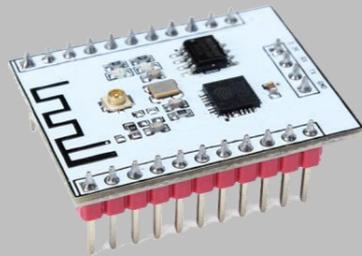
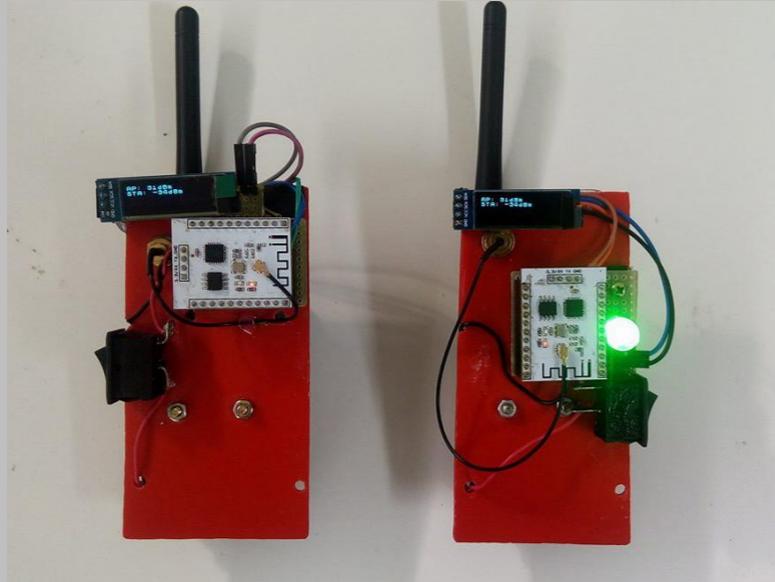


Controle remoto com ESP8266-201

- Medindo sinal



Access Point



Station



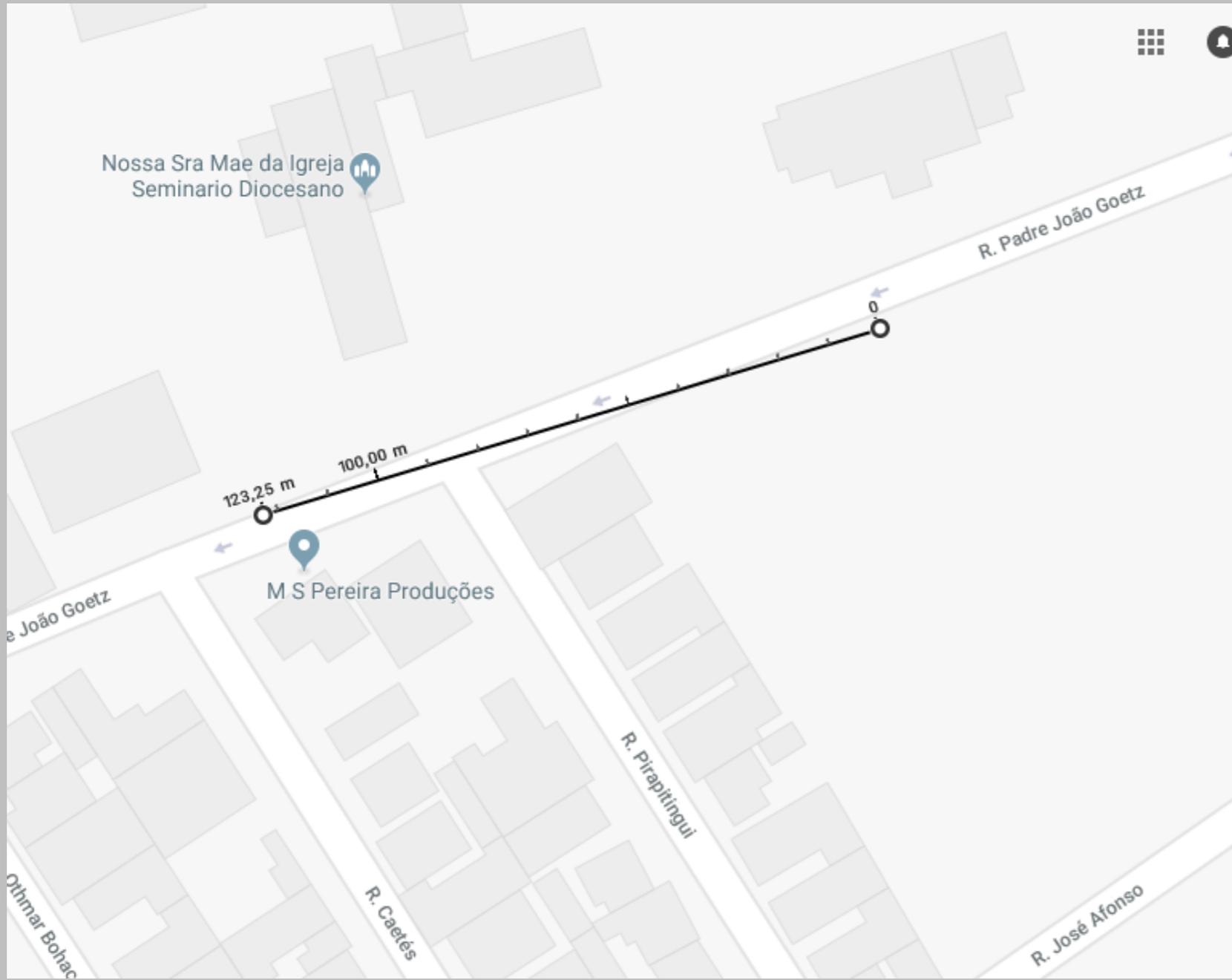
Por Fernando Koyanagi

Intenção dessa aula

- 1. Tornar ESP201 um Access Point.**
- 2. Comunicar remotamente ESP8266-201 client com servidor e medir a força de sinal entre eles(dbm).**

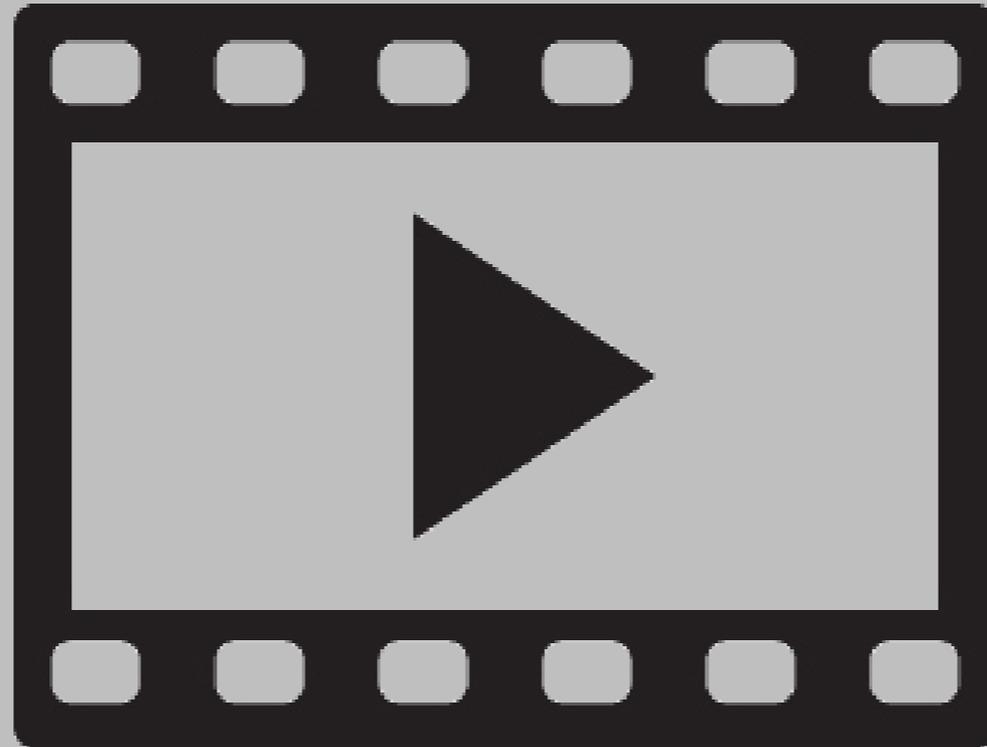


Distância medida



Aprox. 120 metros
Sinal de -96dbm

Demonstração





Em www.fernandok.com

- PRINCIPAL
- SOBRE FERNANDO K
- ARDUINO
- ESP8266
- ESP32
- LORAWAN
- MOTOR
- DISPLAY
- MATERIAIS
- DOWNLOAD

Receba o meu conteúdo GRATUITAMENTE

QUERO RECEBER GRÁTIS



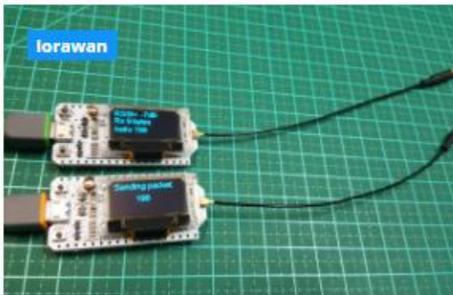
Seu e-mail



Motor de Passo Nema 23 com Driver TB6600 e Arduino Due

by Fernando K Tecnologia - 2:44 PM
Hoje vamos voltar a falar de Motor de Passo. Vamos utilizar um Nema 23 que será controlado por um Driver TB6600 e um Arduino Due. É p...

Leia mais



ESP32 Longa Distância - LoRaWan

by Fernando K Tecnologia - 9:46 AM
Neste artigo vamos tratar da LoRaWAN, uma rede que vai longe gastando pouca energia. Mas, o quanto "longe"? Com o chip que uso no vídeo...

Leia mais



Motor de HD com Arduino

by Fernando K Tecnologia - 2:00 PM

QUAL ASSUNTO VOCÊ TEM

- Arduino
- ESP8266
- ESP32
- Motor
- Display
- Sensor

You may select multiple answers.
Votar Exibir resultados

Votos até o momento: 32
Dias restantes para votar: 49

FACEBOOK



Recursos usados

- 2 ESP 8266-201;
- 2 Displays Oled;
- 2 Antenas para modulo wifi do ESP;
- 2 botões;
- 2 porta pilhas;
- 4 pilhas 8800mA;
- Jumpers.



Código Servidor (Access Point)

Código Server - Includes e configurações

```
#include <ESP8266WiFi.h> //lib para conectar o wifi do ESP201
#include <ESP8266WiFiMulti.h> //lib para as funções addAP e run
#include <U8x8lib.h> //lib usada para comunicação com o display
#include <SPI.h> //protocolo síncrono de dados serial

//SCL (ou serial clock) indica o sinal do clock, que é enviado do servidor para todos os clientes.
// Todos os sinais SPI são síncronos com este sinal
//SDA (ou serial data) é o pino que efetivamente transfere os dados
//O 3º parameto deve ser um inteiro, ele nada mais é do que o pino do reset.
//'U8X8_PIN_NONE' é usado por que este pino não está conectado
U8X8_SSD1306_128X32_UNIVISION_SW_I2C u8x8(SCL, SDA, U8X8_PIN_NONE);

//ip da rede local
IPAddress local_IP(192, 168, 10, 11);
//gateway da rede local
IPAddress gateway(192, 168, 10, 10);
//subnet é uma divisão lógica da rede local
IPAddress subnet(255, 255, 255, 0);

//define servidor na porta 5000 do protocolo TCP
WiFiServer servidor(5000);
```

Código Server - Setup

```
void setup()
{
  //seta o pino do led
  pinMode(13, OUTPUT);
  //inicializa display
  u8x8.begin();
  //desativa o modo de economia de energia do display
  u8x8.setPowerSave(0);
  //configura a fonte do texto que será exibido
  u8x8.setFont(u8x8_font_chroma48medium8_r);

  //aguarda 5 segundos
  delay(5000);

  //escreve no display "Definindo modo"
  escreva("Definindo modo", true, 1000, false);

  //configura modo como access point
  WiFi.mode(WIFI_AP);

  //escreve no display "Pronto"
  escreva("Pronto!", false, 1000, true);
}
```

Código Server - Setup

```
//escreve no display "Definindo AP"
escreva("Definindo AP", true, 1000, false);

//enquanto a interface de rede do access point não for configurada, exibe "."
while (!(WiFi.softAPConfig(local_IP, gateway, subnet)))
    escreva(".", false, 100, false);

//escreve no display "Definindo AP"
escreva("Pronto!", false, 1000, true);

//escreve no display "Definindo rede"
escreva("Definindo rede", true, 1000, false);

//enquanto a rede não for definida, escreve "."
//parametros: WiFi.softAP(nomeDoAccessPoint, senhaRede, canal, redeVisivel)
//canal: canal usado pelo ESP, pode ser do 1 ao 11
//redeVisivel: a rede pode ou não aparecer para outros serviços
while (!(WiFi.softAP("WiFi_Range", "12341234", 6, false)))
    escreva(".", false, 100, false);

//escreve no display "Definindo AP"
escreva("Pronto!", false, 1000, true);
```

Código Server - Setup

```
//escreve no display "Endereco IP"
escreva("Endereco IP", true, 1000, false);

//escreve no display o ip do servidor
escreva(WiFi.softAPIP().toString(), false, 1000, true);

//escreve no display "Iniciando"
escreva("Iniciando", true, 10, false);

//escreve no display "servidor"
escreva("servidor...", true, 1000, false);

//inicializa servidor
servidor.begin();

//define servidor para trabalhar sem delay
//caso seja setado como true, tem o intuito de reduzir o tráfego TCP / IP de pequenos pacotes
servidor.setNoDelay(true);

//escreve no display "Pronto!"
escreva("Pronto!", false, 1000, true);

//escreve no display "Aguardando STA"
escreva("Aguardando STA", true, 1000, false);
}
```

Código Server - Loop

```
void loop()
{
  //enquanto a quantidade de estações conectadas no servidor for zero
  //ou seja, ninguém conectou ainda
  while (WiFi.softAPgetStationNum() == 0)
  {
    //aguarda 1 segundo
    delay(1000);

    //se alguém conectou
    if (WiFi.softAPgetStationNum() != 0)
    {
      //exibe mensagem
      escreva("STA conectada!", true, 1000, false);
    }
  }

  //obtem o cliente que está conectado ao servidor
  WiFiClient client = servidor.available();
}
```

Código Server - Loop

```
if (client)
{
  //limpa tela do display
  u8x8.clear();
  //acende o led
  digitalWrite(13, HIGH);
  //enquanto o cliente estiver conectado
  while (client.connected())
  {
    //se houver dados, leia
    if (client.available())
    {
      //recebe dados do cliente
      String dbmSTA = client.readStringUntil('\n');
      //exibe a força do sinal da estação
      escreva("STA: " + dbmSTA, true, 1000, true);
      //espera até que todos os caracteres de saída no buffer tenham sido enviados
      client.flush();
    }
  }
  //aguarda envio
  delay(1);
  // fecha conexão
  client.stop();
  //apaga led
  digitalWrite(13, LOW);
}
```

Código Server - Loop

```
//aguarda envio
  delay(1);

  // fecha conexão
  client.stop();
  //apaga led
  digitalWrite(13, LOW);
}
}
```

Código Server - Função “escreva”

```
//escreve no display de acordo com os parametros
//nl = pular linha
//intervalo = tempo enviado para o delay
//limpar = limpa a tela do display
void escreva(String texto, boolean nl, int intervalo, boolean limpar)
{
    //se deseja pular linha no display, então pula
    if (nl)
        u8x8.println(texto);
    else //se não, exibe texto sem pular linha
        u8x8.print(texto);

    //aguarda o intervalo passado por parametro
    delay(intervalo);

    //se deseja limpar o display, então limpa
    if (limpar)
        u8x8.clear();
}
```

Código Cliente (Estação)

Código Client- Includes e configurações

```
#include <ESP8266WiFi.h> //lib para conectar o wifi do ESP201
#include <ESP8266WiFiMulti.h> //lib para as funções addAP e run
#include <U8x8lib.h> //lib usada para comunicação com o display
#include <SPI.h> //protocolo síncrono de dados serial

//SCL (ou serial clock) indica o sinal do clock, que é enviado do servidor para todos os clientes.
// Todos os sinais SPI são síncronos com este sinal
//SDA (ou serial data) é o pino que efetivamente transfere os dados
//O 3º parameto deve ser um inteiro, ele nada mais é do que o pino do reset.
//'U8X8_PIN_NONE' é usado por que este pino não está conectado
U8X8_SSD1306_128X32_UNIVISION_SW_I2C u8x8(SCL, SDA, U8X8_PIN_NONE);

//cria uma instância da classe ESP8266WiFiMulti
ESP8266WiFiMulti wifiMulti;

//Criar uma instância da classe IPAddress e define o ip do servidor
IPAddress local_IP(192, 168, 10, 110);
```

Código Client - Setup

```
void setup()
{
  //define pino Led
  pinMode(pinoLed,OUTPUT);

  //inicializa o display
  u8x8.begin();

  //desativa o modo de economia de energia do display
  u8x8.setPowerSave(0);

  //configura a fonte do texto que será exibido
  u8x8.setFont(u8x8_font_chroma48medium8_r);

  //aguarda 1 segundo
  delay(1000);

  //escreve no display "Definindo modo"
  escreva("Definindo modo", true, 1000, false);

  //configura modo como estação
  WiFi.mode(WIFI_STA);

  //escreve no display "Pronto!"
  escreva("Pronto!", false, 1000, true);
}
```

Código Client - Setup

```
//escreve no display "Definindo rede"
escreva("Definindo rede", true, 1000, false);

//parametros: WiFi.softAP(nomeDoAccessPoint, senhaRede)
//redeVisivel: a rede pode ou não aparecer para outros serviços
WiFiMulti.addAP("WiFi_Range", "12341234");

//escreve no display "Pronto!"
escreva("Pronto!", false, 1000, true);

//escreve no display "Conectando"
escreva("Conectando...", true, 1000, false);

//enquanto o cliente não estiver conectado, escreve "."
while (WiFiMulti.run() != WL_CONNECTED)
escreva(".", false, 1000, false);

//escreve no display "Pronto"
escreva("Pronto!", false, 1000, true);
```

Código Client - Setup

```
//escreve no display "Endereco IP:"  
escreva("Endereco IP:", true, 1000, false);  
  
//escreve no display o ip local  
escreva(WiFi.localIP().toString(), false, 1000, true);  
  
//escreve no display "Conectando"  
escreva("Conectando",true,10,false);  
  
//escreve no display "com servidor..."  
escreva("com servidor...",true,10,false);  
}
```

Código Client - Loop

```
void loop()
{
  //porta 5000 do protocolo TCP, deve ser a mesma utilizada pelo servidor
  const uint16_t port = 5000;
  //endereço ip, deve ser o mesmo utilizado pelo servidor
  const char * host = "192.168.10.11";

  //inicializa a lib do cliente
  WiFiClient client;

  //se o cliente não estiver conectado, exibe "Falha..."
  if (!client.connect(host, port))
  {
    escreva("Falha...",true, 1000,true);
    return;
  }
}
```

Código Client - Loop

```
//obtem a força do sinal da conexão entre os ESPs
String _RSSI = String(WiFi.RSSI());

//acende led
digitalWrite(pinoLed,HIGH);

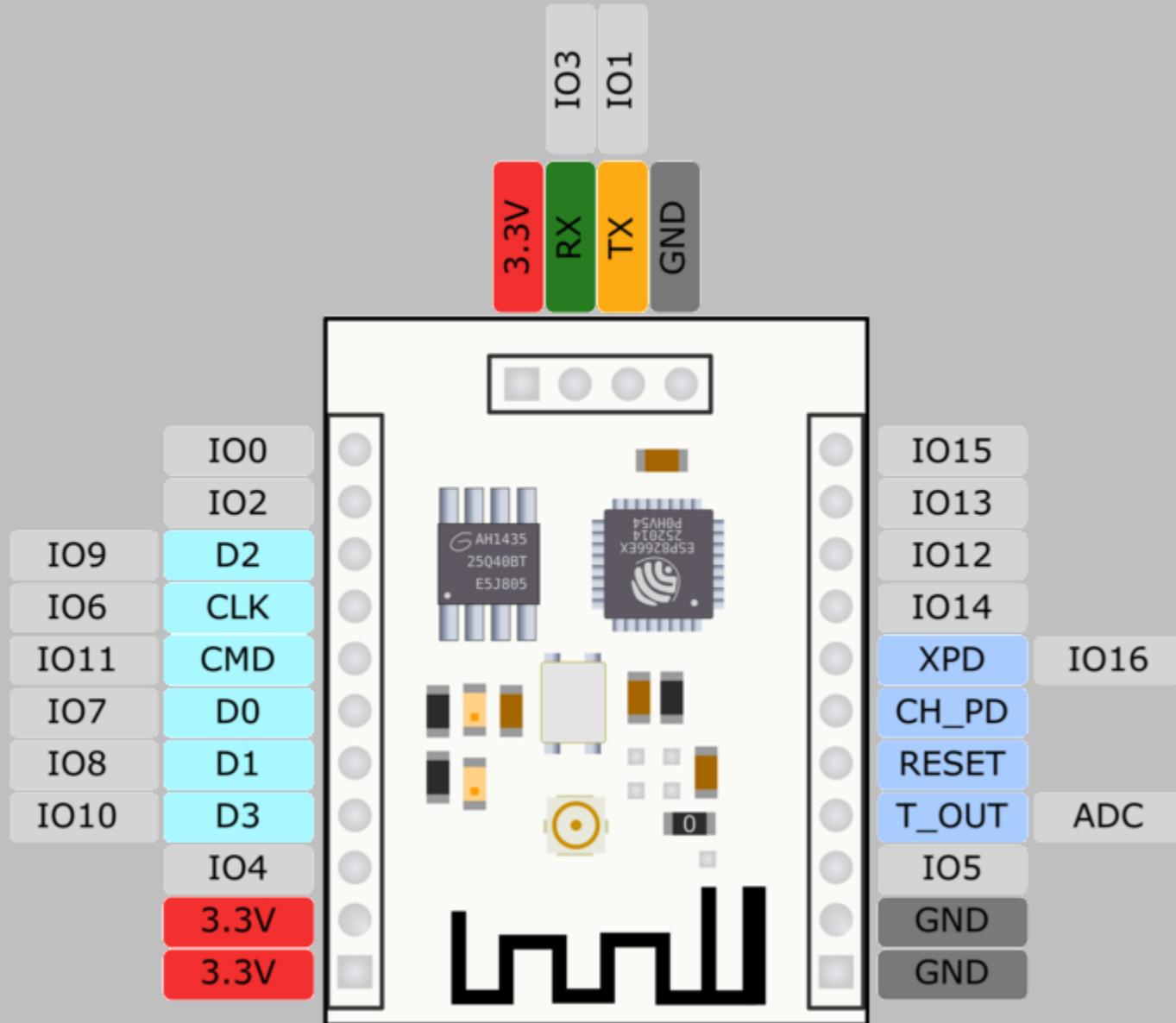
//transmite esta medida em dBm
client.println(_RSSI +"dBm");

//exibe a força do sinal da estação
escreva("STA: " + _RSSI+"dBm",true,1000,true);

//apaga led
digitalWrite(pinoLed,LOW);
}
```

***A função “escreva” é a mesma que a usada no código do servidor.**

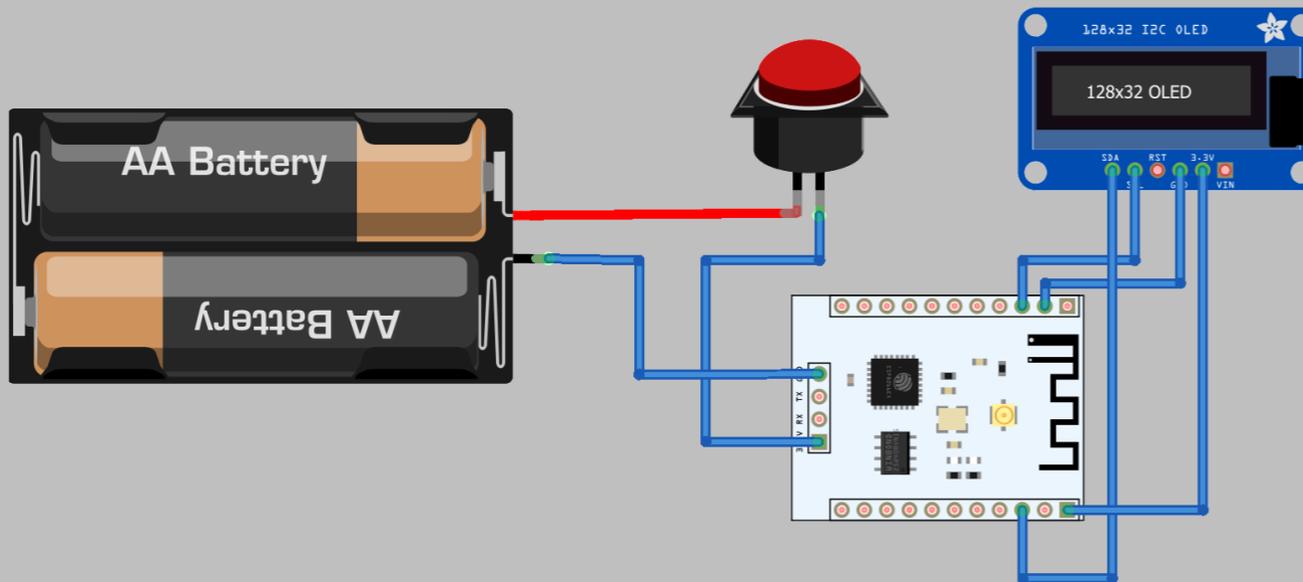
ESP8266-201 Pinout



Ligação de pinos

ESP	Oled
IO4	SDA
IO5	SCK
GND	GND
3,3V	VCC

Esta tabela é a mesma para o Server e Client.



Importante: Alimentar o pino chip_en do ESP com 3.3V

Em www.fernandok.com

Download arquivos PDF e **INO** do código fonte

