Conectar o UNO ao NodeMCU

Via interface Serial. O Uno irá usar os pinos 2 e 3 para Serial e fará apenas um echo neste teste. Tudo que recebe envia de volta.

O codigo é

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(2,3); // rx e tx2 NodeMCU



// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 2 as an output.
  pinMode(13, OUTPUT);
  Serial.begin(115200);
  mySerial.begin(115200);

}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(13, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
  if ( mySerial.available() ) {
    mySerial.write(mySerial.read());
  }
```

Alem de receber e ecoar, fica piscando e o LED.

No caso do NodeMCU precisamos conecta-los. Para isto temos que usar um divisor de tensão com resistores pois o NodeMCU é 3.3 V e o UNO são 5V. Apenas da saida do pino 3 (tx) do UNO passa pelo divisor e conectar no Pino D7 (gpio13) RX2 do Node, O pino D8 (gpio 15) TX2 do Node conecta diretamente no pino 2 do UNO. O GNF deve ser conectado do UNO com o do NodeMCU se voce estiver usando computadores diferentes, se forem alimentados pelo mesmo USB nao precisa.

O codigo do Node é

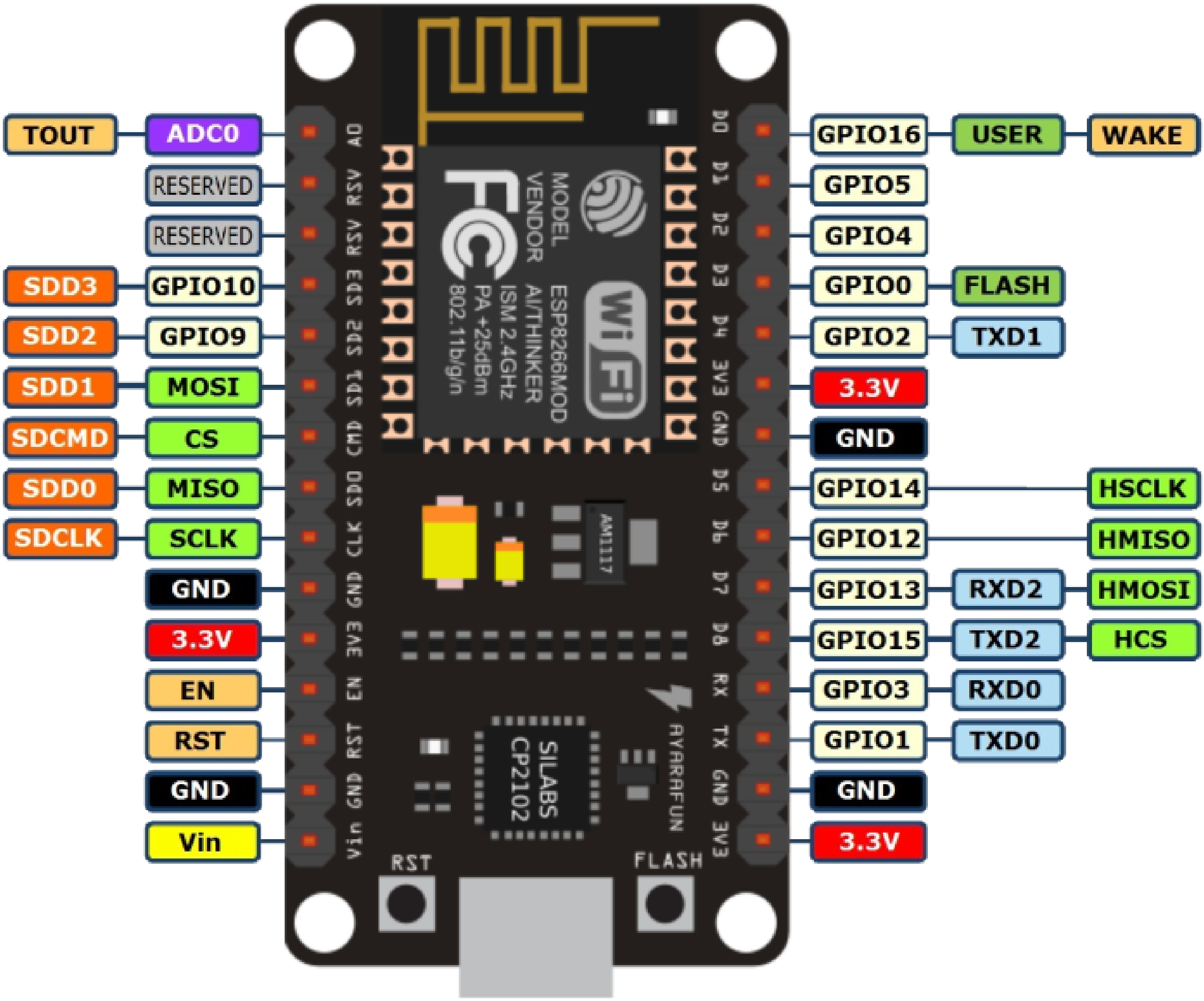
```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(13,15); // rx e tx2 NodeMCU



// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 2 as an output.
  pinMode(2, OUTPUT);
  Serial.begin(115200);
  mySerial.begin(115200);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(2, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(2, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
  if ( Serial.available() ) {
    mySerial.write(Serial.read());
  }
  if ( mySerial.available() ) {
    Serial.write(mySerial.read());
  }

}
```



----

assitir o video

#31 Internet of Things with ESP8266 #3: Interacting with your Smartphone

Toda serie é boa
tem o 33 que ensina watchdog e timers
https://www.youtube.com/watch?v=D_7ciW_TCac

Aqui tem a série completa
https://www.youtube.com/watch?v=NzJ2-siImC0&list=PL3XBzmAj53Rlu3Byy_GkqG6b-nwEpWku0

NodeMcu as AP
do blog http://arduino-er.blogspot.com.br/2016/05/nodemcuesp8266-act-as-ap-access-point_3.html

```
/*
 * NodeMCU/ESP8266 act as AP (Access Point) and simplest Web Server
 * to control GPIO (on-board LED)
 * Connect to AP "arduino-er", password = "password"
 * Open browser, visit 192.168.4.1
 */
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>

const char *ssid = "arduino-er";
const char *password = "password";
int stateLED = LOW;

ESP8266WebServer server(80);

void handleRoot() {
    response();
}

void handleLedOn() {
  stateLED = LOW;
  digitalWrite(LED_BUILTIN, stateLED);
  response();
}

void handleLedOff() {
  stateLED = HIGH;
  digitalWrite(LED_BUILTIN, stateLED);
  response();
}

const String HtmlHtml = "<html><head>"
    "<meta name=\"viewport\" content=\"width=device-width, initial-scale=1\" /></head>";
const String HtmlHtmlClose = "</html>";
const String HtmlTitle = "<h1>Arduino-er: ESP8266 AP WebServer exercise</h1><br/>\n";
const String HtmlLedStateLow = "<big>LED is now <b>ON</b></big><br/>\n";
const String HtmlLedStateHigh = "<big>LED is now <b>OFF</b></big><br/>\n";
const String HtmlButtons =
    "<a href=\"LEDOn\"><button style=\"display: block; width: 100%;\">ON</button></a><br/>"
    "<a href=\"LEDOff\"><button style=\"display: block; width: 100%;\">OFF</button></a><br/>";
```

```
void response(){
  String htmlRes = HtmlHtml + HtmlTitle;
  if(stateLED == LOW){
    htmlRes += HtmlLedStateLow;
  }else{
    htmlRes += HtmlLedStateHigh;
  }

  htmlRes += HtmlButtons;
  htmlRes += HtmlHtmlClose;

  server.send(200, "text/html", htmlRes);
}

void setup() {
    delay(1000);
    Serial.begin(9600);
    Serial.println();

    WiFi.softAP(ssid, password);

    IPAddress apip = WiFi.softAPIP();
    Serial.print("visit: \n");
    Serial.println(apip);
    server.on("/", handleRoot);
    server.on("/LEDOn", handleLedOn);
    server.on("/LEDOff", handleLedOff);
    server.begin();
    Serial.println("HTTP server beginned");
    pinMode(LED_BUILTIN, OUTPUT);
    digitalWrite(LED_BUILTIN, stateLED);
}

void loop() {
    server.handleClient();
}
```