Nodemcu

NodeMCU is an open source IoT platform.[4][5] It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module.[6][7] The term "NodeMCU" by default refers to the firmware rather than the development kits. The firmware uses the Lua 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,[8] and spiffs.[9]

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)[[6]](https://en.wikipedia.org/wiki/NodeMCU#cite_note-Espressif_Systems-6) began production of the ESP8266.[[10]](https://en.wikipedia.org/wiki/NodeMCU#cite_note-10) The ESP8266 is a Wi-Fi SoC integrated with a [Tensilica](https://en.wikipedia.org/wiki/Tensilica) Xtensa LX106 core,[*[citation needed](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed" \o "Wikipedia:Citation needed)*] 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.

As Arduino.cc 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 so that it would be relatively easy to change the IDE to support alternate toolchains to allow Arduino C/C++ to be compiled down to 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 down to the target MCU's machine language. Some creative ESP8266 enthusiasts have developed an Arduino core for the ESP8266 WiFi SoC that is available at the GitHub ESP8266 Core webpage. This is what is popularly called the "ESP8266 Core for the Arduino IDE" and it has become one of the leading software development platforms for the various ESP8266 based modules and development boards, including NodeMCUs

Nodejs

Node.js is an open-source, cross-platform JavaScript run-time environment that executes JavaScript code outside of a browser. Typically, JavaScript is used primarily for client-side scripting, in which scripts written in JavaScript are embedded in a webpage's HTML and run client-side by a JavaScript engine in the user's web browser. Node.js lets developers use JavaScript to write Command Line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm,[6] unifying web application development around a single programming language, rather than different languages for server side and client side scripts.

Though .js is the conventional filename extension for JavaScript code, the name "Node.js" does not refer to a particular file in this context and is merely the name of the product. Node.js has an event-driven architecture capable of asynchronous I/O. These design choices aim to optimize throughput and scalability in web applications with many input/output operations, as well as for real-time Web applications

The Node.js [distributed development](https://en.wikipedia.org/wiki/Distributed_development) project, governed by the Node.js Foundation,[[8]](https://en.wikipedia.org/wiki/Node.js" \l "cite_note-8) is facilitated by the [Linux Foundation](https://en.wikipedia.org/wiki/Linux_Foundation)'s Collaborative Projects program.[[9]](https://en.wikipedia.org/wiki/Node.js#cite_note-9)

Node.js allows the creation of Web servers and networking tools using JavaScript and a collection of "modules" that handle various core functionality.[29][32][45][46][47] Modules are provided for file system I/O, networking (DNS, HTTP, TCP, TLS/SSL, or UDP), binary data (buffers), cryptography functions, data streams, and other core functions.[32][46][48] Node.js's modules use an API designed to reduce the complexity of writing server applications.[32][46]

Though initially the module system was based on commonjs module pattern, the recent introduction of modules in the ECMAScript specification has shifted the direction of using ECMAScript Modules in Node.js by default instead.[49]

Node.js is officially supported on Linux, macOS, Microsoft Windows, SmartOS, FreeBSD, and IBM AIX.[3] The provided source code may also be built on similar operating systems or be modified by third parties to support others such as NonStop[50] and Unix servers. Alternatively, they can be written with CoffeeScript[51] (a JavaScript alternative), Dart or TypeScript (strongly typed forms of JavaScript), or any other language that can compile to JavaScript.[51][52]

Node.js is primarily used to build network programs such as Web servers.[45] The biggest difference between Node.js and PHP is that most functions in PHP block until completion (commands execute only after previous commands finish), while Node.js functions are non-blocking (commands execute concurrently or even in parallel,[53][54] and use callbacks to signal completion or failure).[45]

Node.js operates on a [single thread](https://en.wikipedia.org/wiki/Single_threading) event loop, using [non-blocking I/O](https://en.wikipedia.org/wiki/Non-blocking_I/O) calls, allowing it to support tens of thousands of concurrent connections without incurring the cost of thread [context switching](https://en.wikipedia.org/wiki/Context_switch).[[69]](https://en.wikipedia.org/wiki/Node.js#cite_note-69) The design of sharing a single thread among all the requests that use the [observer pattern](https://en.wikipedia.org/wiki/Observer_pattern) is intended for building highly concurrent applications, where any function performing I/O must use a [callback](https://en.wikipedia.org/wiki/Callback_(computer_programming)). To accommodate the single-threaded event loop, Node.js uses the [libuv](https://en.wikipedia.org/wiki/Libuv) library—which, in turn, uses a fixed-sized thread pool that handles some of the non-blocking asynchronous I/O operations.[[7]](https://en.wikipedia.org/wiki/Node.js#cite_note-readwrite-7)

Angularjs

AngularJS (also written as Angular.js) is a JavaScript-based open-source front-end web application framework mainly maintained by Google and by a community of individuals and corporations to address many of the challenges encountered in developing single-page applications. The JavaScript components complement Apache Cordova, a framework used for developing cross-platform mobile apps. It aims to simplify both the development and the testing of such applications by providing a framework for client-side model–view–controller (MVC) and model–view–viewmodel (MVVM) architectures, along with components commonly used in rich Internet applications. (This flexibility has led to the acronym MVW, which stands for "model-view-whatever" and may also encompass model–view–presenter and model–view–adapter.) In 2014, the original AngularJS team began working on the Angular application platform.

The AngularJS framework works by first reading the HTML page, which has additional custom tag attributes embedded into it. Angular interprets those attributes as directives to bind input or output parts of the page to a model that is represented by standard JavaScript variables. The values of those JavaScript variables can be manually set within the code, or retrieved from static or dynamic JSON resources.

According to JavaScript analytics service Libscore, AngularJS is used on the websites of Wolfram Alpha, NBC, Walgreens, Intel, Sprint, ABC News, and about 12,000 other sites out of 1 million tested in October 2016.[3] AngularJS is currently in the top 100 of the most starred projects on GitHub.[4]

AngularJS is built on the belief that declarative programming should be used to create user interfaces and connect software components, while imperative programming is better suited to defining an application's business logic.[5] The framework adapts and extends traditional HTML to present dynamic content through two-way data-binding that allows for the automatic synchronization of models and views. As a result, AngularJS de-emphasizes explicit DOM manipulation with the goal of improving testability and performance.

AngularJS's design goals include:

to decouple DOM manipulation from application logic. The difficulty of this is dramatically affected by the way the code is structured.

to decouple the client side of an application from the server side. This allows development work to progress in parallel, and allows for reuse of both sides.

to provide structure for the journey of building an application: from designing the UI, through writing the business logic, to testing.

AngularJS implements the MVC pattern to separate presentation, data, and logic components.[6] Using dependency injection, Angular brings traditionally server-side services, such as view-dependent controllers, to client-side web applications. Consequently, much of the burden on the server can be reduced.

Mongodb

MongoDB is a free and open-source cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schemata. MongoDB is developed by MongoDB Inc., and is published under a combination of the Server Side Public License and the Apache License.

Main features

[Ad hoc](https://en.wikipedia.org/wiki/Ad_hoc) queries

Indexing

Replication

[Load balancing](https://en.wikipedia.org/wiki/Load_balancing_(computing))

File storage

Aggregation

Server-side JavaScript execution

Capped collections

Transactions

Literatures survey

1. https://stopsmartmetersau.files.wordpress.com/2013/12/smart-water-meters.pdf
   1. Name smart-water-meters
   2. By- Australian water association

A key issue for utilities and consumers is the frequency and temporal spread of water meter reads. Most utilities record water consumption data manually on a monthly, quarterly or half-yearly basis. While monthly data provides better data set for high levels of water usage, quarterly or half-yearly data collection provides a ‘lumpy’ dataset in which a whole year of water consumption is lumped into only two or four sets. This infrequent data collection is sufficient for billing purposes, but gives limited information on actual water use behaviour, leakage and seasonal variation. The timely collection and analysis of water use data, and the timely relaying of these data to the water user, can result in significant changes in water use behaviour. The benefits include immediate leak detection and consequent remedial action that can save precious quantities of water. The data is also invaluable in designing water efficiency and reuse systems (Butler, 2007) and for the improvement of demand management policies and programs (Giurco et al., 2008). Smart meters are one step closer to bring this dilemma into realtime monitoring of water use, with the added benefit of letting the users know where they use the water most in a dwelling, for example, shower or bath.

1. Intelligent Metering for Urban Water: A Review
   1. [www.mdpi.com/journal/water](http://www.mdpi.com/journal/water)
   2. Intelligent Metering for Urban Water
   3. Metering Technologies There are a range of metering technologies, employing different principles to capture and record water use (see [48] for a comprehensive introduction). Common technologies broadly fall within one of four categories: (i) displacement meters; (ii) velocity meters; (iii) compound (or combination); or (iv) electromagnetic meters . Displacement (or mechanical) meters require the movement of water to mechanically displace components within the meter to record water flow. They have the advantage of being inexpensive and accurate at low to moderate flow rates. Velocity meters, such as multi-jet, magnetic flow and ultrasonic measure the velocity of flow through a meter of a known internal capacity. Speed of flow is then converted into volume of flow for usage [50]. Combination meters utilise the strengths of displacement and velocity technologies in the one meter, in cases of variable flow rates (i.e., both high and low flow rates need to be recorded accurately). Electromagnetic flow meters harness the electromagnetic properties of water, which, as it flows, generates voltage as it crosses the force lines of a magnetic field [49]. With no moving parts, these meters are well-suited to flows with contaminants or debris that could otherwise damage a mechanical meter