

What is ESP32,
how it works and
what you can do with ESP32?

ESP32, like Arduino, is a development board. That means it has all the features you need to create your projects.

In order to better understand this board, you need to know **who created it?** and

what are its main functions?

How ESP32 works? and

what are its technical features.

ESP32 is far superior than [Arduino UNO](#) and an ESP8266, which does not mean that it is ideal for any project, since to get the most out of this microcontroller, let's learn what it is and how it works and what are its special features when compared with other microcontroller boards.

Table of Contents

- What is ESP32?
- Difference between ESP32 and ESP8266
- ESP32 Functional Blocks and Features
- ESP32 Architectural Block diagram
- Wireless Connectivity:
- Core
- Memory
- Encryption hardware accelerators
- Peripheral Features
- ESP32 Pinout diagram and Pins:
- Digital pins

- ADC (Analog to digital converters)
- DAC (Digital to Analog Converters)
- Capacitive Touch GPIOs
- RTC
- SD / SDIO / MMC driver
- UART
- I2C
- SPI
- Infrared remote controller
- PWM
- How to select an ESP32 development board?
- Other ESP32 Development Boards and Specialty Projects
- Conclusion about the ESP32 chip

What is ESP32?

ESP32 is created by Espressif Systems with a series of SoC (System on a Chip) and modules which are low cost with low power consumption.

This new ESP32 is the successor to the well-known ESP8266(became very popular with its inbuilt WiFi). ESP32 not only has Built in WiFi but also has Bluetooth and Bluetooth Low Energy. In other words we can define ESP32 as “ESP8266 on Steroids”.

ESP32 chip ESP32-D0WDQ6 is based on a Tensilica Xtensa LX6 dual core microprocessor with an operating frequency of up to 240 MHz.

The small ESP32 package has a high level of integrations such as:

- Antenna switches
- Balun to control RF
- Power amplifier
- Low noise reception amplifier
- Filters and power management modules

On top of all that, it achieves very low power consumption through power saving features including clock synchronization and multiple modes of operation. The ESP32 chip's quiescent current is less than 5 μA which makes it the ideal tool for your battery powered projects or IoT applications .

Difference between ESP32 and ESP8266

- ESP32 has an additional core compared to ESP8266
- Faster Wi-Fi
- Increased number of GPIO (input/output) pins
- Compatibility with Bluetooth 4.2 and Bluetooth low energy (low energy).

Additionally, the ESP32 comes with touch-sensitive pins that can be used to “wake up” the ESP32 from deep sleep mode and a built-in Hall effect sensor.

Although both boards are extremely cheap, the ESP32 is slightly more expensive than ESP8266. ESP32 deserves it as it has more features than ESP8266.

We have differentiated the main technical specifications between the **ESP8266** and **ESP32** in the below table.

CHARACTERISTICS	ESP8266	ESP32
Microprocessor	Xtensa Single-core 32-bit L106	Xtensa Dual-Core 32-bit LX6 con 600 DMIPS
Wi-Fi (802.11 b/g/n)	HT20	HT40
Bluetooth	Does not have	Bluetooth 4.2 y BLE
Operating frequency (typical value)	80 MHz	160 MHz
SHAME	Not Available	448 KB
Flash	Not Available	520 KB
GPIO	17	34
PWM (hardware)	Not Available	Not Available
PWM (software)	8 channels	16 channels
SPI	2	4
I2C	1	2
I2S	2	2
UART	2	2
ADC	10-bit resolution	12-bit resolution
CAN	No	Yes
Interface MAC Ethernet	No	Yes
Touch sensor	No	Yes
Temperature sensor	No	Yes (only the old versions)
Hall effect sensor	No	Yes
Work temperature	-40°C to 125°C	-40°C to 125°C

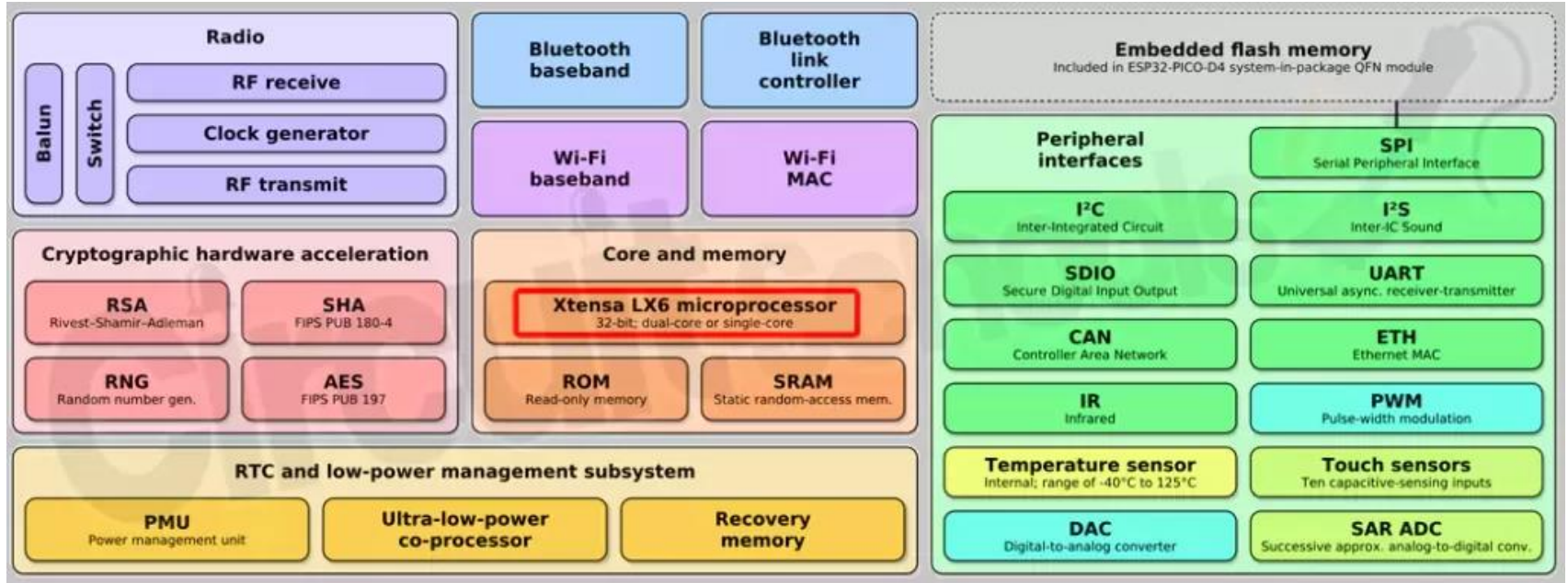
ESP32 Functional Blocks and Features

Although in the previous table you can notice some main technical characteristics of the ESP32, the truth is not everything is in the table. In fact, many details are missing. To get to know all the features of this **magnificent SoC** it is necessary to refer the below documents offered by developer **Espressif**.

- [ESP32 Technical Datasheet](#)
- [ESP32 Technical Reference Manual](#)

ESP32 Architectural Block diagram

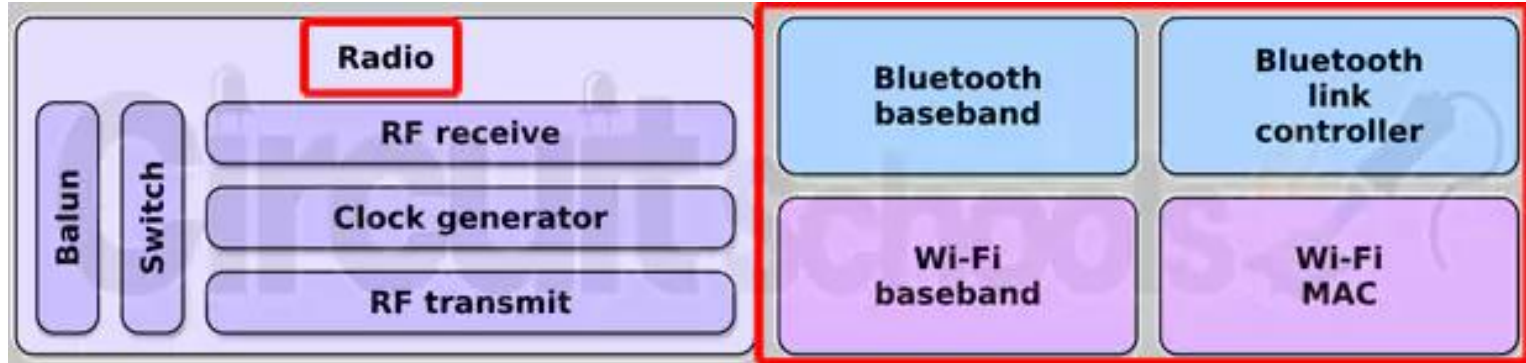
Below is the Architectural block diagram of ESP32 which shows all the functional blocks of ESP32 SOC.



esp32 architectural block diagram lets learn all the individual blocks in detail below.

Wireless Connectivity:

The ESP32 SoC chip has WiFi connectivity, being compatible with 802.11 b / g / n in the 2.4GHz band, reaching speeds of up to 150 Mbits/s. It also includes Bluetooth communication compatible with Bluetooth v4.2 and Bluetooth Low Energy (BLE).



The **radio block** is closely tied to the **wireless communication modules** .

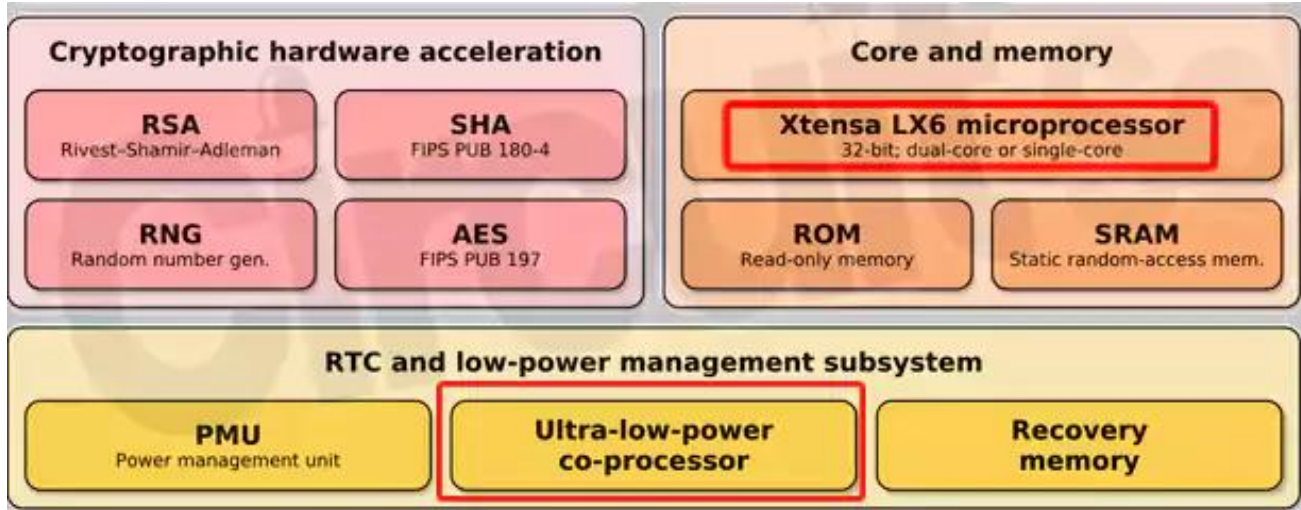
In fact, this is the one that actually transmits and receives the information.

That is, it takes the digital data from the **WiFi** and **Bluetooth modules** and converts them into electromagnetic signals that travel through the air to communicate with your **mobile phone or your *router*** .

It also performs the reverse operation: translate the electromagnetic waves generated by other devices into **digital data** that the **WiFi and Bluetooth modules** are capable of interpreting.

Core

As we have already mentioned that the ESP32 has **dual core low-power Tensilica Xtensa 32-bit LX6** microprocessors.



As you can observe from the above core block image, it has an **ultra-low-power co-processor** that is used to perform **analog-digital conversions** and other operations while the device is operating in deep sleep low-power mode. In this way, a very low consumption by the SoC is achieved.

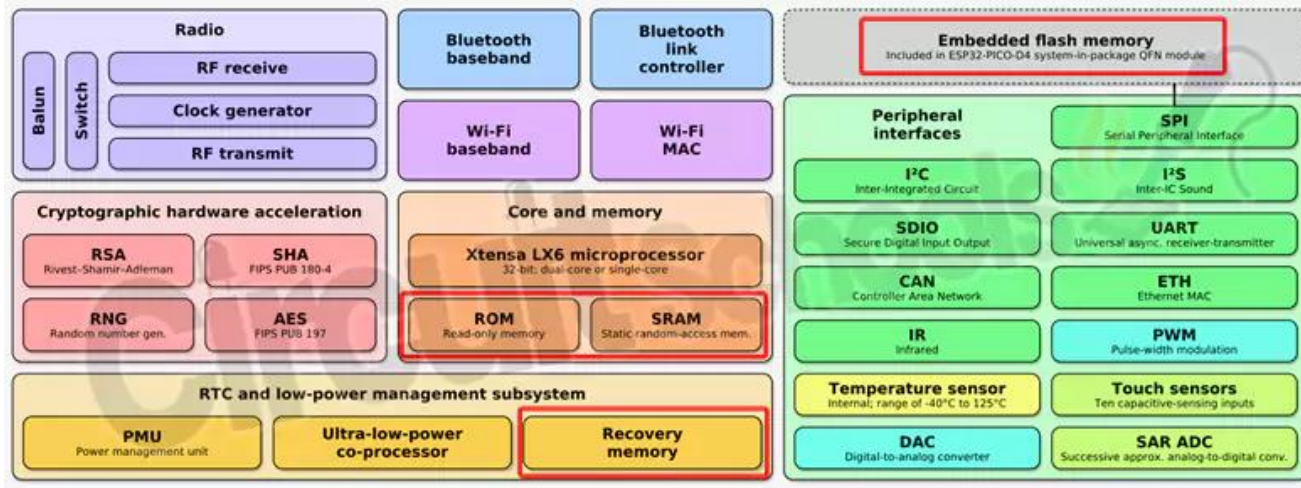
It is important to note that these processors offer great typical advantages of a digital signal processor :

- **Operating frequency** : 240 MHz (executes instructions 15 times faster than an Arduino UNO board)
 - It allows to perform operations with real numbers (numbers with commas) very efficiently.
 - Allows you to multiply large numbers instantly.
-
- Although these characteristics are transparent when programming, they have a great impact on the efficiency and size of the code that is written to the microcontroller.

Memory

In most of the microcontrollers based on Arduino, there are three types of memories:

- **Program memory** : to store the sketch.
- **SRAM memory** : to store the variables that are used in the code.
- **EEPROM memory** : to store variables that do not lose their value even when the device is turned off.



In ESP32 this does not happen, in fact there are more types of memories that are usually classified into internal and external .

The internal memories are those that are already included in the SoC, and the external are those that can be added to expand the capacity of the system .

Many ESP32- based development boards add external memory for a better performing system.

ESP32 Internal memories and their functions:

- ROM memory (448 KiB)** : this memory is write-only, that is, you cannot reprogram it. This is where the codes that handle the Bluetooth stack , the Wi-Fi physical layer control, some general-purpose routines, and the bootloader to start the code from external memory are stored .
- Internal SRAM memory (520 KiB)** : this memory is used by the processor to store both data and instructions. Its advantage is that it is much easier for the processor to access than the external SRAM.
- RTC SRAM (16 KiB)** : this memory is used by the co-processor when the device operates in deep sleep mode .
- Efuse (1 Kilobit)** : 256 bits of this memory are used by the system itself and the remaining 768 bits are reserved for other applications.

• **Flash embedded (Embedded flash)** : This memory is where our application code is stored. The amount of memory varies depending on the chip used:

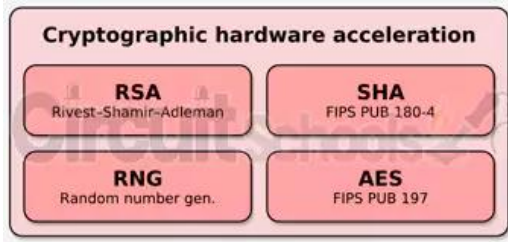
- 0 MB (chips ESP32-D0WDQ6, ESP32-D0WD, ESP32-S0WD)
- 2 MB (chip ESP32-D2WD)
- 4 MB (Chip ESP32-PICO-D4)

For ESP32s that do not have embedded memory or simply when memory is insufficient for your application, it is possible to add more memory externally :

- Up to 16 MB of external flash memory can be added . This way you can develop more complex applications.
- It also supports up to 8 MB of external SRAM memory .

Therefore, it is difficult for you to find yourself limited in memory when implementing an application using this platform.

Encryption hardware accelerators



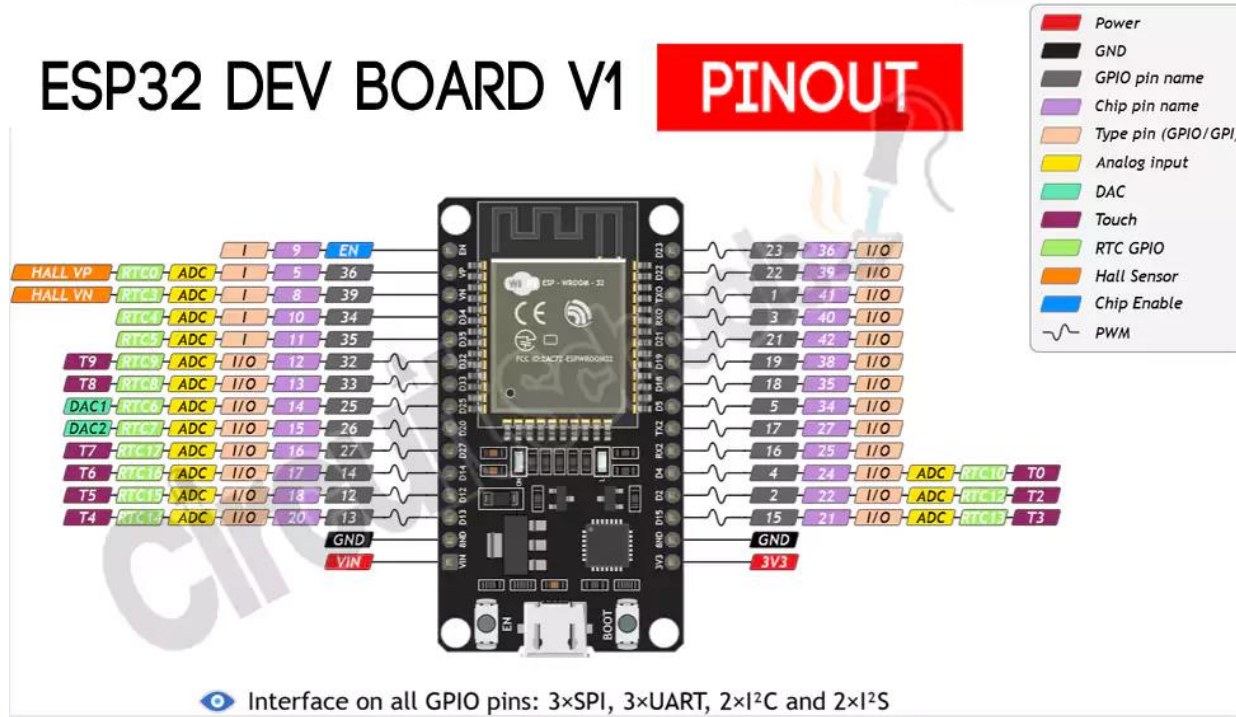
One of the most important factors in any system is security. That is why the **ESP32 has algorithm accelerators** aimed at encryption:

- AES (FIPS PUB 197)
- SHA (FIPS PUB 180-4)
- RSA
- ETC

These accelerators allow to increase the speed of operation and reduce the complexity of the software allowing **encryption and dynamic decryption**. In this way, the system is protected from possible hacking attacks that seek to obtain the stored code.

Peripheral Features

The ESP32 has a great set of peripherals you can see from the above Block diagram



As you can see from the above image of ESP32 WROOM module pinout diagram, all the different types of pins are mentioned in different colors which we are going to explain in detail below .

Digital pins

The ESP32 has a total of 34 digital pins . These pins are similar to Arduino digital pins which allows you to add LED display, OLED display, sensors, buttons, buzzers, etc. to our projects.

Most of these pins support the use of internal pull-up, pull-down, and high impedance status as well. This makes them ideal for connecting buttons and matrix keyboards, as well as for applying LED control techniques such as the well-known Charlieplexing.

ESP32 WROOM module has 25 GPIO pins out of which there are only input pins, pins with input pull up and pins without internal pullup.

Maximum current drawn per a single GPIO is 40mA according to the “Recommended Operating Conditions” section in the ESP32 datasheet.

Input only pins:	Pins with pull up INPUT_PULLUP	Pins without internal pull up
GPIO 34		
GPIO 35	GPIO14	GPIO13
GPIO 36	GPIO16	GPIO25
GPIO 39	GPIO17	GPIO26
	GPIO18	GPIO27
	GPIO19	GPIO32
	GPIO21	GPIO33
	GPIO22	
	GPIO23	

ADC (Analog to digital converters)

Some of the pins listed in the pinout diagram can also be used to interact with analog sensors, same as analog pins of an Arduino board .

For this, the ESP32 has a 12-bit(0-4096 resolution which means when voltage observed is 0 the value is 0 and when max voltage like 3.3v is observed the value goes to 4096), 18-channel analog to digital converter, which means you can take readings from up to 18 analog sensors.

This allows you to develop very compact connected applications, even when using multiple analog sensors.

Analog input pins:

ADC1_CH0 (GPIO 36)

ADC1_CH1 (GPIO 37)

ADC1_CH2 (GPIO 38)

ADC1_CH3 (GPIO 39)

ADC1_CH4 (GPIO 32)

ADC1_CH5 (GPIO 33)

ADC1_CH6 (GPIO 34)

ADC1_CH7 (GPIO 35)

ADC2_CH0 (GPIO 4)

ADC2_CH1 (GPIO 0)

ADC2_CH2 (GPIO 2)

ADC2_CH3 (GPIO 15)

ADC2_CH4 (GPIO 13)

ADC2_CH5 (GPIO 12)

ADC2_CH6 (GPIO 14)

ADC2_CH7 (GPIO 27)

ADC2_CH8 (GPIO 25)

ADC2_CH9 (GPIO 26)

DAC (Digital to Analog Converters)

PWM signals are used on most Arduino boards to generate analog voltages . The ESP32 has two 8 bit digital to analog converters .

This allows two pure analog voltage signals to be generated. These converters can be used to:

- Control an analog circuit
- Manipulate the intensity of an LED
- Can even add a small amp and speaker to your project to play a song.

DAC Pins:

- DAC1 (GPIO25)
- DAC2 (GPIO26)

Capacitive Touch GPIOs

In case if you want to develop applications with **no mechanical buttons**, You can use the touch sensitive pins on ESP32s to achieve it.

These pins are capable of detecting the small variations produced when approaching a finger to the pin. In this way, it is possible to create all kinds of controls such as buttons or slide bars without the need for mechanical components.

Capacitive Touch pins:

T0 (GPIO 4)

T1 (GPIO 0)

T2 (GPIO 2)

T3 (GPIO 15)

T4 (GPIO 13)

T5 (GPIO 12)

T6 (GPIO 14)

T7 (GPIO 27)

T8 (GPIO 33)

T9 (GPIO 32)

RTC

As we already learnt about the RTC GPIO support in the core section. The GPIOs which are routed to the **RTC low-power management subsystem** can be used when the ESP32 is in deep sleep. These RTC GPIOs can be used to wake up the ESP32 from deep sleep when the **Ultra Low Power (ULP) co-processor** is running. The following GPIOs can be used as an external wake up source.

RTC_GPIO0 (GPIO36)

RTC_GPIO3 (GPIO39)

RTC_GPIO4 (GPIO34)

RTC_GPIO5 (GPIO35)

RTC_GPIO6 (GPIO25)

RTC_GPIO7 (GPIO26)

RTC_GPIO8 (GPIO33)

RTC_GPIO9 (GPIO32)

RTC_GPIO10 (GPIO4)

RTC_GPIO11 (GPIO0)

RTC_GPIO12 (GPIO2)

RTC_GPIO13 (GPIO15)

RTC_GPIO14 (GPIO13)

RTC_GPIO15 (GPIO12)

RTC_GPIO16 (GPIO14)

RTC_GPIO17 (GPIO27)

SD / SDIO / MMC driver

This peripheral allows the ESP32 to interact with SD and MMC cards directly. In fact, by combining this controller with the analog digital converter it is possible to improve our little audio player .

UART

Many microcontrollers have UART modules , which on Arduino are known as Serial ports.

These allow asynchronous communications between two devices using only two pins.

The ESP32 has three UART ports :

- UART0
- UART1
- UART2
- All of these are compatible with RS-232 , RS-485 and IrDA protocols .

I2C

The ESP32 have two interfaces I2C or TWI that support the operating modes master and slave. Its features include:

- Standard mode (100 Kbit/s)

- Fast mode (400 Kbit/s)

- 7 and 10 bit addressing

- I2C Pins

- GPIO 21 (SDA)

- GPIO 22 (SCL)

SPI

The ESP32 also has SPI communication . It has three fully functional buses:

- Four transfer modes: this means that it is compatible with all or almost all SPI and QSPI devices available on the market.
- All SPI ports are capable of high speeds (theoretically up to 80 MHz).
- 64-byte buffer for transmission and reception.

By default, the pin mapping for SPI is:

SPI	MOSI	MISO	CLK	CS
VSPI	GPIO 23	GPIO 19	GPIO 18	GPIO 5
HSPI	GPIO 13	GPIO 12	GPIO 14	GPIO 15

PWM

Like the ESP8266, the ESP32 also supports the use of analog outputs using PWM. The big difference is in ESP32 it is possible to use up to 16 pins as PWM outputs where ESP8266 only supports 8 and Arduino UNO board that only supports 6.

PWM pins:

All the PWM pins are indicated with the below symbol in the ESP32 Pinout Diagram above.

How to select an ESP32 development board?

Before selecting an ESP32 development board, you need to take into account certain aspects:

Pin numbers and configuration : it is important to have access to the board's pinout in order to make correct use of it.

Serial -USB interface and voltage regulator : These two features are found in practically all development boards . These are the ones that allow the board to be connected directly to the computer to be energized and programmed.

Battery connector : if you are thinking of venturing into low-consumption systems with batteries, you can opt for boards that already include battery connectors.

Extra functions : many development boards for ESP32 come with extra features such as cameras, OLED displays, LoRa modules , etc.

Other ESP32 Development Boards and Specialty Projects

There are other development boards with particular characteristics that can be useful in more specific projects:

ESP32-CAM : it is a development board based on ESP32 that has a camera, microSD card slot and several GPIOs.

TTGO T-Call ESP32 with SIM800L GSM / GPRS : development board that combines an ESP32 module with one SIM800L GSM / GPRS.

M5Stack : it is a very economical development board that has a 2-inch color LCD display, a nice housing, USB type C connector for programming and power, I2C connectivity and a group of GPIOs pins to connect it to all kinds of devices. .

ESP-EYE : is a development board for image recognition and audio processing, which can be used in various AIoT applications. It has an ESP32 chip, a 2 megapixel camera and a microphone. ESP-EYE offers a lot of storage, with 8 Mbyte of SRAM and 4 Mbyte of flash memory. It also supports image transmission over Wi-Fi and debugging through a Micro-USB port.

Conclusion about the ESP32 chip

- This is the article about the ESP32, we learnt what is ESP32, How ESP32 works, ESP32 functional Block diagram with explanation, ESP 32 pins and their functions, ESP32 Security etc.,
- It is a new generation of boards and microcontrollers for the world of IoT, home automation and connected objects.
- You can check our Projects related to ESP32 here like ECG monitoring system, plant watering system and more: [Best ESP32 Projects](#)
- And although as you have seen in this article its power makes it a robust and affordable microcontroller, the ESP8266 will still continue for a few more years with all of us.