

ROTEIRO 00 - DEMONSTRAÇÃO

CUIDADO!

- O CI ESP8266 requer uma tensão de alimentação de 3,3 volts. Não deve ser alimentado com 5 volts como outras placas Arduino.
- A Placa NodeMCU ESP-12E dev pode ser conectada ao conector micro USB 5V ou pino Vin disponível na placa.
- Os pinos de E/S do ESP8266 se comunicam ou usam entrada/saída máxima de apenas 3,3 volts. Isto é, os pinos NÃO são entradas tolerantes de 5V.

Projeto *open hardware*:

<https://github.com/nodemcu/nodemcu-devkit>
https://www.nodemcu.com/index_en.html

Esquemáticos:

<http://fritzing.org/download/>
<https://github.com/squix78/esp8266-fritzing-parts> (nodemcu-v1.0/NodeMCUV1.0.fzpz)

Programação de firmware:

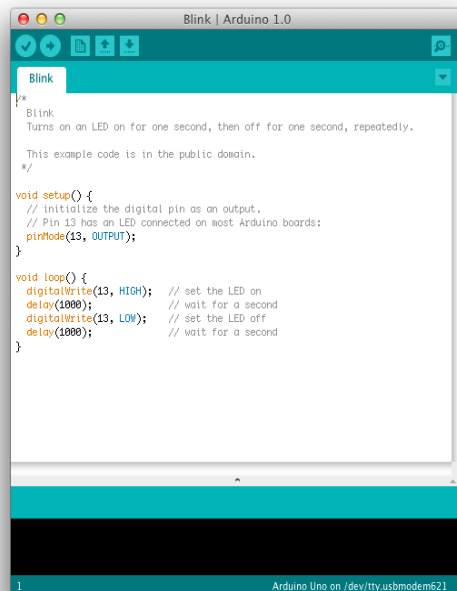
<https://www.arduino.cc/en/Main/Software> (*Download the Arduino IDE*)

Referência de pinos (pins_arduino.h, common.h):

```
// Entrada e saída digital (saída analógica PWM)
static const uint8_t D0 = 16; // LED_BUILTIN
static const uint8_t D1 = 5;
static const uint8_t D2 = 4;
static const uint8_t D3 = 0;
static const uint8_t D4 = 2;
static const uint8_t D5 = 14;
static const uint8_t D6 = 12;
static const uint8_t D7 = 13;
static const uint8_t D8 = 15;
static const uint8_t D9 = 3;
static const uint8_t D10 = 1;
// Entrada analógica
static const uint8_t A0 = 17;
```

Preparação

1. Instalar o Arduino IDE:



2. Garanta que seu sistema operacional tem o driver apropriado (<https://github.com/nodemcu/nodemcu-devkit/tree/master/Drivers>) para o chip CH341 (comunicação serial com a placa NodeMCU):

- Se você estiver no Windows, a instalação do driver é **imprescindível**.
- Se você estiver no Linux, as distribuições recentes já fazem a comunicação; entretanto você deve adicionar o seu usuário ao grupo dialout:

```
sudo usermod -a -G dialout $USER
sudo reboot
```

3. No Arduino IDE, instale as placas com ESP8266:

- No menu *Preferences*, coloque http://arduino.esp8266.com/stable/package_esp8266com_index.json no campo *Additional Board Manager URLs*.
- Em *Tools > Board ... > Boards Manager*, encontrar “*esp8266 by ESP8266 Community*” e instalar.
- Conecte a placa NodeMCU a uma porta USB do computador.
- De acordo com sua placa, fazer no menu *Tools* as configurações necessárias: escolha da *board*, e.g. NodeMCU 1.0 (ESP-12E Module); escolha da *upload speed*, e.g. 9600 (bauds); escolha da *port*, e.g. a que aparece quando você conecta sua placa à porta USB do micro.
- Entre em *File > New* e apague tudo do documento. Copie e cole o código a seguir:

```

#include "ESP8266WiFi.h"

void setup() {
  Serial.begin(9600);
  // Set WiFi to station mode and disconnect from an AP if it was previously connected
  WiFi.mode(WIFI_STA);
  WiFi.disconnect();
  delay(2000);
  Serial.println("Setup done");
}

void loop() {
  Serial.println("scan start");

  int n = WiFi.scanNetworks(); // WiFi.scanNetworks will return the number of networks found
  Serial.println("scan done");
  if (n == 0)
    Serial.println("no networks found");
  else
  {
    Serial.print(n);
    Serial.println(" networks found");
    for (int i = 0; i < n; ++i)
    {
      // Print SSID and RSSI for each network found
      Serial.print(i + 1);
      Serial.print(": ");
      Serial.print(WiFi.SSID(i));
      Serial.print(" (");
      Serial.print(WiFi.RSSI(i));
      Serial.print(")");
      Serial.println((WiFi.encryptionType(i) == ENC_TYPE_NONE) ? " ":"*");
      delay(10);
    }
  }
  Serial.println("");

  // Wait a bit before scanning again
  delay(5000);
}

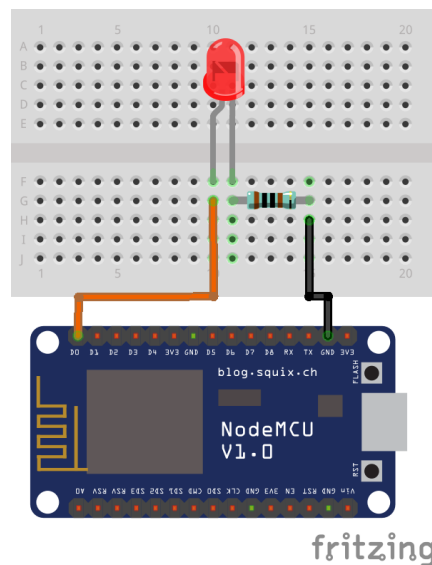
```

- Acesse *Tools > Serial Monitor*. Deixe a janela de lado por enquanto.
- Na janela principal, acesse o menu *Sketch*, acesse *Upload* (CTRL+U ou o ícone da “setinha”). Você estará compilando o código fonte do seu firmware e carregando a placa com o binário.
- Depois de transmitido o firmware, volte para a janela do *Serial Monitor*. Se estiver mostrando informações das redes Wi-Fi do entorno, sua IDE e sua placa estão prontas para seus projetos.

4. Monte o projeto a seguir.

Projeto 00 (demonstração)

Monte o seguinte circuito:



Copie e cole o código a seguir em um novo documento na sua IDE:

```
void setup() {  
  pinMode(D0, OUTPUT);    // Initialize the LED_BUILTIN pin as an output  
}  
  
// the loop function runs over and over again forever  
void loop() {  
  digitalWrite(D0, LOW);  // Turn the LED on (Note that LOW is the voltage level  
                          // but actually the LED is on; this is because  
                          // it is active low on the ESP-01)  
  delay(1000);            // Wait for a second  
  digitalWrite(D0, HIGH); // Turn the LED off by making the voltage HIGH  
  delay(2000);            // Wait for two seconds (to demonstrate the active low LED)  
}
```

Se depois do *Upload* o LED estiver piscando, tudo está correto. Tente mudar o pino do LED, ou mesmo alterar os valores no delay. Discuta a respeito.