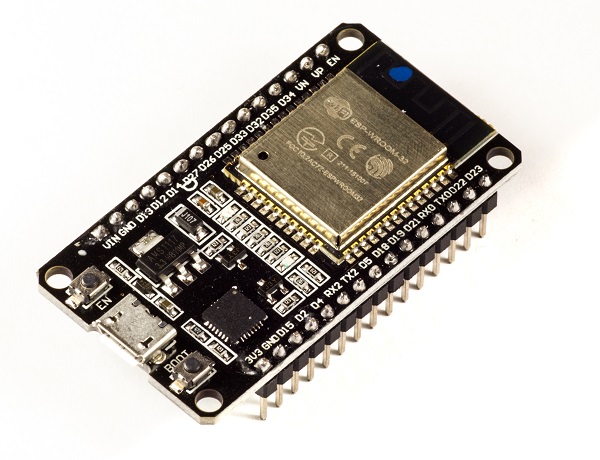
**ESP32 WROOM 32**

The ESP32 is way advanced compared to the ESP-12e. Among several features, the ESP32 packs a CPU core, a faster Wi-Fi, more GPIOs (especially increased analog pins that we all desired), supports Bluetooth 4.2 and Bluetooth low energy. The board also comes with touch-sensitive pins, alongside a built-in Hall Effect and temperature sensors.

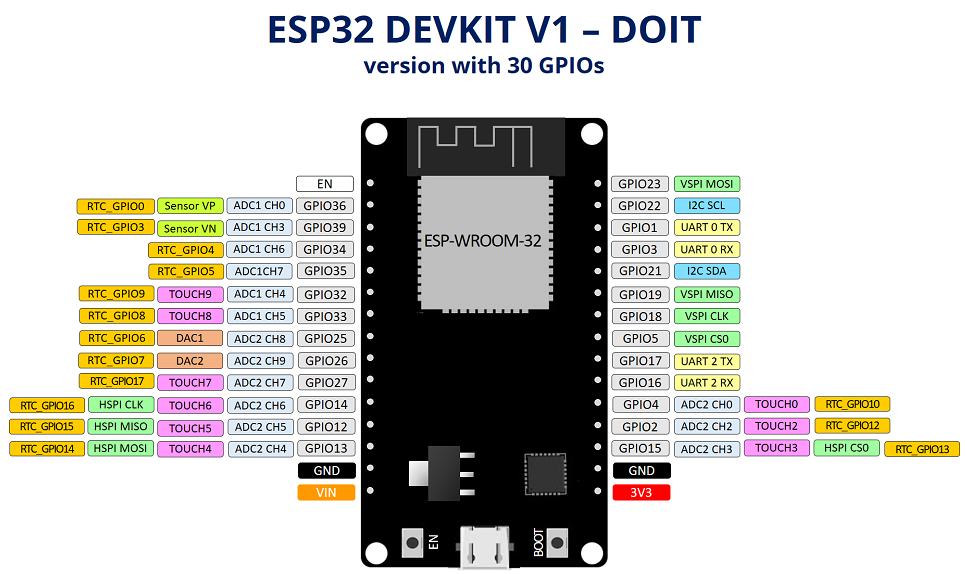


The specs listed below belong to the ESP32 WROOM 32

* Integrated Crystal− 40 MHz
* Module Interfaces− UART, SPI, I2C, PWM, ADC, DAC, GPIO, pulse counter, capacitive touch sensor
* Integrated SPI flash− 4 MB
* ROM− 448 KB (for booting and core functions)
* SRAM− 520 KB
* Integrated Connectivity Protocols− WiFi, Bluetooth, BLE
* On−chip sensor− Hall sensor
* Operating temperature range− −40 − 85 degrees Celsius
* Operating Voltage− 3.3V
* Operating Current− 80 mA (average)

The development board equips the ESP-WROOM-32 module containing **Tensilica Xtensa® Dual-Core 32-bit LX6 microprocessor**. This processor is similar to the ESP8266 but has two CPU cores (can be individually controlled), operates at **80 to 240 MHz** adjustable clock frequency and performs at up to **600 DMIPS** (Dhrystone Million Instructions Per Second).

**ESP-WROOM-32 Chip**



* Xtensa® Dual-Core 32-bit LX6
* Upto 240MHz Clock Freq.
* 520kB internal SRAM
* 4MB external flash
* 802.11b/g/n Wi-Fi transceiver
* Bluetooth 4.2/BLE
* The ESP32 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. The ESP32 supports WiFi Direct as well, which is a good option for peer-to-peer connection without the need of an access point. The **WiFi Direct** is easier to setup and the data transfer speeds are much better than Bluetooth.
* The chip also has dual mode Bluetooth capabilities, meaning it **supports both Bluetooth 4.0 (BLE/Bluetooth Smart) and Bluetooth Classic (BT)**, making it even more versatile.

**Power Requirement**

As the operating voltage range of ESP32 is **2.2V 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 ESP32 pulls as much as **250mA 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.

**Power to the ESP32 development board** 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 ESP32 and its peripherals.

Also the sleep current of the ESP32 chip is less than 5 µA, making it suitable for battery powered and wearable electronics applications.

**Peripherals and I/O**

Although the ESP32 has total **48 GPIO pins**, only 25 of them are 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:

* **15 ADC channels** – 15 channels of 12-bit SAR ADC’s. The ADC range can be set, in firmware, to either 0-1V, 0-1.4V, 0-2V, or 0-4V
* **2 UART interfaces** – 2 UART interfaces. One is used to load code serially. They feature flow control, and support IrDA too!
* **25 PWM outputs** – 25 channels of PWM pins for dimming LEDs or controlling motors.
* **2 DAC channels** – 8-bit DACs to produce true analog voltages.
* **SPI, I2C & I2S interface** – There are 3 SPI and 1 I2C interfaces to hook up all sorts of sensors and peripherals, plus two I2S interfaces if you want to add sound to your project.
* **9 Touch Pads** – 9 GPIOs feature capacitive touch sensing.

The ESP32’s **pin multiplexing feature** (Multiple peripherals multiplexed on a single GPIO pin). Meaning a single GPIO pin can act as an ADC input/DAC output/Touch pad.

Pin D34, D35, VP and VN cannot be configured as outputs, but they can be used as either digital inputs, analog inputs, or for other unique purposes. Also note that they do not have internal pull-up or pull-down resistors, like the other GPIO pins.

Also GPIO pins VP and VN are an integral part of the ultra-low-noise pre-amplifier for the ADC, which help to configure the sampling time and noise of the pre-amp.

## On-board Switches & LED Indicators

The ESP32 development board features two buttons. One marked as **EN** located on the top left corner is the Reset button, used of course to reset the ESP32 chip. The other **Boot** button on the bottom left corner is the download button used while downloading the new sketch/programs.

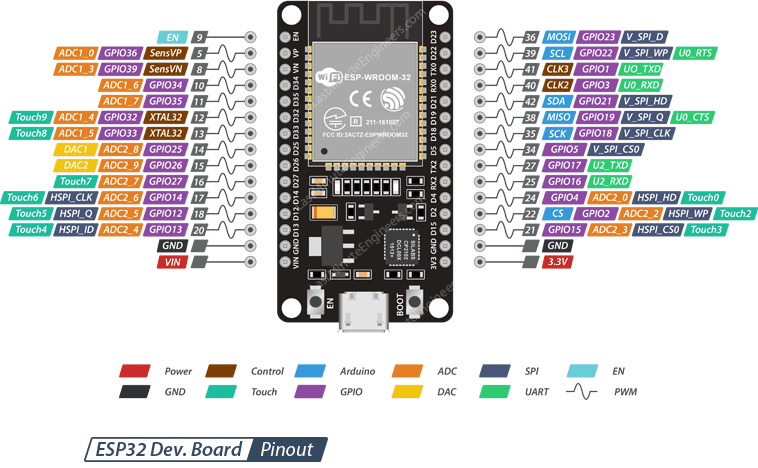
The board also has **2 LED indicators** viz. **Red LED & Blue LED**. A Red LED indicates that the board is powered up and has 3.3V from the regulator. The Blue LED is user programmable and is connected to the D2 pin of the board.

## Serial Communication

The board includes CP2102 USB-to-UART Bridge Controller from [Silicon Labs](http://www.silabs.com/), which converts USB signal to serial and allows your computer to program and communicate with the ESP32 chip.

## ESP32 Development Board Pin-out:

The ESP32 development board has total 30 pins that interface it to the outside world. The connections are as follows:



**Power Pins** There are two power pins viz. VIN pin & 3.3V pin. The VIN pin can be used to directly supply the ESP32 and its peripherals, if you have a regulated 5V voltage source. The 3.3V pin is the output of an on-board voltage regulator. This pin can be used to supply power to external components.

**GND** is a ground pin of ESP32 development board.

**Arduino Pins** are nothing but ESP32’s hardware I2C and SPI pins to hook up all sorts of sensors and peripherals in your project.

**GPIO Pins** ESP32 development board has 25 GPIO pins which can be assigned to various functions 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.

**ADC Channels** The board integrates 12-bit SAR ADCs and supports measurements on 15 channels (analog enabled pins). Some of these pins can be used to build a programmable gain amplifier which is used for the measurement of small analog signals. The ESP32 is also designed to measure the voltages while operating in the sleep mode.

**DAC Channels** The board features two 8-bit DAC channels to convert digital signals into true analog voltages. This dual DAC can drive other circuits.

**Touch Pads** The board offers 9 capacitive sensing GPIOs which detect capacitive variations introduced by the GPIO’s direct contact or close proximity with a finger or other objects.

**UART Pins** ESP32 development board has 2 UART interfaces, i.e. UART0 and UART2, which provide asynchronous communication (RS232 and RS485) and IrDA support, and communicate at up to 5 Mbps. UART provides hardware management of the CTS and RTS signals and software flow control (XON and XOFF) as well.

**SPI Pins** SPI Pins ESP32 features three SPIs (SPI, HSPI and VSPI) 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

All SPIs can also be used to connect to the external Flash/SRAM and LCD.

**PWM Pins** The board has 25 channels (Nearly All GPIO pins) of PWM pins controlled by Pulse Width Modulation (PWM) controller. The PWM output can be used for driving digital motors and LEDs. The controller consists of PWM timers and the PWM operator. Each timer provides timing in synchronous or independent form, and each PWM operator generates the waveform for one PWM channel.

**EN Pin** is used to enable ESP32. The chip is enabled when pulled HIGH. When pulled LOW the chip works at minimum power.