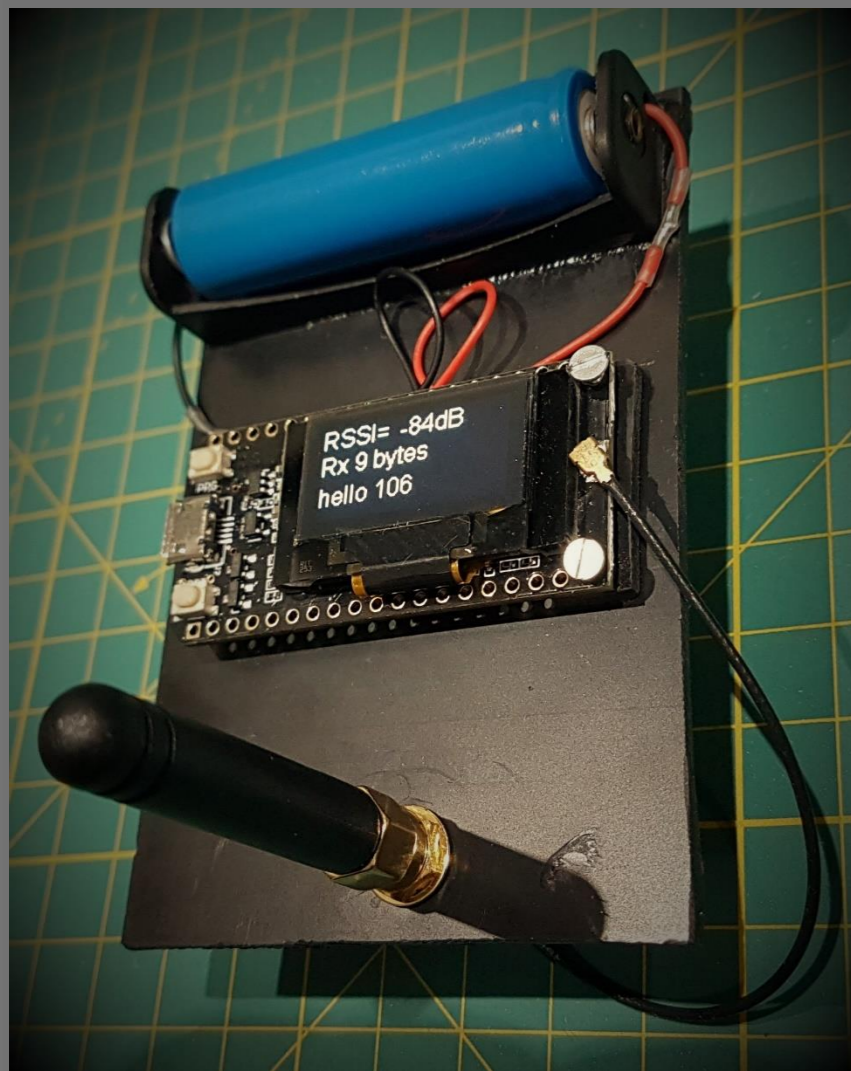
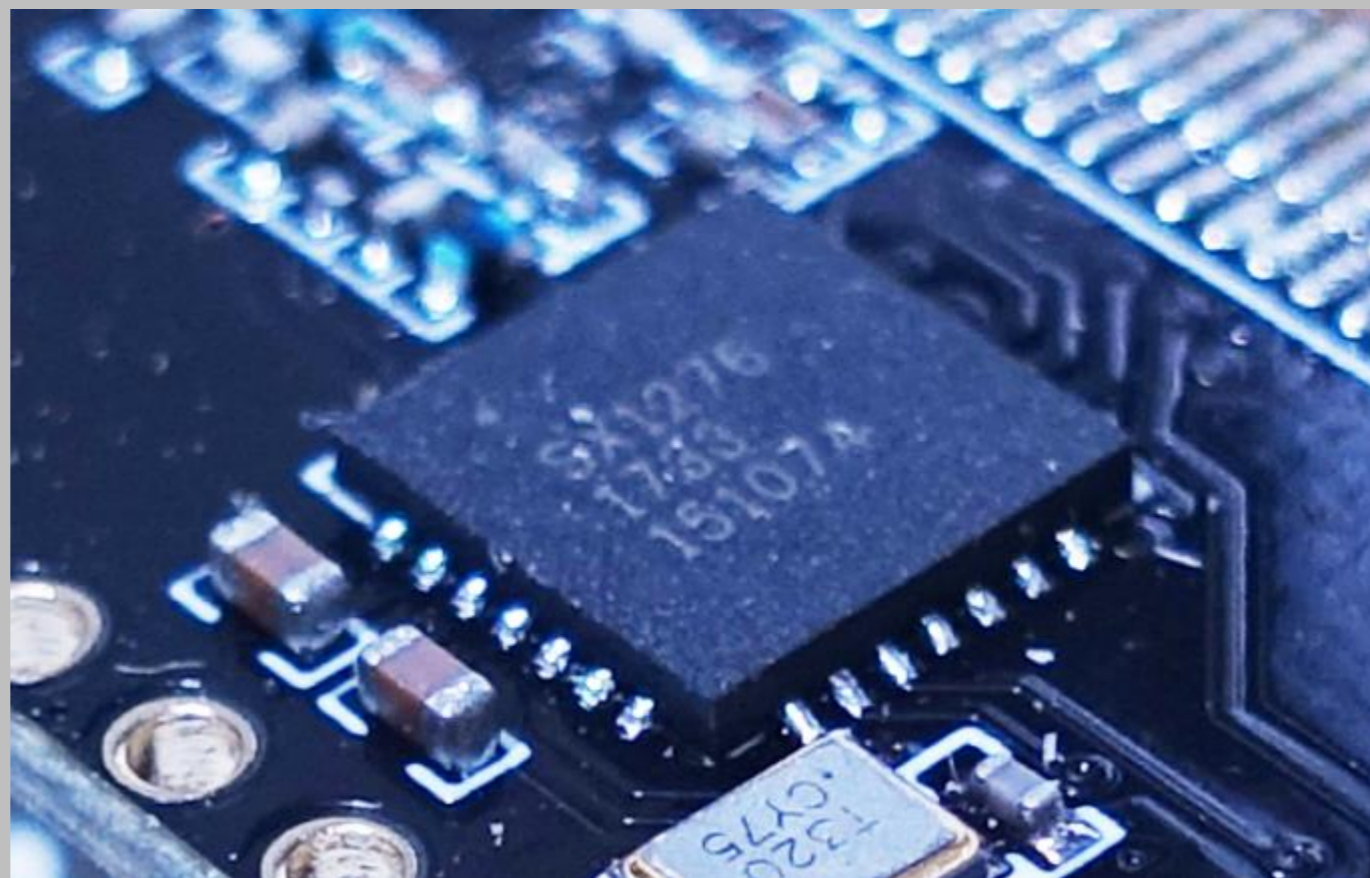


# ESP32 LoRaWan - Longa Distância



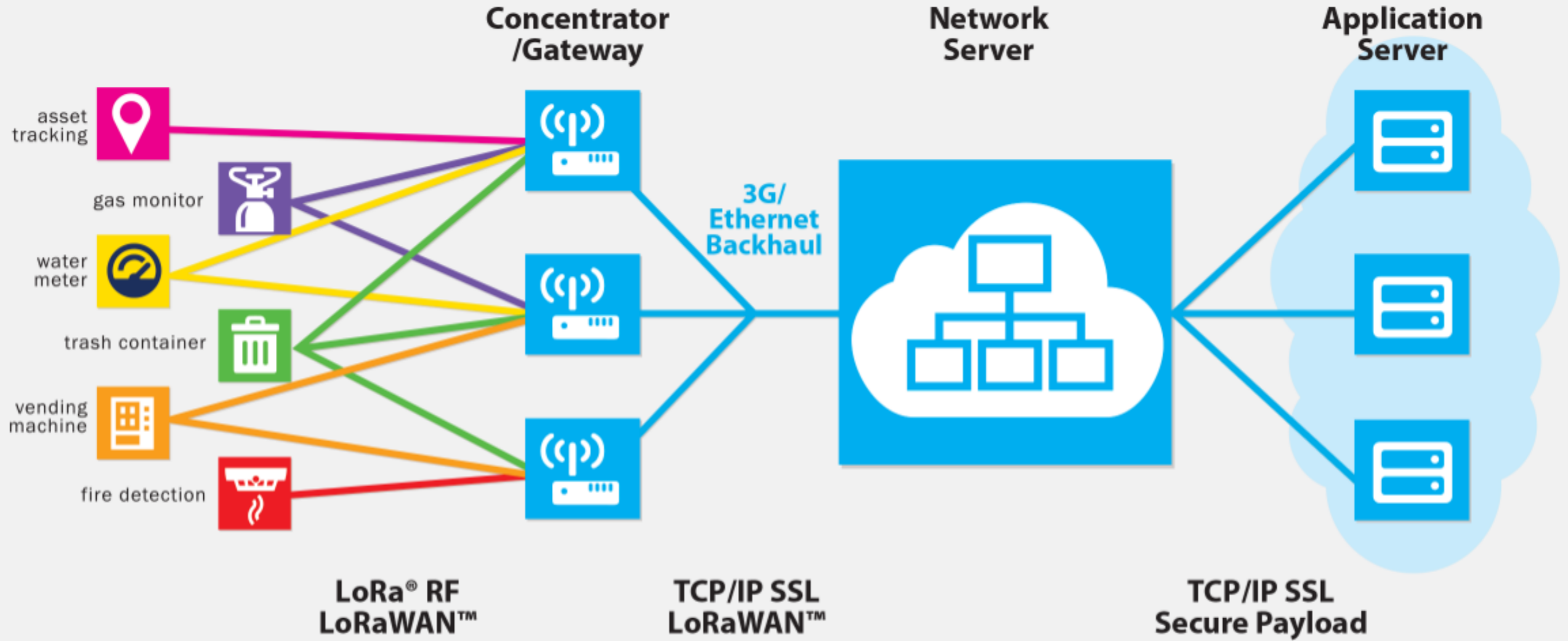
ESP32 + chip SX1276



Por Fernando Koyanagi



# Network Diagram



## Principais recursos da tecnologia LoRa e do protocolo LoRaWAN

### GEOLocalização

Permite aplicações de rastreamento de baixo consumo de GPS e de baixa potência

### BAIXO CUSTO

Reduz os custos de três maneiras: investimento em infra-estrutura, despesas operacionais e sensores de nodos finais

### PADRONIZADO

A interoperabilidade global aprimorada acelera a adoção e implantação de redes baseadas em LoRaWAN e aplicativos IoT

### BAIXO PODER “Consumo”

Protocolo projetado especificamente para baixo consumo de energia que aumenta a vida útil da bateria até 20 anos

### LONGO ALCANCE

A estação base única proporciona penetração profunda em regiões urbanas / interiores densas, além de conectar áreas rurais até 30 milhas de distância

### SEGURO

Criptografia AES128 integrada de ponta a ponta

### ALTA CAPACIDADE

Suporta milhões de mensagens por estação base, ideal para operadores de rede pública atendendo a muitos clientes

## Protocolo LoRaWAN

LoRaWAN é uma especificação de protocolo construída em cima da tecnologia LoRa desenvolvida pela **Aliança LoRa**. Ele usa espectro radioelétrico sem licença nas bandas Industrial, Científica e Médica (ISM) para permitir a comunicação de baixa potência e ampla área entre sensores remotos e gateways conectados à rede. Esta abordagem baseada em padrões para a construção de um LPWAN permite a instalação rápida de redes IoT públicas ou privadas em qualquer lugar usando hardware e software que seja bidirecionalmente seguro, interoperável e móvel, fornece uma localização precisa e funciona do jeito que você espera. A especificação está disponível gratuitamente para [download no site da LoRa Alliance](#).

# Chip sx1276

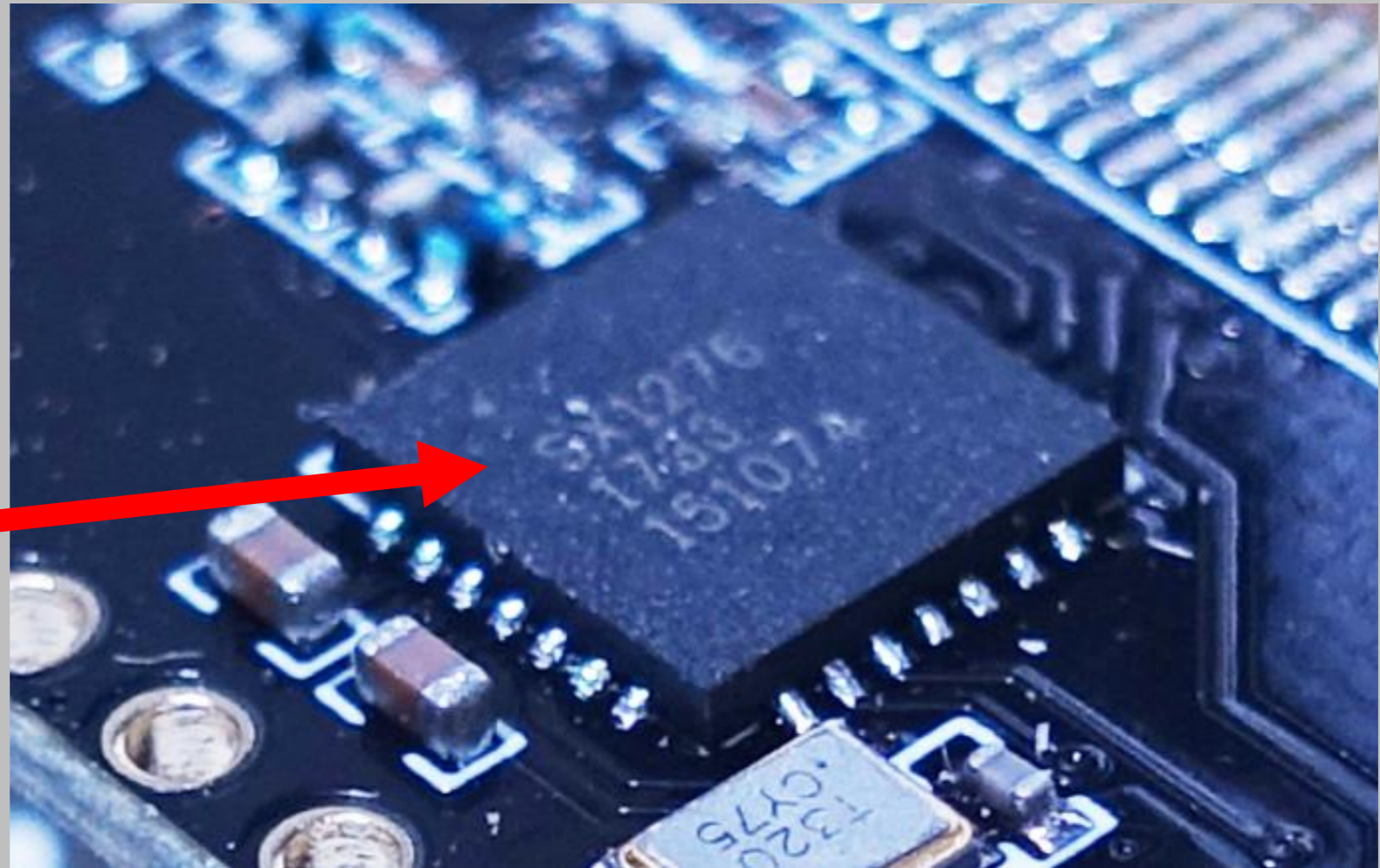
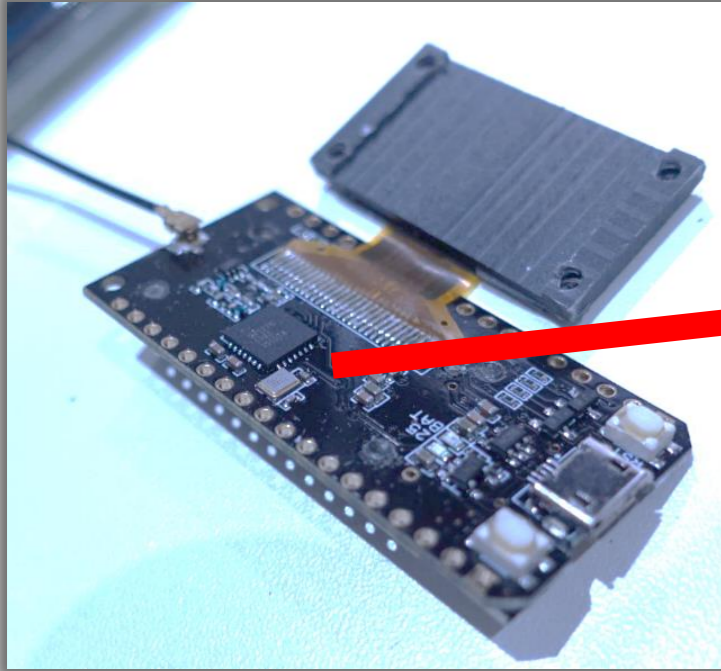
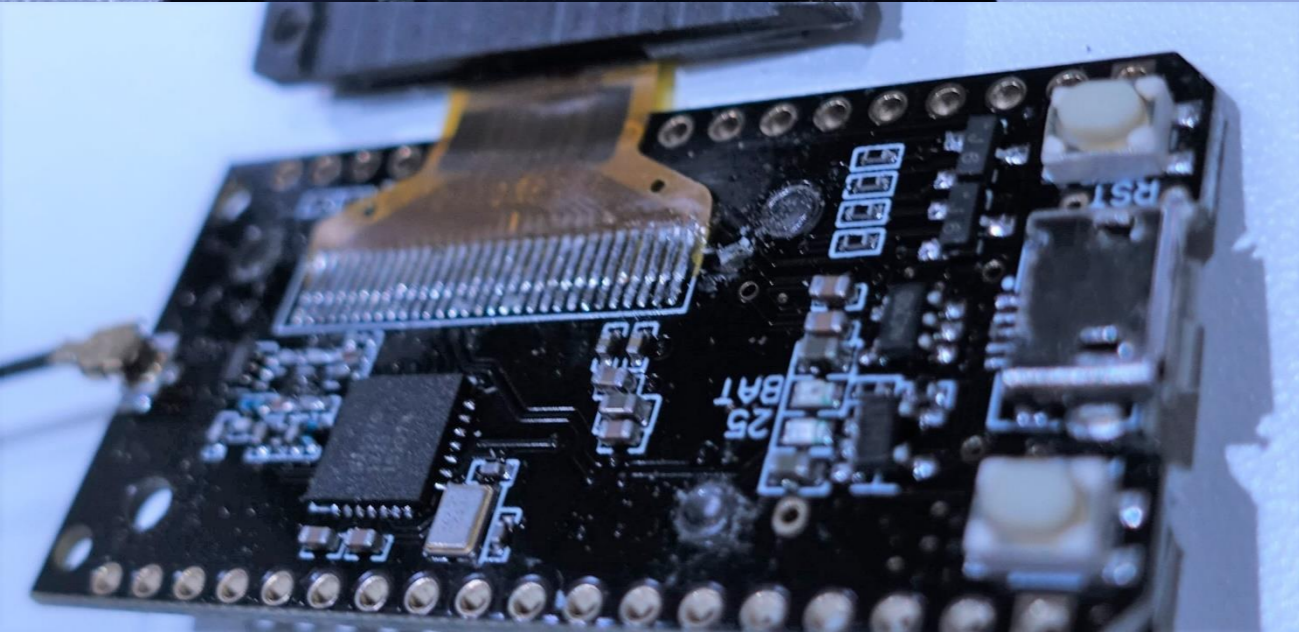
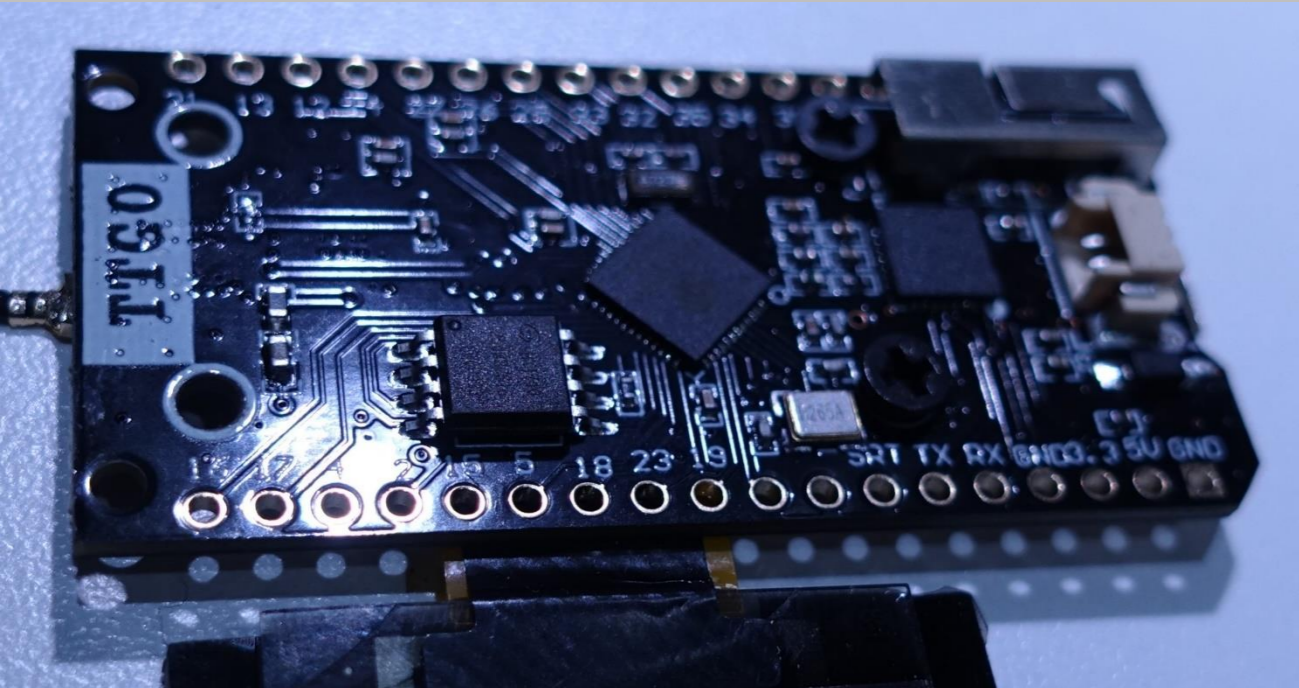


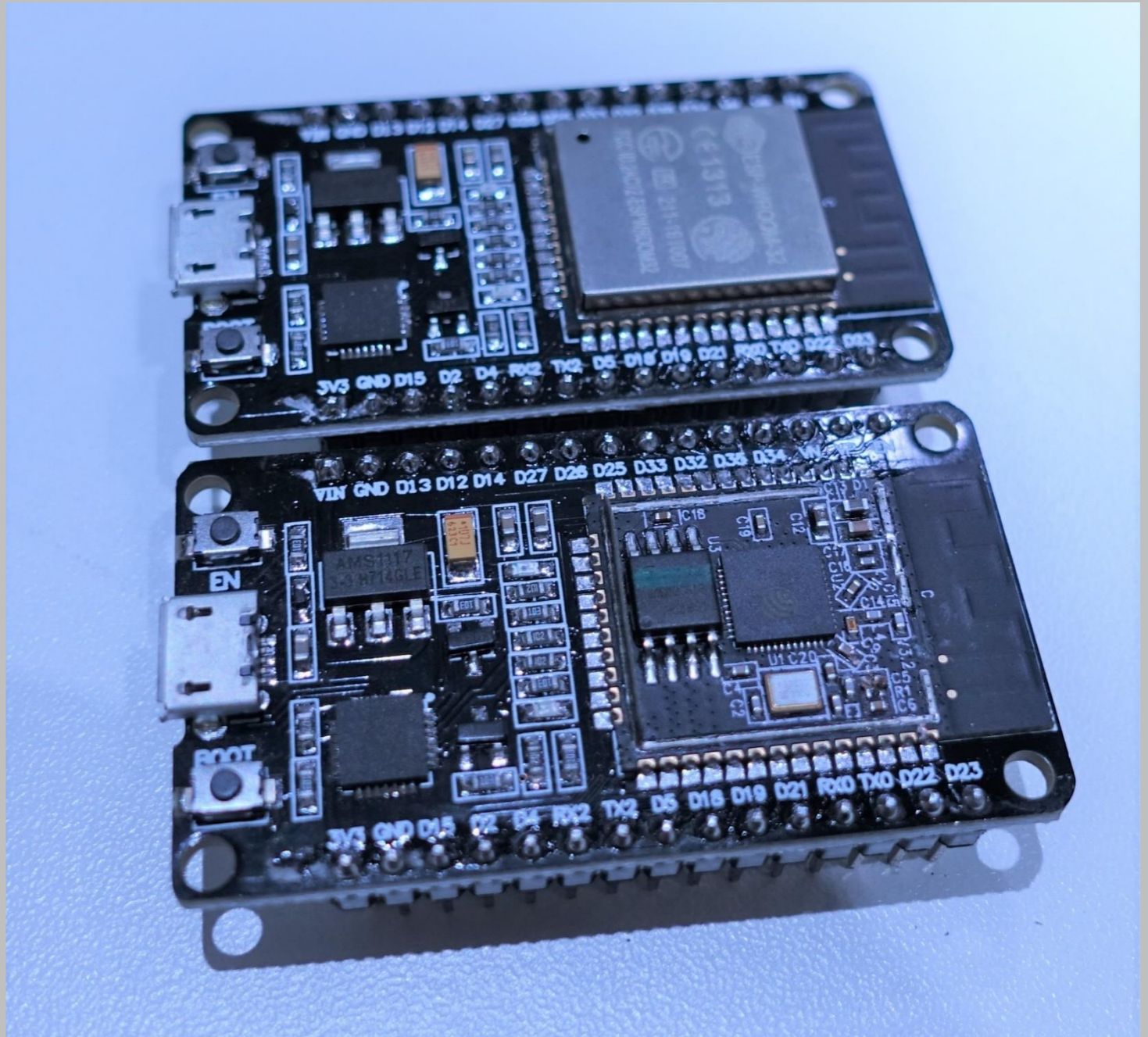
Table 1 SX1276/77/78/79 Device Variants and Key Parameters

Part Number	Frequency Range	Spreading Factor	Bandwidth	Effective Bitrate	Est. Sensitivity
SX1276	137 - 1020 MHz	6 - 12	7.8 - 500 kHz	.018 - 37.5 kbps	-111 to -148 dBm
SX1277	137 - 1020 MHz	6 - 9	7.8 - 500 kHz	0.11 - 37.5 kbps	-111 to -139 dBm
SX1278	137 - 525 MHz	6- 12	7.8 - 500 kHz	.018 - 37.5 kbps	-111 to -148 dBm
SX1279	137 - 960MHz	6- 12	7.8 - 500 kHz	.018 - 37.5 kbps	-111 to -148 dBm

# ESP32 LoRa



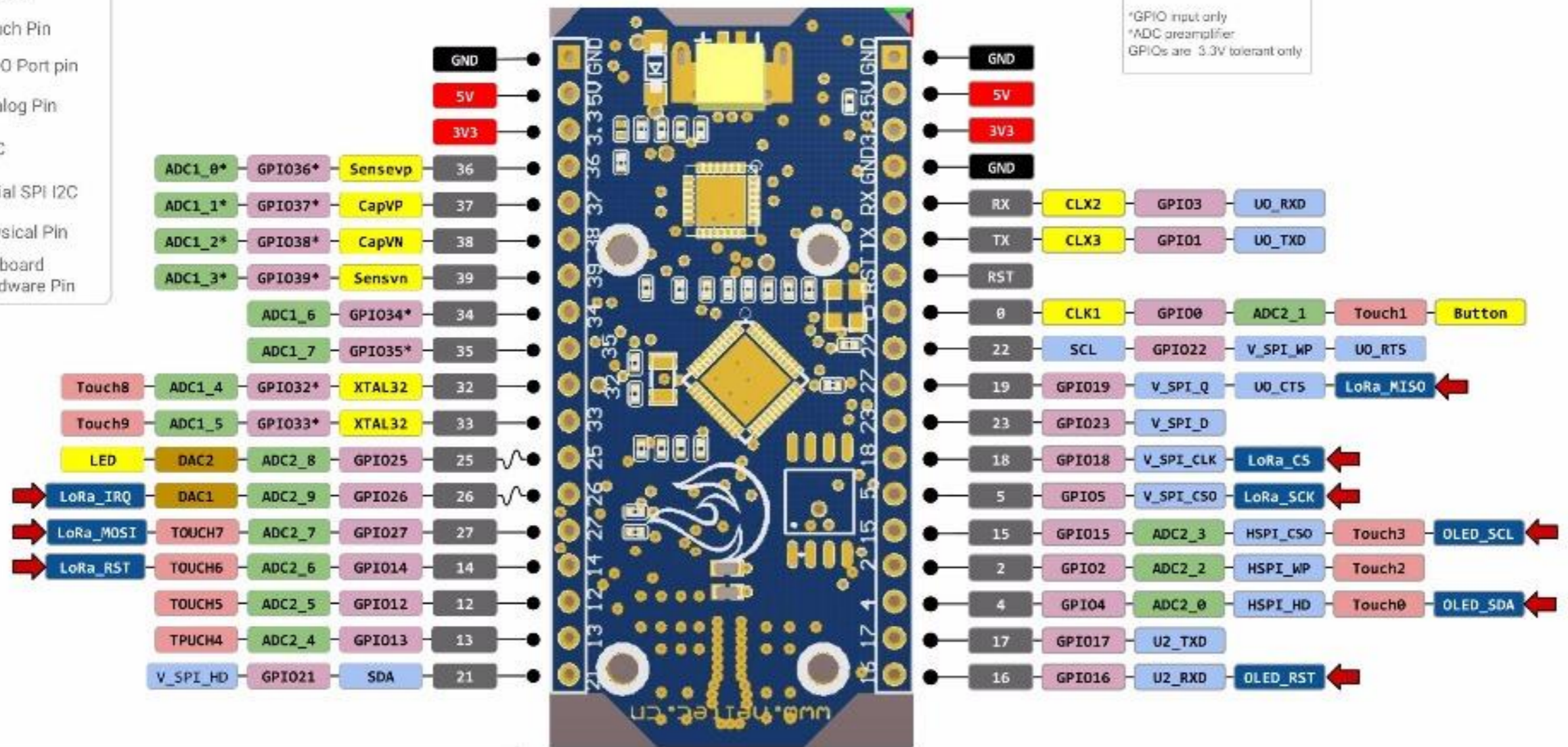
# ESP32 WiFi



# WiFi LoRa 32 Pinout Diagram

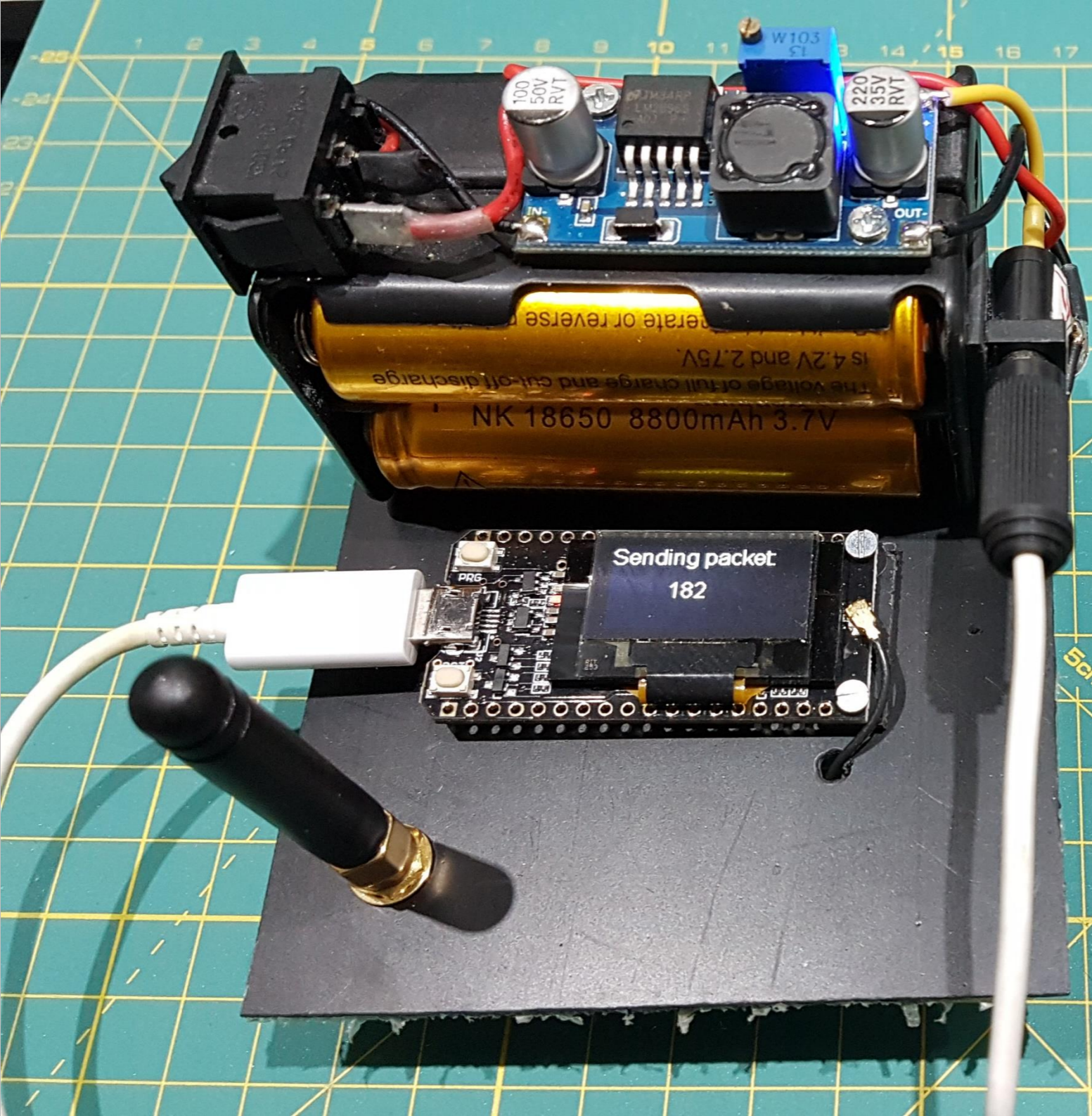
- GND
- Power
- Control
- Touch Pin
- GPIO Port pin
- Analog Pin
- DAC
- Serial SPI I2C
- Physical Pin
- On-board hardware Pin

Notes:  
 \*GPIO input only  
 \*ADC preamplifier  
 GPIOs are 3.3V tolerant only

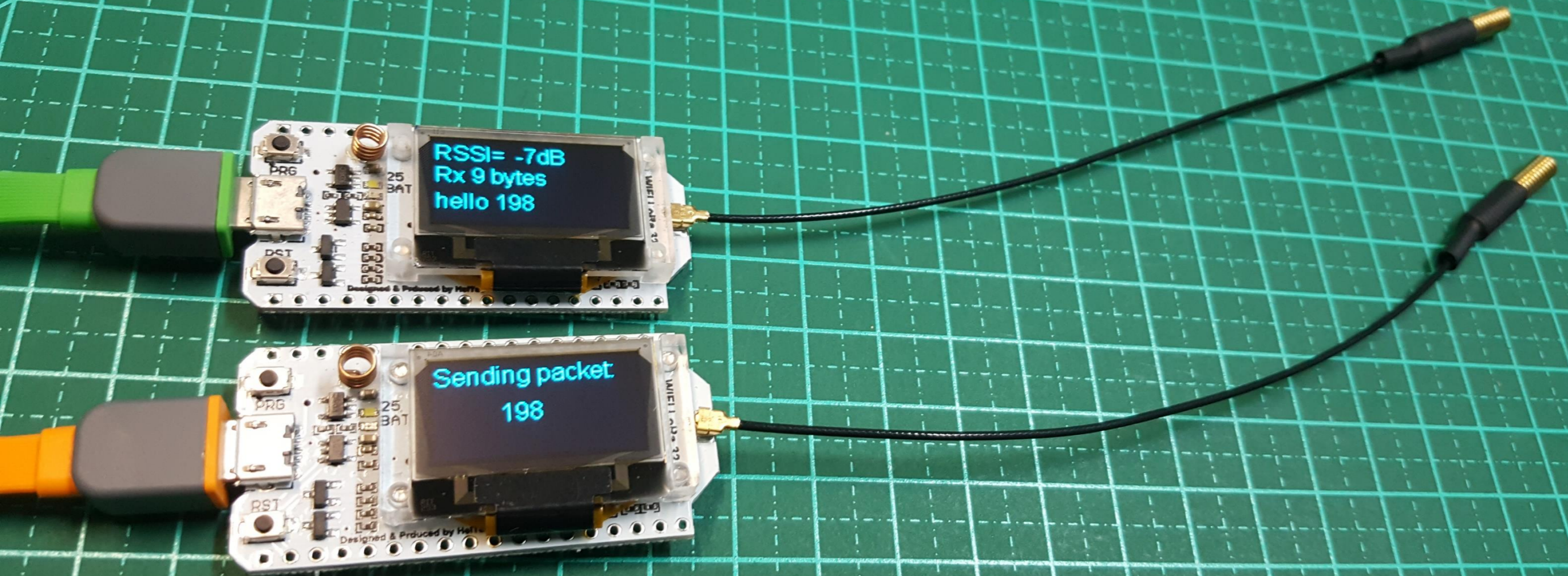


➔ Pins with this arrow are used by on-board OLED or LoRa, they must not be used for other purpose unless you know what you are doing!

# ESP32 LoRaWan ttgo



# ESP32 LoRaWan - HELTEC





# Lib e Exemplos da Heltec

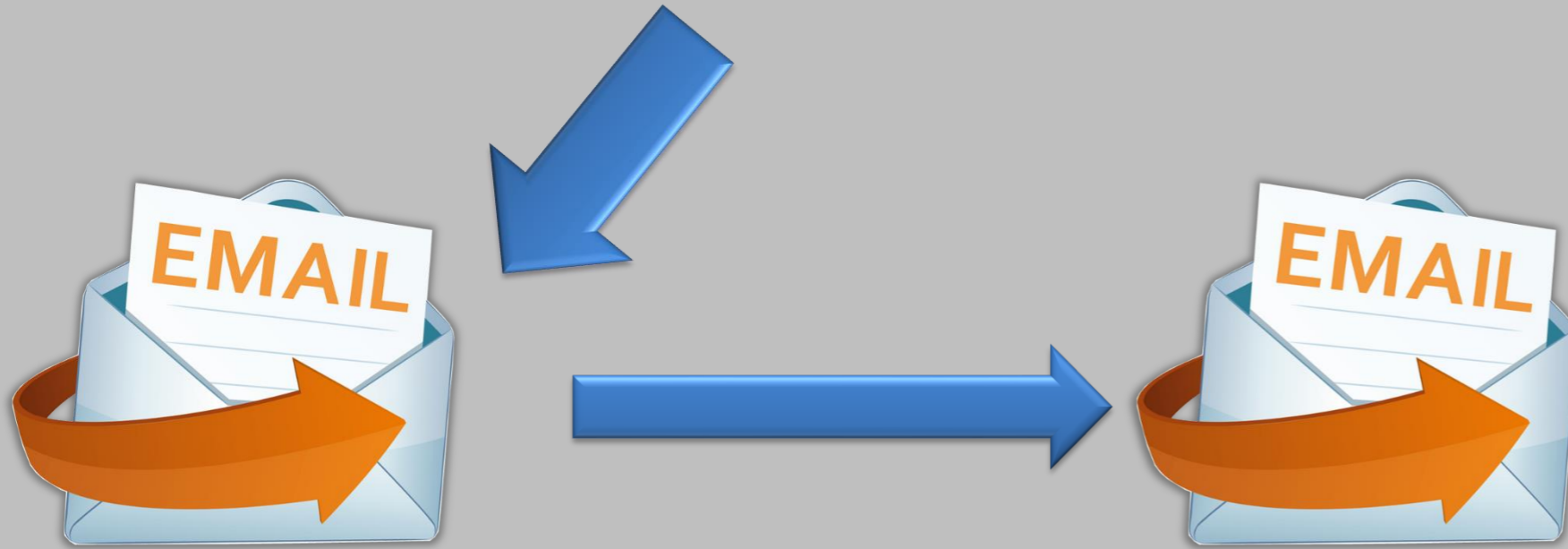
<https://github.com/Heltec-Aaron-Lee/WiFi Kit series>



**No seu computador veja os exemplos !**

**C:\Program Files (x86)\Arduino\hardware\heltec\esp32\libraries\LoRa\examples\OLED\_LoRa\_Sender**

# Cadastre-se na minha lista de e-mail !



E-mail de confirmação

Para Receber o link do download



Como Instalar o Arduino IDE ESP32 Lora - Heltec no Windows.pdf

Entre no endereço:

# 1º Passo

[pdf.fernandok.com](http://pdf.fernandok.com)

Fernando K Tecnologia

Como instalar o ESP32 no Arduino IDE no Windows

ESP32 LoRaWan - HELTEC

Fernando Koyanagi

Inscreva-se agora e receba o PDF

Digite seu Nome

Digite seu E-mail

DOWNLOAD PDF

\*Verifique sua caixa de spam

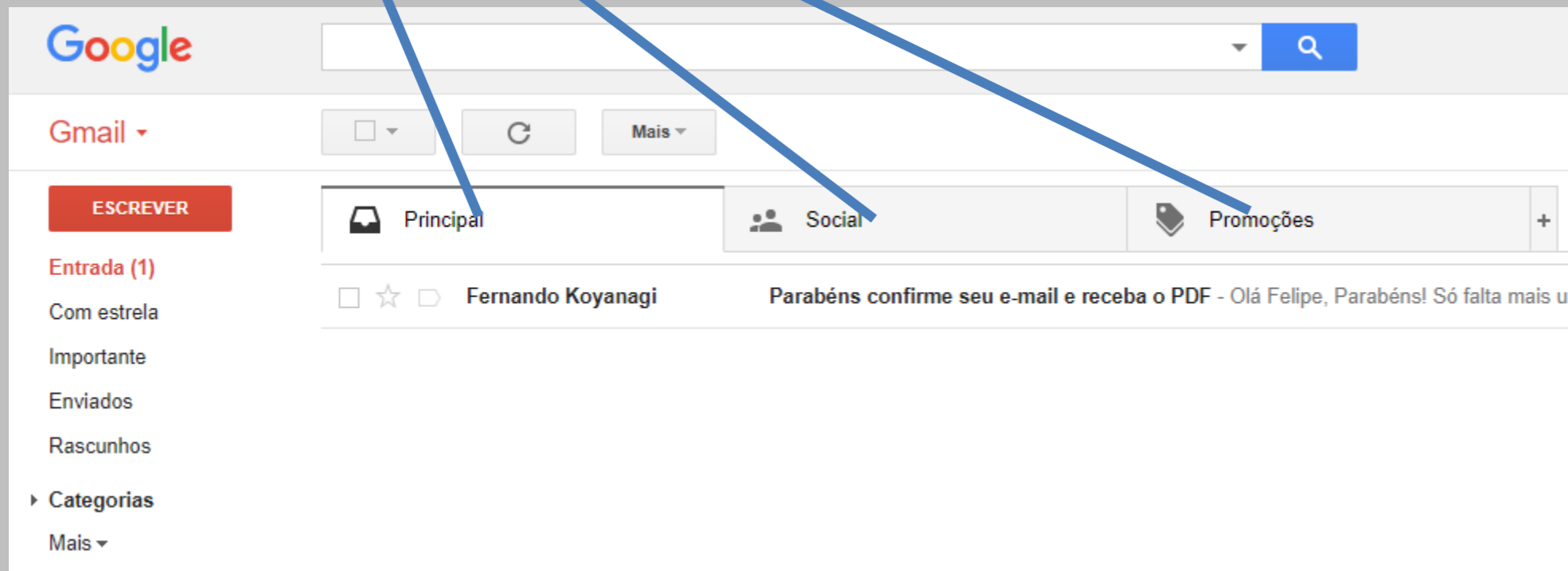
TRUSTe CERTIFIED PRIVACY

Norton by Symantec

fernandok.com

# Achar o e-mail

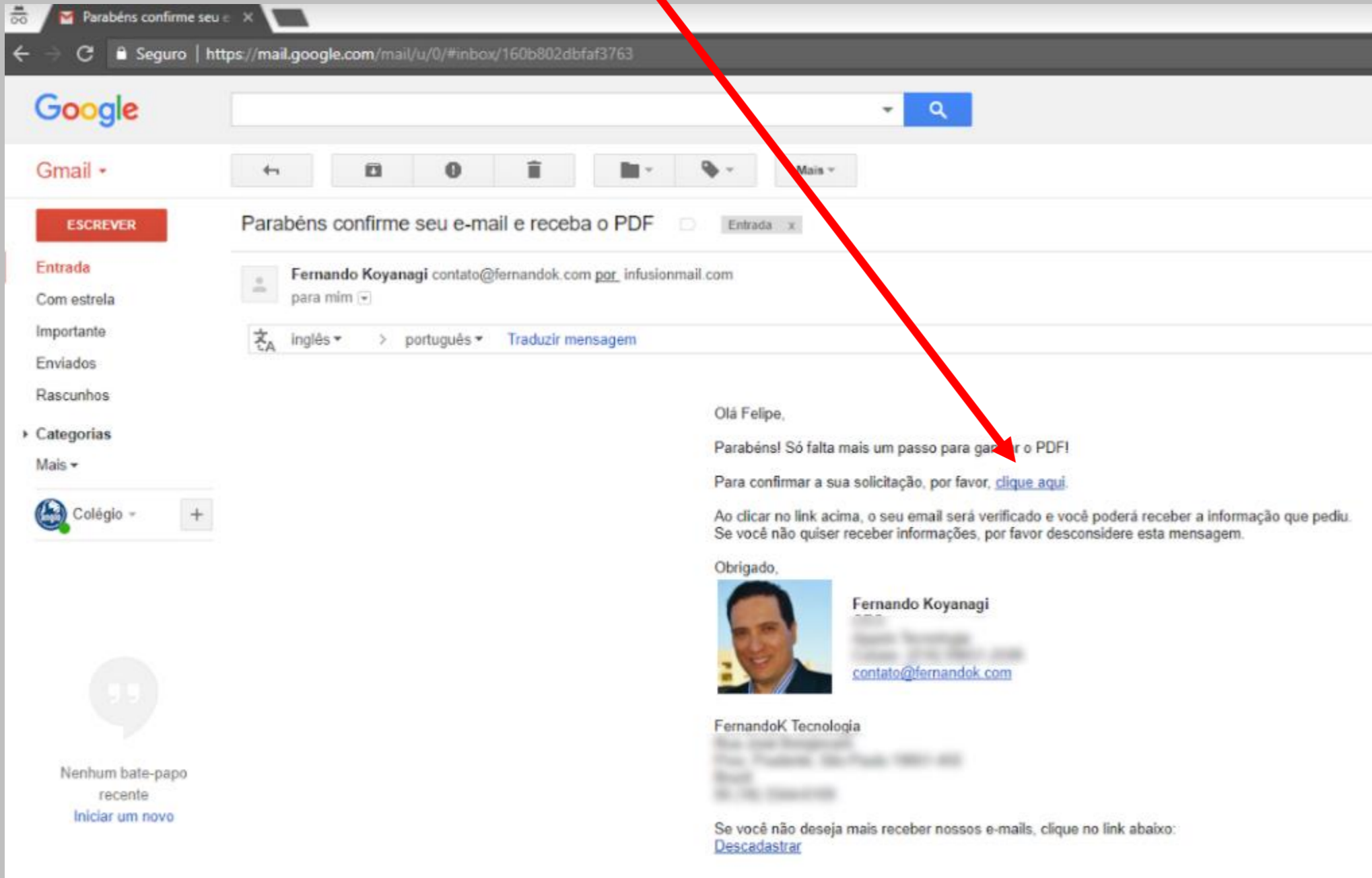
## 2º Passo



**Veja se esta no spam ou atualizações !**

# Clicar no **Clique aqui**

# 3º Passo



The screenshot shows a Gmail interface with a message from Fernando Koyanagi. A red arrow points from the text 'Clique aqui' in the main heading to the underlined link 'clique aqui' in the email body.

Parabéns confirme seu e-mail e receba o PDF

Fernando Koyanagi contato@fernandok.com por infusionmail.com para mim

Inglês > português Traduzir mensagem


Olá Felipe,

Parabéns! Só falta mais um passo para ganhar o PDF!

Para confirmar a sua solicitação, por favor, [clique aqui](#).

Ao clicar no link acima, o seu email será verificado e você poderá receber a informação que pediu. Se você não quiser receber informações, por favor desconsidere esta mensagem.

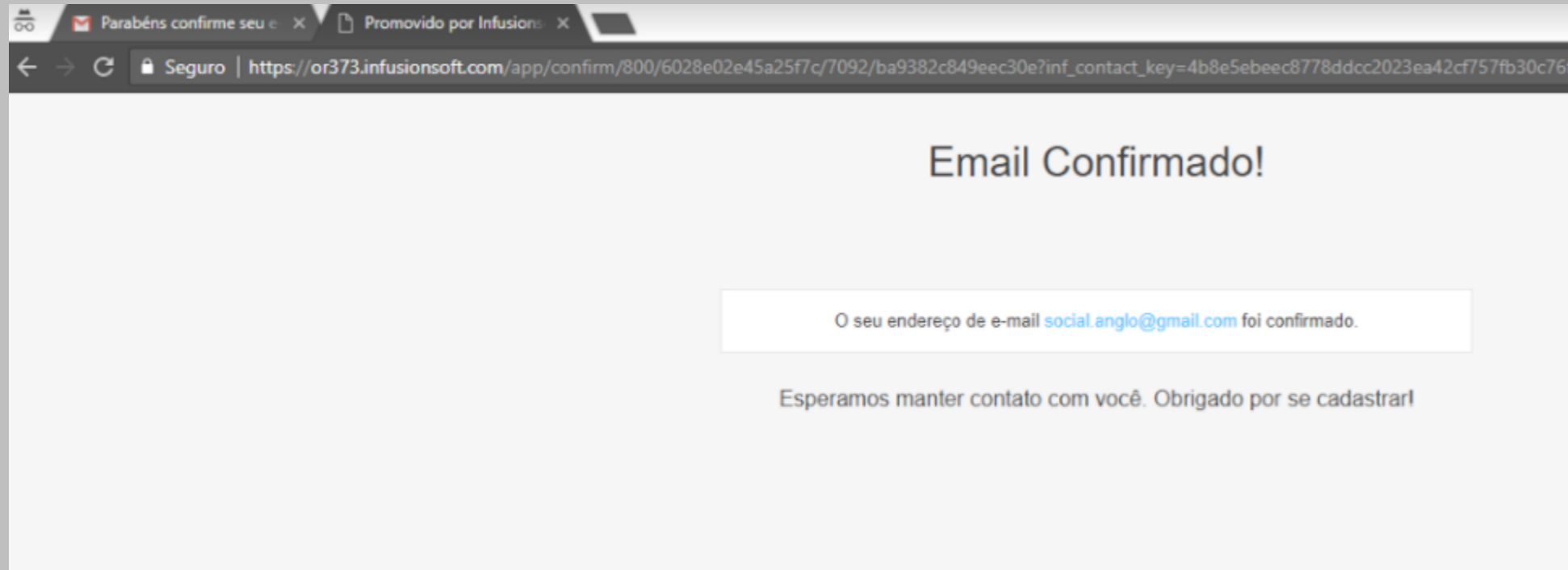
Obrigado,

 **Fernando Koyanagi**  
contato@fernandok.com

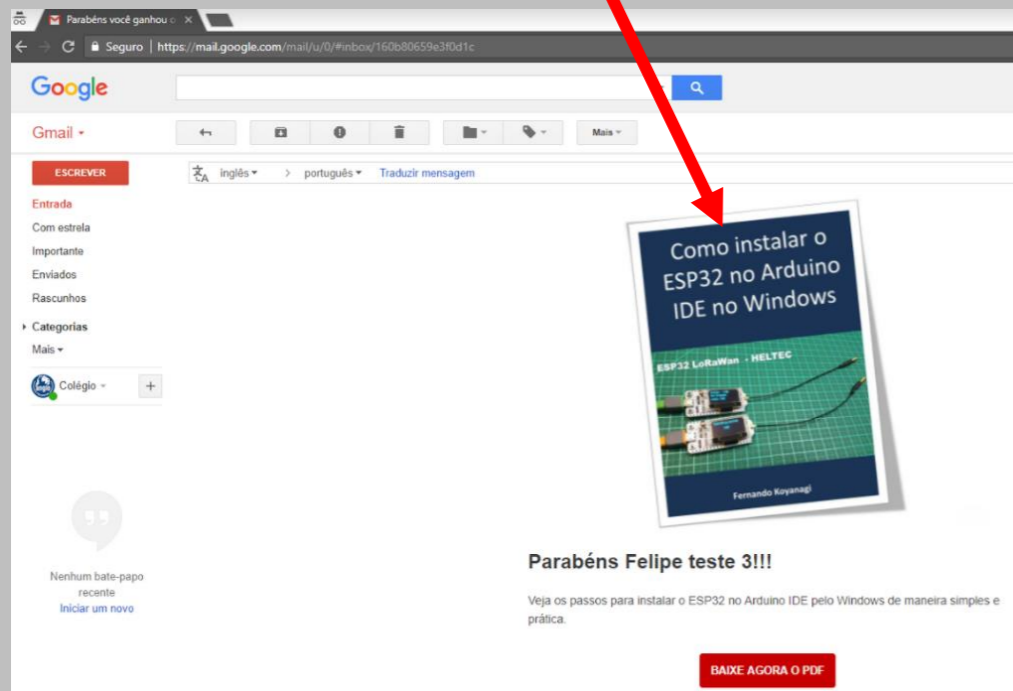
FernandoK Tecnologia

Se você não deseja mais receber nossos e-mails, clique no link abaixo:  
[Descadastrar](#)

# Em 3 minutos você vai receber **4º Passo**



# Em 3 minutos você vai receber **5º Passo**



# Sender e Receiver

**OLED\_LoRa\_Reciver.ino**

**OLED\_LoRa\_Sender.ino**

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

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

#define BAND 915E6 //Frequencia do radio - podemos utilizar ainda : 433E6,
868E6, 915E6
#define PABOOST true
```



## Sender : OLED\_LoRa\_Sender.ino

```
//variável responsável por armazenar o valor do contador  
(enviaremos esse valor para o outro Lora)  
unsigned int counter = 0;  
  
//parametros: address,SDA,SCL  
SSD1306 display(0x3c, 4, 15); //construtor do objeto que  
controlaremos o display  
  
String rssi = "RSSI --";  
String packSize = "--";  
String packet ;
```

# Setup : **OLED\_LoRa\_Sender.ino**

```
void setup()
{
//configura os pinos como saida
pinMode(16,OUTPUT); //RST do oled
pinMode(25,OUTPUT);
digitalWrite(16, LOW); // reseta o OLED
delay(50);
digitalWrite(16, HIGH); // enquanto o OLED estiver ligado, GPIO16 deve estar HIGH

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

//imprime o logo na tela
logo();

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

## Setup ( continuação ) : **OLED\_LoRa\_Sender.ino**

```
SPI.begin(SCK,MISO,MOSI,SS); //inicia a comunicação serial com o Lora
LoRa.setPins(SS,RST,DI00); //configura os pinos que serão utilizados
pela biblioteca (deve ser chamado antes do LoRa.begin)
//inicializa o Lora com a frequencia específica.
if (!LoRa.begin(BAND,PABOOST))
{
display.drawString(0, 0, "Starting LoRa failed!");
display.display(); //mostra o conteúdo na tela
while (1);
}
//indica no display que inicilizou corretamente.
display.drawString(0, 0, "LoRa Initial success!");
display.display(); //mostra o conteúdo na tela
delay(1000);
}
```

## Loop: **OLED\_LoRa\_Sender.ino**

```
void loop()
{
  //apaga o conteúdo do display
  display.clear();
  display.setTextAlignment(TEXT_ALIGN_LEFT);
  display.setFont(ArialMT_Plain_16);
  display.drawString(0, 0, "Sending packet: ");
  display.drawString(40, 26, String(counter));
  display.display(); //mostra o conteúdo na tela

  //beginPacket : abre um pacote para adicionarmos os dados para envio
  LoRa.beginPacket();
  //print: adiciona os dados no pacote
  LoRa.print("hello ");
  LoRa.print(counter);
  //endPacket : fecha o pacote e envia
  LoRa.endPacket(); //retorno= 1:sucesso | 0: falha
```

## Loop (continuação): **OLED\_LoRa\_Sender.ino**

```
//beginPacket : abre um pacote para adicionarmos os dados para envio
LoRa.beginPacket();
//print: adiciona os dados no pacote
LoRa.print("hello ");
LoRa.print(counter);
//endPacket : fecha o pacote e envia
LoRa.endPacket(); //retorno= 1:sucesso | 0: falha

//incrementa uma unidade no contador
counter++;

digitalWrite(25, HIGH); // liga o LED indicativo
delay(500); // aguarda 500ms
digitalWrite(25, LOW); // desliga o LED indicativo
delay(500); // aguarda 500ms
}
```

# Imprime - Logotipo: **OLED\_LoRa\_Sender.ino**

```
//essa função apenas imprime o logo na tela do display
void logo()
{
//apaga o conteúdo do display
display.clear();
//imprime o logo presente na biblioteca "images.h"
display.drawXbm(0,5,logo_width,logo_height,logo_bits);
display.display();
}
```

```
#define logo_width 128
#define logo_height 53
static char logo_bits[] = {
0x00, 0x00, 0x38, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x70, 0x00, 0xF8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xE0, 0x01, 0xF0, 0x03,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0xC0, 0x07, 0xF0, 0x07, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xC0, 0x0F, 0xF0, 0x0F, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xC0, 0x1F, 0xF0, 0x1F,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0xE0, 0x1F, 0xF0, 0x1F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xE0, 0x3F, 0xF8, 0x3F, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xE0, 0x1F, 0xF8, 0x3F,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0xE0, 0x1F, 0xF8, 0x3F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0xF0, 0x1F, 0xF8, 0x1F, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xF0, 0x1F, 0xFC, 0x1F,
0x80, 0xFF, 0x8F, 0x7F, 0xC0, 0xFF, 0x7F, 0xC0, 0xFF, 0x03, 0xC0, 0x3F,
0xF0, 0x1F, 0xFC, 0x1F, 0xE0, 0xFF, 0x87, 0x3F, 0xC0, 0xFF, 0x7F, 0xF0,
```



#include "images.h"

## Setup : **OLED\_LoRa\_Reciver.ino**

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

//indica no display que inicilizou corretamente.
display.drawString(0, 0, "LoRa Initial success!");
display.drawString(0, 10, "Wait for incomm data...");
display.display();
delay(1000);

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

## Setup : **OLED\_LoRa\_Reciver.ino**

```
void loop() {  
    //parsePacket: checa se um pacote foi recebido  
    //retorno: tamanho do pacote em bytes. Se retornar 0 (ZERO) nenhum  
    pacote foi recebido  
    int packetSize = LoRa.parsePacket();  
    //caso tenha recebido pacote chama a função para configurar os dados que  
    serão mostrados em tela  
    if (packetSize) {  
        cbk(packetSize); //função responsável por recuperar o conteúdo do  
        pacote recebido  
        delay(10);  
    }  
}
```





## Print no OLED :

### **OLED\_LoRa\_Receiver.ino**

```
//função responsável por configurar os dados que serão exibidos em tela.  
//RSSI : primeira linha  
//RX packSize : segunda linha  
//packet : terceira linha  
void loraData(){  
  display.clear();  
  display.setTextAlignment(TEXT_ALIGN_LEFT);  
  display.setFont(ArialMT_Plain_16);  
  display.drawString(0 , 18 , "Rx "+ packSize + " bytes");  
  display.drawStringMaxWidth(0 , 39 , 128, packet);  
  display.drawString(0, 0, rssi);  
  display.display();  
}
```

Em [www.fernandok.com](http://www.fernandok.com)

Download arquivo **PDF** dos diagramas  
Download arquivo **INO** do código fonte

