Architecture of NodeMCU

An Overview of NodeMCU’s Design and Components

Engr. Mon Arjay Malbog

Intended Learning Outcomes

At the end of the session, the students will be able to:

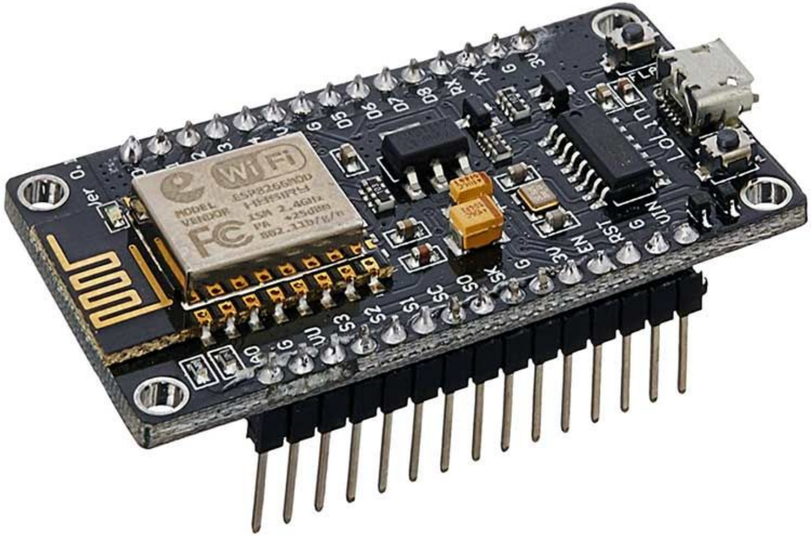
- identify and explain the hardware components of NodeMCU
- gain an in-depth understanding of the ESP8266 microcontroller, including its CPU, memory, Wi-Fi capabilities, and power management features, and how these are used to build IoT applications.
- learn how NodeMCU's firmware architecture, including the Lua interpreter and APIs for GPIO, networking, and file system management, allows easy control of peripherals and communication with external devices.
- familiarize NodeMCU's Wi-Fi networking modes

What is NodeMCU?

NodeMCU is an open-source IoT platform based on the ESP8266 microcontroller.

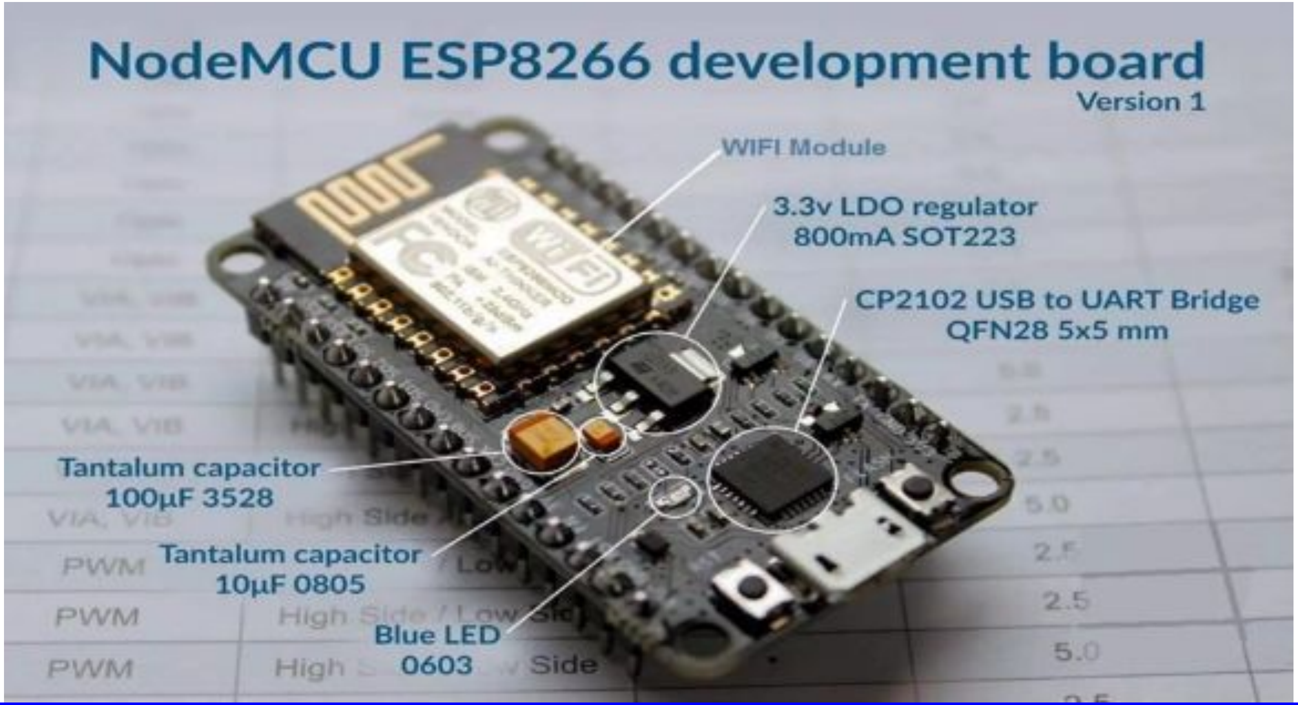
It includes firmware that runs on the ESP8266 and hardware based on the ESP-12 module.

Built-in support for Lua scripting language.



https://components101.com/development-boards/nodemcu-esp8266-pinout-features-and-datasheet

Main Component of NodeMCU



https://www.slideshare.net/slideshow/introduction-to-node-mcu/234762414#9

Key Components of NodeMCU

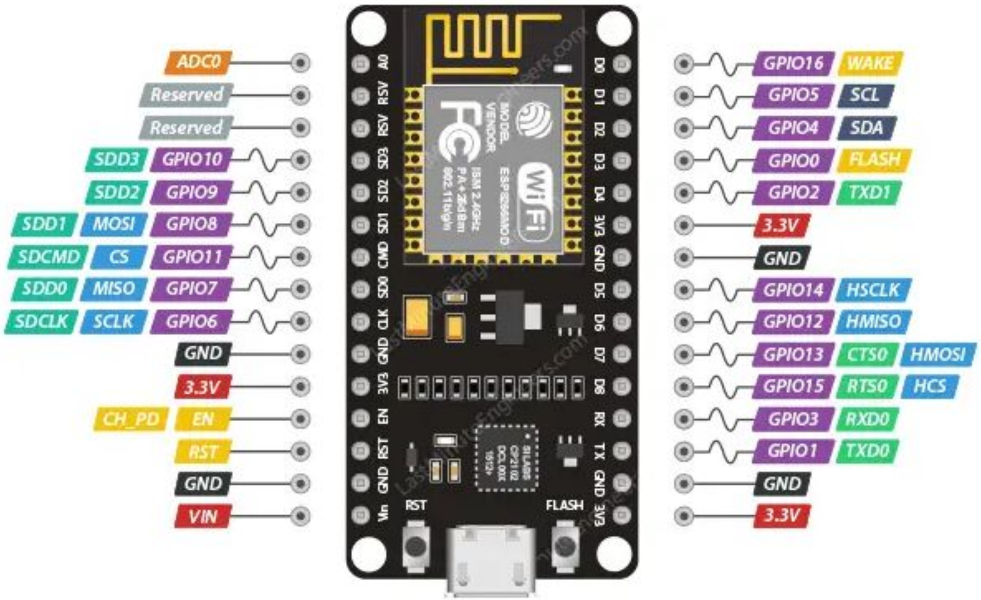
ESP8266 SoC: The core microcontroller with built-in Wi-Fi capability.

Flash Memory: Stores firmware and user code.

GPIO Pins: General-purpose input/output pins for connecting sensors, actuators, etc.

Power Supply: Voltage regulator to convert 5V input to 3.3V.

UART, I2C, SPI: Communication interfaces.



https://lastminuteengineers.com/esp8266-pinout-reference/

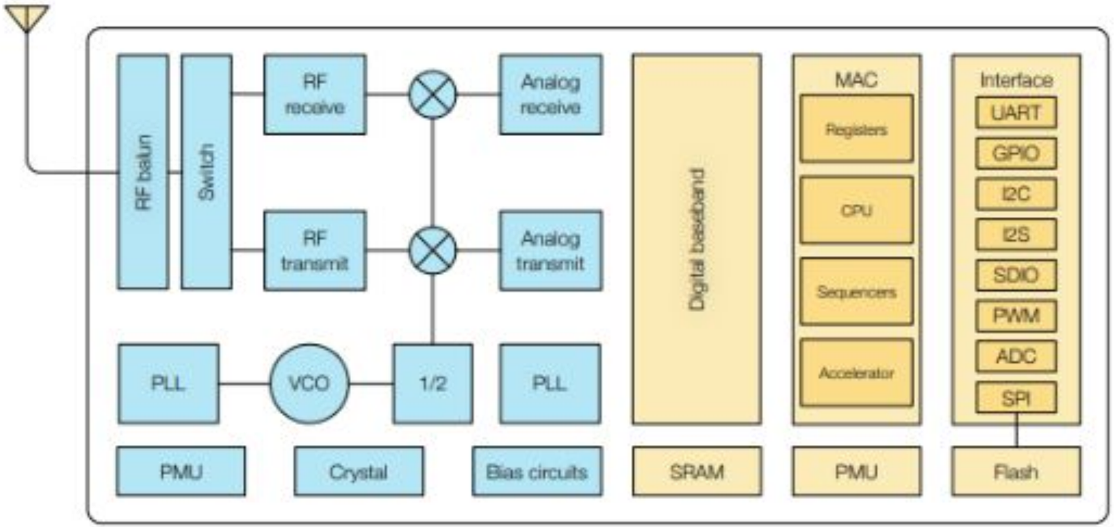
Inside the ESP8266

RISC CPU (Tensilica L106): 32-bit low-power processor running at 80 MHz.

Memory: 64 KB of instruction RAM, 96 KB of data RAM.

Wi-Fi Module: Integrated 802.11 b/g/n Wi-Fi transceiver.

System Clock: Supports deep sleep mode and low-power operations.



https://annefou.github.io/IoT_introduction/02-ESP8266/index.html

NodeMCU Firmware Overview

Lua Interpreter: Built-in Lua scripting engine to run user code.

API for GPIO, Wi-Fi, and Networking: Pre-built libraries for networking, sensor control, and peripheral communication.

File System (SPIFFS): SPI Flash File System for storing scripts and data.

OTA (Over-the-Air) Updates: Supports firmware updates without the need for physical access.

GPIO and Communication Protocols

GPIO Pins: Configurable for digital input/output.

I2C and SPI Buses: Interfaces for communicating with external sensors and peripherals.

UART: Serial communication interface, useful for debugging.

PWM: Pulse-width modulation support for controlling motors, LEDs, etc.

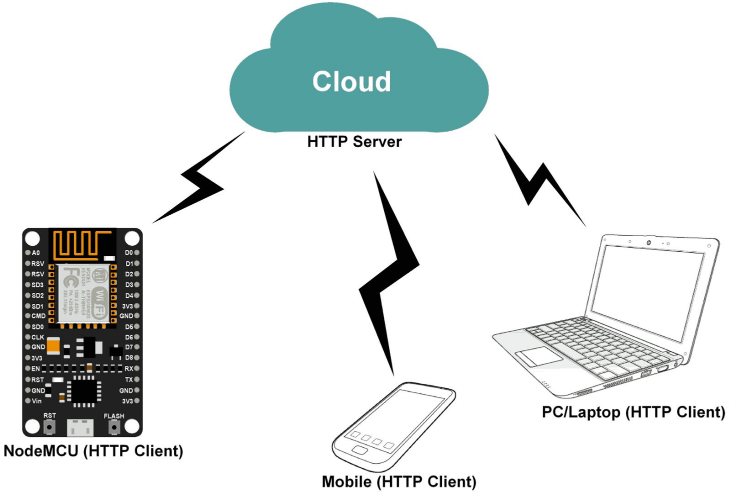
Networking Features

Access Point and Station Mode: Can act as both a client and a Wi-Fi hotspot.

TCP/IP Stack: Built-in networking protocols.

HTTP/MQTT Support: Enables communication with cloud servers or IoT platforms.

Power-Saving Modes: Supports deep sleep for low-power IoT applications.



https://www.electronicwings.com/nodemcu/http-client-on-nodemcu-with-arduino-ide

Recap of NodeMCU Architecture

- Combines a powerful microcontroller with built-in Wi-Fi.
- Easy to program using Lua or Arduino IDE.
- Ideal for IoT applications due to its small size, low power consumption, and network capabilities.

Next Step: Explore and experiment with NodeMCU for your IoT projects.

End