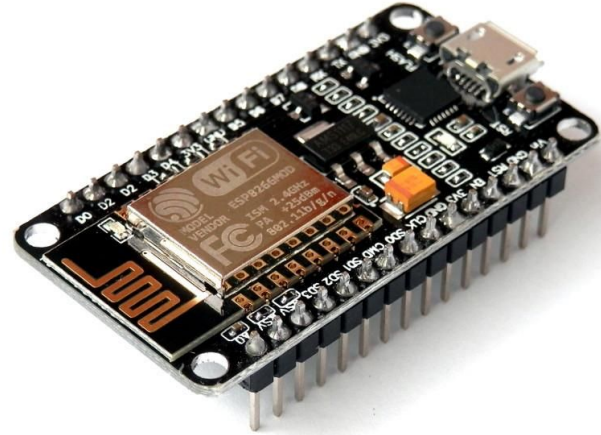


Minicurso

IoT

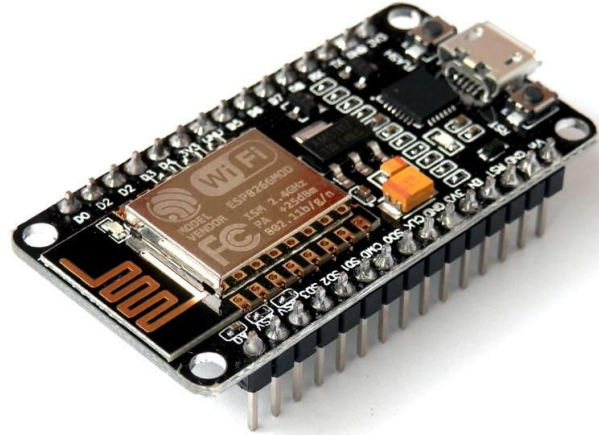
ESP8266

Aula 1



Jimmy PETECO

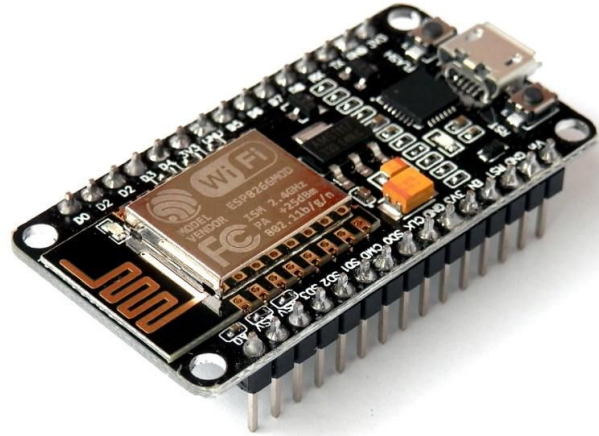
<https://github.com/jimmytsoares>
jimmy@alunos.utfpr.edu.br



ATENÇÃO!

Login: sem_RA

Senha: \$s4pp0\$



1.

Conteúdo

O que a gente vai aprender

- NodeMCU
- Instalando a biblioteca
- Piscar LED da placa
- Uso como ponto de acesso/servidor
- Uso do Blynk com o NodeMCU
- Uso do WiFiManager
- Integração com IFTTT

No final do minicurso (se a internet da UTF cooperar) vocês poderão acionar as portas do NodeMCU com a voz pela internet! Exemplo de aplicação:

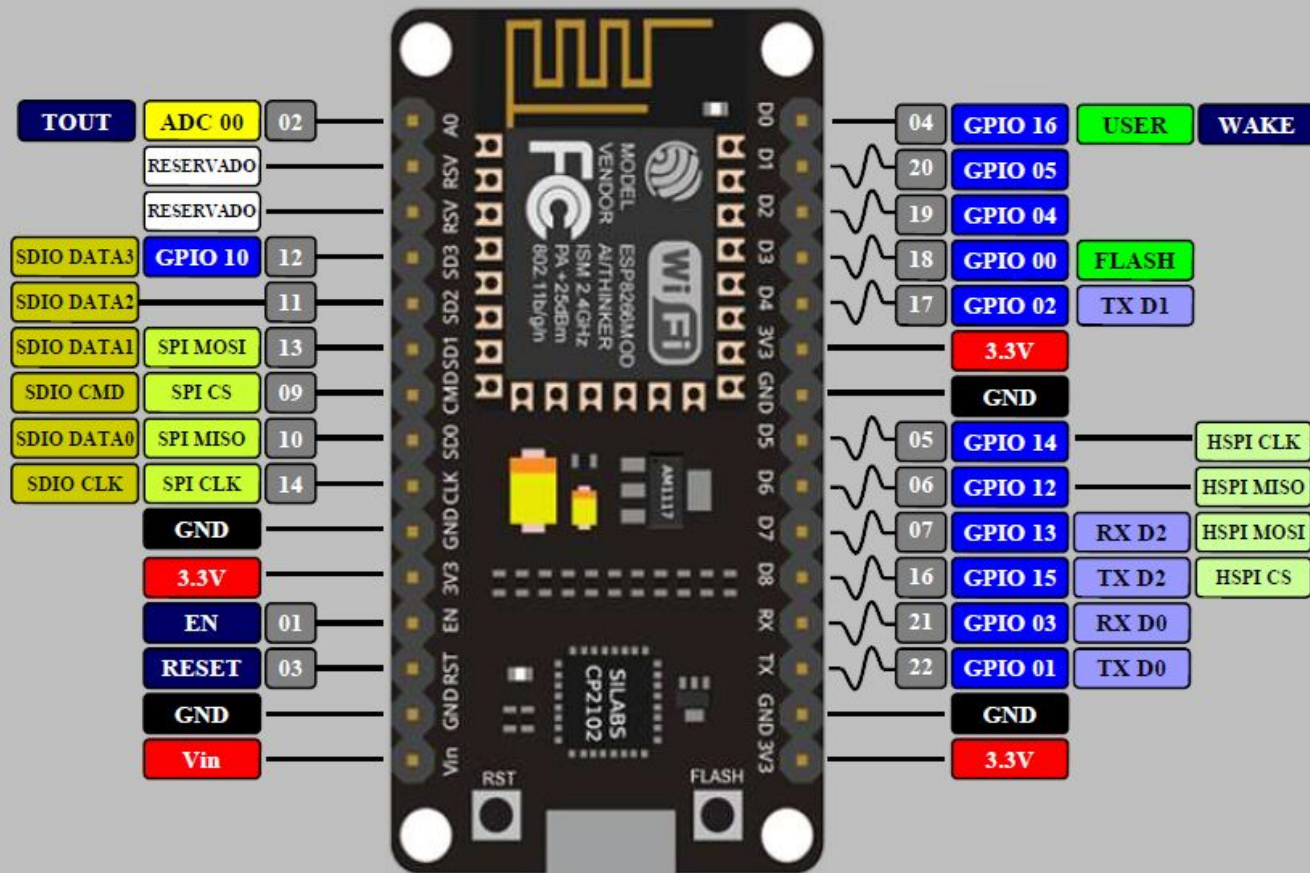
<https://youtu.be/XR6OoohzRPc>

2.

NodeMCU

A placa que vamos usar

- Placa de desenvolvimento
- Microcontrolador ESP8266
- WiFi integrado tamanho reduzindo
- Programável com o C do Arduino
- Tensão de operação **3,3V**



Pinos para prestar atenção



GPIO16

Possui a função WAKE. O ESP pode dormir por um tempo para economizar energia (muito importante caso utilize bateria na sua aplicação), para acordar ele utiliza esse pino conectado ao RESET.

A0

O ADC do ESP não é muito bom, embora tenha é aconselhado o uso de um expensor de portas analógicas.

GPIO 4 e 5

Utilizados para I2C.

GPIO4 SDA

GPIO5 SCL

3.

**Instalando a
biblioteca para
usar com Arduino
IDE**

1. Abrir a IDE do Arduino

2. File -> Preferences

3. Em *Additional Boards Manager URLs* adicionar o seguinte endereço:

http://arduino.esp8266.com/stable/package_esp8266com_index.json

Não retire se já tiver algo, apenas adicione no final separando com vírgula

4. Tools -> Board -> Board Manager...

5. Procurar por esp8266 e instalar a última versão (deve ser 2.4.0)

4.

Piscando LED

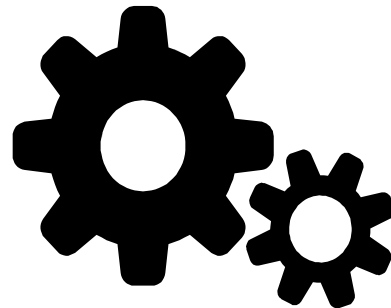
Só pra ver se ta funcionando

- 1. Abrir a IDE do Arduino**
- 2. Selecionar o NodeMCU 1.0 como a placa utilizada em Tools -> Board**
- 3. File ou Arduino -> Examples -> Examples for NodeMCU -> ESP8266 -> Blink**
- 4. V ->, se piscou, ok!**

5.

Uso como Access Point com Server

Você se conecta ao ESP



Nesse exemplo nós utilizamos a placa como um ponto de acesso direto (AP), ou seja, a placa cria uma rede local wireless (WLAN).

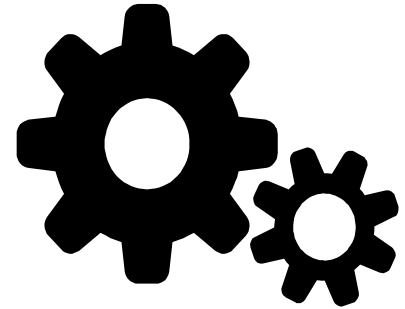
Dessa maneira será possível ver pelo celular a rede criada e se conectar nela, porém não é muito útil na maioria das aplicações.

- 1. Abrir a IDE do Arduino**
- 2. Selecionar o NodeMCU 1.0 como a placa utilizada em Tools -> Board**
- 3. File -> Examples -> Examples for NodeMCU -> ESP8266WiFi -> WiFiAccessPoint**
- 4. Mudar o SSID e Password para algo mais único, a senha deve ter no mínimo 8 caracteres para funcionar!**
- 5. V ->**
- 6. Acessar pelo celular, acessar 192.168.4.1 no navegador**
- 7. Se a mensagem apareceu, tudo certo!**

6.

Server + GPIO

Algo um pouco mais útil



Agora a placa vai se comportar como uma estação, irá conectar a uma rede e funcionar como um servidor que outros dispositivos da mesma rede conseguem utilizar.

A utilidade melhora um pouco pois uma placa configurada na sua rede local sempre poderá ser acessada se você tiver acesso a essa rede. Ou seja, se conectar na sua casa, sempre que sua WiFi de casa conectar, irá funcionar!

1. Abrir a IDE do Arduino
2. Selecionar o NodeMCU 1.0 como a placa utilizada em Tools -> Board
3. Pegar o código em <https://github.com/jimmytsoares/Minicurso-IoT-NodeMCU/>
4. Alterar a linha *if (MDNS.begin("esp8266"))* com outro nome para ser único (seu nome por exemplo)
5. V ->, acessar com <http://seunome.local>
6. Se funcionou, ok! Agora tente entender o código

“mDNS é um DNS multicast. Ele resolve nomes de máquinas para IP dentro de redes pequenas que não tem um servidor de nomes local. O funcionamento é bem simples: quando um cliente mDNS precisa resolver um nome, ele envia um pacote multicast perguntando quem tem aquele nome. A máquina que tem o nome pesquisado responde com um novo pacote multicast que inclui seu IP. Todas as máquinas do grupo multicast recebem a informação e a armazenam em seus caches.”

Exercício

- Utilizar o exemplo dado para criar um webserver com 3 páginas
- A primeira, o root (“/”) deve apenas mostrar a mensagem “Minicurso IoT”
- A segunda, em /LED, deve mostrar o número de acessos na página e acender ou apagar o LED se o número de acessos for par ou ímpar
A página deve mostrar o seguinte:
“Numero de acessos: 2
Estado do LED: Desligado”
- A terceira para qualquer outra coisa, o 404



Dicas:

Mudar para “text/plain”
dentro do server.send
para poder mandar
qualquer string

Mudar em server.on para
HTTP_GET para
conseguir acessar a
pagina /LED