

Relatório de Telecomunicações - COE353

Comunicação à longa distância utilizando a tecnologia LoRa

João Pedro Brandão e Silva¹, Pedro Maciel Xavier¹

¹Escola Politécnica – Universidade Federal do Rio de Janeiro (UFRJ)
21941-909 – Rio de Janeiro – RJ – Brazil

{jp.brs,pedromxavier}@poli.ufrj.br

Resumo. *Este artigo tem por objetivo reunir as informações necessárias para compreender a comunicação machine-to-machine utilizando o protocolo LoRa; bem como descrever o funcionamento de um protótipo que utiliza o mesmo protocolo para fazer a comunicação entre dois microcontroladores ESP32.*

1. Informações Gerais

LoRa, de acrônimo **Long Range**, é um protocolo de comunicação difundido no meio de Internet das Coisas (IOT) justamente por conta da sua capacidade de transmissão de pequenas quantidades de dados à longa distância com um baixo custo de energia. Isso caracteriza o protocolo como um protocolo de *Low Power Wide Area Network* (LPWAN), que será melhor explicado na próxima seção.

1.1. LPWAN

Wide Area Networks são feitas de maneira muito eficiente utilizando ondas de rádio como por exemplo as estações de rádio que utilizam modulação AM ou FM cuja cobertura abrange às vezes uma cidade inteira. Porém, esse tipo de sinal requer um gasto de energia muito grande, algo que é inviável diante do paradigma de Internet of Things, cujos dispositivos possuem limitações de energia e processamento. Outra coisa que torna esse tipo de transmissões de rádio não muito útil para o paradigma IOT é o fato de que a quantidade de informações transmitidas por estações AM/FM é infinitamente superior a quantidade de dados geralmente enviada por redes de sensores sem fio, cujo payload muitas vezes possui apenas um byte.

Sendo assim, pensando justamente nessas necessidades peculiares de dispositivos IoT foram criados diversos protocolos de comunicação que abrangem uma área relativamente grande e que possuem um consumo energético relativamente baixo, os protocolos para Low Power Wide Area Network, como Zigbee, Sigfox e LoRa. Uma característica desses protocolos que justamente faz com que eles consigam transmitir ondas eletromagnéticas de longa distância com um menor consumo energético é a diminuição da largura da banda, pondo muito menos informações em uma onda do que outros protocolos de rede.

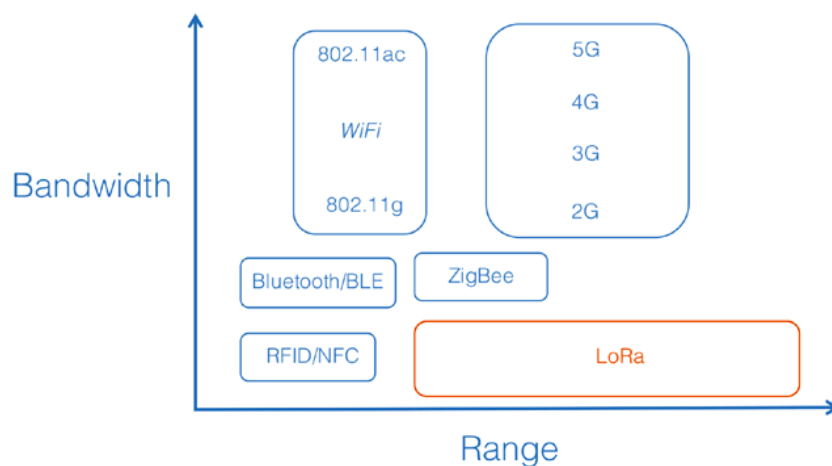


Figure 1. Comparação entre protocolos de rede

2. Radio LoRa

A tecnologia de rádio LoRa é uma técnica de modulação Spread Spectrum derivada da tecnologia Chirp Spread Spectrum(CSS); é a primeira implementação de CSS a baixo custo para uso comercial. O sinal LoRa é marcado por um preâmbulo, que sinaliza aos receptores que a onda eletromagnética recebida é justamente um sinal LoRa. Este preâmbulo pode ser observado nas figuras abaixo como as primeiras linhas perfeitamente paralelas.

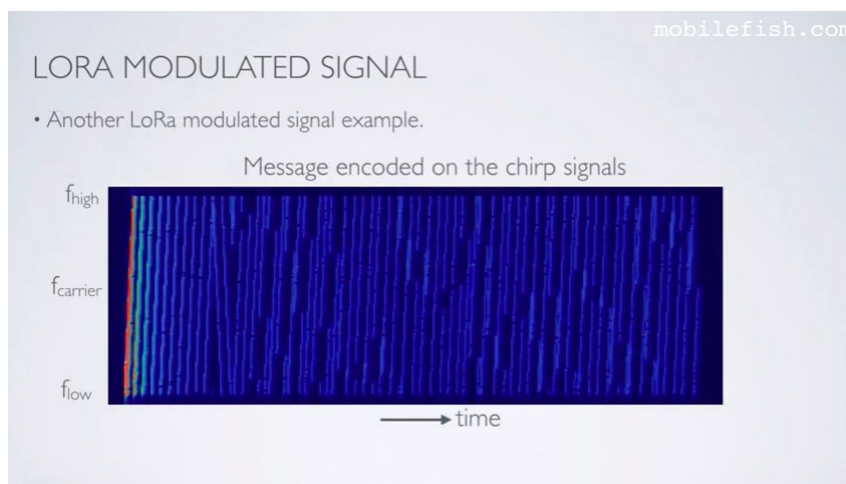


Figure 2. Sinal LoRa. Fonte : mobilefish.com

A frequência padrão inicial das transmissões das ondas LoRa pode ser diferente dependendo do chip a ser utilizado. Por questões legais, em cada país existe uma faixa de frequência destinada à pesquisas e portanto é a frequência utilizada pelo LoRa; no caso do Brasil e dos Estados Unidos essa frequência é de 915MHz e portando o chip que foi utilizado nesse trabalho foi o RFM95HW 915MHz, como mostrado na figura abaixo. O preço médio desse chip em Dezembro de 2019 é de R\$16,00 em sites chineses.

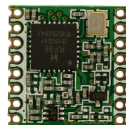


Figure 3. Chip RFM95HW 915MHz utilizado no projeto

2.1. O Pacote LoRa

O pacote LoRa, como já dito, começa com um preâmbulo para ser identificado pelos dispositivos receptores e além disso possui uma mensagem de sincronização que pode variar de 0x00(0) a 0xff(255), isso faz com que o dispositivo transmissor e o receptor tenham uma determinada filtragem dos dados, dando uma determinada garantia de que apenas a informação relevante da aplicação está sendo avaliada pelo receptor. Como existem apenas 255 opções de sync message, em uma área que se possua muitos dispositivos LoRa é recomendado utilizar outro tipo de garantia de filtragem dos dados pela parte da aplicação, visto que a probabilidade de duas aplicações utilizarem a mesma sync message começa a aumentar dado o número de dispositivos.

Além do Sync message o pacote LoRa também possui uma parte destinada obviamente ao payload, cujo tamanho máximo pode atingir 243 bytes, além de um Code Redundance Check(CRC), que serve para tratar eventuais perdas de informação durante a transmissão dos dados pelo ar.

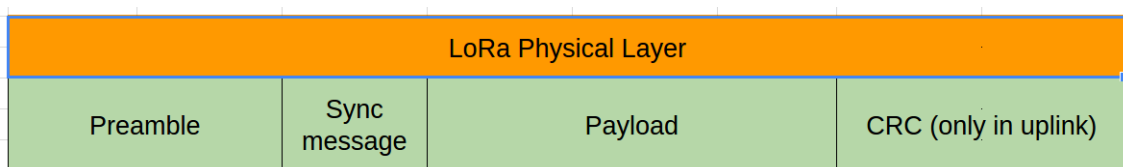


Figure 4. Pacote LoRa

Também é válido dizer que qualquer tipo de criptografia deve ser utilizada na parte da aplicação, a tecnologia Radio LoRa não encripta nenhuma mensagem transmitida e portando caso um receptor possua uma antena e um módulo LoRa com a mesma frequência que o seu transmissor e a mensagem de sincronização seja a mesma dentre 256 possibilidades, o receptor poderá ter acesso aos dados transmitidos sem nenhum problema.

Por esses motivos e para garantir a conectividade e a comunicação segura entre aplicações que utilizam LoRa e a Internet, que a LoRa WAN busca oferecer comunicação utilizando especificades implementadas em camadas de aplicação que utilizam o protocolo LORAWAN, que usa da tecnologia Radio LoRa na camada física.

3. LoRa WAN

LoRaWAN é uma rede conectada a internet que utiliza LoRaWAN gateways para ligar dispositivos LoRa à rede LoRaWAN e à Internet. O funcionamento da rede LoRaWAN pode ser visto no diagrama abaixo.

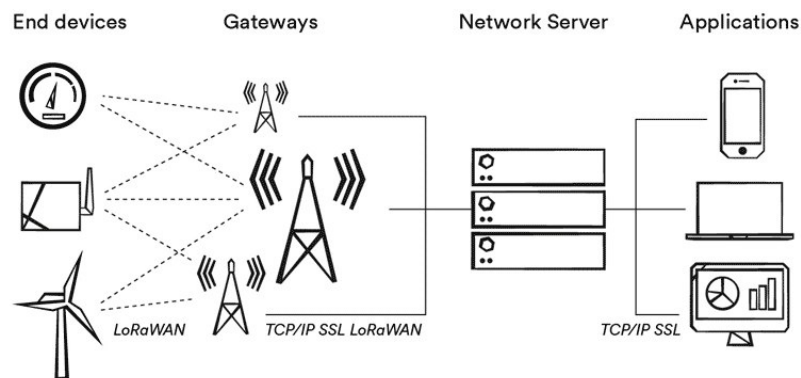


Figure 5. Funcionamento da Rede LoRaWAN

A LoRa Alliance, é a organização sem fins lucrativos que visa garantir a interoperabilidade entre diversos dispositivos que utilizam o protocolo de comunicação LoRaWAN. As empresas da LoRa Alliance espalham gateways LoRaWAN pelo mundo para serem utilizados publicamente. Para se conectar à um desses gateways LoRaWAN é necessário a utilização do protocolo LoRaWAN desenvolvido pela LoRa Alliance e o registro em algum servidor de redistribuição do tráfego da rede LoRaWAN como o The Things Network, que possui milhares de servidores LoRaWAN pelo mundo.

Caso o leitor desejar utilizar a rede LoRaWAN em sua aplicação IoT, porém não houver nenhum gateway da rede próximo o suficiente do local da aplicação, é possível criar seu próprio gateway LoRaWAN utilizando um RaspberryPi, um adaptador LoRa e um software disponibilizado pela The Things Network para redirecionar o tráfego do Gateway para a internet via LoRaWAN.



Figure 6. Empresas colaboradoras da LoRa Alliance

A rede LoRaWAN The Things Network, principal rede LoRaWAN do mundo possui milhares de Gateways espalhados pelo mundo, porém na América Latina ainda há bastante carência desses dispositivos, o que tornou inviável o teste da conectividade LoRaWAN entre o dispositivo feito nesse trabalho e a rede LoRaWAN. Em toda a cidade do Rio de Janeiro há apenas 1 Gateway LoRaWAN, que fica na Zona Sul do Rio de

Janeiro, porém sua disponibilidade não é estável, o que dificultou a completude do trabalho. Porém, como já dito, isso não implica que a tecnologia não seja utilizada, visto que tanto na Europa quanto na América do Norte existem milhares de Gateways.

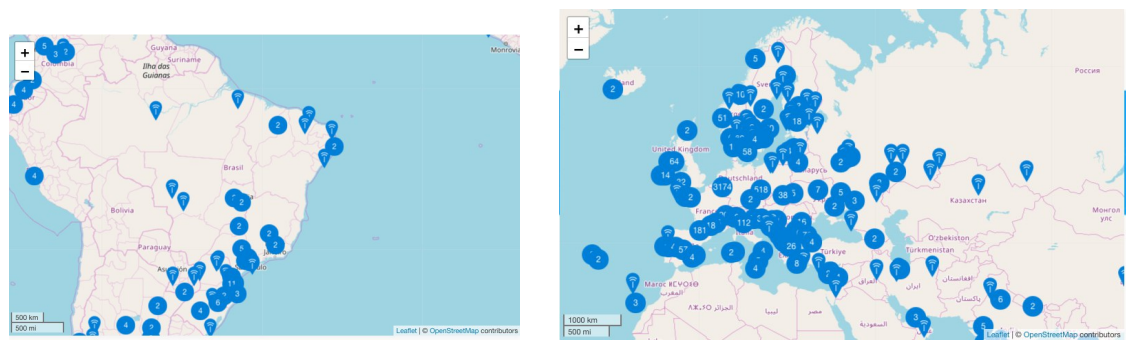


Figure 7. Comparação entre número de Gateways nos continentes

O protocolo LoRaWAN utiliza de toda a estrutura criada pela tecnologia Radio LoRa, além de fornecer uma conexão orientada e criptografada possível através de uma troca de chaves de segurança como AppKey, DevUI e AppUI, trocadas e conferidas antes da transmissão dos dados da aplicação.

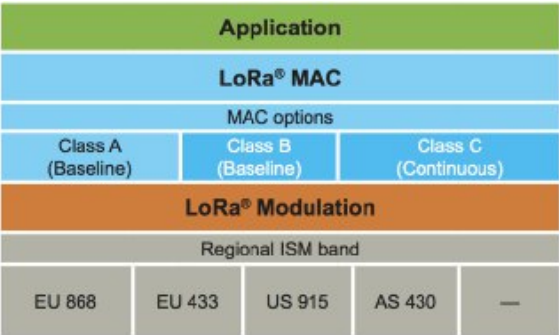


Figure 8. Pilha do pacote LoRaWAN

4. Modulação

4.1. Spread Spectrum

Desenvolvido em tempos de Guerra, o *Spread Spectrum* nada mais é do que a comunicação através de diferentes faixas de banda, escolhidas de maneira alternada, como vemos na Figura 9. A intenção era fazer com que um receptor desavisado encarasse uma possível interceptação do sinal como se fosse mero vislumbre de puro ruído.

No contexto da transmissão LoRa, alternar as faixas de frequência torna o sinal enviado menos suscetível aos efeitos de interferência, principalmente aquela causada pela presença de obstáculos ou pela proximidade com o solo.

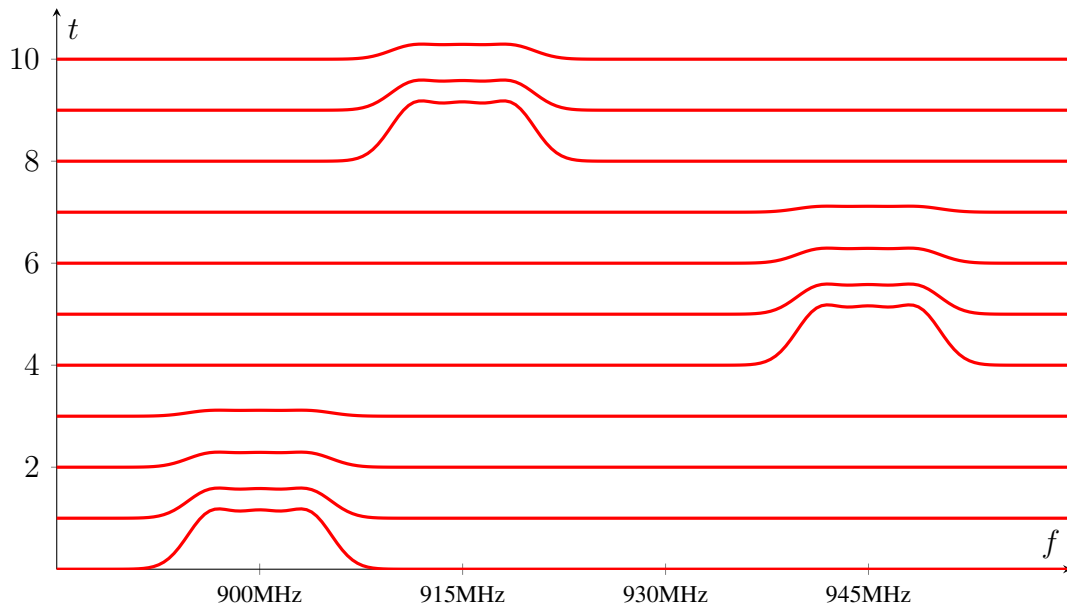


Figure 9. Espectrograma de uma emissão LoRa

4.2. Chirp

O *Chirp* consiste em variar rapidamente a frequência do sinal de forma linear no tempo, seja de forma ascendente (*up-chirp*) ou descendente (*down-chirp*). No *up-chirp*, partimos de uma frequência inicial $f_c - \frac{\omega}{2}$, indo até $f_c + \frac{\omega}{2}$, onde ω é a largura de banda. Isso nos dá a figura serrilhada característica desse protocolo.

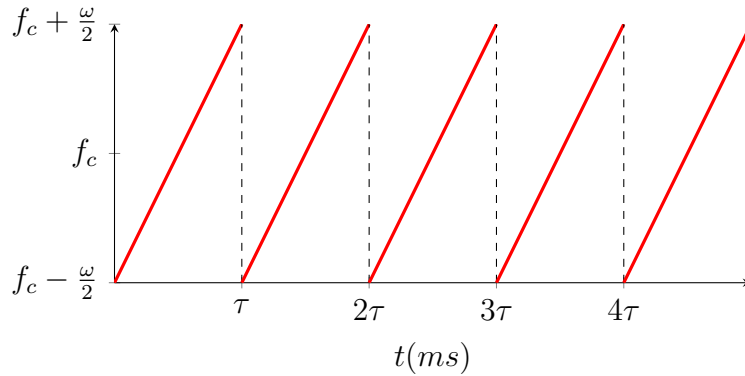


Figure 10. Frequência instantânea, onda em serra

A transmissão de dados se dá através da modulação digital, que pode assumir $M = 2^n$ figuras distintas. Uma figura, segundo a nomenclatura corrente entre as pessoas que utilizam esta tecnologia, diz respeito ao comportamento da frequência em um intervalo de tempo $[k\tau, (k+1)\tau]$, $k \in \mathbb{N}$. A codificação da informação é feita através de "saltos" na frequência, o que caracteriza uma modulação por fase.

Para fins práticos, podemos dizer que somos capazes de transmitir inteiros $a \in \{0, 1, \dots, M-1\}$, que no fim das contas são M sequências binárias de tamanho

n . O inteiro escolhido define a frequência onde se inicia a figura da seguinte forma: A frequência instantânea $f_i(t; a)$ é calculada no intervalo $[0, \tau]$ da seguinte forma:

$$f_i(t; a) = \omega \cdot \left[\frac{a}{M} + \frac{t}{\tau} - u(t - \tau_a) \right] \quad t \in [0, \tau]$$

Onde temos que $u(t)$ é a função degrau

$$u(t) \triangleq \begin{cases} 1, & t \geq 0 \\ 0, & t < 0 \end{cases}$$

$$\text{e } \tau_a = \tau \cdot \left(1 - \frac{a}{M} \right)$$

Iniciando a modulação em $t = 0$, a fase $\phi(t; a), t \in [0, \tau[$ é dada por:

$$\begin{aligned} \phi(t; a) &\triangleq 2\pi \int_0^t f(s; a) ds \\ &= 2\pi\omega \left[\frac{a \cdot t}{M} + \frac{t^2}{2\tau} - (t - \tau_a) \cdot u(t - \tau_a) \right] \end{aligned}$$

Por exemplo, supomos que temos $M = 4$, isto é, $a \in \{0, 1, 2, 3\}$. Codificando a mensagem 3 – 1 – 2 – 0 – 2 nós terminaríamos por criar a seguinte sequência de figuras:

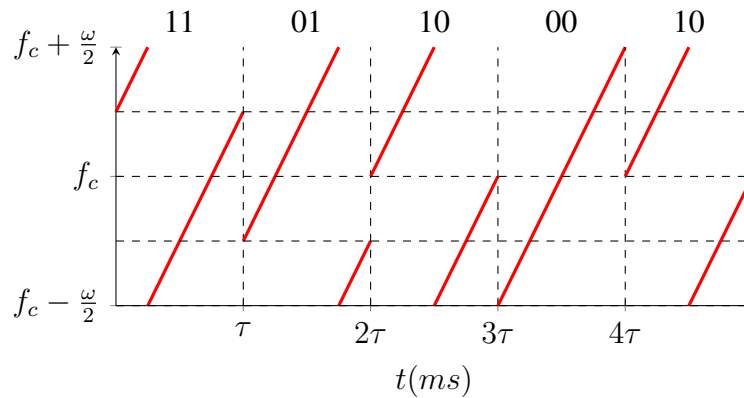


Figure 11. Exemplo: Frequência instantânea para modulação.

4.3. A onda portadora

O sinal da onda portadora (Figura 10) se distribui igualmente no domínio da frequência, numa lagura de banda ω ao redor de f_c .

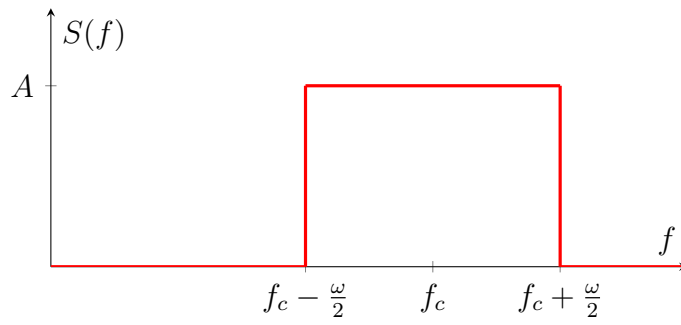


Figure 12. Onda portadora no domínio da frequência.

Para obter a onda portadora no domínio do tempo, calculamos a transformada inversa de Fourier:

$$\begin{aligned}
 \mathcal{F}^{-1}\{S(f)\} &= \int_{-\infty}^{\infty} S(f) \cdot e^{2\pi j f t} df \\
 &= \int_{f_c - \frac{\omega}{2}}^{f_c + \frac{\omega}{2}} A \cdot e^{2\pi j f t} df \\
 &= \frac{A}{2\pi j t} \cdot [e^{2\pi j f t}]_{f_c - \frac{\omega}{2}}^{f_c + \frac{\omega}{2}} \\
 &= \frac{A}{2\pi j t} \cdot [e^{2\pi j (f_c + \frac{\omega}{2}) t} - e^{2\pi j (f_c - \frac{\omega}{2}) t}] \\
 &= \frac{A}{\pi t} \cdot e^{2\pi j f_c t} \cdot \frac{e^{2\pi j \frac{\omega}{2} t} - e^{-2\pi j \frac{\omega}{2} t}}{2j} \\
 S(t) &= \frac{A}{\pi t} \cdot [\cos(2\pi f_c t) + j \sin(2\pi f_c t)] \cdot \sin(2\pi \frac{\omega}{2} t) \\
 &\approx \frac{A}{\pi t} \cdot \cos(2\pi f_c t) \cdot \sin(2\pi \frac{\omega}{2} t)
 \end{aligned}$$

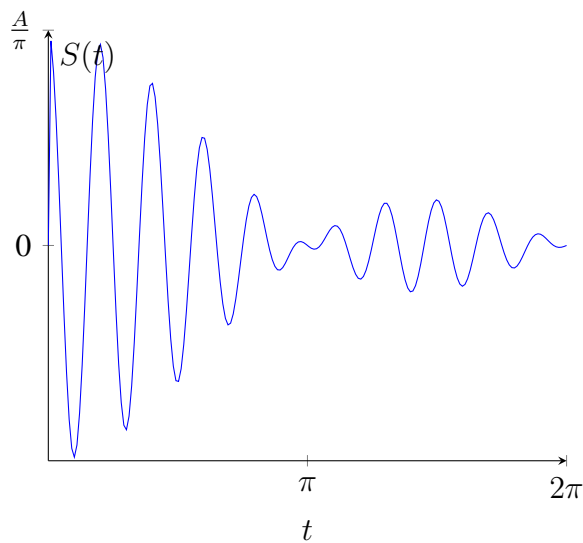


Figure 13. Onda portadora no domínio do tempo

Maiores detalhes sobre a modulação podem ser encontrados em [1]

5. Componentes do projeto

Para estabelecer a comunicação utilizando LoRa foi utilizado em cada ponto da comunicação um microcontrolador ESP32, onde foi possível programar o comportamento da aplicação e controlar o chip LoRa RFM95HW de frequência 915MHz utilizado para fazer a modulação do sinal LoRa. O microcontrolador ESP32 foi programado utilizando linguagem C++ pela IDE Arduino, onde foi possível importar a biblioteca LoRa.h, que serviu para tornar mais fácil de configurar o RFM95HW para transmitir os dados desejados. As conexões entre as GPIO's e o chip RFM95HW podem ser vistas na figura 10, que mapeia as conexões se baseando no manual da biblioteca LoRa.h.

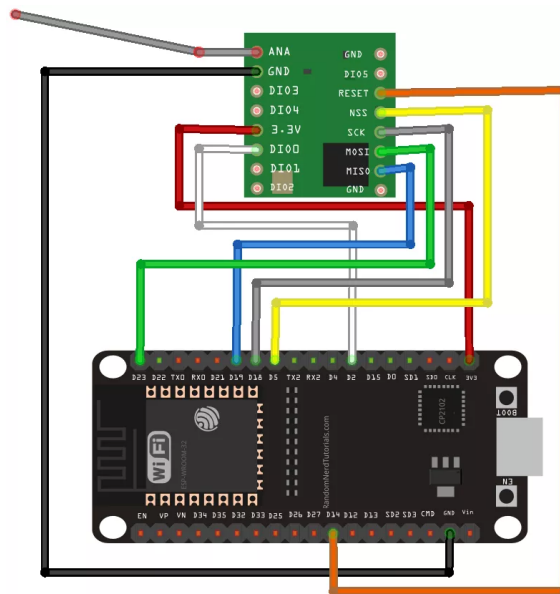


Figure 14. Esquemático das Conexões entre ESP32 e RFM95HW

6. Código

Aqui cada parte do código utilizado tanto no receptor quanto no transmissor será explicada. Os códigos podem ser encontrados em : <https://github.com/jpbrs/UFRJ-Projects/tree/master/COE364%20-%20LoRa>

6.1. Transmissor

```
1 #include<iostream>
2
3 using namespace std;
4
5 int
6 main(int argc, char* argv[]){
7     /* Vem uma galera aqui em casa mais tarde
8        * to puto
9        * amigo dos meus familiares kkkk
```

```

10      * to ouvindo brega igual um corno
11      * KKKKKKKKKKKKK
12      * Estava nessa fase semana passada
13      * mano
14      * tinha uma garota de cabelo curto
15      * no onibus de niteroi que veio falar comigo
16      * ela e mto gente boa mas eu preciso de um tempo pra
17      mim
18      * que merda
19      * te entendo, mano, ja passei por isso. Uma gatinha
20      veio me dar mole na academia nacional de medicina e
21      eu nem tchum. Mas agora ja to bem, ate pegaria ela,
22      mas esta namorando kkkk (triste demais)
23      * kkkkkkkkkkkkkkkkkkk foda parcerro
24      */
25      cout << "Telecom" << endl;
26      return 0;
27  }

```

Listing 1. Código

O código começa por importar as bibliotecas necessárias tanto por sincronizar os componentes do ESP32 com o Chip(biblioteca SPI.h), tanto por facilitar a codificar e controlar o RFM95HW(biblioteca LoRa.h)

```

1 #include <SPI.h>
2 #include <LoRa.h>

```

Listing 2. Bibliotecas

Abaixo definimos os pinos que se conectarão ao chip em cada ponto, atribuindo o nome das variáveis às entradas do RFM95HW e o número da GPIO do ESP32 associada.

```

1 #define ss 5
2 #define rst 14
3 #define dio0 2

```

Listing 3. Mapa de Pinos

Agora definimos as configurações do ESP32, primeiro estabelecemos a frequência da porta serial, para podermos ver por um monitor serial disponibilizado pela Arduino IDE, logo depois utilizamos a biblioteca LoRa.h para associar os pinos do ESP32 às entradas do RFM95HW, com o módulo setPins. Por fim, iniciamos o Chip LoRa utilizando a frequência 915MHz e indicamos que a SyncWord de comunicação será 0xF3, ou 243.

```

1 void setup() {
2     Serial.begin(115200);
3     while (!Serial);

```

```

4   Serial.println("LoRa Sender");
5
6   LoRa.setPins(ss, rst, dio0);
7
8   while (!LoRa.begin(915E6)) {
9       Serial.println(".");
10      delay(500);
11  }
12
13  LoRa.setSyncWord(0xF3);
14  Serial.println("LoRa Initializing OK!");
15 }

```

Listing 4. Configurações

E finalmente chega a parte de envio de dados, onde à cada iteração do Loop é acrescido de 1 o valor enviado da variável counter, que indica no monitor serial do receptor que os pacotes estão chegando em ordem e o código funciona normalmente.

```

1 void loop() {
2     Serial.print("Enviando pacote: ");
3     Serial.println(counter);
4
5     LoRa.beginPacket();
6     LoRa.print("Hello Telecom! Pacote numero : ");
7     LoRa.print(counter);
8     LoRa.endPacket();
9
10    counter++;
11
12    delay(5000);
13 }

```

Listing 5. Envio dos dados

6.2. Receptor

Existem basicamente 2 diferenças do código do receptor para o transmissor. A primeira obviamente é o Loop, enquanto um envia os dados o outros chega por recebimento. A segunda diferença é a configuração extra de um pino GPIO para acender um LED sempre que houver algum dado LoRa recebido.

Então, abaixo pode ser visto o código do loop de recebimento dos dados. Primeiro atribui-se uma variável packetSize ao pacote LoRa que foi recebido pelo chip, caso esse pacote exista, ou seja, possua tamanho maior que zero, um Led é aceso e apagado num intervalo de 200ms, e a mensagem é mostrada no monitor serial do receptor juntamente com a intensidade do sinal recebido.

```

1 void loop() {
2
3   int packetSize = LoRa.parsePacket();
4   if (packetSize) {
5     Serial.print("Pacote recebido ");
6     digitalWrite(4, HIGH);
7     delay(200);
8     digitalWrite(4, LOW);
9
10    while (LoRa.available()) {
11      String LoRaData = LoRa.readString();
12      Serial.print(LoRaData);
13    }
14
15    Serial.print(" with RSSI ");
16    Serial.println(LoRa.packetRssi());
17  }
18 }

```

Listing 6. Recebimento dos dados

7. Aspectos Físicos Considerados para os testes

Aqui descreveremos aspectos físicos que nos fizeram tomar algumas medidas durante a realização dos testes.

7.1. Ganho da Antena

Como possuíamos uma antena helicoidal, seu ganho era direcionado para frente. Isso fez com que apontássemos as antenas dos dispositivos uma para a outra, de forma a maximizar o desempenho da aplicação otimizando a potência do sinal.

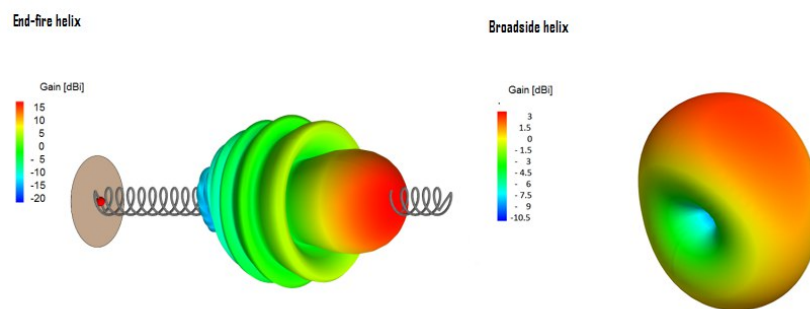


Figure 15. Ganho da antena Helicoidal(à esquerda) e ganho da antena multidirecional(à direita)

7.2. Zona de Fresnel

Outro fator que foi levado em conta no momento dos testes foi a Zona de Fresnel, que é um elipsóide onde tudo que se encontra dentro interfere na propagação do sinal através de interferências ou reflexões.

Para o cálculo do Raio da Zona de Fresnel utilizamos a fórmula abaixo, onde utilizamos D em Km, r em metros e f em GHz.

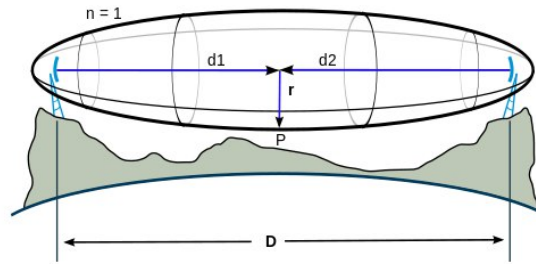


Figure 16. Zona de Fresnel

$$r = 8.657 \sqrt{\frac{D}{f}}$$

8. Testes

Como descrito anteriormente para a otimização dos testes levamos em conta a Zona de Fresnel e o ganho da antena. Os principais testes foram feitos em dois ambientes com os dispositivos finais, cujo custo total foi de R\$37 cada um.

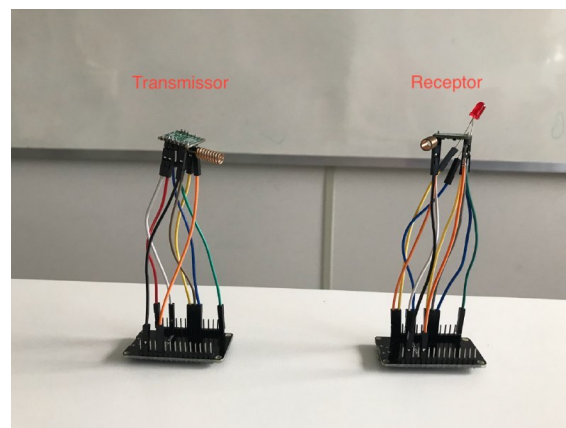


Figure 17. Dispositivos finais

8.1. Teste de Distância Indoor

Para o teste indoor fomos ao local coberto e reto com o maior comprimento da faculdade e fizemos o seguinte teste: Em uma ponta estava o receptor e em outra o transmissor, ambos separados por 377 metros e incerteza de medição de 10m.

No teste 100% dos pacotes transmitidos chegaram ao receptor sem nenhum tipo de mudança no payload, além do mais só não foi possível atingir distâncias maiores pois não havia mais espaço para continuar avançando com teste. Vale ressaltar que a intensidade do sinal final foi de -67 RSSI, visto que em outro teste Outdoor a perda dos pacotes só começou a ocorrer quando o RSSI chegava muito próximo de 100, ainda havia a possibilidade de continuar com o teste indoor por maiores distâncias.

8.2. Teste de Distância Outdoor

O teste de distância Outdoor foi feito posicionando a antena receptora à aproximadamente 30m de altura enquanto a antena transmissora se movimentava para pontos distantes em



Figure 18. Corredor onde foi realizado o teste

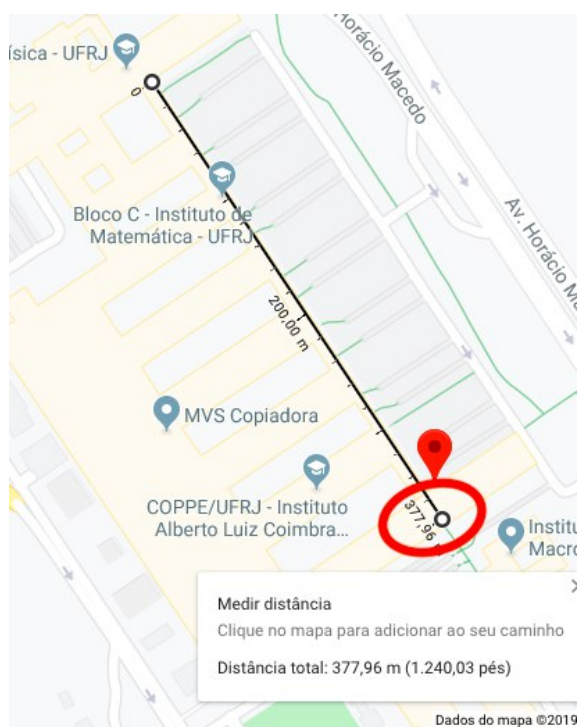


Figure 19. Distância Máxima Indoor

um campo aberto. Ao final o desempenho do teste foi de 650m com incerteza de 10m, que era justamente quando 25% dos pacotes eram recebidos alterados, ou seja com perda de informação, ou simplesmente não chegavam. Vale acrescentar que apesar de um teste outdoor ideal posicionar uma antena em um ponto alto e não haver nenhum obstáculo entre o receptor e o transmissor, isso não foi possível em vista que conforme a distância entre as antenas aumentava menores eram nossas possibilidades de caminhos. Sendo assim, o resultado final de 650m possui como obstáculo dois edifícios de aproximadamente 25m entre as duas antenas.

References

- [1] Marco Chiani & Ahmed Elzanaty, **On the LoRa Modulation for IoT: Waveform Properties and Spectral Analysis**, *IEEE Internet of Things Journal*, 2019.

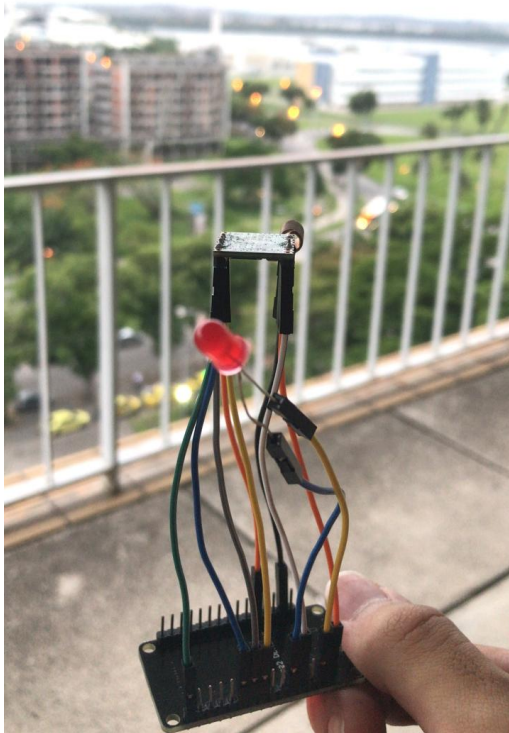


Figure 20. A esquerda podemos ver os prédios entre os dispositivos e à direita a distância entre eles