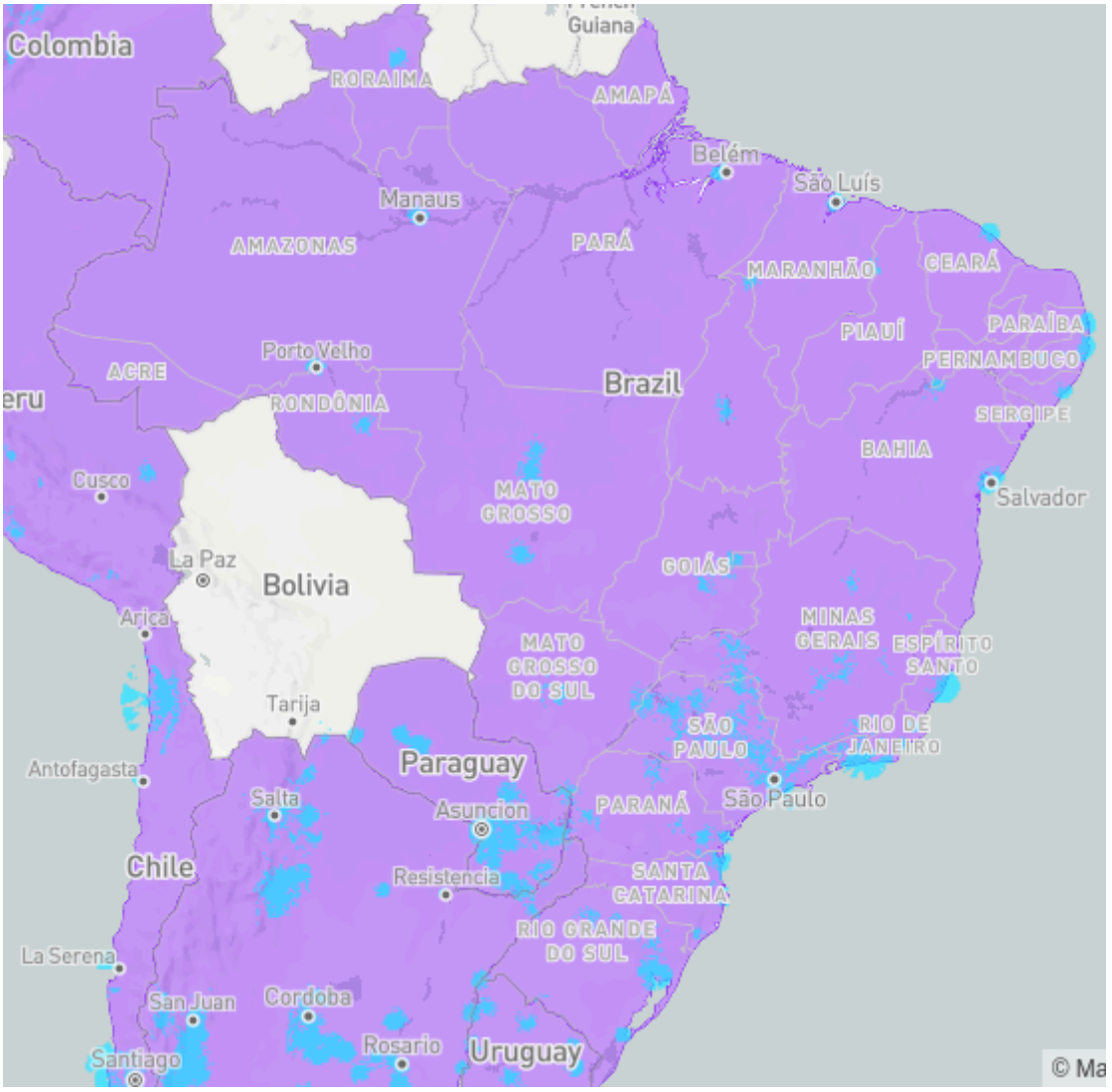


Fig. 4.12 - Mapa de cobertura da rede Sigfox no Brasil. Fonte: sigfox.com/couverage/

Além disso, assim como as redes móveis que vimos anteriormente, é necessário contratar um pacote para sua utilização. No Brasil, a operadora Sigfox é a empresa WND.

O protocolo SigFox é minimalista, projetado especificamente para dados em rajadas, proporcionando eficiência na transmissão de informações. Além disso, a compatibilidade com tecnologias estabelecidas, como Bluetooth, GPS, 2G/3G/4G e Wi-Fi, amplia a versatilidade da rede, permitindo a integração suave com uma variedade de dispositivos.

1.8.1 Arquitetura e funcionamento

A rede SigFox adota uma arquitetura em estrela, composta por dispositivos SigFox Ready, estações base e a plataforma SigFox Cloud. Cada mensagem é transmitida três vezes em frequências diferentes, garantindo diversidade de tempo, espaço e frequência. As bases stations desempenham um papel crucial na mediação entre dispositivos e a nuvem, responsável pela gestão da rede e comunicação. O limite de mensagens por dia, tanto para uplink quanto para downlink, visa otimizar o uso da rede. A Figura 4.13 abaixo traz a exemplificação dessa arquitetura.

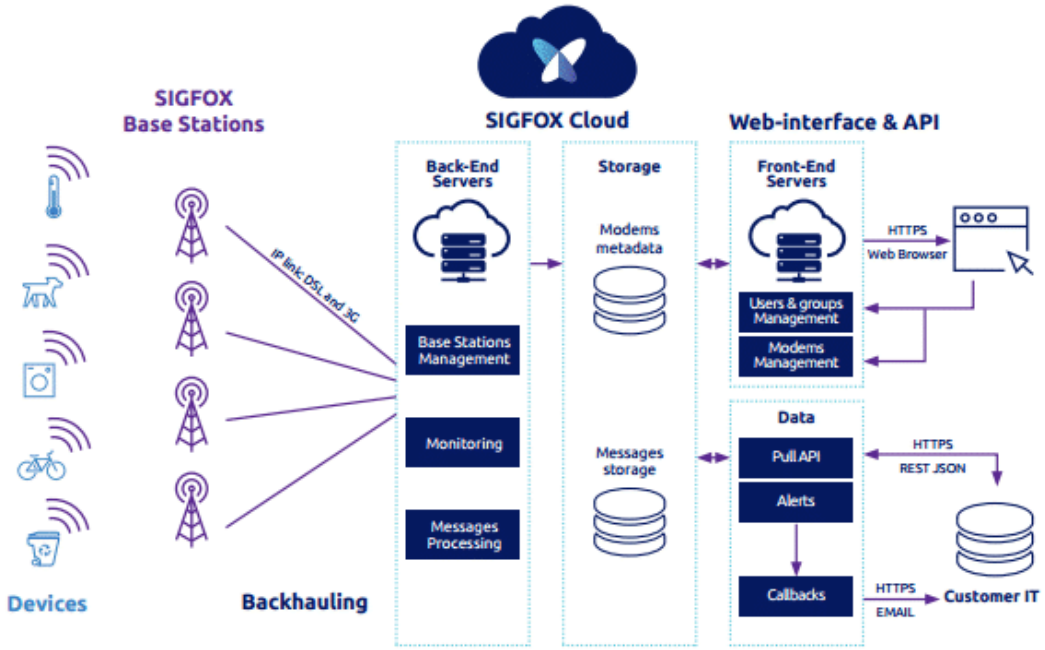


Figura 4.13 – Arquitetura plana (https://embarcados.com.br/uma-visao-tecnica-da-rede-sigfox/)

Um ponto importante é que a rede Sigfox foi desenvolvida para o envio de informações dos dispositivos IoT até a nuvem, de forma que esses dados possam ser acessados. Desta forma, não é possível enviar informações para um dispostivo da rede.

1.8.2 Segurança

A segurança é uma prioridade na rede SigFox, evidenciada pela autenticação *end-to-end* com chave secreta e ID do dispositivo. A sequência numérica e a utilização de múltiplas frequências dificultam interceptações e falsificações, garantindo a integridade das comunicações. A banda ultra estreita

oferece alta resistência a interferências, enquanto a comunicação é iniciada pelo dispositivo, mantendo a rede em uma configuração privada.

1.8.3 Aplicações

Dado as características da rede SigFox, as aplicações envolvem, principalmente, soluções que necessitem de uma longa área de alcance.

- **Monitoramento de Ativos -** Utilizado para rastrear a localização e o status de ativos valiosos em logística e cadeia de suprimentos, como contêineres, veículos e equipamentos.
- **Medição Inteligente -** Aplicado em medidores de utilidades (água, eletricidade, gás) para monitoramento remoto do consumo, facilitando a leitura automatizada de medidores e a detecção de anomalias.
- **Agricultura Inteligente** - Empregado para monitorar condições agrícolas, como umidade do solo, temperatura e níveis de nutrientes, ajudando os agricultores a otimizar o uso de recursos.
- **Gestão de Resíduos -** Usado em sistemas de gestão de resíduos para monitorar o nível de enchimento de contêineres de lixo, otimizando as rotas de coleta.
- **Monitoramento Ambiental** - Empregado para rastrear condições ambientais, como qualidade do ar e níveis de poluição, contribuindo para iniciativas de cidades inteligentes.
- **Monitoramento de Infraestrutura -** Usado para monitorar a condição de infraestruturas críticas, como pontes e oleodutos, para detecção precoce de danos ou desgaste.

1.9 NFC

NFC (Near Field Communication) é um padrão de comunicação sem fio de curtíssimo alcance, desenvolvido para facilitar a comunicação simples e intuitiva entre dois equipamentos eletrônicos. Esta tecnologia relativamente nova foi desenvolvida no início dos anos 2000, visando eliminar a necessidade de cabos ou contato físico entre dispositivos.

Funcionando por meio da tecnologia RFID (Radio Frequency Identification), o NFC permite a comunicação entre dois dispositivos por meio de ondas de rádio, com uma distância de comunicação de até 4 centímetros.

1.9.1 Funcionamento do NFC

O funcionamento do NFC (Near Field Communication) é caracterizado por sua simplicidade, ocorrendo em três etapas principais: Aproximação, Autenticação e Comunicação.

Aproximação

Dois dispositivos equipados com NFC precisam estar próximos um do outro, a uma distância de até 4 centímetros. Neste momento, inicia-se a comunicação por meio da tecnologia RFID (Radio Frequency Identification). Os dispositivos possuem um chip RFID que gera ondas de rádio, e suas antenas são capazes de receber essas ondas de outros dispositivos.

Autenticação

Após a aproximação, os dispositivos iniciam a troca de sinais de rádio para se identificarem e estabelecerem uma conexão segura. Em seguida, ocorre a etapa de autenticação, onde os dispositivos verificam suas identidades mutuamente. Essa verificação é realizada por meio de criptografia, um processo que transforma os dados em um formato ilegível para pessoas não autorizadas, garantindo a segurança do processo.

Comunicação

Uma vez estabelecida a conexão segura e autenticada, os dispositivos podem trocar dados. Esses dados podem ser de diversos tipos, como texto, números ou imagens. A comunicação efetiva entre os dispositivos é possibilitada pelo sucesso das etapas anteriores.

Os dispositivos desta tecnologia são caracterizados como ativos, passivos ou ativos/passivos (Figura 4.14). Essa característica define qual dispositivo inicia o processo de comunicação ou responde a ele. As características de cada tipo de dispositivos são explicados a seguir.

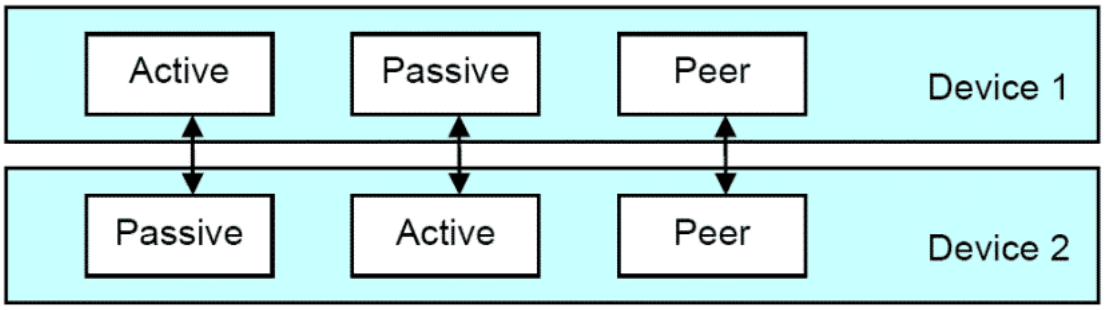


Figura 4.14 – Dispositivos NFC. Fonte:https://embarcados.com.br/nfc-near-field-communication/

Dispositivos Ativos

Um dispositivo ativo possui a capacidade de iniciar a comunicação NFC. Exemplos comuns incluem smartphones, tablets, maquininhas de cartão de crédito/débito ou outros dispositivos eletrônicos que têm a função NFC incorporada. Esses dispositivos podem enviar comandos, solicitar informações e iniciar interações com outros dispositivos NFC, sejam eles ativos ou passivos.

O dispositivo ativo é responsável por energizar o dispositivo passivo para que seja possível o envio, por meio de ondas de rádio, das informações gravadas na etiqueta NFC. O cartão de crédito que possui a tecnologia NFC, por exemplo, consegue efetuar transações a partir da função NFC presente na maquininha (responsável por energizar a etiqueta do cartão, que por sua vez envia as informações por onda de rádio)

Em contextos específicos, como no modo ponto a ponto, ambos os dispositivos envolvidos podem ser ativos, trocando papéis de iniciador e responder durante a comunicação.

Dispositivos passivos

Dispositivos passivos, por outro lado, não iniciam a comunicação por conta própria. Eles respondem aos comandos e solicitações recebidos de dispositivos ativos. Exemplos de dispositivos passivos incluem etiquetas NFC, cartões de crédito/débito ou outros objetos que incorporam a tecnologia NFC sem a capacidade de iniciar interações.

As etiquetas NFC são frequentemente usadas no modo de escrita e leitura (falaremos do modo de operação a seguir), onde um dispositivo ativo, como um smartphone, lê ou grava dados na etiqueta passiva.

1.9.2 Contexto de Operações

O contexto de operação do NFC (Near Field Communication) é essencial para compreender como os dispositivos interagem entre si, sendo definido por três modos distintos: escrita e leitura, emulação de cartão e ponto a ponto.

Escrita e leitura

No modo de escrita e leitura, um dispositivo NFC tem a capacidade de ler ou gravar dados em uma etiqueta NFC. Essas etiquetas podem conter diversas informações, como texto, números, imagens ou links. Este modo é fundamental para interações simples de leitura e gravação entre dispositivos NFC e etiquetas.

Emulação de cartão

O modo de emulação de cartão permite que um dispositivo NFC se comporte como um cartão inteligente. Isso implica que o dispositivo pode autenticar usuários, acessar sistemas ou facilitar pagamentos sem contato. Essa funcionalidade torna o NFC aplicável em uma variedade de cenários, desde autenticação em sistemas de segurança até pagamentos eficientes com cartão de crédito/débito.

Ponto a ponto

No modo ponto a ponto, dois dispositivos NFC podem trocar dados diretamente entre si. Quando utilizamos aplicativos como o *Apple Pay* – uma espécie de carteira dos celulares IOS onde podemos cadastrar nossos cartões de crédito/débito – para pagar uma conta, utiliza-se o contexto de ponto a ponto. Tanto a maquininha quanto o celular podem trocar dados diretamente entre si, pois são dispositivos ativos/passivos. Essa capacidade de comunicação direta é valiosa para compartilhar arquivos, sincronizar dados ou configurar dispositivos de maneira eficaz. O NFC, nesse contexto, oferece uma solução prática para comunicação entre dispositivos próximos.

1.9.3 Etiquetas NFC

As etiquetas NFC são peças de hardware que incorporam a tecnologia NFC. Elas desempenham um papel crucial ao adicionar funcionalidades NFC a produtos e equipamentos. Com diversos formatos e tamanhos disponíveis, as etiquetas podem ser personalizadas para armazenar uma ampla gama de informações, tornando-as versáteis em várias aplicações.

A figura x apresenta uma etiqueta NFC que pode ser comparada a um adesivo. Ela é composta pelo DIE (pastilha de silício onde o circuito do NFC é construído) e trilhas metálicas que formam um circuito junto com uma espécie de antena. São essas etiquetas que estão presentes nos cartões de crédito, por exemplo.

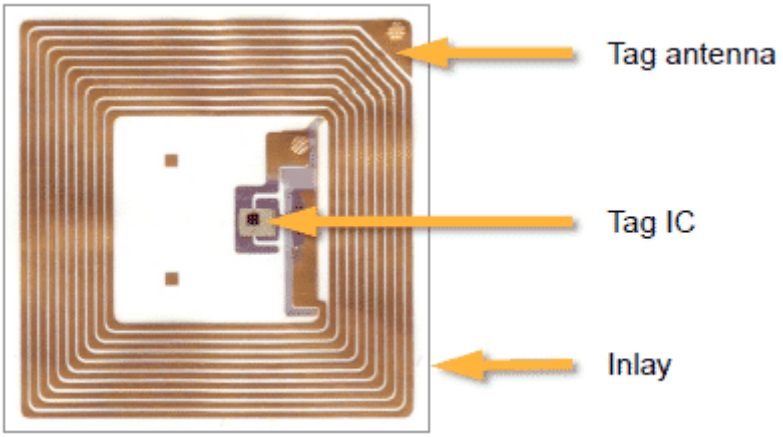


Figura 4.15 – Etiqueta NFC no formato INLAY Fonte: https://embarcados.com.br/nfc-near-field-communication/

1.9.4 Segurança

A segurança é uma prioridade no NFC. A tecnologia utiliza criptografia para proteger os dados transmitidos, transformando as informações em um formato ilegível para pessoas não autorizadas. Além da criptografia, são aplicadas medidas como autenticação e verificação de integridade para assegurar que os dados não foram alterados durante a transmissão.

1.9.5 Aplicações

NFC em Pagamentos: Uma das aplicações mais notáveis do NFC é em pagamentos sem contato. Cartões de crédito e débito com NFC permitem que

usuários efetuem pagamentos simplesmente aproximando o cartão do terminal de pagamento.

NFC em Identificação: A tecnologia pode ser utilizada para identificar pessoas, objetos ou equipamentos. Por exemplo, em ambientes corporativos, pode-se empregar o NFC para identificar funcionários, controlar o acesso a edifícios ou rastrear produtos.

NFC em Autenticação: O NFC é empregado para autenticar usuários ou dispositivos. Pode ser utilizado para autenticar usuários em computadores ou desbloquear smartphones, proporcionando uma camada adicional de segurança.

NFC em Compartilhamento de Dados: Além disso, o NFC possibilita o compartilhamento de dados entre dois dispositivos, como fotos, vídeos ou contatos.

1.10 ZigBee

O ZigBee é um protocolo de comunicação sem fio amplamente utilizado para dispositivos IoT, com um foco particular em eficiência energética e baixa potência. Operando na frequência de 2,4 GHz, compartilhada com o Wi-Fi, mas com potência reduzida, o ZigBee é especialmente projetado para dispositivos que têm uma demanda energética limitada, proporcionando autonomia que pode se estender por anos com uma única bateria.

1.10.1 Funcionamento do ZigBee

O ZigBee destaca-se por sua aplicação em ambientes que necessitam de baixa potência e com um curto alcance, sendo especialmente projetado para automação residencial e industrial. A seguir vamos explorar suas principais características de funcionamento.

Operando na frequência de 2,4 GHz, a mesma utilizada por Wi-Fi e Bluetooth, o Zigbee utiliza uma técnica de modulação diferente que resulta em um consumo de energia significativamente inferior. Seu alcance máximo é cerca de 100 metros, tornando-o ideal para aplicações com distâncias curtas.

O ZigBee é fundamentado no padrão IEEE 802.15.4, que define tanto a camada física quanto a camada de enlace de dados do protocolo. A camada

física é responsável pela transmissão de dados pelo ar, enquanto a camada de enlace encapsula e transmite os dados entres os nós da rede.

A pilha do protocolo ZigBee (Figura 4.16) é definida sobre duas camadas: Camada de aplicação (*Application Layer*) e a camada de rede NWK (*Network Layer*).

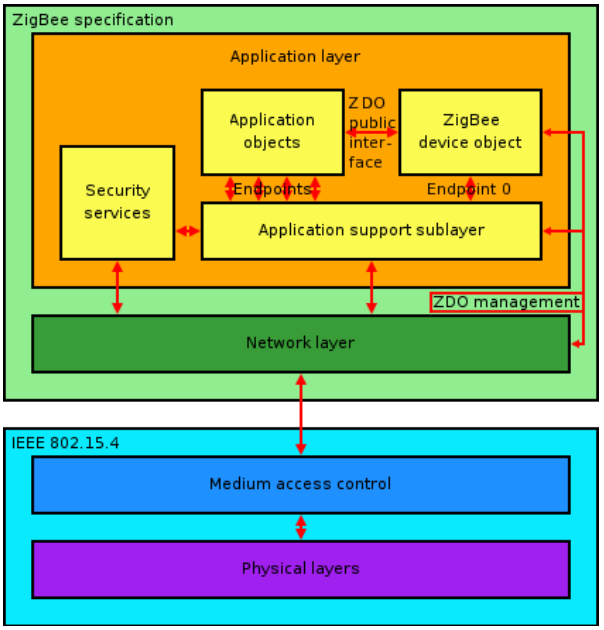


Figura 4.16: Pilha do protocolo ZigBee Fonte: https://www.gta.ufrj.br/ensino/eel879/trabalhos_vf_2017_2/802154/zigbee.html

A partir desta configuração, o ZigBee oferece quatro tipos de serviços:

- **Serviços de Encriptação Adicionais** - A implementação de criptografia AES (Padrão Avançado de Criptografia) de 128 bits é estendida para as chaves de aplicação e de rede, proporcionando uma camada extra de segurança.
- **Associação e Autenticação** - A rede permite a associação apenas de nós válidos, garantindo autenticação. Apenas dispositivos autorizados têm permissão para se juntar à rede, reforçando a segurança do sistema.
- **Protocolo de Roteamento -**: AODV (*Ad Hoc On-Demand Distance Vector*): Para facilitar o roteamento e encaminhamento de dados em qualquer nó da rede, foi implementado o protocolo AODV. Este é um protocolo ad hoc reativo, adaptável a cenários de alta mobilidade. Sua

eficiência visa evitar o desperdício de largura de banda, minimizar o processamento nos nós e manter rotas de pacotes de forma otimizada.

- **Serviços de Aplicação** - Introdução do Conceito "*Cluster*": Um conceito abstrato chamado "*cluster*" é incorporado aos serviços de aplicação. Cada nó pertence a um cluster predefinido, com a capacidade de executar um número específico de ações. Por exemplo, um "conjunto de sistemas de iluminação residencial" pode realizar ações predefinidas, como "acender as luzes" e "desligar as luzes". Essa abstração simplifica a gestão e controle de dispositivos, oferecendo uma organização lógica e eficiente no ambiente da rede.

1.10.1.1 Modo de operação

Inicialmente, é crucial destacar que existem três tipos de componentes (nós) na rede ZigBee. São eles:

- **Coordenador** - dispositivo mestre que governa toda a rede e permite a associação de outros dispositivos.
- **Roteadores** - roteiam as informações dos dispositivos finais.
- **Dispositivo final** - sensores ou outros dispositivos IoT que coletam dados do ambiente

O ZigBee é uma camada projetada para organizar uma rede. Quando um nó (que pode ser um roteador ou um dispositivo final) deseja ingressar na rede, o primeiro passo é solicitar ao coordenador um endereço de rede de 16 bits, como parte do processo de associação. Durante esta etapa, são realizados procedimentos de autenticação e criptografia. A Figura 4.17 a seguir apresenta a estrutura dos nós da rede ZigBee. É perceptível na imagem a presença dos coordenados em todas as topologias possíveis da rede.

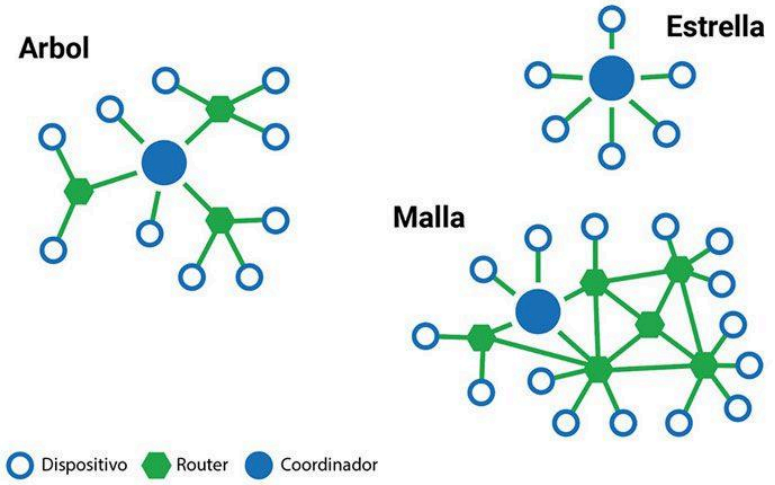


Figura 4.17 - Topologias e configuração da rede ZigBee. Fonte: https://blog.ledbox.es/que-es-el-protocolo-zigbee/

Após a adesão à rede, um nó pode enviar informações para outros nós por meio dos roteadores, que estão sempre prontos para receber pacotes. Quando o roteador recebe um pacote e verifica que o destino está dentro de sua área de cobertura, ele primeiro verifica se o dispositivo final de destino está ativo ou em modo de suspensão. Se estiver ativo, o roteador encaminha o pacote para o dispositivo final. No entanto, se estiver em modo de suspensão, o roteador mantém o pacote protegido até que o dispositivo final acorde e solicite as informações.

O Zigbee oferece flexibilidade por meio de seus dois modos de operação principais: beaconing e non-beaconing. O modo beaconing é destinado a redes com alta mobilidade, onde os roteadores enviam mensagens beacon periodicamente, confirmando sua presença e fornecendo informações cruciais sobre o estado da rede. Já o modo non-beaconing é a escolha para redes com baixa mobilidade, permitindo que os nós se comuniquem diretamente, eliminando a necessidade de roteadores.

1.10.2 Segurança

O ZigBee, como protocolo de comunicação sem fio, enfrenta desafios de segurança, sendo suscetível a ataques de espionagem e adulteração. Isso acontece porque não é necessário ter acesso físico ao fio para participar das comunicações.

As limitações de segurança adicionais decorrem da natureza das redes ad hoc, que envolvem dispositivos de baixo custo com recursos limitados em termos de energia, computação e armazenamento. Além disso, essas redes não podem confiar em uma infraestrutura fixa e frequentemente envolvem relações de curto prazo entre dispositivos que nunca antes se comunicaram.

O ZigBee utiliza criptografia de chave simétrica, onde as chaves são fornecidas por processos de camada superior. O estabelecimento e a manutenção dessas chaves não são tratados pelo ZigBee.

No aspecto criptográfico, o ZigBee oferece serviços específicos, como confidencialidade dos dados, autenticidade dos dados e proteção contra repetição. Isso significa garantir que a informação transmitida seja apenas divulgada às partes destinatárias, verificar a fonte das informações transmitidas e evitar repetições indesejadas.

A proteção do quadro real, adaptada quadro a quadro, oferece diferentes níveis de autenticidade de dados e confidencialidade opcional, ajustando-se às necessidades específicas de segurança.

1.10.3 Aplicações

O ZigBee encontra utilidade em diversas áreas, facilitando a vida cotidiana de maneiras simples e eficientes:

Automação Residencial: Imagine poder controlar dispositivos domésticos de qualquer lugar. Com o ZigBee, isso é possível. Você pode gerenciar remotamente lâmpadas, termostatos, fechaduras e alarmes, tornando sua casa mais inteligente e conveniente.

Automação Industrial: No cenário industrial, o ZigBee entra em ação para automatizar processos e coletar dados de equipamentos. Isso não apenas otimiza as operações, mas também melhora a eficiência na produção.

Monitoramento de Redes Elétricas: A gestão de redes elétricas inteligentes torna-se descomplicada com o ZigBee. Detectar falhas e controlar remotamente a temperatura se torna possível, contribuindo para uma distribuição de energia mais eficiente e confiável.

Exercícios de Fixação

1. Explique o que é o modelo OSI e como ele está estruturado.
2. Explique os seguintes conceitos
    a. IP
    b. Máscara de Rede
    c. Gateway
    d. Porta
    e. MAC
3. O que são redes LAN, PAN, WAN e LPWAN?
4. Crie um quadro comparativo relacionados as tecnologias estudadas, apresentando as principais diferenças e aplicações.
5. Cite possíveis aplicações para diferentes tecnologias de rede.

Referências

A. G. D. S. Junior, L. M. G. Gonçalves, G. A. De Paula Caurin, G. T. B. Tamanaka, A. C. Hernandes and R. V. Aroca, "**BIPES: Block Based Integrated Platform for Embedded Systems**," in IEEE Access, vol. 8, pp. 197955-197968, 2020, doi: 10.1109/ACCESS.2020.3035083.
Full text: https://ieeexplore.ieee.org/document/9246562

SANTOS, Bruno P.; SILVA, Luiz A. M.; CELES, Carla S. F. S.; BORGES NETO, João B.; PERES, Bruno S.; VIEIRA, Marcelo A. M.; VIEIRA, Luiz F. M.; GOUSSEVSKAIA, Olga N.; LOUREIRO, Antônio A. F. (2016). **Internet das Coisas: da Teoria à Prática**. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS [SBRC], 31., 30 de maio de 2016, Salvador. Anais [...]. Salvador: UFMG, 2016. p. 1-50.

ASHTON, Kevin. **Entrevista exclusiva com o criador do termo "Internet das Coisas"**. Finep, 2015. Disponível em: <http://finep.gov.br/noticias/todas-noticias/4446-kevin-ashton-entrevista-exclusiva-com-o-criador-do-termo-internet-das-coisas>. Acesso em: 26 de outubro de 2023.

TOTVS. **Aplicações da Internet das Coisas**. Blog Totvs, 2023. Disponível em: <https://www.totvs.com/blog/inovacoes/aplicacoes-da-internet-das-coisas/>. Acesso em: 8 de novembro de 2023.

Usemobile. **IoT: 9 exemplos de aplicativos bem-sucedidos**. Usemobile, 2023. Disponível em: <https://usemobile.com.br/iot-9-exemplos-de-aplicativos/>. Acesso em: 8 de novembro de 2023.

Google. **Breaking down language barriers with augmented reality**. YouTube, 2023. Disponível em: <https://www.youtube.com/watch?v=lj0bFX9HXeE>. Acesso em: 8 de novembro de 2023.

ZUP. **A arquitetura da Internet das Coisas**. 30 de agosto de 2023. Disponível em: https://www.zup.com.br/blog/a-arquitetura-da-internet-das-coisas. Acesso em: 28 de novembro de 2023.

AMAZON WEB SERVICES. **O que é MQTT?** 20 de julho de 2023. Disponível em: https://aws.amazon.com/pt/what-is/mqtt/. Acesso em: 28 de novembro de 2023.

GTA/UFRJ. **COAP: Protocolo de comunicação para a Internet das Coisas**. 8 de março de 2019. Disponível em: https://www.gta.ufrj.br/ensino/eel878/redes1-2019-1/vf/coap/. Acesso em: 28 de novembro de 2023.

HIVEMQ. **MQTT vs HTTP: Protocolos para IoT e IIoT**. 21 de junho de 2023. Disponível em: https://www.hivemq.com/article/mqtt-vs-http-protocols-in-iot-iiot/. Acesso em: 28 de novembro de 2023.

EMBARCADOS. **AMQP: Protocolo de comunicação para IoT**. 10 de maio de 2023. Disponível em: https://embarcados.com.br/amqp-protocolo-de-comunicacao-para-iot/. Acesso em: 28 de novembro de 2023.

CONCEIÇÃO JÚNIOR, André Lisboa da. **Redes sem Fio: Protocolo Bluetooth Aplicado em Interconexão entre Dispositivos**. Disponível em: https://www.teleco.com.br/tutoriais/tutorialredespbaid/pagina_5.asp. Acesso em: 8 jan. 2024.

EMBARCADOS. **Protocolos de Rede sem Fio de IoT**. Disponível em: https://embarcados.com.br/protocolos-de-rede-sem-fio-de-iot/ Acesso em: 8 jan. 2024.

ARAÚJO, André Silva de; VASCONSELLOS, Pedro de. **Conclusão - Bluetooth Low Energy (BLE)**. Disponível em: https://www.gta.ufrj.br/ensino/eel879/trabalhos_vf_2012_2/bluetooth/conclusao.htm. Acesso em: 8 jan. 2024.

ELEMENT14 COMMUNITY. **Tech Spotlight: SigFox - A Wide Area Network Protocol for IoT.** Disponível em: https://community.element14.com/learn/learning-center/the-tech-connection/w/documents/3897/tech-spotlight-sigfox----a-wide-area-network-protocol-for-iot. Acesso em: 8 jan. 2024.

EMBARCADOS. **Uma Visão Técnica da Rede Sigfox**. Disponível em: https://embarcados.com.br/uma-visao-tecnica-da-rede-sigfox/. Acesso em: 8 jan. 2024.

GTA UFRJ. **Protocolos de Rede para Redes de Sensores Sem Fio**. Disponível em: https://www.gta.ufrj.br/grad/10_1/rssf/protocolos.html. Acesso em: 8 jan. 2024.

EMBARCADOS. **NFC (Near Field Communication) – Aplicações e uso**. Disponível em: https://embarcados.com.br/nfc-near-field-communication/ Acesso em: 23 jan. 2024.

GTA UFRN. **Protocolo Zigbee.** Disponível em: https://www.gta.ufrj.br/ensino/eel879/trabalhos_vf_2017_2/802154/zigbee.html. Acesso em: 24 jan. 2024.

BOM CURSO!

Instituto de Tecnologia