ESP8266 and ESP32 Series of SoC Microcontrollers

Hakkı SOY
Necmettin Erbakan University

Introduction

Nowadays, Internet of Things (IoT) heavily permeates the daily lives of humans through smart devices that can be monitored or controlled remotely. Basically, IoT contains the embedded devices can be accessed through Internet to meet the users' demand coming from separate applications (Al-Fuqaha et al., 2015). In parallel to increasing penetration of IoT-enabled devices in various domains, several manufacturers have started to produce the RF transceiver chips and modules with different protocols. Espressif Systems is one of the main actors in IoT market which was founded in 2008, based in Shanghai, China. Espressif offers well-integrated, low-power, low-cost, high performance, energy-efficient wireless IoT chips and modules that are widely used in mobile devices, home appliances and industrial applications. The well-known products from Espressif are the ESP8266 and ESP32 series chips, modules and development boards. Espressif uses a system-on-chip (SoC) architecture, which refers a single integrated circuit (IC) composed of multiple components. Table 1 shows the specifications of the ESP8266 and ESP32 Series SoCs (Espressif, 2021a).

Table 1. The Technical Specifications of the SoCs from Espressif Systems.

SoC	Core	MCU	Wi-Fi	BLE	Pins	RAM	ROM	Flash
ESP32-C6	Single	160 MHz	2.4 GHz	5.0	32	400 KB RAM	384KB	--
ESP32-S3	Dual	240 MHz	2.4 GHz	5.0	56	512 KB SRAM	384 KB	--
ESP32-S2	Single	240 MHz	2.4 GHz	--	56	320 KB SRAM	128 KB	--
ESP32-C3	Single	160 MHz	2.4 GHz	5.0	32	400 KB RAM	384 KB	4 MB
ESP32-D0WD-V3	Dual	240 MHz						--
ESP32-D2WD	Dual	160 MHz						2 MB
ESP32-U4WDH	Single	160 MHz						4 MB
ESP32-S0WD	Single	160 MHz						--
ESP32-PICO-V3	Dual	240 MHz						4 MB
ESP32-PICO-D4	Dual	240 MHz	2.4 GHz	4.2	48	520 KB SRAM	448 KB	4 MB
ESP8266EX	Single	160 MHz	2.4 GHz	--	32	160 KB RAM	--	--

ESP8266

ESP8266 (2014), officially known as ESP8266EX, is a Wi-Fi enabled SoC that integrates Cadence Tensilica L106 32-bit RISC processor with a TCP/IP stack. It is also embedded with memory controller, including on-chip SRAM and ROM. Figure 1 shows the building blocks of the ESP8266 SoC (Espressif, 2021b; Kolban, 2016).

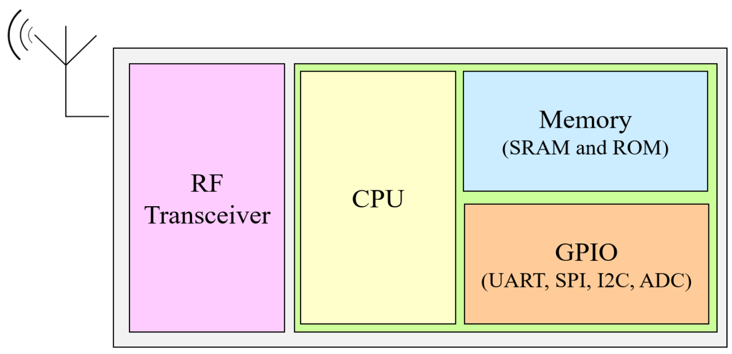


Figure 1. Building Blocks of the ESP8266 SoC.

ESP8266 SoC has a quad-flat no-leads (QFN)-32 package with 5 x 5 mm size, which means that there are no external pins, but instead there are pads at the bottom of the chip. ESP8266 SoC pinout diagram is shown by Figure 2. The processor communicates with the peripherals through standard interfaces, i.e., UART, SPI (Espressif, 2020a).

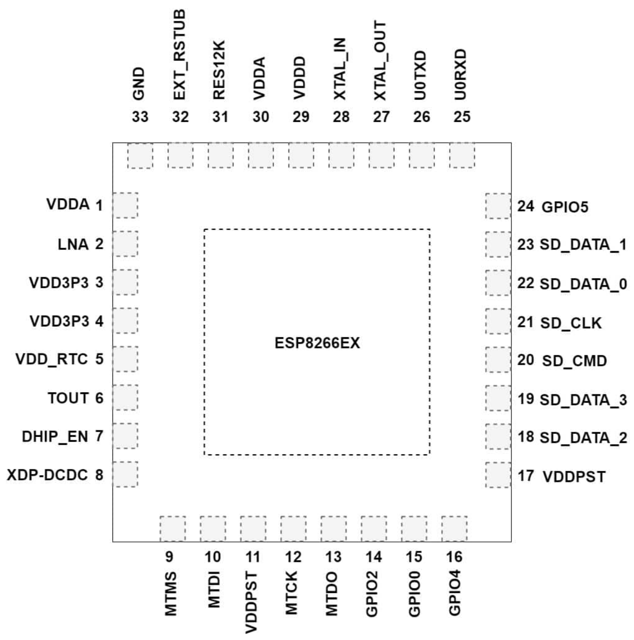


Figure 2. ESP8266 SoC Pinout Diagram

ESP8266 SoC typically draws about 80 mA in idle state and 170 mA while in operation. The operating voltage ranges from 2.5 V to 3.6 V DC. It also supports the light sleep, modem sleep and deep sleep modes with current of 0.5 mA, 15 mA and 0.1 mA, respectively. The peak operating current can reach to 320 mA in “boot up” mode or ‘wakes up’ from a sleep mode. When compared with ordinary Wi-Fi adapters, the ESP8266 SoC has a relatively lower power consumption that ensures the battery life can be typically up to 3 months (Espressif, 2020b).

Although ESP8266 is the name of the basic chip, it can be also bought in two different varieties including the module and development board, as shown in Figure 3. ESP8266

SoC is manufactured unshielded and its package is usually used by soldering onto a circuit board. So, it is unsuitable for hobbyists and it is recommended to be used as a module when mass production is not required. The ESP8266 modules are surface-mountable and enable the microcontrollers to connect to the Internet by using IEEE 802.11 b/g/n Wi-Fi standards with WPA/WPA2 certifications. On the other hand, the development boards are produced by several manufacturers with different specifications to develop IoT projects rapidly.

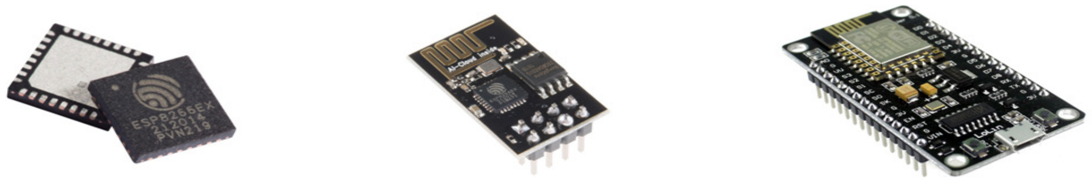


Figure 3. ESP8266 SoC, Module (ESP-01) and Development Board (NodeMCU v3).

Today, Espressif produces the ESP-WROOM-02 (Espressif, 2020c) and ESP-WROOM-S2 (Espressif, 2020d) modules, based on ESP8266 SoC. Besides, several third party manufacturers started to produce custom modules and boards based on the Espressif’s ESP8266 SoC as the main controller. Ai-Thinker is an officially licensed manufacturer of Espressif modules that are labeled ESP-XX. The ESP-01 is the most basic and cheapest Wi-Fi SoC made by Ai–Thinker in August 2014 (Ai-Thinker, 2018a). Thanks to small form factor, it fits into any enclosure that has limited spacing. However ESP-01 module has only two digital general-purpose input/output pins (GPIO0 and GPIO2) and no analog pin. It can be integrated with the sensors and other application specific devices via GPIO pins. Besides, two additional pins (GPIO1-Tx and GPIO3-Rx) are available for serial communication. The pinout diagram of ESP-01 module and its breadboard adapters are shown by Figure 4.

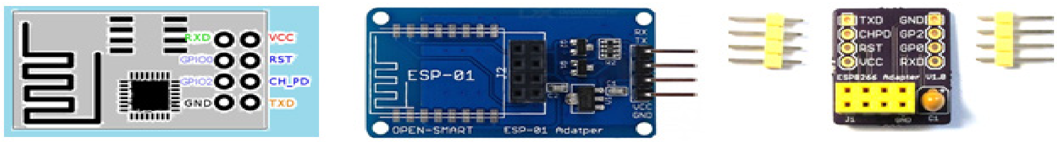


Figure 4. ESP-01 Module Pinout Diagram and Two Different Breadboard Adapters.

In response to high demand for ESP-01 module, updated versions from ESP-02 to ESP-14 were released by Ai-Thinker. All of these modules are designed for the needs of IoT applications and they are working on the ESP8266 SoC. A typical ESP-XX module enables Internet connectivity that can be required by host applications and to create a Wi-Fi network that can be used to exchange data and instructions. But, the main differences are the number of GPIO pins, pitch (space between pins), form factor and antenna connection type. The specifications of available modules are given in Table 2.

Table 2. The ESP8266 Modules Based on ESP8266 SoC.

Vendor	Module	GPIO	Pitch	Form factor	Antenna
Espressif	ESP-WROOM-02	18	1.5 mm	2×9 castellated	PCB trace
	ESP-WROOM-02D	18	1.5 mm	2×9 castellated	PCB trace
	ESP-WROOM-02U	18	1.5 mm	2×9 castellated	U.FL socket
	ESP-WROOM-S2	20	1.5 mm	2×10 castellated	PCB trace
Ai-Thinker	ESP-01	6	0.1 in	2×4 DIL	PCB trace
	ESP-02	6	0.1 in	2×4 castellated	U.FL socket
	ESP-03	10	2 mm	2×7 castellated	Ceramic
	ESP-04	10	2 mm	2×4 castellated	External
	ESP-05	3	0.1 in	1×5 SIL	U.FL socket
	ESP-06	11	misc	4×3 dice	External
	ESP-07	14	2 mm	2×8 pinhole	Ceramic + U.FL socket
	ESP-08	10	2 mm	2×7 castellated	External
	ESP-09	10	misc	4×3 dice	External
	ESP-10	3	2 mm	1×5 castellated	External
	ESP-11	6	1.27 mm	1×8 pinhole	Ceramic
	ESP-12	14	2 mm	2×8 castellated	PCB trace
	ESP-13	16	1.5 mm	2×9 castellated	PCB trace
	ESP-14	22	2 mm	2×8 castellated +6	PCB trace

ESP-07 is an improved version of the ESP-01 module with more functionalities (Ai-Thinker, 2018b). It comes with a UFL connector that enables to connect an external antenna in case of need to boost the received Wi-Fi signal. ESP-07 module has 16 pins and 11 of them operate as a GPIO pins. It has also an analog input pin with an input range of 0 to 1V. GPIO 1 and GPIO 3 pins are used for TX and RX of serial communication interface. Besides, SPI interface works on GPIO12 (MISO), GPIO13 (MOSI) and GPIO14 (CLK) pins. Figure 5 shows the ESP-07 and its pinout diagram. Since the ESP-07 has a 2.0mm spacing between the pads, it is not breadboard friendly. But, it can be transformed to a breadboard compatible form through a PCB shield with 2.54 mm spacing between the pins. This makes the prototyping easier for lots of cases. Also, it is possible to program the ESP-07 module via FTDI programmer based on USB to TTL serial converter module, i.e., FT232RL chip.

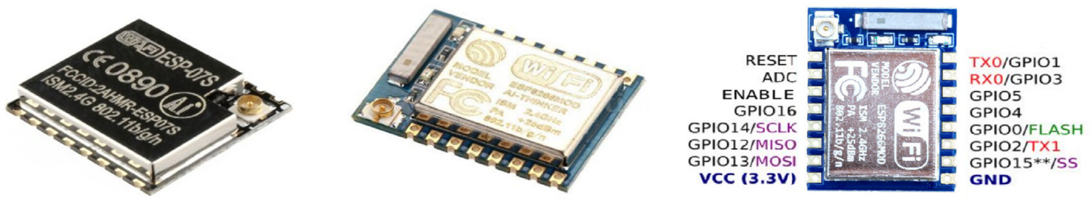


Figure 5. ESP-07 Module and Its Pinout Diagram.

When more pins are needed in the application to be implemented, ESP-12 module is an ideal choice to provide reliable Wi-Fi connectivity. ESP-12 is now outdated, although it rarely exists on the market. ESP-12E (Ai-Thinker, 2015) and ESP-12F (Ai-Thinker, 2018c) modules are the improved versions of the ESP8266-12. They are basically identical to each other, with the difference that ESP-12F has a better antenna design which gives a longer range. ESP-12E includes 11 GPIO pins, analog input serial communication and SPI interfaces. Besides, ESP-12S (Ai-Thinker, 2019a) module is a compact version of the ESP-12F, which has only 9 GPIO pins. Figure 6 shows the ESP-12 series modules and the pinout diagram of popular ESP-12E module. As seen in ESP-01 module, GPIO0 and GPIO2 pins are used to control the bootloader mode. These pins should be taken into account when preparing the application firmware.

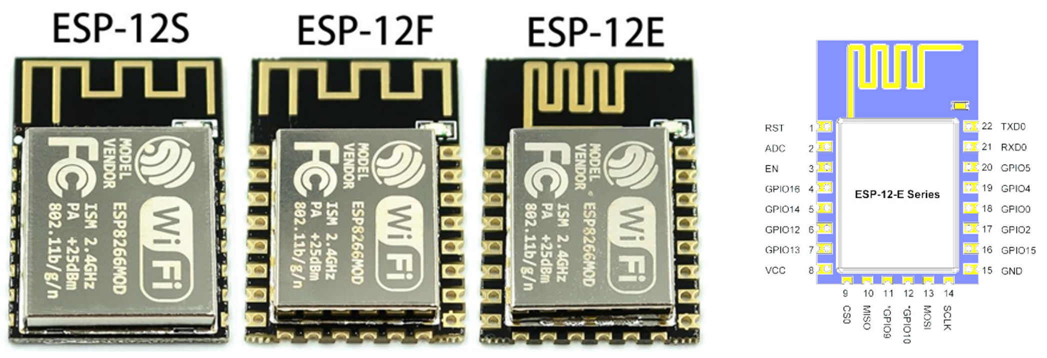


Figure 6. ESP-12 Series Modules and ESP-12E Pinout Diagram.

When a developer does not have enough experience working with ESP8266 module, employment of a development board is a more convenient for complex IoT projects. There are wide variety of ESP8266 development board options are available in the market. NodeMCU is a widely used development board, which is based on ESP8266 as well as ESP32 SoC. It makes the prototyping stage much easier. As shown in Figure 7, NodeMCU boards currently produced by three different producers, namely Amica, LoLin and DOIT.

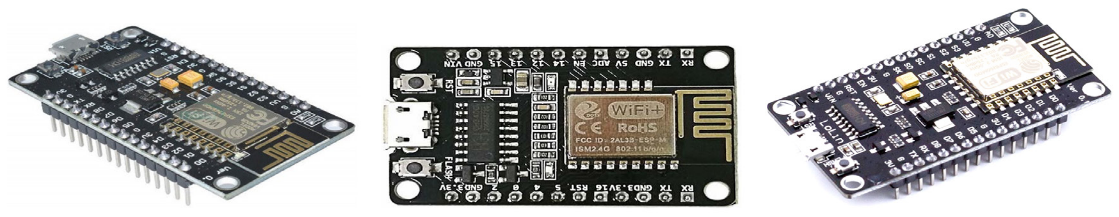


Figure 7. NodeMCU Development Boards from Amica, DOIT and Lolin.

A typical NodeMCU board has got two more chips for USB to serial converter and voltage regulator from 5V to 3.3V. According to their hardware configurations, NodeMCU development boards have been separated into two different generations and three different versions, currently ranging from v1 to v3. The first generation NodeMCU v1 board has CP2102 USB to UART bridge chip, but it is currently outdated. The second generation NodeMCU v2 and NodeMCU v3 boards are produced by Amica/DOIT and LoLin, respectively. Note that, while the breadboard friendly NodeMCU v2 boards have CP210x, the larger sized NodeMCU v3 is equipped with CH34x USB to serial chip

(Stör, 2015). The pinout diagrams of different version NodeMCU boards are shown by Figure 8.

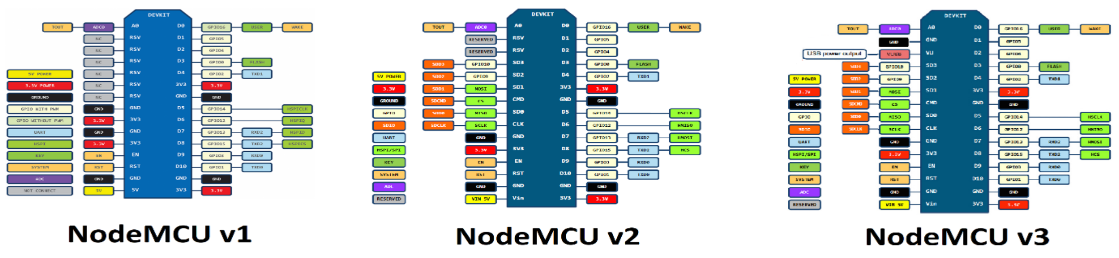


Figure 8. Pinout Diagrams of Different Nodemcu Development Board Versions.

NodeMCU boards can be supplied with the on-board USB connector. They have an onboard LED (connected to GPIO2 pin) and two buttons for Flash and Reset operations. The build-in blue LED is programmable by users to test the certain routines in the implemented algorithm. Flash button is used to download a new program and reset button restarts the microcontroller operation. It is easy to program the NodeMCU board by directly plugging it into the computer via Micro-B type USB connector. Additionally, it is possible to update the firmware of ESP8266 modules wirelessly by using Over-The-Air (OTA) programming feature. Clearly, the OTA functionality also allows uploading a new program to ESP8266 module by using Wi-Fi connection without requiring to connect the NodeMCU board to a computer via USB cable. As long as they share the same network, the firmware of multiple ESP8266 modules can be updated simultaneously from a central server (Bakolia, 2019).

ESP32

ESP32 (2016) is a hybrid Wi-Fi and Bluetooth SoC that integrates Tensilica Xtensa LX series 32-bit RISC processor with a TCP/IP stack. As a successor of the ESP8266, ESP32 series SoCs have dual core processor, more GPIO pins, faster Wi-Fi and BLE protocol support (Kurniawan, 2019). The processor communicates with the peripherals through standard interfaces, i.e., UART, SPI and I2C. In normal (active) mode, ESP32 SoC requires 160-260 mA current to operate at 3.3 V DC supply voltage. When Wi-Fi and Bluetooth functions are open, the current drawn can spike up to 790 mA. Although the ESP32 SoC is relatively power-hungry, the current drawn can be cut down by leveraging one of its sleep modes (modem sleep, light sleep, deep sleep and hibernation). Figure 9 shows the ESP32 SoC housed in QFN-49 package and its pinout diagram (Kolban, 2018).

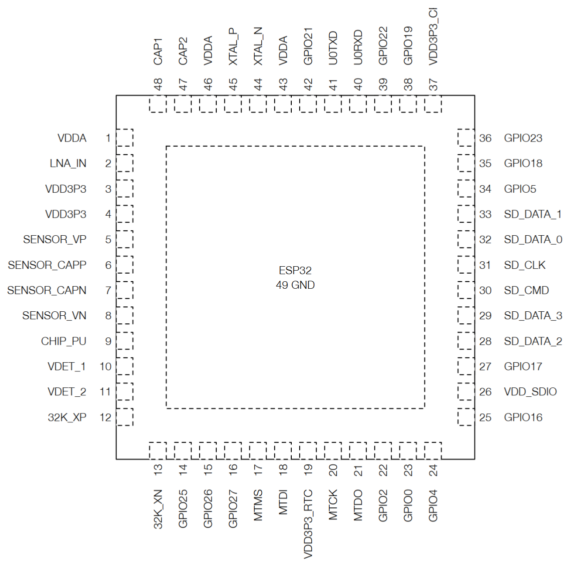


Figure 9. ESP32 SoC Pinout Diagram

ESP32 can be bought as a chip as well as module or development board forms, as shown in Figure 10. There are a number of variants of basic ESP32 SoC (ESP32-D0WD-V3, ESP32-D0WDQ6-V3, ESP32-D0WD, ESP32-D0WDQ6, ESP32-D2WD, ESP32-S0WD and ESP32-U4WDH) have been introduced to meet the different needs of embedded system developers (Espressif, 2021c). Nevertheless, Espressif Systems has been upgraded the ESP32 SoC in 2020. The latest ESP32 V3 SoC, is known as ECO V3, offers a new silicon wafer as well as the secure boot and flash encryption. The ESP32-D0WD-V3, ESP32-D0WDQ6-V3 and ESP32-U4WDH SoCs are produced based on ECO V3 wafer (Espressif, 2020e). It is noteworthy that all of the ESP32 SoC variants share the same software development tools and they are largely code-compatible. So, the firmware developed for one particular ESP32 SoC can be run on all others without significant modification.

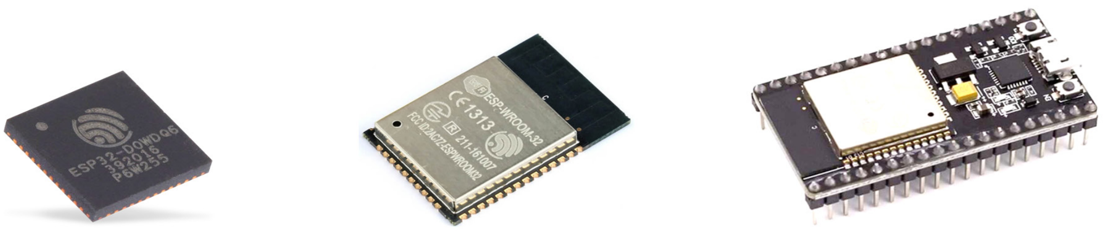


Figure 10. ESP32 SoC, Module (ESP32-WROOM-32) and Development Board (NodeMCU ESP32-S).

The specifications of the advanced ESP32 SoC versions can be summarized as follows:

- ESP32-S2 (2019, 43 programmable GPIOs) is a single-core Xtensa LX7 processor which has only Wi-Fi 4 (802.11 b/g/n) connectivity (Espressif, 2021d).
- ESP32-S3 (2020, 44 programmable GPIOs) has a dual core XTensa LX7 processor with integrated Wi-Fi 4 and Bluetooth 5 (LE) connectivity. It has been designed to address the needs of the Artificial Intelligence of Things (AIoT) applications

(Espressif, 2021e).

- ESP32-C3 (2020, 22 programmable GPIOs) is a reduced version of the ESP32 and it has single-core RISC-V processor with Wi-Fi 4 and Bluetooth 5 (LE) connectivity (Espressif, 2021f).

- ESP32-C6 (2021, 22 programmable GPIOs) integrates a single core RISC-V processor with Wi-Fi 6 (802.11ax) and Bluetooth 5 (LE) connectivity. Thanks to the 802.11ax mode support, ESP32-C6 offers longer transmission range and lower power consumption for battery-driven IoT devices. Furthermore, it supports both the OFDMA (Orthogonal Frequency Division Multiple Access) and MU-MIMO (Multi User, Multiple Input Multiple Output) antenna techniques to provide the high spectral efficiency and low transmission latency in congested environments (Espressif, 2021g).

ESP32 modules offer an extended flexibility and customization opportunity in the revision of firmware for embedded system applications. ESP32-WROOM-32 is a popular module that can be used a wide variety of IoT applications. It is equipped with Wi-Fi and Bluetooth capabilities powered with ESP32-D0WDQ6 SoC and 4 MB integrated SPI Flash memory. It integrates a rich set of peripherals through ADC, DAC, SPI, UART, I²S, I²C, PWM, SD card and Ethernet interfaces (Espressif, 2021h). ESP32-WROOM-32 module is produced with different antenna types. While ESP32-WROOM-32U has a connector that allows us to connect an external antenna, ESP32-WROOM-32D, ESP32-WROOM-32E and ESP32-WROOM-32UE modules come with MIFA, PCB and IPEX antennas, respectively. Different ESP32-WROOM-32 module variants are shown in Figure 11.

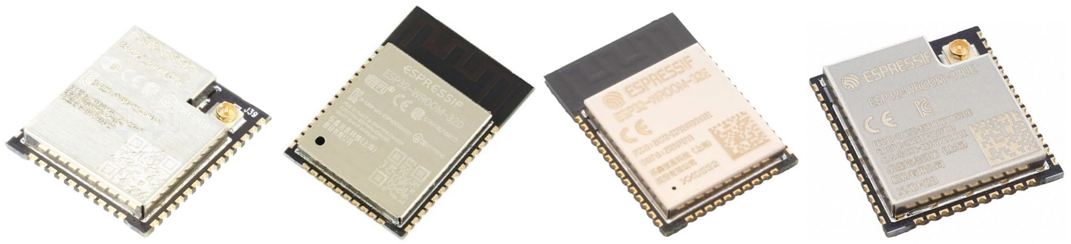


Figure 11. U, D, E and UE Variants of ESP32-WROOM-32 Module.

There is also another ESP32 module variant from Espressif Systems, namely ESP32-WROVER, which has dual core ESP32-D0WDQ6 SoC and its clock frequency can be adjustable from 80 MHz to 240 MHz. The distinctive feature is that it has an additional 8 MB Pseudo-Static RAM (PSRAM) besides the 4 MB external SPI Flash memory. Note that the ESP32-WROVER-B version is equipped with ESP32-D0WD SoC and it offers 8 MB or 16 MB Flash memory options when the customer needs more resources (Espressif, 2021i). The front and back views of ESP32-WROVER and WROVER-B modules are shown by Figure 12.

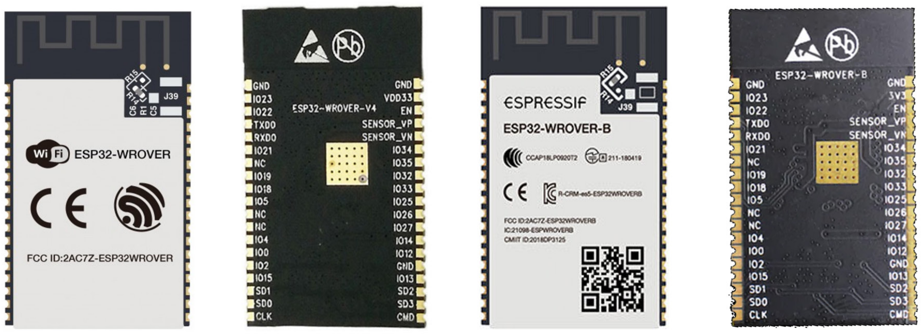


Figure 12. ESP32-WROVER and WROVER-B Modules.

On the other side, Ai-Thinker produces the ESP32-S which is an original ESP32 series module with dual core Xtensa 32-bit LX6 processor. The clock frequency is adjustable from 80 MHz to 240 MHz. It supports UART, SPI, SDIO, I2C, PWM, I2S, IR, ADC and DAC interfaces with 22 GPIO pins. It also supports the Wi-Fi (802.11 b/g/n/e/i) and Bluetooth (dual mode BR/EDR + BLE v4.2) connectivity. In terms of hardware capabilities, ESP32-S can be considered as equivalent to ESP-WROOM-32 module of Espressif Systems. Note that ESP32-S module comes with on-board PCB antenna and IPEX connector for external antenna (Ai-Thinker, 2016). The front and back views of ESP32-S module are shown by Figure 13.

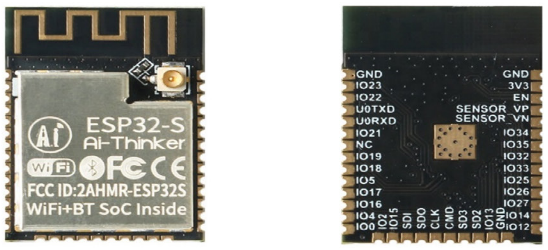


Figure 13. The front and back Views of Ai-Thinker ESP32-S Module.

When considering the challenges of working with chips and modules, the ESP32 SoC based development boards help embedded designers to rapidly prototype and test their IoT applications. Ai-Thinker NodeMCU ESP32-S is a compact prototyping board, which has 38 programmable GPIOs and built-in 2-channel 12-bit high-precision ADC with up to 18 channels (Ai-Thinker, 2019b). The pinout of the NodeMCU ESP32-S board can be seen in Figure 14.

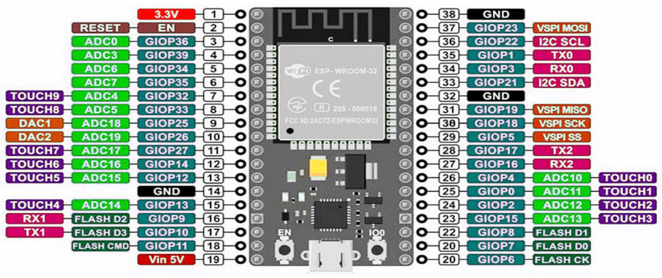


Figure 14. Ai-Thinker NodeMCU ESP32-S Development Board Pinout Diagram.

Programming the ESP8266 and ESP32 SoC Microcontrollers

Espressif provides an official software development framework, called as Espressif ESP-IDF (IoT Development Framework), to develop the IoT applications with wireless connectivity (Espressif, 2021j). The 'Getting Started Guide' can be seen from the below link:

https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started

It is possible to download the ESP-IDF from GitHub repository:

https://github.com/espressif/esp-idf

Although it has complex programming environment, ESP-IDF offers maximum functionality and compatibility. ESP-IDF contains the ESP32-specific API (software libraries and source code), but it requires a toolchain to build the application. Eclipse is an ideal development environment. ESP-IDF also requires some prerequisite tools to build firmware for supported SoCs. ESP-IDF Tools Installer is simple solution to install the prerequisites. It can be downloadable from the following link:

https://dl.espressif.com/dl/esp-idf/

Espressif has also been developed the ESP-AT firmware based on ESP-IDF, which contains a set of AT commands to integrate connectivity for IoT applications. AT commands is the most primitive way to program the Espressif SoC microcontrollers (Espressif, 2021k). The ESP-AT firmware has in-built TCP/IP stack and data buffering functions. By this way, it is possible to quickly join the wireless network, connect to the cloud platform, realize data transmission and remote control functions and realize the interconnection of everything through wireless communication easily. The host microcontroller is connected to the ESP SoC via UART interface to sends/receives AT commands/responses. The following link contains several firmware files in binary format:

https://www.espressif.com/en/support/download/at

Each ESP-AT-Bin file contains several binaries for some specific functions and the binary file in factory folder is the combination of all binaries. The guide in (Espressif, 2020f) demonstrates how to download AT firmware and flash it into an ESP SoC microcontroller.

Furthermore, ESP-IDF allows the FreeRTOS to efficiently use of the processor(s) and manage the built-in peripherals. Both the ESP8266 and ESP32 SoCs are capable of support the FreeRTOS, which is a light-weight RTOS (Real-Time Operating System). Thanks to the RTOS scheduler, multiple threads are executed simultaneously by switching between tasks. So, multitasking can be performed effectively (Espressif, 2021m).

ESP8266 and ESP32 SoC microcontrollers can be programmed in various programming languages: Arduino, MicroPython, LUA and JavaScript.

Arduino Programming

Both ESP8266 and ESP32-based boards can be programmed with the Arduino IDE based on C++ programming language. The 'Getting Started Guide' can be seen from the below link:

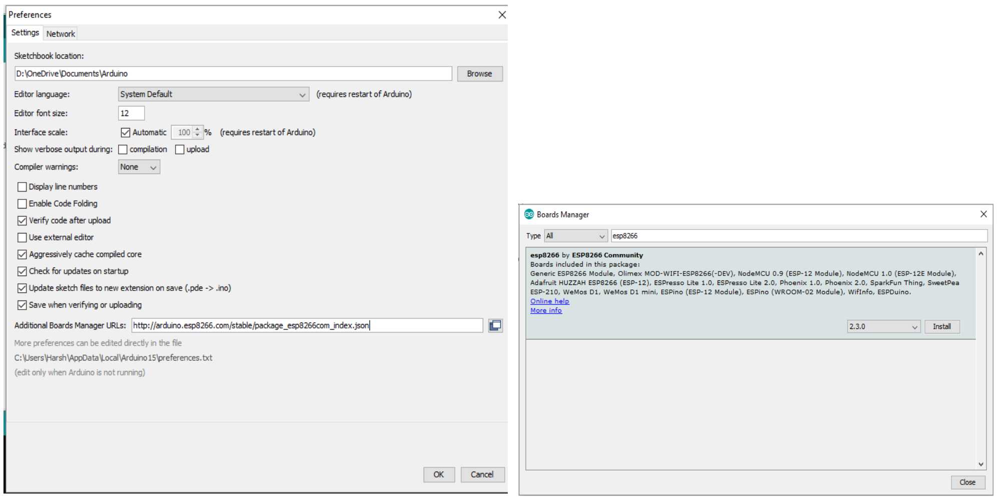
https://github.com/espressif/arduino-esp32

In order to code by using Arduino IDE, firstly the NodeMCU should be installed in supported boards. For this, the following links are added to File > Preferences window of Arduino IDE:

https://dl.espressif.com/dl/package_esp32_index.json

http://arduino.esp8266.com/stable/package_esp8266com_index.json

After that, NodeMCU boards are updated using Tools > Board > Boards Manager window. When clicked the Install button, Arduino IDE downloads all the required libraries for development boards. So, ESP32 boards are listed in the Tools > Board menu. Figure 15 shows the installing procedure of the ESP8266 package into Arduino IDE.


Preferences

Settings | Network

Sketchbook location: D:\OneDrive\Documents\Arduino [Browse]

Editor language: System Default (requires restart of Arduino)

Editor font size: 12

Interface scale: [✓ Automatic] 100 % (requires restart of Arduino)

Show verbose output during: ☐ compilation ☐ upload

Compiler warnings: None

☐ Display line numbers

☐ Enable Code Folding

☑ Verify code after upload

☐ Use external editor

☑ Aggressively cache compiled core

☑ Check for updates on startup

☑ Update sketch files to new extension on save (.pde -> .ino)

☑ Save when verifying or uploading

Additional Boards Manager URLs: http://arduino.esp8266.com/stable/package_esp8266com_index.json

More preferences can be edited directly in the file

C:\Users\Harsh\AppData\Local\Arduino15\preferences.txt

(edit only when Arduino is not running)

[OK] [Cancel]

Boards Manager

Type: All | esp8266

esp8266 by **ESP8266 Community**

Boards included in this package:

Generic ESP8266 Module, Olimex MOD-WIFI-ESP8266(-DEV), NodeMCU 0.9 (ESP-12 Module), NodeMCU 1.0 (ESP-12E Module), Adafruit HUZZAH ESP8266 (ESP-12), ESPresso Lite 1.0, ESPresso Lite 2.0, Phoenix 1.0, Phoenix 2.0, SparkFun Thing, SweetPea ESP-210, WeMos D1, WeMos D1 mini, ESPino (ESP-12 Module), ESPino (WROOM-02 Module), WifInfo, ESPDuino.

Online help

More info

2.3.0 [Install]

[Close]

Figure 15. Installing the ESP8266 Package into Arduino IDE.

MicroPython Programming

MicroPython is a programming language written in C language, which is optimized to run on popular microcontrollers. As a lightweight implementation of Python 3, it includes a small subset of the Python standard library (Magda, 2017). MicroPython aims to simplify the programming of microcontrollers while simultaneously improving the performance and efficiency of code blocks. MicroPython runs on both the ESP8266 and ESP32 SoC microcontrollers. When programming the microcontrollers with MicroPython firmware, the programmer needs an IDE as an essential tool which is used for writing, testing, debugging and compiling program codes. Thonny is an open-source IDE which is used to write and upload the MicroPython codes to microcontroller. It can be downloaded from https://thonny.org link. The latest and stable versions of the firmware for ESP8266 and ESP32-based boards can be downloaded from following links:

https://micropython.org/download/esp8266

https://micropython.org/download/esp32

NodeMCU Firmware with LUA Programming

LUA is an open source scripting language which has many application programming interface (API) for network programming. LUA includes a stand-alone interpreter and works embedded in a host application within the application. The further details can be seen from the below link:

http://www.lua.org

NodeMCU is an LUA based firmware for the Espressif SoC microcontrollers. It can be downloadable from GitHub repository:

https://github.com/nodemcu/nodemcu-firmware

NodeMCU supports many Espressif modules, but only the required module is selected due to resource constraints. Also, the automated custom firmware build service can be used to get the specific firmware configuration. This service can be accessible via the below link:

https://nodemcu-build.com

ESPlorer is a Lua based IDE used to develop applications for NodeMCU. It allows to establish serial communication with Espressif modules, send commands and upload code. Note that, ESPlorer requires a Java Runtime Environment (JRE) to run. The latest ESPlorer IDE can be found in the following GitHub repository:

https://github.com/4refr0nt/ESPlorer

NodeMCU Flasher is a firmware programmer to program the Espressif modules. It enables the easily transferring the firmware files to the Espressif modules. The download link is given below:

https://github.com/nodemcu/nodemcu-flasher

The Arduino IDE uses the GPIO number to address each pin. However, GPIO pins renamed for NodeMCU firmware on LUA language. The GPIO pin mapping from NodeMCU to Arduino is shown by Table 3.

Table 3. NodeMCU to Arduino Pin Mapping.

NodeMCU	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12
Arduino	16	5	4	0	2	14	12	13	5	3	1	9	10

JavaScript Programming

JavaScript is another language option to program the ESP8266 and ESP32-based boards. But, as in the use of MicroPython and LUA programming, JavaScript requires a firmware that defines the code blocks to microcontroller. Clearly, the firmware run like as an operating system of the computer. Espruino is a JavaScript interpreter which is designed to run on microcontrollers with constrained memory limits, i.e., 128 kB Flash and 8 kB RAM. The latest version of Espruino IDE (also Espruino Web IDE) can be found in following link:

https://www.espruino.com/

But, the setup of Espruino IDE depends on the node.js with NPM (Node Package Manager). Node.js is an open source and cross-platform runtime environment that executes JavaScript code blocks outside a web browser. Besides, NPM is the official package manager for Node.js environment. The details can be seen from https://nodejs.org link. Also, a detailed guideline can be read on https://www.espruino.com/programming web site.

Conclusion

This chapter has been discussed the technical specifications and programming abilities of ESP8266 and ESP32 series of SoC microcontrollers from Espressif Systems. Thanks to Wi-Fi and Bluetooth based wireless connectivity capabilities, these microcontrollers can connect the smart things to the Internet. Clearly, IoT projects can be easily prototyped with the help of ESP8266 and ESP32 SoC microcontrollers. This chapter has also presented an overview of the different available IDEs. Although there are so many programming environments, all of them have generally similar characteristics. Hence, embedded developers can adapt themselves to new environments. So they can rapidly evolve their suite of IoT devices and services.

References

Ai-Thinker. (2015). ESP-12E Datasheet v1. https://docs.ai-thinker.com/_media/esp8266/docs/esp12e_datasheet.pdf

Ai-Thinker. (2016). ESP-32S Datasheet. https://cdn.ozdisan.com/ETicaret_Dosya/632827_168142.pdf

Ai-Thinker. (2018a). ESP-01E Datasheet v1. https://docs.ai-thinker.com/_media/esp8266/docs/esp-01e_product_specification_en.pdf

Ai-Thinker. (2018b). ESP-07S Datasheet v1. https://docs.ai-thinker.com/_media/esp8266/docs/esp-07s_product_specification_en.pdf

Ai-Thinker. (2018c). ESP-12F Datasheet v1. https://docs.ai-thinker.com/_media/esp8266/docs/esp-12f_product_specification_en.pdf

Ai-Thinker. (2019a). ESP-12S Datasheet v1. https://docs.ai-thinker.com/_media/esp8266/docs/esp-12s_product_specification_en.pdf

Ai-Thinker. (2019b). Nodemcu-32s Datasheet v1. https://docs.ai-thinker.com/_media/esp32/docs/nodemcu-32s_product_specification.pdf

Al-Fuqaha, A., Guizani, M., Mohammadi, M., & Aledhari, M. (2015). Internet of Things: A Survey on Enabling Technologies, Protocols and Applications. IEEE Communications Surveys & Tutorials, 17(4), 2347 - 2376. https://doi.org/10.1109/COMST.2015.2444095

Bakolia, A. (2019). Programming NodeMCU ESP8266 Over-the-Air (OTA) using Arduino IDE. https://circuitdigest.com/microcontroller-projects/esp8266-ota-update-programming-using-arduino-ide

Espressif Systems. (2020a). ESP8266 Technical Reference. https://www.espressif.com/sites/default/files/documentation/esp8266-technical_reference_en.pdf

Espressif Systems. (2020b). ESP8266EX Datasheet v6.6. https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf

Espressif Systems. (2020c). ESP-WROOM-02 Datasheet v3.2. https://www.espressif.com/sites/default/files/documentation/0c-esp-wroom-02_datasheet_en.pdf

Espressif Systems. (2020d). ESP-WROOM-S2 Datasheet v2.4. https://www.espressif.com/sites/default/files/documentation/esp-wroom-s2_datasheet__en.pdf

Espressif Systems. (2020e). ESP32 ECO V3 User Guide v1.1. https://www.espressif.com/sites/default/files/documentation/ESP32_ECO_V3_User_Guide__EN.pdf

Espressif Systems. (2020f). Downloading Guide. https://docs.espressif.com/projects/esp-at/en/latest/Get_Started/Downloading_guide.html

Espressif Systems. (2021a). SoCs. https://www.espressif.com/en/products/socs

Espressif Systems. (2021b). ESP8266. https://www.espressif.com/en/products/socs/esp8266

Espressif Systems. (2021c). ESP32 Series Datasheet v3.7. https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

Espressif Systems. (2021d). ESP32-S2 Family Datasheet v1.3. https://www.espressif.com/sites/default/files/documentation/esp32-s2_datasheet_en.pdf

Espressif Systems. (2021e). ESP32-S3. https://www.espressif.com/en/products/socs/esp32-s3

Espressif Systems. (2021f). ESP32C3 Family Datasheet v1.0. https://www.espressif.com/sites/default/files/documentation/esp32-c3_datasheet_en.pdf

Espressif Systems. (2021g). Announcing ESP32-C6: a Wi-Fi 6 + Bluetooth 5 (LE) SoC. https://www.espressif.com/en/news/ESP32_C6

Espressif Systems. (2021h). ESP32WROOM32 Datasheet v3.1. https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf

Espressif Systems. (2021i). ESP32WROVER Datasheet v2.4. https://www.espressif.com/sites/default/files/documentation/esp32-wrover_datasheet_en.pdf

Espressif Systems. (2021j). Official IoT Development Framework. https://www.espressif.com/en/products/sdks/esp-idf

Espressif Systems. (2021k). ESP-AT. https://www.espressif.com/en/products/sdks/esp-at/overview

Espressif Systems. (2021m). FreeRTOS. https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/freertos.html

Kolban, N. (2016). Kolban's book on the ESP8266. https://leanpub.com/ESP8266_ESP32

Kolban, N. (2018). Kolban's book on ESP32. https://leanpub.com/kolban-ESP32

Kurniawan, A. (2019). Internet of Things Projects with ESP32. Packt Publishing.

Magda, Y. (2017). Programming ESP8266-based Wireless Systems in MicroPython.

Stör, M. (2015). Comparison of ESP8266 NodeMCU development boards. https://frightanic.com/iot/comparison-of-esp8266-nodemcu-development-boards

About the Authors

Hakkı SOY was born in Konya, Turkey in 1978. He received the B.S. degree in Electronics Engineering from Uludağ University, Bursa, Turkey in 1999 and the Ph.D. degree in Electrical and Electronics Engineering from Selçuk University, Konya, Turkey in 2013. He joined the Department of Electrical and Electronics Engineering, Necmettin Erbakan University, Konya, Turkey, where he is currently an Assistant Professor. His research interests include wireless networks, embedded systems, smart sensors, cyber-physical systems, mobile communication and IoT applications.

E-mail: hakkisoy@erbakan.edu.tr, Orcid: 0000-0003-3938-038

Similarity Index

The similarity index obtained from the plagiarism software for this book chapter is 14%.

To Cite This Chapter:

Soy, H. (2021). ESP8266 and ESP32 Series of SoC Microcontrollers. In S. Kocer, O. Dundar & R. Butuner (Eds .), *Programmable Smart Microcontroller Cards* (pp. 110 –125). ISRES Publishing.