

# Introdução

NodeMCU

Fabiano  
Sardenberg  
Kuss

Introdução

A plataforma NodeMCU é uma placa open source baseada nas funcionalidade providas pelo chip de baixo custo com suporte a redes sem fio 802.11 ESP8266 que utiliza o microprocessador Xtensa.

Esta plataforma oferece um ambiente adequado para o desenvolvimento de dispositivos que implementam funcionalidades para atuarem em um conceito de IoT de forma simples. Pode ser visto como uma evolução da estratégia de desenvolvimento utilizando Arduino

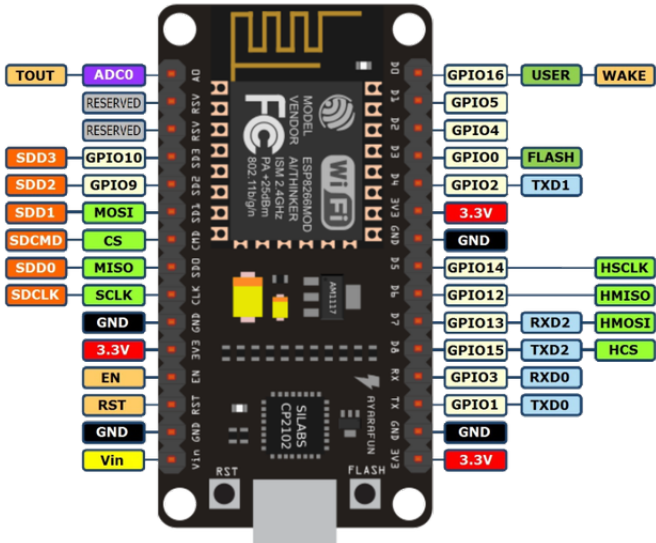


# Apresentando a Placa

NodeMCU

Fabiano  
Sardenberg  
Kuss

Introdução



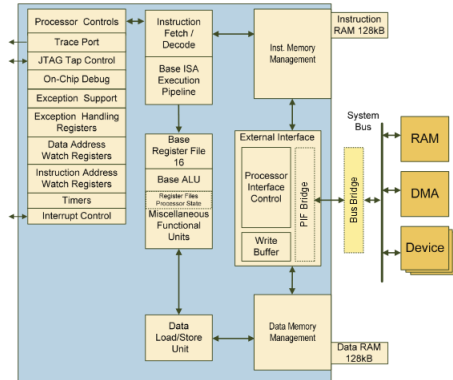
# ESP8266

NodeMCU

Fabiano  
Sardenberg  
Kuss

Introdução

- Microcontrolador: Xtensa L106 (32-bit) 80Mhz
- Memória Interna: 128K para instruções; 128K para dados
- Memória Flash: 4Mb.
- I/O: 16 Pinos GPIO
- Tensão: 3.3 VDC
- WI-Fi: 802.11 b/g/n



The Diamond Standard 106Micro

# Sintaxe básica

NodeMCU

Fabiano  
Sardenberg  
Kuss

Introdução

Referência da linguagem:

<https://www.lua.org/manual/5.2/pt/manual.htm>

Algumas características:

- Dinamicamente tipada
- Variáveis sempre globais
- Tipo table para qualquer tipo de array

# Sintaxe básica

NodeMCU

Fabiano  
Sardenberg  
Kuss

Introdução

## Operadores:

- + adição
- - subtração
- \* multiplicação
- / divisão
- % módulo

## Operadores Lógicos:

- == comparador igualdade
- != comparador desigualdade
- or
- and
- false
- true
- nil Valor nulo

# Sintaxe básica

NodeMCU

Fabiano  
Sardenberg  
Kuss

Introdução

Algumas funções relevantes:

- `tonumber(var)`
- `strlen(str)`
- `type(var)` –Retorna o tipo da variável
- `return`

Criando uma função:

```
f = function () corpo end
```

Utiliza o operador : para manter o controle da operação utilizando uma sintaxe mais simples.

— Sem arquivo fecha ísada ãpadro

```
o.close ([file])  
file:close ([file])
```

—Intera sobre as linhas

```
io.input():lines()
```

Pode ser utilizado para socket como por exemplo

```
srv=net.createServer(net.TCP)  
srv:listen()
```

# Programando em Lua

NodeMCU

Fabiano  
Sardenberg  
Kuss

Introdução

Download da ferramenta que permite comunicação serial usando Python

```
git clone https://github.com/pyserial/pyserial.git
```

Download da ferramenta de flash esptool.py em

```
git clone https://github.com/espressif/esptool
```

Copiar a pasta serial de pyserial-2.7

```
cp -r pyserial/pyserial-2.7/serial para a pasta esptool/
```

Compilar o próprio módulo ou fazer download de um módulo pronto de

<https://github.com/nodemcu/nodemcu-firmware/releases> Para a compilação será necessário

Copiar o firmware para o dispositivo

```
./esptool.py -p /dev/ttyUSB0 write_flash 0x00000 \n~/Downloads/nodemcu_float_0.9.6-dev_20150704.bin
```



# Oi mundo

NodeMCU

Fabiano  
Sardenberg  
Kuss

Introdução

## Escrever um oi mundo em lua

```
while true do
  print("Oi_Mundo")
  tmr.delay(1000 * 1000)
end
```

## Copiar o programa para o dispositivo

```
./luatool.py --port /dev/ttyUSB0 --src /tmp/script.lua --dofile
```

# Comunicando com a serial

NodeMCU

Fabiano  
Sardenberg  
Kuss

Introdução

Para comunicar com a porta serial para visualizar o resultado é possível utilizar um script python lembrando que a pasta serial tem que estar no mesmo diretório que o script

```
import serial
ser = serial.Serial('/dev/ttyUSB0', 9600, timeout=1)
for i in range(0,10):
    print ser.readline()
```

# Conectar em um AP

NodeMCU

Fabiano  
Sardenberg  
Kuss

Introdução

## Criando uma aplicação conectada ao Access Point

```
wifi.setmode(wifi.STATION)
wifi.sta.config("SSID","senha")

ip, nm, gw=wifi.sta.getip()

while true do
  print("\nIP_Info:\nIP_Address:_" .. ip .. "_" .. "\nNetmask:_" .. nm .. "\nGateway_Addr:_" .. gw .. "\n")
  tmr.delay(2000*1000)
end
```

# Requisições HTTP

NodeMCU

Fabiano  
Sardenberg  
Kuss

Introdução

## Atendendo requisições HTTP

```
srv=net.createServer(net.TCP)
srv:listen(80,function(conn)
  conn:on("receive",function(conn,payload)
    print(payload)
    conn:send("<h1>OLA_ALUNO_DO_MINICURSO</h1>")
  end)
  conn:on("sent",function(conn) conn:close() end)
end)
```

## Aplicação completa

NodeMCU

## Introdução

# WEB e LED

```
LED_PIN = 1

gpio.mode(LED_PIN, gpio.OUTPUT)

wifi.setmode(wifi.STATION)
wifi.sta.config("pereskuss", "meloquita")

ip, nm, gw=wifi.sta.getip()

print("\nIP_Info:\nIP_Address:_"..ip.."_"\nNetmask:_"..nm.."_"\nGateway_Addr:_"..gw.."_"\n")
tmr.delay(2000*1000)

srv=net.createServer(net.TCP)
srv:listen(80,function(conn)
    conn:on("receive",function(conn,payload)
        print(payload)
        conn:send("<h1>OLA_ALUNO_DO_MINICURSO</h1>")
        gpio.write(LED_PIN, gpio.HIGH)
        tmr.delay(2000*1000)
        gpio.write(LED_PIN, gpio.LOW)
    end)
    conn:on("sent",function(conn) conn:close() end)
end)
```

# Aplicação completa (Continuação)

NodeMCU

Fabiano  
Sardenberg  
Kuss

Introdução

```
get={}

if string.find(valor , "GET") == 1 then
  print("OK")
end

for w in string.gfind (valor , "%S+") do
  table.insert(get, w)
end

qstring = get[2]

init=string.find(qstring , "?")

substr = string.sub(qstring , init+1, string.len(qstring))

for w, k in string.gfind(qstring , '([^&=?]-)([^\&=?]+)' ) do
  print(w, k)
end
```

# Software Libre

NodeMCU

## Introdução

```
LED_PIN = 1

gpio.mode(LED_PIN, gpio.OUTPUT)

wifi.setmode(wifi.STATION)
wifi.sta.config("pereskuss","meloquita")

ip, nm, gw=wifi.sta.getip()

print("\nIP_Info:\nIP_Address: " .. ip .. "\nNetmask: " .. nm .. "\nGateway_Addr: " .. gw .. "\n")
tmr.delay(2000*1000)

srv=net.createServer(net.TCP)
srv:listen(80,function(conn)
    conn:on("receive",function(conn,payload)
        print("PROCESS_QS")
        for w, k in string.gfind(payload, '([^&=?]-)=([^&=?]+)' ) do print(w, k) end
        conn:send("<h1>OLA_ALUNO_DO_MINICURSO</h1>")

        gpio.write(LED_PIN, gpio.HIGH)
        tmr.delay(2000*1000)
        gpio.write(LED_PIN, gpio.LOW)
    end)
    conn:on("sent",function(conn) conn:close() end)
end)
```