



Melyssa Mariana

[melyssa.mariana32@gmail.com](mailto:melyssa.mariana32@gmail.com)

11/11/20

# TECNOLOGIAS DE COMUNICAÇÃO IOT:

Da teoria à prática

1

IOT

X

EMBARCADOS

Conexão entre  
objetos físicos  
utilizando sensores,  
chips e softwares.

Sistema microprocessado  
no qual o computador é  
completamente  
encapsulado ou dedicado ao  
dispositivo ou sistema que  
ele controla.

# Modelo OSI x Modelo TCP/IP



Modelo de Referência OSI



Modelo de Referência TCP/IP 12



Conexão e protocolos de transporte de dados



Endereço (IP) /Roteamento



Modem, cabos

# CAPACIDADE DE UM CANAL DE COMUNICAÇÃO

1 dispositivo  
2 canais



1 dispositivo  
1 canal



$$C = B \cdot \log_2 \{1 + [S / (N_o \cdot B)]\}$$

Capacity (maximum throughput), bit-per-second

Bandwidth, Hz

Received signal power, W

Noise power density, W/Hz

# O QUE SÃO PROTOCOLOS?

- x Um protocolo de comunicações é um conjunto de normas que estão obrigadas a cumprir todas as máquinas e programas que intervêm em uma comunicação de dados entre computadores sem os quais, a comunicação seria caótica e, portanto impossível.



Standard	802.11	Bluetooth	IEEE 802.15.1	IEEE 802.15.4	IEEE 802.16	Z-Wave Alliance	GSM GPRS	LoRaWA R1.0	ECMA-340 and ISO/IEC 18092
Tipo de rede:	LAN	PAN	LAN	LAN	MAN	LAN	MAN	LAN	-
Topologia	Estrela	Estrela	Mesh, Estrela, Árvore	Mesh, Estrela	Mesh	Mesh	Mesh	Estrela	Par a par
Frequencia	5–60 GHz	2.4 GHz	2.4 GHz	2.4 GHz	2.5-GHz 3.5-GHz 5.8-GHz	908.42 MHz	2.4 GHz	868/900 MHz	13.56 MHz
Consumo de energia	Baixo Alto	Baixo	Muito baixo	Muito baixo	Alto	Muito baixo	Alto	Muito Baixo	Médio
Taxa de envio de dados	1.3 Gbps	1–24 Mb/s	250 kb/s	300 kb/s	11-100mb/s	40kb/s	0.1–1 Gb/s	0.3–50 Kb/s	424 Kb/s
Alcance	100m	8–10 m	10-20m	10-20m	50km	100m	Área do sinal	<30 Km	10cm
Tipo de aplicacao	Qualquer dispositivo com capacidade de conexão	Rede para troca de dados	Sensor network de automação industrial	Sensor network de automação industrial	Conexão de Internet de marca ampla na área metropolitana	Automação residencial	Celulares	Smart cities	Cartão de crédito/ônibus
Tamanho da rede	Médio	Pequena	Muito grande	Muito grande	Grande	Grande	Muito grande	Médio	Muito pequena
Custo	Alto	Baixo	Médio	Baixo - Médio	Alto	Baixo - Médio	Médio	Médio	Baixo

# 1- NFC



- x NFC é uma sigla para “Near Field Communication”
- x tecnologia de troca de dados sem fio por aproximação entre dois dispositivos
- x - **Leitura e gravação:** tendo como base a comunicação passiva, permite leitura ou alteração de dados existentes em um dispositivo NFC, como um receptor que desconta créditos registrados em um cartão de viagens (como o Bilhete Único da cidade de São Paulo);
- x - **Peer-to-peer:** é um modo para troca bidirecional de informações entre os dois dispositivos, ou seja, cada um pode tanto receber quanto enviar dados para o outro. Pode ser útil, por exemplo, para a troca de arquivos entre dois smartphones;
- x - **Emulação de cartão:** neste modo, o dispositivo NFC pode se passar por um cartão inteligente, de forma que o aparelho leitor não consiga distinguir um do outro.



# 1- NFC

NFC





## 2- ZIGBEE

- x O Zigbee é um conjunto de especificações desenvolvido pela Zigbee Alliance para utilização em residências inteligentes (smart home) e IoT, que define as camadas subsequentes às camadas estabelecidas pelo IEEE 802.15.4, oferecendo serviços de segurança, tolerância a erros e conexão de novos dispositivos. O Zigbee abrange as camadas referentes a Internet, transporte e aplicação do modelo TCP/IP.



### 3- WIFI

- X Wireless Fidelity,
- X WLAN (Wireless Local Area Network).
- X Protocollo 802.11 (a, b, g e n,)



## 4- WIFI



## 5- BLUETOOTH

**Bluetooth** é uma especificação de rede **sem fio** de âmbito pessoal (*Wireless personal area networks* – PANs) consideradas do tipo **PAN** ou mesmo WPAN



## 6- GPRS

- x GPRS é a sigla de General Packet Radio Services, ou Serviços Gerais de Pacote por Rádio. GPRS é uma tecnologia que tem o objetivo de aumentar as taxas de transferência de dados entre celulares, facilitando a comunicação e o acesso a redes.



## 7- LORA

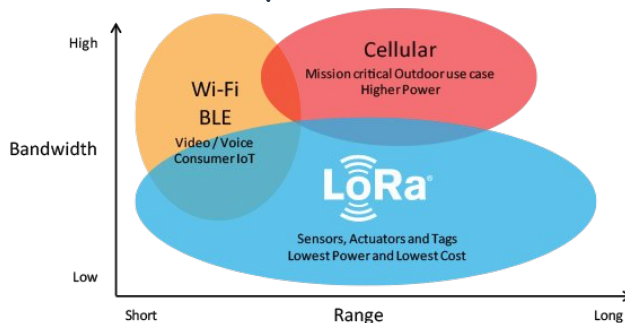
LoRa (**L**ong **R**ange) é uma tecnologia de rede de área ampla de baixa potência. Baseia-se em técnicas de modulação de espectro de propagação derivadas da tecnologia **chirp spread spectrum** (CSS).



## 7- LORA



- X Baixo consumo de energia
- X Longo alcance (Geralmente 2~7Km)
- X Possibilidade de montar sua própria rede
- X Solução para IoT em ambientes remotos
- X Frequências baixas (433~968MHz)



## 7- LORA



### Minha experiência com LoRa



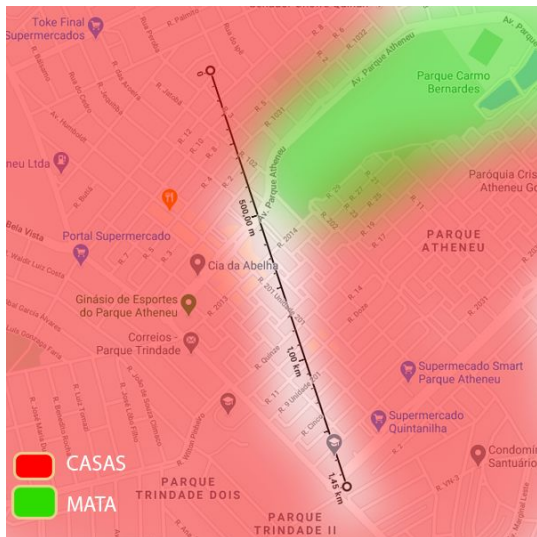
- ✗ Dispositivo IoT para monitorar vazão e fluxo de água.
- ✗ Alcance necessário: min 2.5Km
- ✗ Alcance esperado: +4Km.



# 7- LORA



## Minha experiência com LoRa



X Alcance na cidade: 1.5Km

(Altura do solo)

## 7- LORA



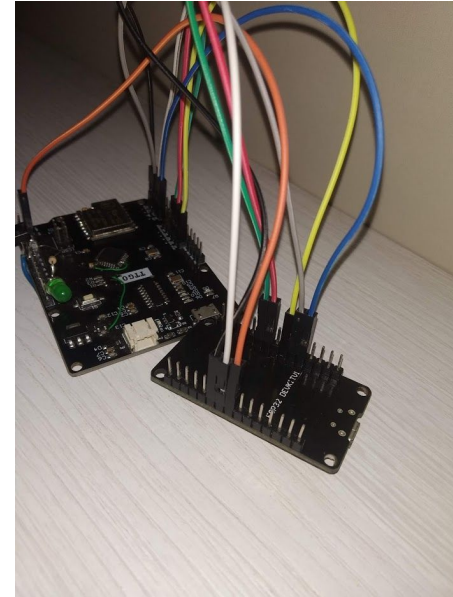
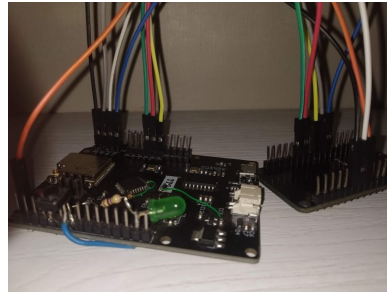
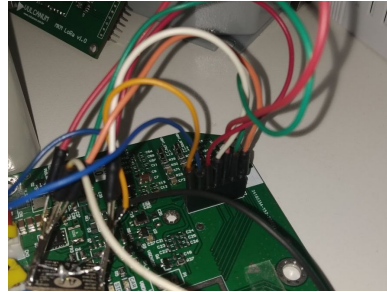
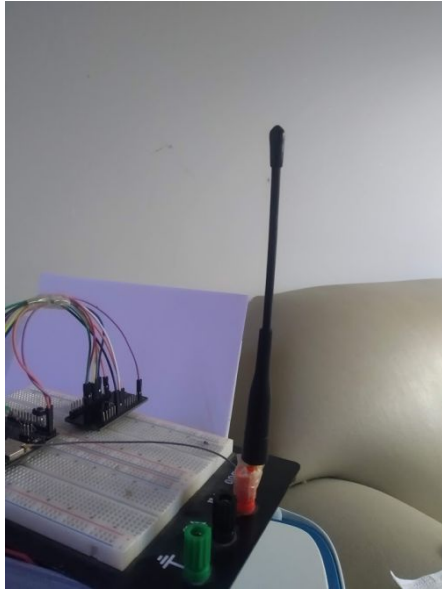
### Minha experiência com LoRa



# 7- LORA



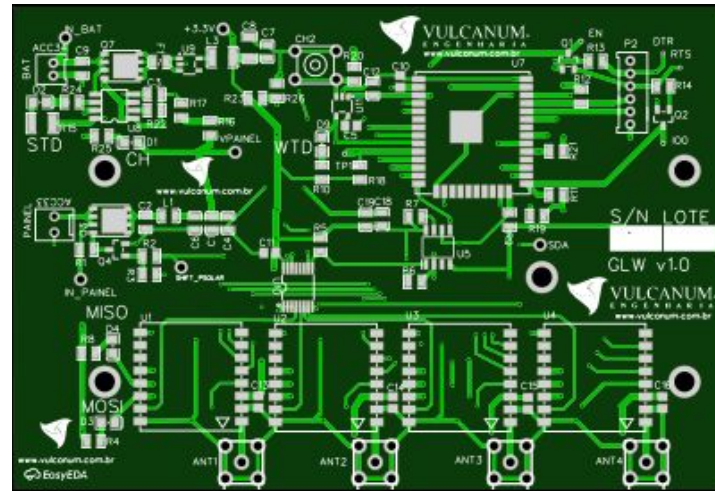
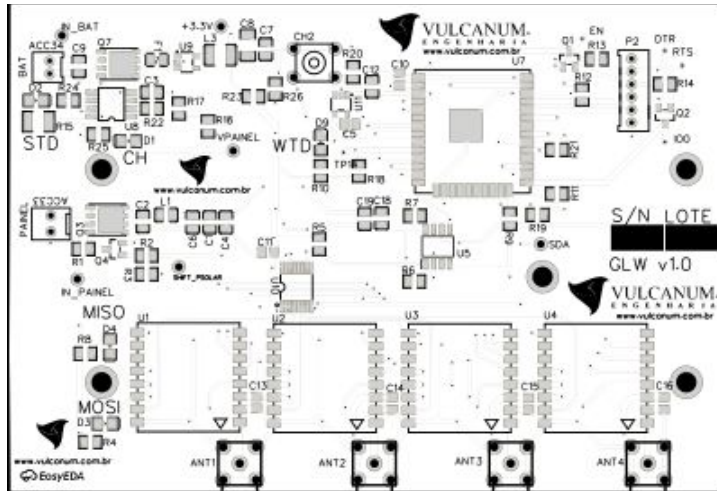
## Minha experiência com LoRa



# 7- LORA



## Minha experiência com LoRa



## 7- LORA



Principais benefícios:

- X Rede privada
- X Longo alcance pelo custo (Módulo RA-02 433MHz ~ R\$40,00)
- X Rede MESH possibilita maior alcance
- X Sem necessidade de infraestrutura

## 7- LORA



### Implementação

- X Biblioteca para arduino
- X Protocolo LoraWan
- X Código dispositivo remoto
- X Código Gateway
- X Testes de envio entre os dois

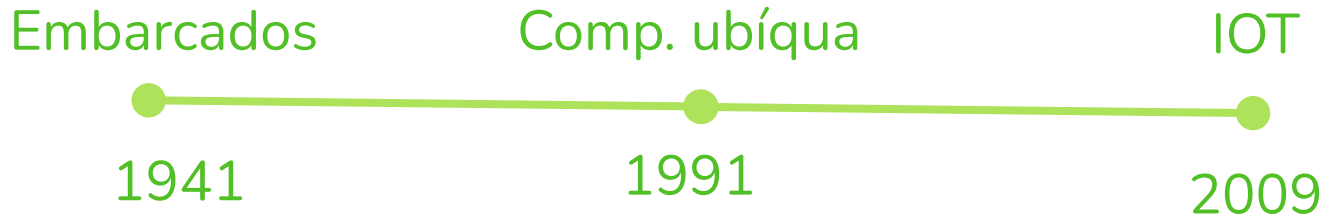
## 7- LORA

Diferenças



# COMPUTAÇÃO UBÍQUA

- Termo criado por Mark Weiser em 1991
- Dispositivos conectados em todos os lugares de forma tão transparente para o ser humano que acabaremos por não perceber que eles estão lá.





Colaboração:

[antonio\\_pereira@outlook.com](mailto:antonio_pereira@outlook.com)

18/11/20

# TECNOLOGIAS DE COMUNICAÇÃO IOT: PARTE PRÁTICA

## 2ª ETAPA

### ESP-NOW

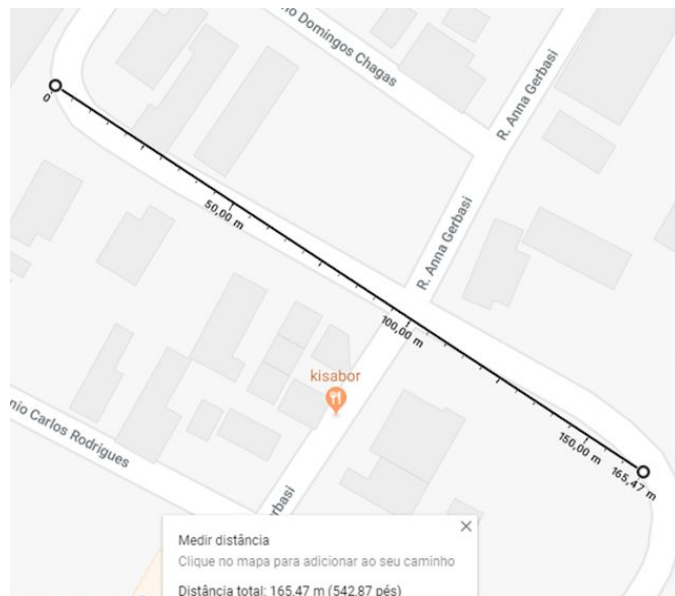
- x Semelhante à conectividade sem fio de baixa potência de **2,4 GHz**
- x Controle direto e com baixa latência
- x Sem necessidade de conexão Wi-Fi
- x Comunicação entre dispositivos ESP
- x Até 200m de distância com antena onboard

# ESP-NOW



# ESP-NOW

## Distância obtida



Fernando K, 3/2018

# ESP-NOW

- x Útil para automação residencial (baixas distâncias).
- x Pode ser útil para o AirPure!

**ESP NOW**

# ESP-NOW(PAI)

```
void setup() {  
  // Init Serial Monitor  
  Serial.begin(115200);  
  
  // Set device as a Wi-Fi Station  
  WiFi.mode(WIFI_STA);  
  
  // Init ESP-NOW  
  if (esp_now_init() != ESP_OK) {  
    Serial.println("Error initializing ESP-NOW");  
    return;  
  }  
  
  // Once ESPNow is successfully Init, we will register for Send CB to  
  // get the status of Trasnmitted packet  
  esp_now_register_send_cb(OnDataSent);  
  
  // Register peer  
  esp_now_peer_info_t peerInfo;  
  memcpy(&peerInfo.peer_addr, broadcastAddress, 6);  
  peerInfo.channel = 0;  
  peerInfo.encrypt = false;  
  
  // Add peer  
  if (esp_now_add_peer(&peerInfo) != ESP_OK) {  
    Serial.println("Failed to add peer");  
    return;  
  }  
}
```

```
void loop() {  
  // Set values to send  
  strcpy(myData.a, "THIS IS A CHAR");  
  myData.b = random(1,20);  
  myData.c = 1.2;  
  myData.d = "Hello";  
  myData.e = false;  
  
  // Send message via ESP-NOW  
  esp_err_t result = esp_now_send(broadcastAddress, (uint8_t *) &myData, sizeof(myData));  
  
  if (result == ESP_OK) {  
    Serial.println("Sent with success");  
  }  
  else {  
    Serial.println("Error sending the data");  
  }  
  delay(2000);  
}
```

# ESP-NOW(FILHO)

```
// callback function that will be executed when data is received
void OnDataRecv(const uint8_t * mac, const uint8_t *incomingData, int len) {
    memcpy(myData, incomingData, sizeof(myData));
    Serial.print("Bytes received: ");
    Serial.println(len);
    Serial.print("Char: ");
    Serial.println(myData.a);
    Serial.print("Int: ");
    Serial.println(myData.b);
    Serial.print("Float: ");
    Serial.println(myData.c);
    Serial.print("String: ");
    Serial.println(myData.d);
    Serial.print("Bool: ");
    Serial.println(myData.e);
    Serial.println();
}
```

```
void setup() {
    // Initialize Serial Monitor
    Serial.begin(115200);

    // Set device as a Wi-Fi Station
    WiFi.mode(WIFI_STA);

    // Init ESP-NOW
    if (esp_now_init() != ESP_OK) {
        Serial.println("Error initializing ESP-NOW");
        return;
    }

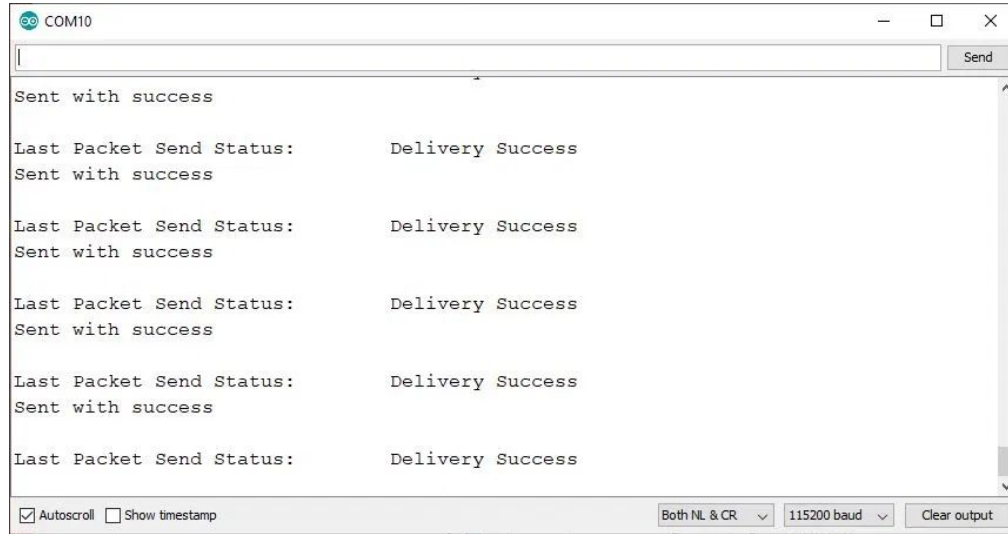
    // Once ESPNow is successfully Init, we will register for recv CB to
    // get recv packer info
    esp_now_register_recv_cb(OnDataRecv);
}
```

CÓDIGO

PARA BAIXAR O CÓDIGO,  
CLIQUE AQUI.



# FUNCIONAMENTO (FILHO)

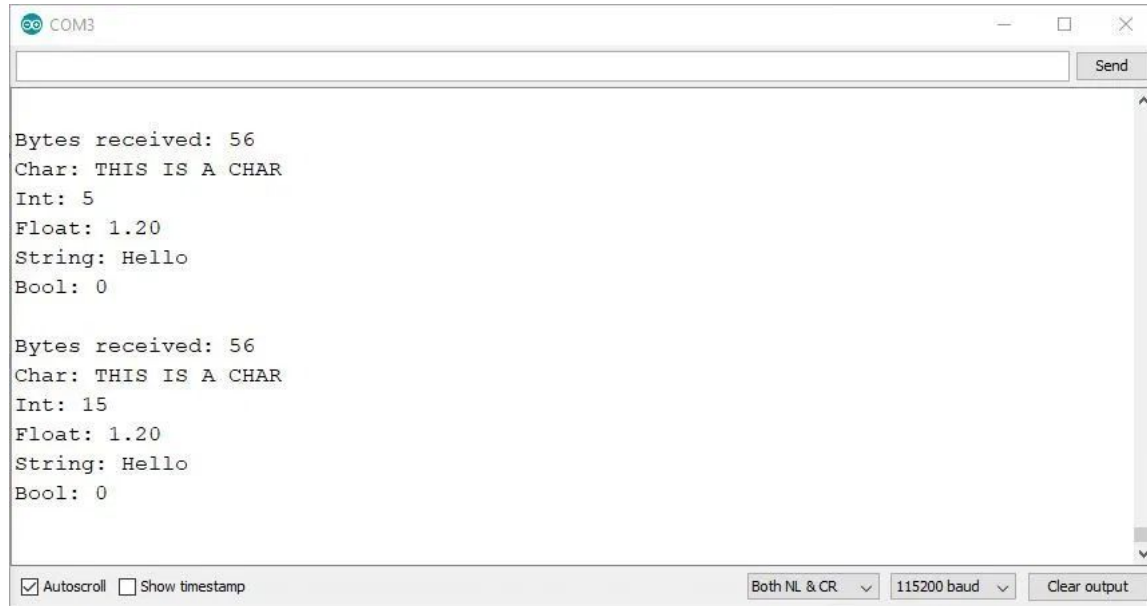


The screenshot shows a terminal window titled "COM10". It displays a series of status messages indicating successful packet transmissions. The messages are as follows:

```
Sent with success  
  
Last Packet Send Status:      Delivery Success  
Sent with success  
  
Last Packet Send Status:      Delivery Success  
Sent with success  
  
Last Packet Send Status:      Delivery Success  
Sent with success  
  
Last Packet Send Status:      Delivery Success  
Sent with success  
  
Last Packet Send Status:      Delivery Success
```

At the bottom of the window, there are several controls: a checked "Autoscroll" checkbox, an unchecked "Show timestamp" checkbox, a baud rate dropdown set to "115200 baud", and a "Clear output" button. The window also has a "Send" button in the top right corner.

# FUNCIONAMENTO (PAI)



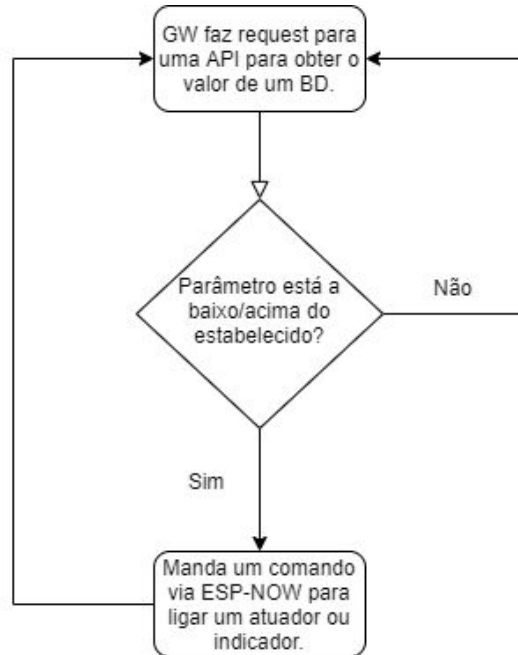
```
COM3

Bytes received: 56
Char: THIS IS A CHAR
Int: 5
Float: 1.20
String: Hello
Bool: 0

Bytes received: 56
Char: THIS IS A CHAR
Int: 15
Float: 1.20
String: Hello
Bool: 0

Autoscroll Show timestamp Both NL & CR 115200 baud Clear output
```

# FLUXOGRAMA DE UMA APLICAÇÃO



## 2º ETAPA

### LORA (SENDER - RECEIVER)

- x Utilizando o chip SX1278;
- x Funciona com Arduino ou ESP32;
- x Método mais simples de funcionamento.

# LORA



433MHZ

# LORA - SENDER

```
#include <SPI.h>
#include <LoRa.h>

int counter = 0;

void setup() {
  Serial.begin(9600);
  while (!Serial);

  Serial.println("LoRa Sender");

  if (!LoRa.begin(915E6)) {
    Serial.println("Starting LoRa failed!");
    while (1);
  }
}
```

```
void loop() {
  Serial.print("Sending packet: ");
  Serial.println(counter);

  // send packet
  LoRa.beginPacket();
  LoRa.print("hello ");
  LoRa.print(counter);
  LoRa.endPacket();

  counter++;

  delay(5000);
}
```

# LORA - RECEIVER

```
#include <SPI.h>
#include <LoRa.h>

void setup() {
  Serial.begin(9600);
  while (!Serial);

  Serial.println("LoRa Receiver");

  if (!LoRa.begin(915E6)) {
    Serial.println("Starting LoRa failed!");
    while (1);
  }
}
```

```
void loop() {
  // try to parse packet
  int packetSize = LoRa.parsePacket();
  if (packetSize) {
    // received a packet
    Serial.print("Received packet ");

    // read packet
    while (LoRa.available()) {
      Serial.print((char)LoRa.read());
    }

    // print RSSI of packet
    Serial.print(" with RSSI ");
    Serial.println(LoRa.packetRssi());
  }
}
```

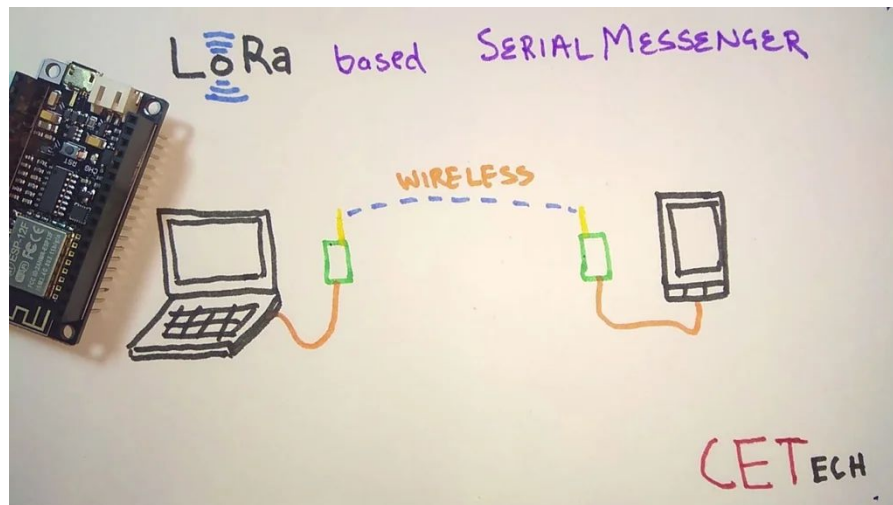
CÓDIGO

PARA BAIXAR O CÓDIGO,  
CLIQUE AQUI.



2º ETAPA

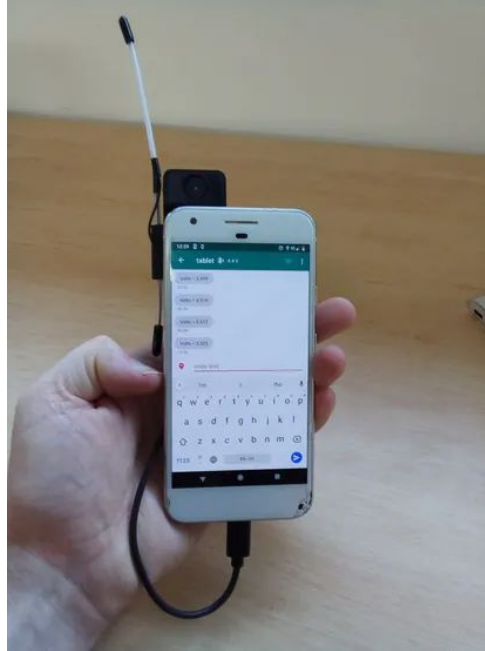
## LORA (MENSAGEIRO)



# LORA (MENSAGEIRO)

- x Utilizando o chip SX1278;
- x Utiliza um aplicativo chamado “Ripple Messenger” para comunicar com o ESP32.
- x O ESP32 comunica com o outro nodo, que por sua vez comunica com o outro celular.

# LORA (MENSAGEIRO)

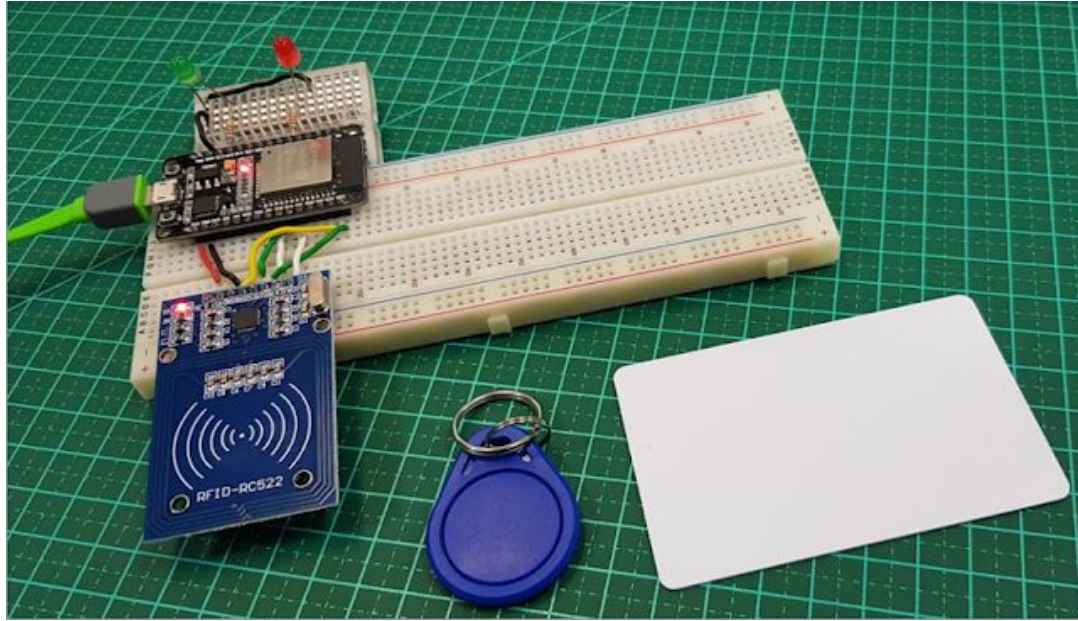


scottpowell69/Hackster.io

CÓDIGO

PARA BAIXAR O CÓDIGO,  
CLIQUE AQUI.

# NFC

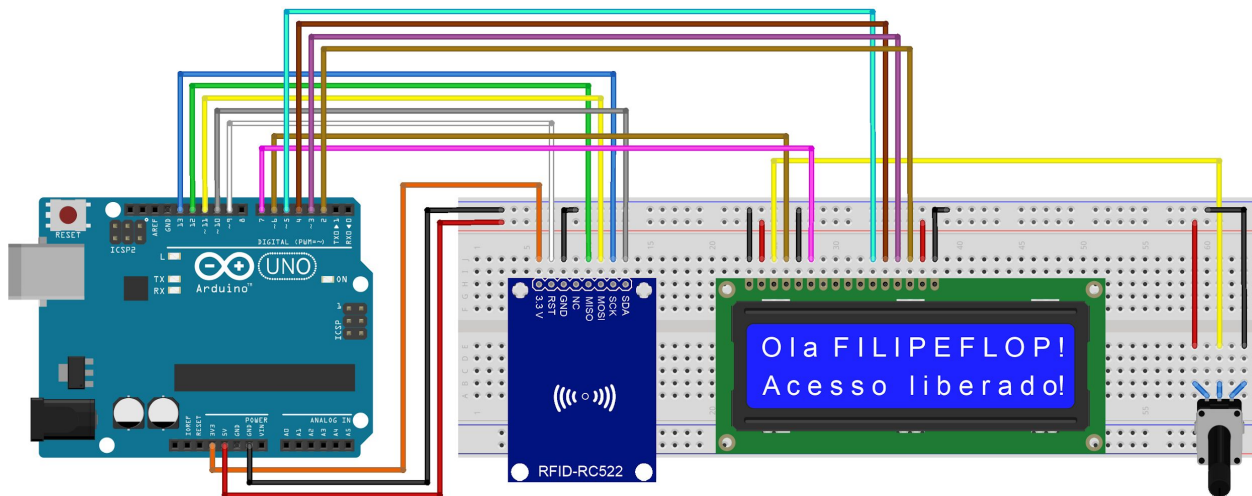


# NFC

- x Diversas utilidades no ramo da segurança por se assemelhar com uma “chave”.
- x Crescente utilização no ramo de pagamentos (Ex. Apple Pay, Android Pay, etc).

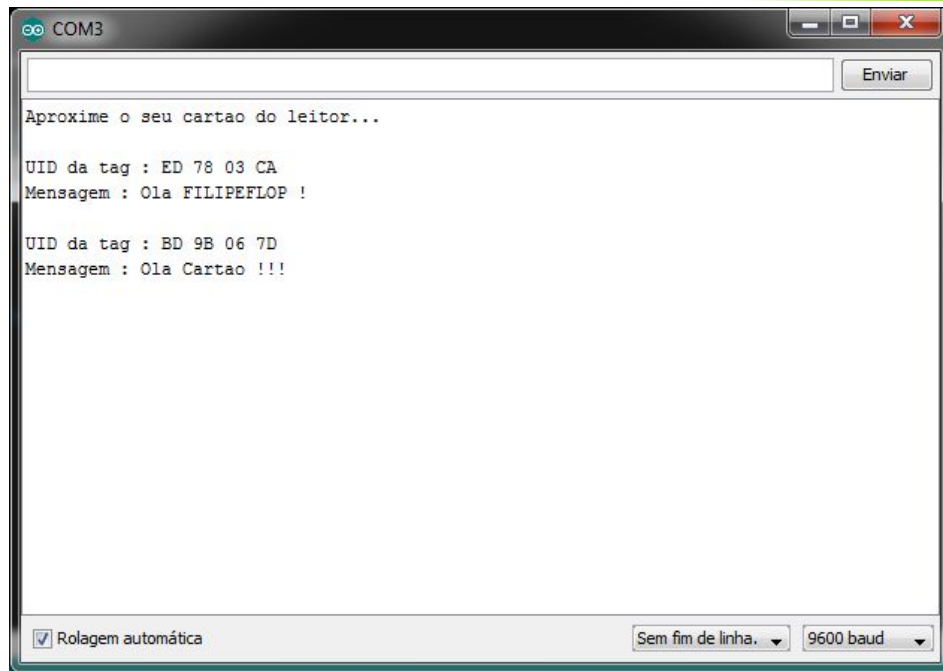
# NFC

## DIAGRAMA DE LIGAÇÃO



Filipeflop, 2014.

# FUNCIONAMENTO





CÓDIGO

PARA BAIXAR O CÓDIGO,  
CLIQUE AQUI.

# GPS



**Triangulação:** Determina a localização do receptor na Terra.

- x três satélites enviam o sinal para o receptor, que calcula quanto tempo cada sinal demorou a chegar nele.

# GPS

- x Precisão de até 20m.
- x Retorno rápido da informação.
- x Amplo alcance de grande parte do mundo.

# GPS



# GPS

Coloque suas credenciais wi-fi (nome da rede e senha) no código-fonte, nas constantes `ssid_wifi` e `password_wifi`.

Compile o projeto e o grave na placa (utilizando como board o ESP32 Dev Module).

Com um cliente MQTT qualquer no seu smartphone ou computador, conecte-se ao broker público `broker.hivemq.com` na porta 1883 e se inscreva ao tópico MQTT do seu código-fonte (definido na constante `topico_publish_mqtt`). Segue abaixo algumas sugestões de clientes MQTT:

- Computador: MQTT Lens
- Smartphone Android: MyMQTT

Observe no cliente MQTT as mensagens com a localização (latitude e longitude) e horário obtidos via GPS (considerando GMT 0, ou seja, fuso-horário de Greenwich) chegando na temporização configurada no seu código-fonte (no `define TEMPO_ENTRE_POSICOES_GPS`).

CÓDIGO

PARA BAIXAR O CÓDIGO,  
CLIQUE AQUI.