INTRODUCTION TO NODEMCU (ESP8266), CONNECTIVITY OF SENSORS AND ACTUATORS WITH NODEMCU,

Today, IOT applications are on the rise, and connecting objects are getting more and more important. There are several ways to connect objects such as Wi-Fi protocol.

NodeMCU is an open source platform based on ESP8266 which can connect objects and let data transfer using the Wi-Fi protocol. In addition, by providing some of the most important features of microcontrollers such as GPIO, PWM, ADC, and etc, it can solve many of the project's needs alone. The general features of this board are as follows:

- Easy to use
- Programmability with Arduino IDE or IUA languages
- Available as an access point or station
- practicable in Event-driven API applications
- Having an internal antenna
- Containing 13 GPIO pins, 10 PWM channels, I2C, SPI, ADC, UART,

NodeMCU is an open-source LUA based firmware developed for the ESP8266 wifi chip. By exploring functionality with the ESP8266 chip, NodeMCU firmware comes with the ESP8266 Development board/kit i.e. NodeMCU Development board. NodeMCU is an open-source platform, its hardware design is open for edit/modify/build. The **ESP8266** is a low-cost Wi-Fi chip developed by Espressif Systems with TCP/IP protocol.
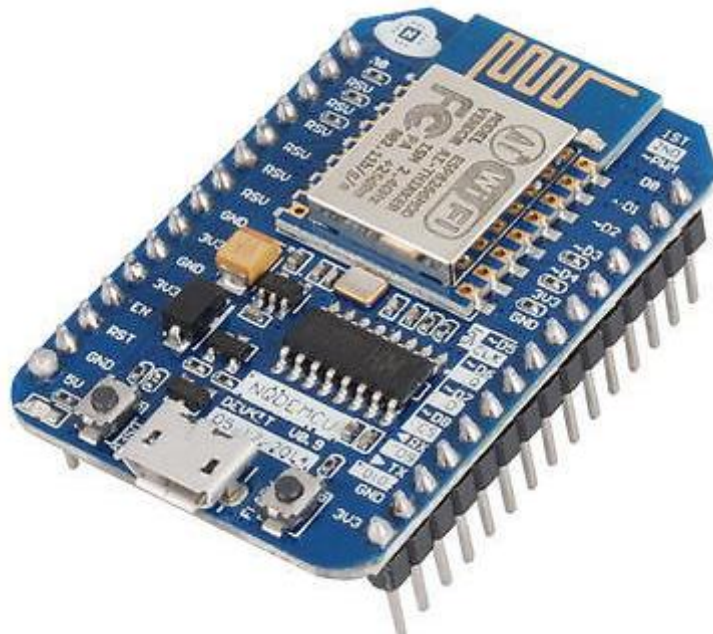
NodeMCU Development Board/kit v0.9 (Version1)

NodeMCU Development board is featured with wifi capability, analog pin, digital pins, and serial communication protocols. To get started with using NodeMCU for IoT applications first we need to know about how to write/download NodeMCU firmware in NodeMCU Development Boards. And before that where this NodeMCU firmware will get as per our requirement.

ESP8266 NodeMCU Features & Using It With Arduino IDE

The Internet of Things (IoT) has been a trending field in the world of technology. It has changed the way we work. Physical objects and the digital world are connected now more than ever. Keeping this in mind, Espressif Systems (A Shanghai-based Semiconductor Company) has released an adorable, bite-sized WiFi enabled microcontroller – ESP8266, it can monitor and control things from anywhere in the world – perfect for just about any IoT project.

1. **ESP-12E Module :** The development board equips the ESP-12E module containing ESP8266 chip having Tensilica Xtensa® 32-bit LX106 RISC microprocessor which operates at 80 to 160 MHz adjustable clock frequency and supports RTOS.
   - Tensilica Xtensa® 32-bit LX106
   - 80 to 160 MHz Clock Freq.
   - 128kB internal RAM
   - 4MB external flash
   - 802.11b/g/n Wi-Fi transceiver


ESP-12E Chip

2.4GHz Antenna
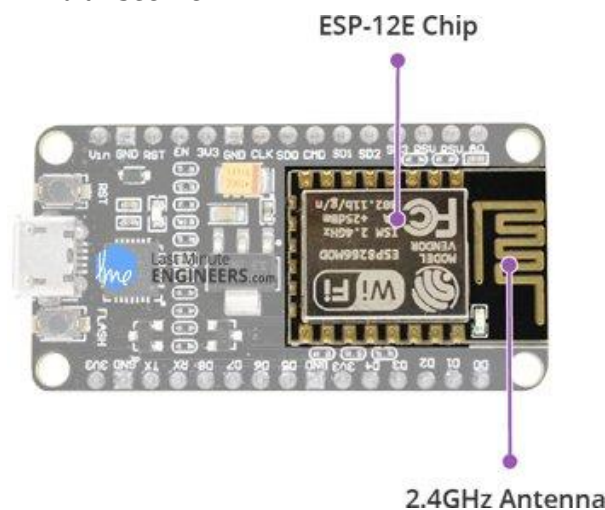
There's also 128 KB RAM and 4MB of Flash memory (for program and data storage) just enough to cope with the large strings that make up web pages, JSON/XML data,

and everything we throw at IoT devices nowadays. The ESP8266 Integrates 802.11b/g/n HT40 Wi-Fi transceiver, so it can not only connect to a WiFi network and interact with the Internet, but it can also set up a network of its own, allowing other devices to connect directly to it. This makes the ESP8266 NodeMCU even more versatile.

2. **Power Requirement :** As the operating voltage range of ESP8266 is 3V to 3.6V, the board comes with a LDO voltage regulator to keep the voltage steady at 3.3V. It can reliably supply up to 600mA, which should be more than enough when ESP8266 pulls as much as 80mA during RF transmissions. The output of the regulator is also broken out to one of the sides of the board and labeled as 3V3. This pin can be used to supply power to external components.

   - Operating Voltage: 2.5V to 3.6V
   - On-board 3.3V 600mA regulator
   - 80mA Operating Current
   - 20 µA during Sleep Mode
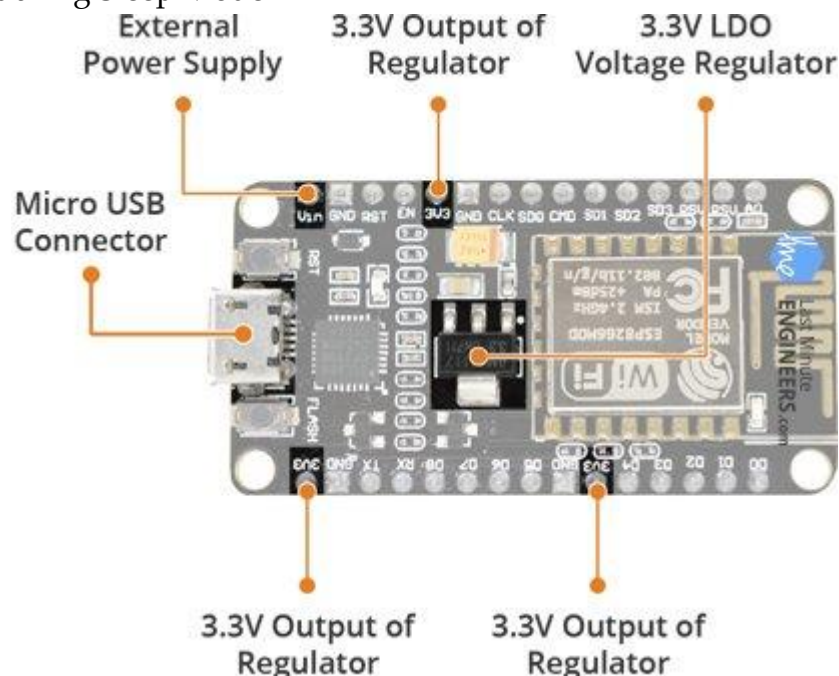
Power to the ESP8266 NodeMCU is supplied via the on-board MicroB USB connector. Alternatively, if you have a regulated 5V voltage source, the VIN pin can be used to directly supply the ESP8266 and its peripherals.

The ESP8266 requires a 3.3V power supply and 3.3V logic levels for communication. The GPIO pins are not 5V-tolerant! If you want to interface the board with 5V (or higher) components, you'll need to do some level shifting.

3. **Peripherals and I/O :** The ESP8266 NodeMCU has total 17 GPIO pins broken out to the pin headers on both sides of the development board. These pins can be assigned to all sorts of peripheral duties, including:
   - ADC channel – A 10-bit ADC channel.
   - UART interface – UART interface is used to load code serially.
   - PWM outputs – PWM pins for dimming LEDs or controlling motors.
   - SPI, I2C & I2S interface – SPI and I2C interface to hook up all sorts of sensors and peripherals.
   - I2S interface – I2S interface if you want to add sound to your project.

   Multiplexed I/Os
   - 1 ADC channels
   - 2 UART interfaces
   - 4 PWM outputs
   - SPI, I2C & I2S interface

   Thanks to the ESP8266's pin multiplexing feature (Multiple peripherals multiplexed on a single GPIO pin). Meaning a single GPIO pin can act as PWM/UART/SPI.
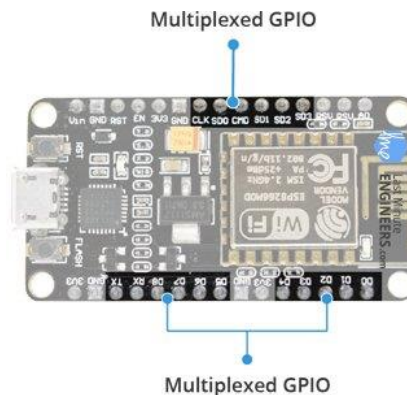
4. **On-board Switches & LED Indicator :** The ESP8266 NodeMCU features two buttons. One marked as RST located on the top left corner is the Reset button, used of course to reset the ESP8266 chip. The other FLASH button on the bottom left corner is the download button used while upgrading firmware.
   - RST – Reset the ESP8266 chip
   - FLASH – Download new programs
   - Blue LED – User Programmable
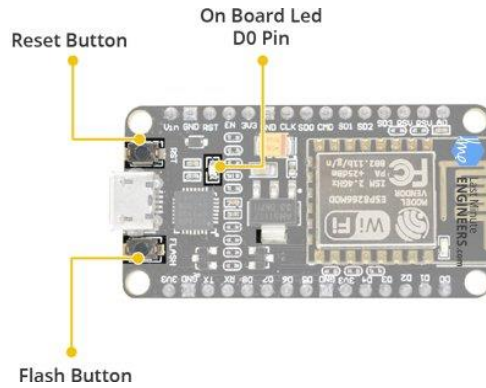
The board also has a LED indicator which is user programmable and is connected to the D0 pin of the board.

5. **Serial Communication :** The board includes CP2102 USB-to-UART Bridge Controller from Silicon Labs, which converts USB signal to serial and allows your computer to program and communicate with the ESP8266 chip.
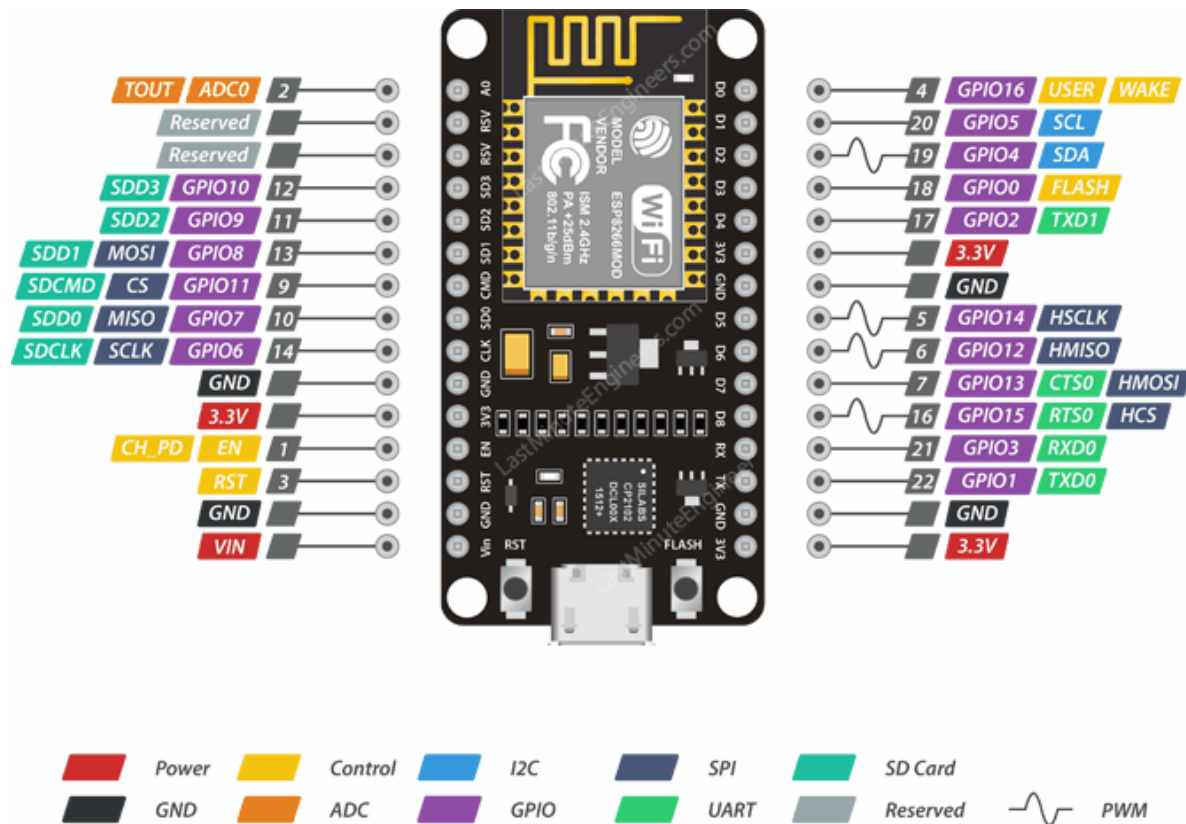   - CP2102 USB-to-UART converter
    - 4.5 Mbps communication speed
    - Flow Control support
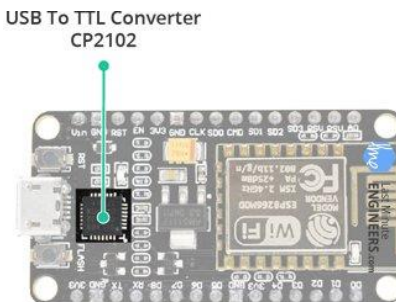
ESP8266 NodeMCU Pinout

The ESP8266 NodeMCU has total 30 pins that interface it to the outside world. The connections are as follows:

1. Power Pins There are four power pins viz. one VIN pin & three 3.3V pins. The VIN pin can be used to directly supply the ESP8266 and its peripherals, if you have a regulated 5V voltage source. The 3.3V pins are the output of an on-board voltage regulator. These pins can be used to supply power to external components.

2. GND is a ground pin of ESP8266 NodeMCU development board.

3. I2C Pins are used to hook up all sorts of I2C sensors and peripherals in your project. Both I2C Master and I2C Slave are supported. I2C interface functionality can be realized programmatically, and the clock frequency is 100 kHz at a maximum. It should be noted that I2C clock frequency should be higher than the slowest clock frequency of the slave device.

4. GPIO Pins ESP8266 NodeMCU has 17 GPIO pins which can be assigned to various functions such as I2C, I2S, UART, PWM, IR Remote Control, LED Light and Button programmatically. Each digital enabled GPIO can be configured to internal pull-up or pull-down, or set to high impedance. When configured as an input, it can also be set to edge-trigger or level-trigger to generate CPU interrupts.

5. ADC Channel The NodeMCU is embedded with a 10-bit precision SAR ADC. The two functions can be implemented using ADC viz. Testing power supply voltage of

VDD3P3 pin and testing input voltage of TOUT pin. However, they cannot be implemented at the same time.

6. UART Pins ESP8266 NodeMCU has 2 UART interfaces, i.e. UART0 and UART1, which provide asynchronous communication (RS232 and RS485), and can communicate at up to 4.5 Mbps. UART0 (TXD0, RXD0, RST0 & CTS0 pins) can be used for communication. It supports fluid control. However, UART1 (TXD1 pin) features only data transmit signal so, it is usually used for printing log.

7. SPI Pins ESP8266 features two SPIs (SPI and HSPI) in slave and master modes. These SPIs also support the following general-purpose SPI features:
   - 4 timing modes of the SPI format transfer
   - Up to 80 MHz and the divided clocks of 80 MHz
   - Up to 64-Byte FIFO

8. SDIO Pins ESP8266 features Secure Digital Input/Output Interface (SDIO) which is used to directly interface SD cards. 4-bit 25 MHz SDIO v1.1 and 4-bit 50 MHz SDIO v2.0 are supported.

9. PWM Pins The board has 4 channels of Pulse Width Modulation (PWM). The PWM output can be implemented programmatically and used for driving digital motors and LEDs. PWM frequency range is adjustable from 1000 µs to 10000 µs, i.e., between 100 Hz and 1 kHz.

10. Control Pins are used to control ESP8266. These pins include Chip Enable pin (EN), Reset pin (RST) and WAKE pin.
    - EN pin – The ESP8266 chip is enabled when EN pin is pulled HIGH. When pulled LOW the chip works at minimum power.
    - RST pin – RST pin is used to reset the ESP8266 chip.