INTRODUCTION TO NODE MCU

**NodeMCU** is an open source [IoT](https://en.wikipedia.org/wiki/Internet_of_Things) platform. It includes [firmware](https://en.wikipedia.org/wiki/Firmware) which runs on the [ESP8266](https://en.wikipedia.org/wiki/ESP8266) [Wi-Fi](https://en.wikipedia.org/wiki/Wi-Fi) [SoC](https://en.wikipedia.org/wiki/System_on_a_chip) from Espressif Systems, and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the development kits. The firmware uses the [Lua](https://en.wikipedia.org/wiki/Lua_(programming_language)) scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson and [SPIFFS](https://en.wikipedia.org/w/index.php?title=SPIFFS&action=edit&redlink=1).



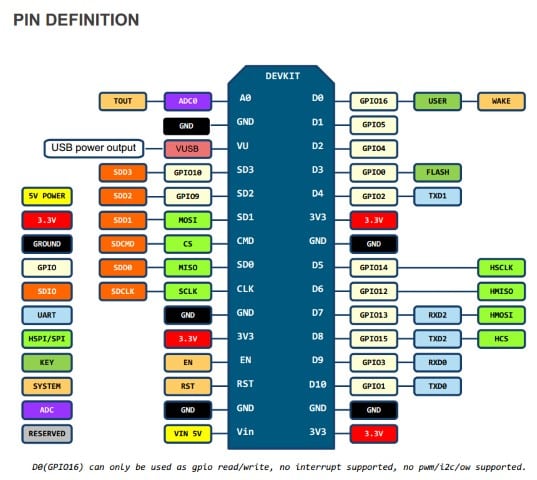
NodeMCU was created shortly after the [ESP8266](https://en.wikipedia.org/wiki/ESP8266) came out. On December 30, 2013, [Espressif Systems](https://en.wikipedia.org/w/index.php?title=Espressif_Systems&action=edit&redlink=1" \o "Espressif Systems (page does not exist)) began production of the ESP8266. The ESP8266 is a Wi-Fi SoC integrated with a [Tensilica](https://en.wikipedia.org/wiki/Tensilica" \o "Tensilica) Xtensa LX106 core, widely used in IoT applications (see [related projects](https://en.wikipedia.org/wiki/NodeMCU#Related_projects)). NodeMCU started on 13 Oct 2014, when Hong committed the first file of nodemcu-firmware to GitHub. Two months later, the project expanded to include an open-hardware platform when developer Huang R committed the [gerber](https://en.wikipedia.org/wiki/Gerber_format" \o "Gerber format) file of an ESP8266 board, named devkit v0.9. Later that month, Tuan PM ported [MQTT](https://en.wikipedia.org/wiki/MQTT) client library from [Contiki](https://en.wikipedia.org/wiki/Contiki) to the ESP8266 SoC platform, and committed to NodeMCU project, then NodeMCU was able to support the MQTT IoT protocol, using Lua to access the MQTT broker. Another important update was made on 30 Jan 2015, when Devsaurus ported the u8glib to NodeMCU project, enabling NodeMCU to easily drive LCD, Screen, OLED, even VGA displays.

In summer 2015 the creators abandoned the firmware project and a group of independent contributors took over. By summer 2016 the NodeMCU included more than 40 different modules. Due to resource constraints users need to select the modules relevant for their project and build a firmware tailored to their needs.



**ESP8266 Arduino Core**

As [Arduino.cc](https://en.wikipedia.org/wiki/Arduino) began developing new MCU boards based on non-[AVR](https://en.wikipedia.org/wiki/AVR_microcontrollers) processors like the ARM/SAM MCU and used in the Arduino Due, they needed to modify the [Arduino IDE](https://en.wikipedia.org/wiki/Arduino_IDE) so that it would be relatively easy to change the IDE to support alternate toolchains to allow Arduino C/C++ to be compiled for these new processors. They did this with the introduction of the Board Manager and the SAM Core. A "core" is the collection of software components required by the Board Manager and the Arduino IDE to compile an Arduino C/C++ source file for the target MCU's machine language. Some ESP8266 enthusiasts developed an Arduino core for the ESP8266 WiFi SoC, popularly called the "ESP8266 Core for the Arduino IDE".This has become a leading software development platform for the various ESP8266-based modules and development boards, including NodeMCUs.



It comes with 10 GPIOs, and interestingly each GPIO can be used for PWM, I2C and 1-wire. It also has 3.3V pins, GND pins, RX, TX, SCLK, MISO, MOSI and so on. Add the fact that it’s powered by a 5V micro-USB port .

## **Pins**

NodeMCU provides access to the [GPIO](https://en.wikipedia.org/wiki/General-purpose_input/output) (General Purpose Input/Output) and a pin mapping table is part of the API documentation.

|  |  |
| --- | --- |
| **I/O index** | **ESP8266 pin** |
| 0 [\*] | GPIO16 |
| 1 | GPIO5 |
| 2 | GPIO4 |
| 3 | GPIO0 |
| 4 | GPIO2 |
| 5 | GPIO14 |
| 6 | GPIO12 |
| 7 | GPIO13 |
| 8 | GPIO15 |
| 9 | GPIO3 |
| 10 | GPIO1 |
| 11 | GPIO9 |
| 12 | GPIO10 |

[\*] D0 (GPIO16) can only be used for GPIO read/write. It does not support open-drain/interrupt/PWM/I²C or 1-Wire.

REGARDS:

NARSIMHA REDDY

DHEERAJ T

RAGASRI YARLAGADDA