

# IoT usando o ESP8266



Usando o módulo WiFi como uma *custom board* Arduino

*Daniel Junho*

# Tópicos

O que é a IoT?

Arduino

ESP8266

Arduino com placas customizadas

ESP8266 como Arduino

Bibliotecas

Conexão

Plataforma ThingSpeak

Exemplos

Dúvidas

---

**Internet das  
Coisas**

# O que é IoT?

Internet das Coisas

Conexão com dispositivos e uso dos dados gerados.

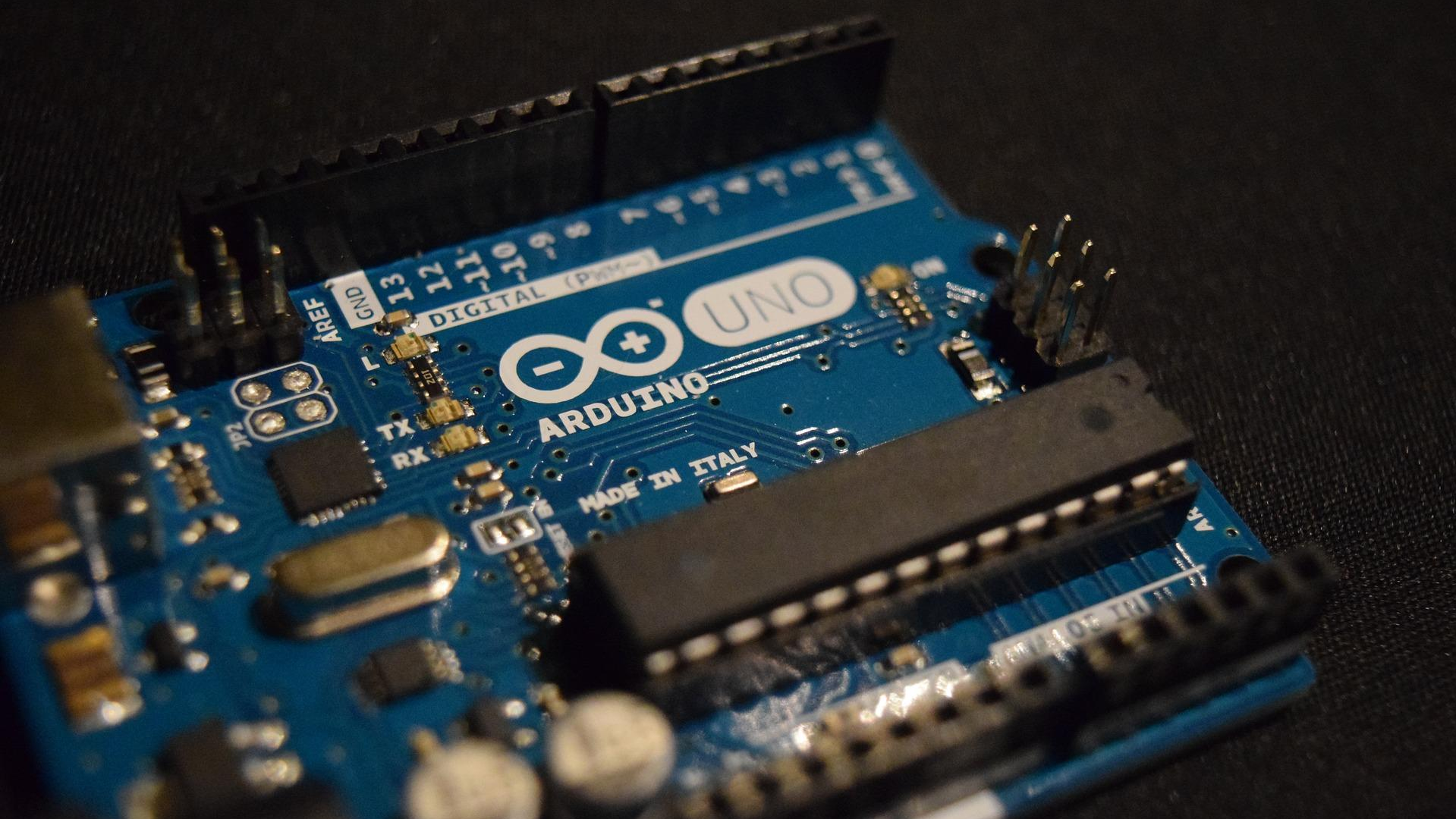
Sistemas de sistemas

Conversa entre sistemas para tomada de decisões.

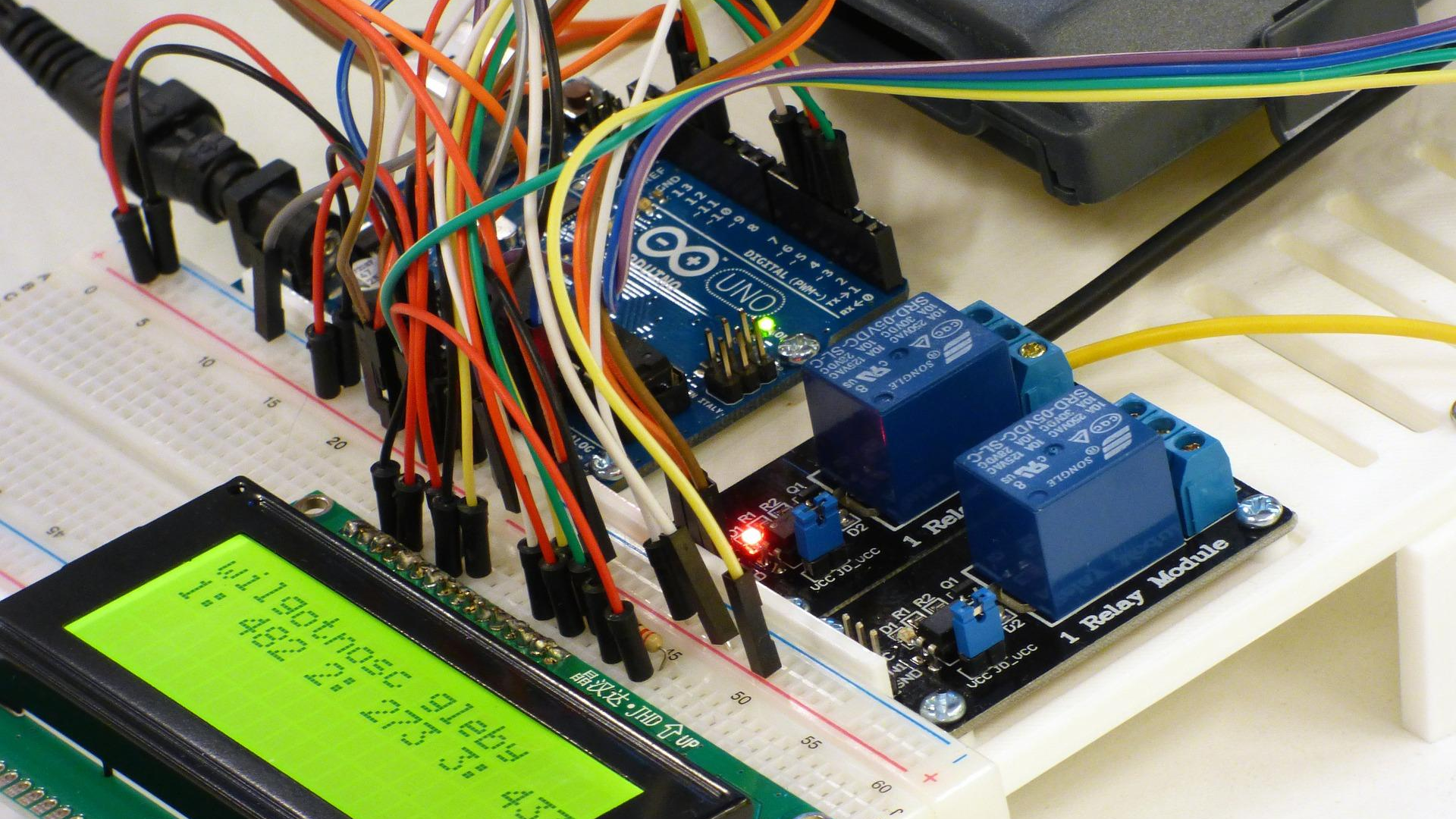
Ethernet, WiFi, RFID, Xbee, ZigBee, Bluetooth, 2G, 3G, 4G, 5G, ...



# Arduino



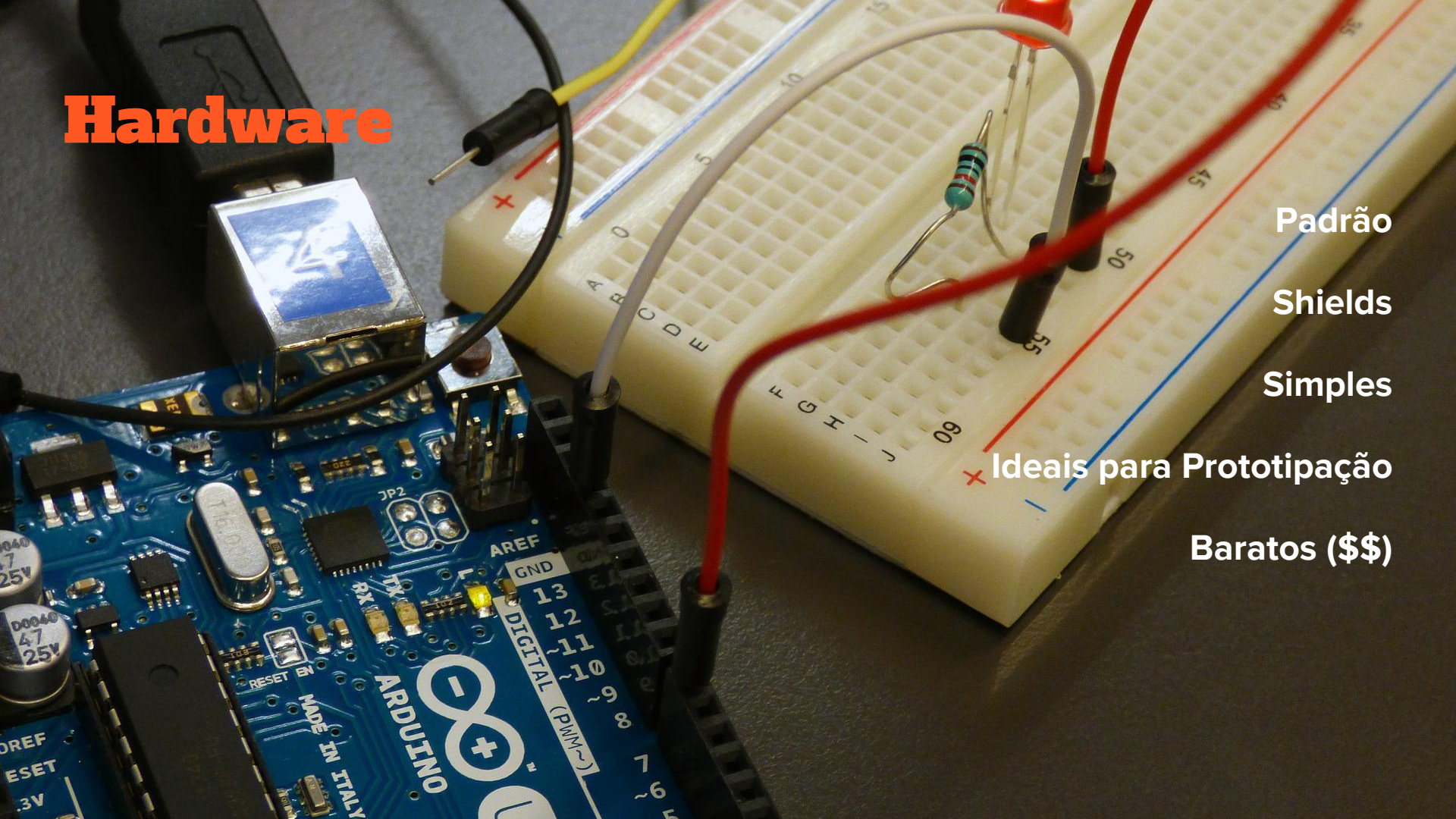




Wi190t05C 91eb9  
1: 482 2: 273 3: 43



# Hardware



Padrão

Shields

Simples

Ideais para Prototipação

Baratos (\$\$)

# Software

Opensource

Multiplataforma

Padrão

Funções


pinMode(), digitalWrite(), delay(), ...

Arquitetura

setup() e loop()

Semelhante com o “C”

Fácil de carregar

A screenshot of the Arduino IDE interface. The title bar at the top reads "Blink | Arduino 1.6.8". Below the title bar is a menu bar with "Arquivo", "Editar", "Sketch", "Ferramentas", and "Ajuda". Under "Arquivo", there are icons for opening, saving, and printing. The main editor area shows the "Blink" sketch. The code is as follows:

```
1  /*  
2   Blink  
3   Turns on an LED on for one second, then off for one second, repeatedly.  
4  
5   Most Arduinos have an on-board LED you can control. On the Uno and  
6   Leonardo, it is attached to digital pin 13. If you're unsure what  
7   pin the on-board LED is connected to on your Arduino model, check  
8   the documentation at http://www.arduino.cc  
9  
10  This example code is in the public domain.  
11  
12  modified 8 May 2014  
13  by Scott Fitzgerald  
14  */  
15  
16  
17  // the setup function runs once when you press reset or power the board  
18  void setup() {  
19    // initialize digital pin 13 as an output.  
20    pinMode(13, OUTPUT);  
21  }  
22  
23  // the loop function runs over and over again forever  
24  void loop() {  
25    digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)  
26    delay(1000);           // wait for a second  
27    digitalWrite(13, LOW);  // turn the LED off by making the voltage LOW  
28    delay(1000);           // wait for a second  
29  }
```

**ESP8266**

# ESP8266

Espressif

[espressif.com](http://espressif.com)

[bbs.espressif.com](http://bbs.espressif.com)

[esp8266.com](http://esp8266.com)

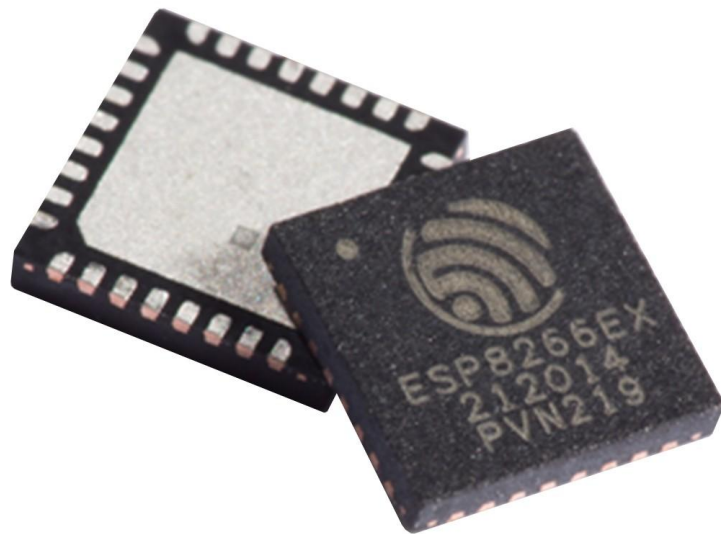
SoC WiFi

Tem um micro interno

Memória flash externa

Tensilica L106 32-bit micro controller

80MHz



# Especificações do Hardware

Tensão: 3V3

WiFi Protocol: 802.11 b/g/n (2.5GHz)

Periféricos: UART/SDIO/SPI/I2C/I2S/IR Remote Control/GPIO/ADC/PWM

Wi-Fi mode: station/softAP/SoftAP+station

Security: WPA/WPA2

Encryption: WEP/TKIP/AES

Firmware Upgrade: UART Download / OTA (via network)



# Módulos

Starter kits

Prontos para prototipação

Solda SMD

AT Commands

SDK

ESP8266



ESP-02



ESP-03



ESP-04



ESP-05



ESP-06



ESP-07



ESP-08



ESP-09



ESP-10



ESP-11



ESP-12



# **Arduino em placas customizadas**

# Placas customizadas no Arduino

Adafruit (proxy)

Sparkfun (modificações em arquivos)

ESP8266 (IDE separada)

# Custom boards

Funcionalidade disponível a partir da versão 1.6 do Arduino IDE.

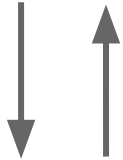
Funciona com requisições “json”

- Trata dependências

- Baixa os pacotes e arquivos necessários

- Os arquivos definem os menus e opções

# Custom Boards - Funcionamento



Requisição



Informações obtidas:

Versões;

Onde obtém os arquivos;

Nome das placas;

Ferramentas (compiladores, upload tool, ...).



# Custom Boards - Interface

Arquivos importantes:

- “boards.txt” - Definições da interface (menu)

- “platform.txt” - Definições e regras de compilação

# **ESP8266 como Arduino**

# ESP8266

[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)

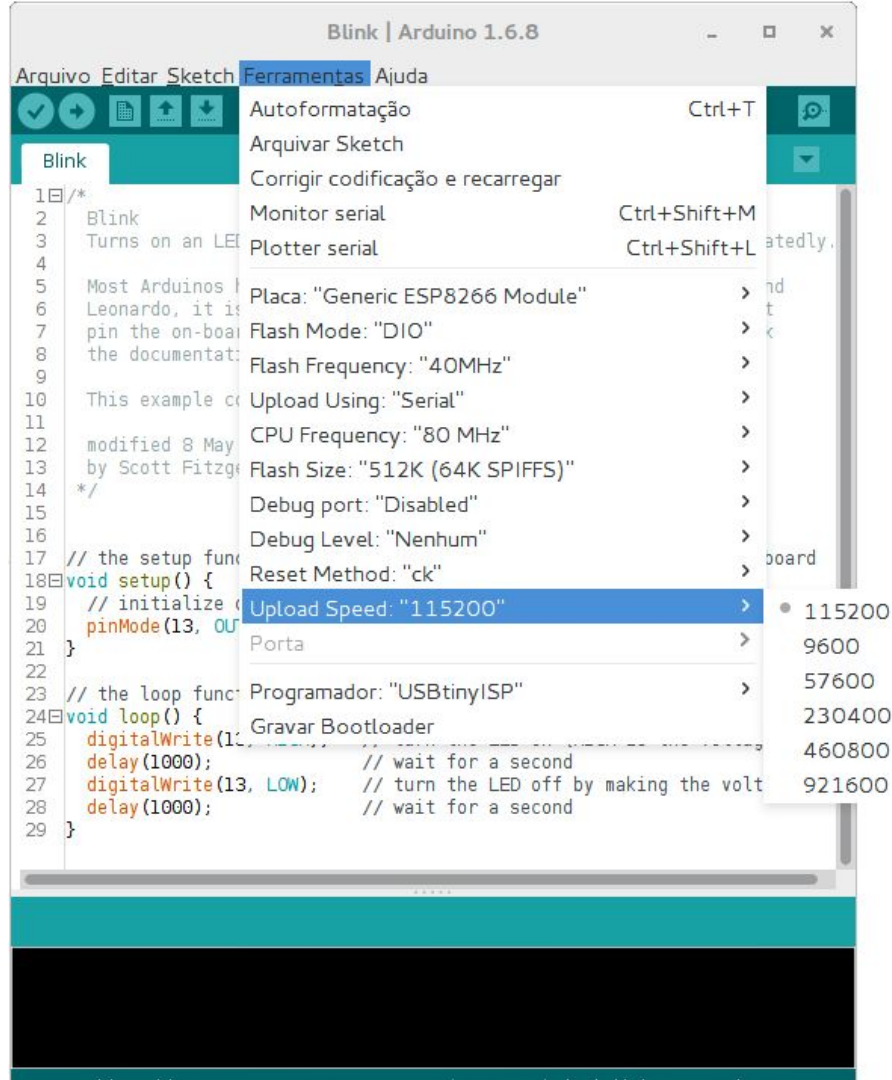
Oferece opções do *menu* para carga

Flash;

Frequência;

Tamanho de memória;

Entre outras coisas.



**Bibliotecas**

# Bibliotecas portadas

WiFi (ESP8266WiFi library)

Ticker

EEPROM

I2C (Wire library)

SPI

SoftwareSerial

ESP-specific APIs

mDNS and DNS-SD responder (ESP8266mDNS library)

SSDP responder (ESP8266SSDP)

DNS server (DNSServer library)

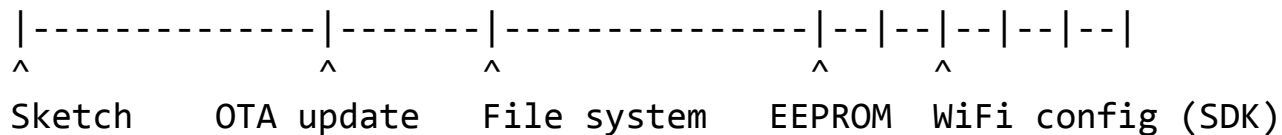
Servo

Other libraries (not included with the IDE)



# Sistema de Arquivos

## Documentação



| Board          | Flash chip size, bytes | File system size, bytes |
|----------------|------------------------|-------------------------|
| Generic module | 512k                   | 64k, 128k               |
| Generic module | 1M                     | 64k, 128k, 256k, 512k   |
| Generic module | 2M                     | 1M                      |
| Generic module | 4M                     | 3M                      |

# OTA (Over the air)

Tem suporte a carga de firmware OTA

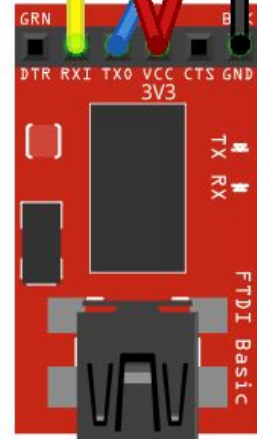
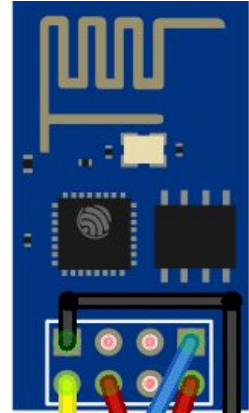
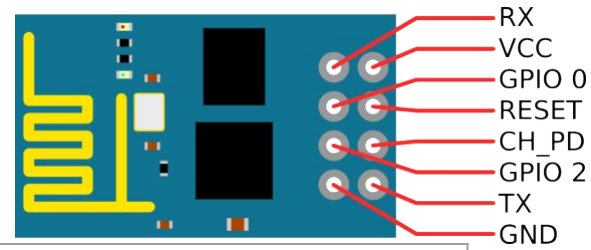
Arduino IDE

Web Browser

HTTP Server

**Conexão**

# Conexão



fritzing

|               |  |
|---------------|--|
| <b>Vcc:</b>   | Tensão de alimentação 3,3V. Módulo consome até 300 mA;   |
| <b>GND:</b>   | Sinal de Terra GND;  |
| <b>Tx:</b>    | Sinal de Tx do módulo, a ser conectado no Rx do microcontrolador (Sinal em 3,3V);  |
| <b>RST:</b>   | Sinal de Reset/Restart acionado em nível baixo (GND);  |
| <b>CH_PD:</b> | Sinal de habilitação do chip (chip enable), usado para gravação de firmware ou atualização. Deve ser mantido em nível ALTO para operação normal; |
| <b>GPIO0:</b> | Pode ser controlado pelo firmware, e <b>deve ser colocado em nível baixo (GND) para modo update, ou em nível alto para operação normal;</b>      |
| <b>GPIO2:</b> | I/O que pode ser controlada pelo firmware;   |
| <b>LED:</b>   | Quando está ligado, fica aceso em cor Vermelha, e aciona a cor Azul para indicar atividade. Pisca uma vez para indicar momento de boot.          |

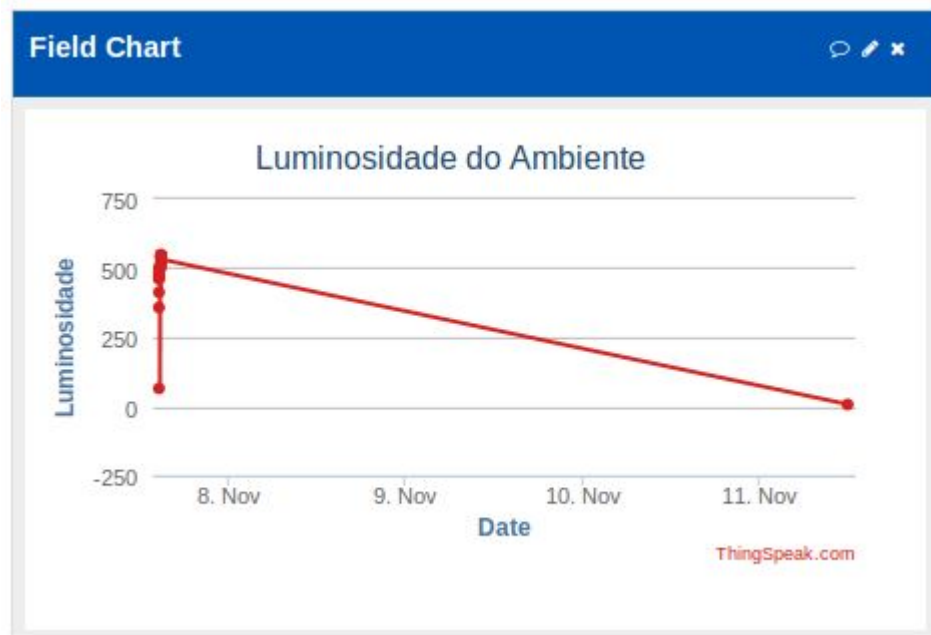
**Plataforma  
ThingSpeak**

# ThingSpeak



Plataforma *open source* para Internet das Coisas.

- Coleta
- Analisa
- Ação



# Exemplo de uso

Comandos HTTP (Get e Post)

Dados gerais:

<http://api.thingspeak.com/channels/104699/feed.json?key=IXOAZM8K253UMN6M>

Lendo:

<https://api.thingspeak.com/channels/104699/fields/1/last.txt>

Escrevendo:

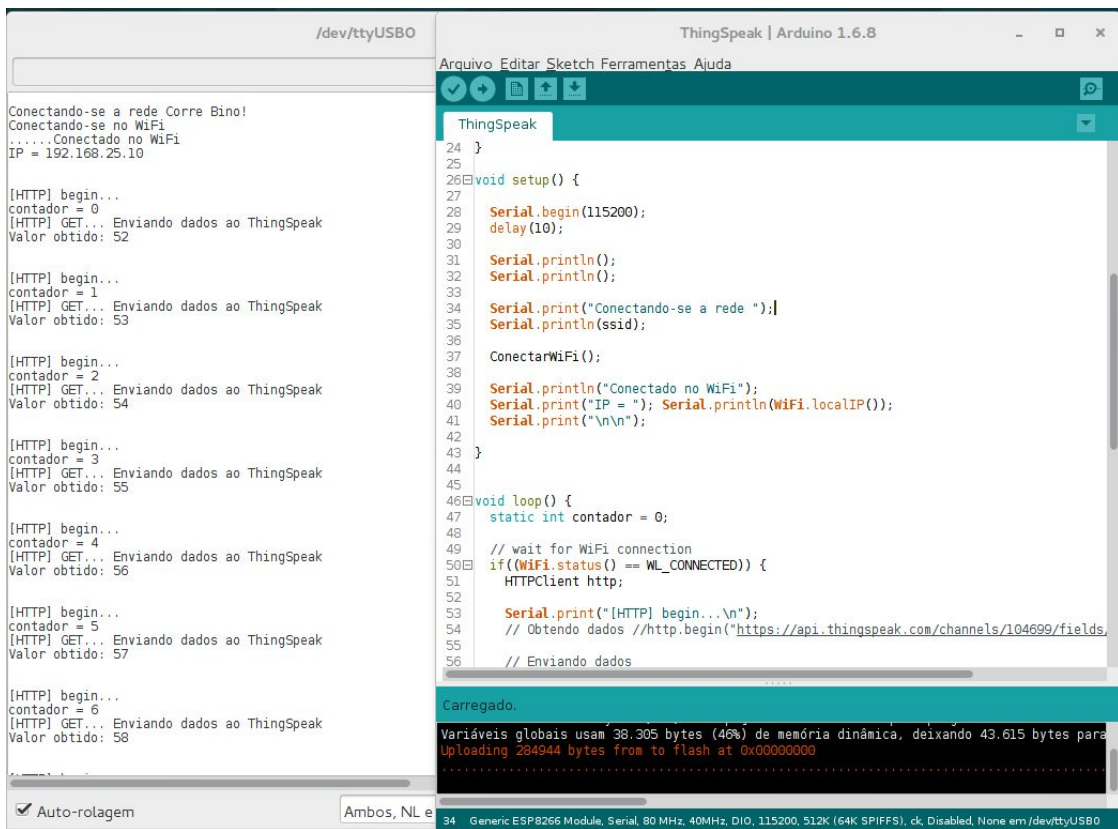
<http://api.thingspeak.com/update?key=GYVDZ7HQLLI1WRYB&field1=0>



**Exemplo**

# Aplicação de exemplo

Código disponibilizado  
no [Github](#).



The image shows a screenshot of an Arduino IDE window titled "ThingSpeak | Arduino 1.6.8". The left pane displays the serial monitor output for the "/dev/ttyUSB0" port, showing the program's execution logs. The right pane displays the sketch code, which includes setup and loop functions for connecting to a WiFi network and sending data to ThingSpeak.

```
Conectando-se a rede Corre Bino!  
Conectando-se no WiFi  
.....Conectado no WiFi  
IP = 192.168.25.10  
  
[HTTP] begin...  
contador = 0  
[HTTP] GET... Enviando dados ao ThingSpeak  
Valor obtido: 52  
  
[HTTP] begin...  
contador = 1  
[HTTP] GET... Enviando dados ao ThingSpeak  
Valor obtido: 53  
  
[HTTP] begin...  
contador = 2  
[HTTP] GET... Enviando dados ao ThingSpeak  
Valor obtido: 54  
  
[HTTP] begin...  
contador = 3  
[HTTP] GET... Enviando dados ao ThingSpeak  
Valor obtido: 55  
  
[HTTP] begin...  
contador = 4  
[HTTP] GET... Enviando dados ao ThingSpeak  
Valor obtido: 56  
  
[HTTP] begin...  
contador = 5  
[HTTP] GET... Enviando dados ao ThingSpeak  
Valor obtido: 57  
  
[HTTP] begin...  
contador = 6  
[HTTP] GET... Enviando dados ao ThingSpeak  
Valor obtido: 58
```

```
24 }  
25  
26 void setup() {  
27  
28   Serial.begin(115200);  
29   delay(10);  
30  
31   Serial.println();  
32   Serial.println();  
33  
34   Serial.print("Conectando-se a rede ");  
35   Serial.println(ssid);  
36  
37   ConectarWiFi();  
38  
39   Serial.println("Conectado no WiFi");  
40   Serial.print("IP = "); Serial.println(WiFi.localIP());  
41   Serial.print("\n\n");  
42  
43 }  
44  
45  
46 void loop() {  
47   static int contador = 0;  
48  
49   // wait for WiFi connection  
50   if((WiFi.status() != WL_CONNECTED)) {  
51     HTTPClient http;  
52  
53     Serial.print("[HTTP] begin...\n");  
54     // Obtendo dados //http.begin("https://api.thingspeak.com/channels/104699/fields");  
55  
56     // Enviando dados
```

Carregado.

Variáveis globais usam 38.305 bytes (46%) de memória dinâmica, deixando 43.615 bytes para uploading 284944 bytes from to flash at 0x00000000

34 Generic ESP8266 Module, Serial, 80 MHz, 40MHz, DIO, 115200, 512K (64K SPIFFS), ck, Disabled, None em /dev/ttyUSB0

**Duvidas?**

# Obrigado!

**Daniel Junho**

[djunho@gmail.com](mailto:djunho@gmail.com)

[danieljunho.com](http://danieljunho.com)