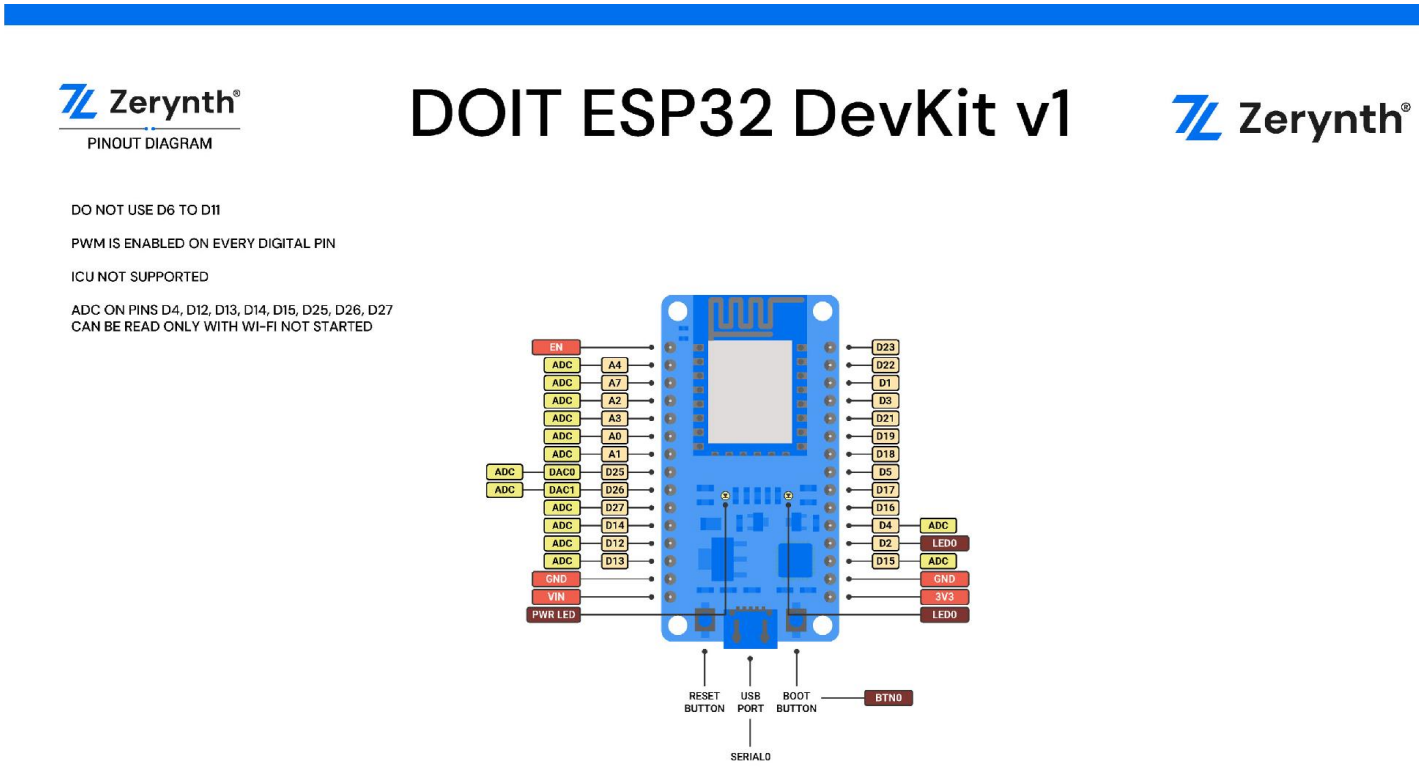
DOIT Esp32 DevKit v1

The DOIT Esp32 DevKit v1 is one of the development board created by DOIT to evaluate the ESP-WROOM-32 module. It is based on the ESP32 microcontroller that boasts Wifi, Bluetooth, Ethernet and Low Power support all in a single chip.

Pin Mapping



More info about DOIT Esp32 DevKit v1 can be found here.

Flash Layout

The internal flash of the ESP32 module is organized in a single flash area with pages of 4096 bytes each. The flash starts at address 0x00000, but many areas are reserved for Esp32 IDF SDK and Zerynth VM. There exist two different layouts based on the presence of BLE support.

In particular, for non-BLE VMs:

Start address	Size	Content
0x00009000	16Kb	Esp32 NVS area
0x0000D000	8Kb	Esp32 OTA data
0x0000F000	4Kb	Esp32 PHY data
0x00010000	1Mb	Zerynth VM
0x00110000	1Mb	Zerynth VM (FOTA)
0x00210000	512Kb	Zerynth Bytecode
0x00290000	512Kb	Zerynth Bytecode (FOTA)
0x00310000	512Kb	Free for user storage
0x00390000	448Kb	Reserved

For BLE VMs:

Start address	Size	Content
0x00009000	16Kb	Esp32 NVS area
0x0000D000	8Kb	Esp32 OTA data
0x0000F000	4Kb	Esp32 PHY data
0x00010000	1216Kb	Zerynth VM
0x00140000	1216Kb	Zerynth VM (FOTA)
0x00270000	320Kb	Zerynth Bytecode
0x002C0000	320Kb	Zerynth Bytecode (FOTA)
0x00310000	512Kb	Free for user storage
0x00390000	448Kb	Reserved

Device Summary

- Microcontroller: Tensilica 32-bit Single-/Dual-core CPU Xtensa LX6
- Operating Voltage: 3.3V
- Input Voltage: 7-12V
- Digital I/O Pins (DIO): 25
- Analog Input Pins (ADC): 6
- Analog Outputs Pins (DAC): 2
- UARTs: 3
- SPIs: 2
- I2Cs: 3
- Flash Memory: 4 MB
- SRAM: 520 KB
- Clock Speed: 240 Mhz
- Wi-Fi: IEEE 802.11 b/g/n/e/i:
    - Integrated TR switch, balun, LNA, power amplifier and matching network
    - WEP or WPA/WPA2 authentication, or open networks

Power

Power to the DOIT Esp32 DevKit v1 is supplied via the on-board USB Micro B connector or directly via the “VIN” pin. The power source is selected automatically.

The device can operate on an external supply of 6 to 20 volts. If using more than 12V, the voltage regulator may overheat and damage the device. The recommended range is 7 to 12 volts.

Connect, Register, Virtualize and Program

The DOIT Esp32 DevKit v1 comes with a serial-to-usb chip on board that allows programming and opening the UART of the ESP32 module. Drivers may be needed depending on your system (Mac or Windows) and can be download from the official Espressif documentation page. In Linux systems, the DevKit v1 should work out of the box.

Note

For Linux Platform: to allow the access to serial ports the user needs read/write access to the serial device file. Adding the user to the group, that owns this file, gives the required read/write access:

- **Ubuntu** distribution –> dialout group
- **Arch Linux** distribution –> uucp group

Once connected on a USB port, if drivers have been correctly installed, the DevKit v1 device is recognized by Zerynth Studio. The next steps are:

- **Select** the DevKit v1 on the **Device Management Toolbar** (disambiguate if necessary);
- **Register** the device by clicking the "Z" button from the Zerynth Studio;
- **Create** a Virtual Machine for the device by clicking the "Z" button for the second time;
- **Virtualize** the device by clicking the "Z" button for the third time.

Note

No user intervention on the device is required for registration and virtualization process

After virtualization, the DevKit v1 is ready to be programmed and the Zerynth scripts **uploaded**. Just **Select** the virtualized device from the "Device Management Toolbar" and **click** the dedicated "upload" button of Zerynth Studio.

Note

No user intervention on the device is required for the uplink process.

Firmware Over the Air update (FOTA)

The Firmware Over the Air feature allows to update the device firmware at runtime. Zerynth FOTA in the DevKitC device is available for bytecode and VM.

Flash Layout is shown in table below:

Start address	Size	Content
0x00010000	1Mb	Zerynth VM (slot 0)
0x00110000	1Mb	Zerynth VM (slot 1)
0x00210000	512Kb	Zerynth Bytecode (slot 0)
0x00290000	512Kb	Zerynth Bytecode (slot 1)

For BLE VMs:

Start address	Size	Content
0x00010000	1216Kb	Zerynth VM (slot 0)
0x00140000	1216Kb	Zerynth VM (slot 1)

Start address	Size	Content
0x00270000	320Kb	Zerynth Bytecode (slot 0)
0x002C0000	320Kb	Zerynth Bytecode (slot 1)

For Esp32 based devices, the FOTA process is implemented mostly by using the provided system calls in the IDF framework. The selection of the next VM to be run is therefore a duty of the Espressif bootloader; the bootloader however, does not provide a failsafe mechanism to revert to the previous VM in case the currently selected one fails to start. At the moment this lack of a safety feature can not be circumvented, unless by changing the bootloader. As soon as Espressif relases a new IDF with such feature, we will release updated VMs.

Secure Firmware

Secure Firmware feature allows to detect and recover from malfunctions and, when supported, to protect the running firmware (e.g. disabling the external access to flash or assigning protected RAM memory to critical parts of the system).

This feature is strongly platform dependent; more information at Secure Firmware - ESP32 section.

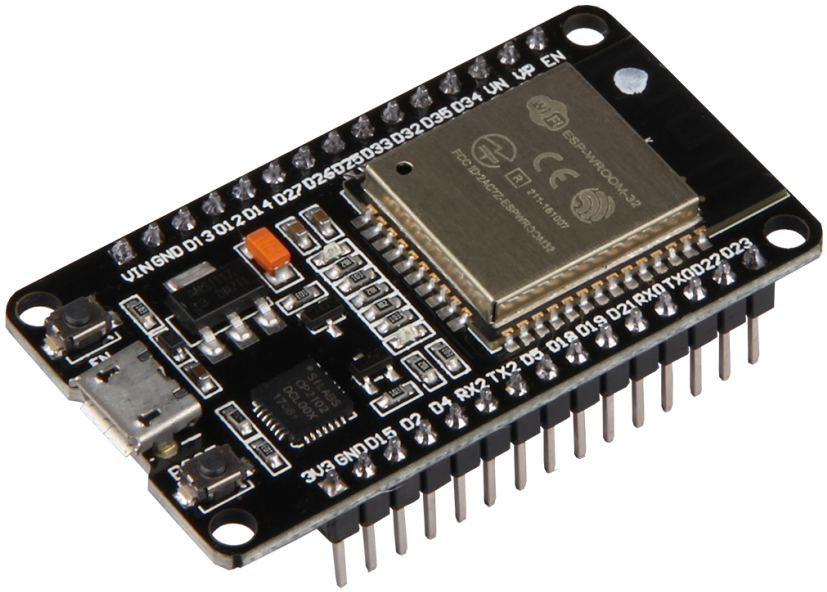
Zerynth Secure Socket

To be able to use Zerynth Secure Socket on esp32 boards `NATIVE_MBEDTLS: true` must be used instead of `ZERYNTH_SSL: true` in the `project.yml` file.

Missing features

Not all IDF features have been included in the Esp32 based VMs. In particular the following are missing but will be added in the near future:

- Touch detection support



NodeMCU ESP32

Manual

TABLE OF CONTENTS

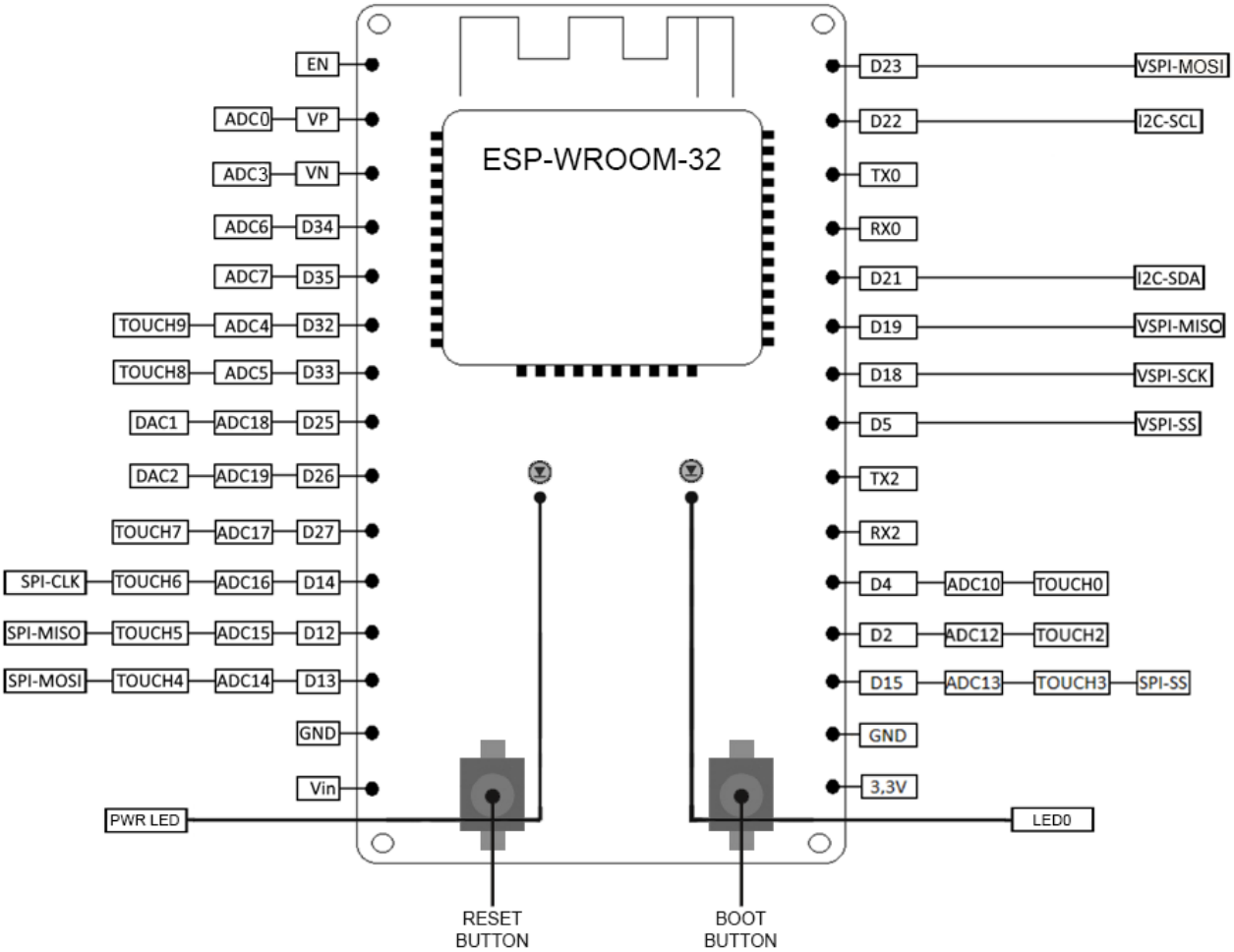
1. INTRODUCTION

Dear customer,
thank you for choosing our product.
In the following, we will show you what to observe during commissioning and use.
Should you encounter any unexpected problems during use, please do not hesitate to contact us.

2. OVERVIEW

The NodeMCU ESP32 module is a compact prototyping board and can be easily programmed via the Arduino IDE. It features 2.4 GHz dual-mode Wi-Fi and Bluetooth. Also integrated on the microcontroller development board are 512kb SRAM and 16MB memory.
PWM is enabled on every digital pin.
An overview of the available pins can be seen in the following figure:

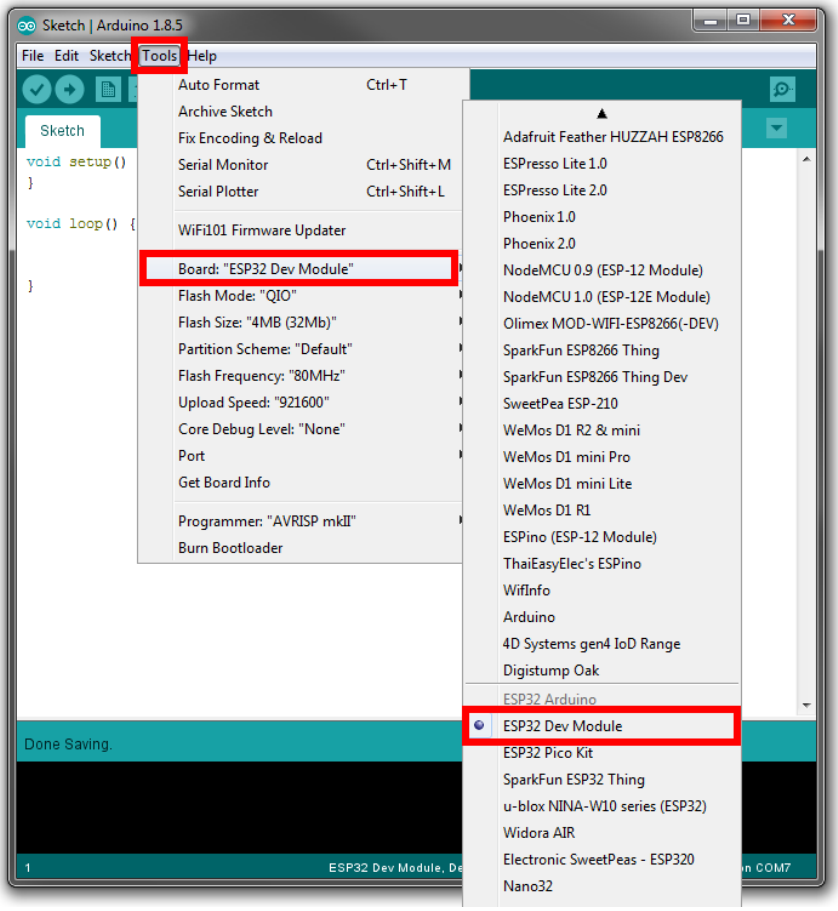
2x DAC	15x ADC	1x SPI
1x I²C	2x UART	



3. MODULE INSTALLATION

If you have not yet installed the Arduino IDE on your computer, download and install it first.

Now open the Arduino IDE and change the board setting to "ESP32 Dev Module":



Download and install the updated CP210x USB-UART driver for your operating system.
The next step is to download the official ESP32 library.
Unpack the contents of the archive into the "hardware" folder of your Arduino folder ("C:\Users\[your username]\Documents\Arduino\hardware").
Now restart your Arduino IDE. The module installation is now complete.

Attention! After the initial installation, the baud rate may have changed to "921600". This may cause problems. In this case, select the baud rate "115200" to avoid any problems.

4. USAGE

Your NodeMCU ESP32 is now ready to use. Simply connect it to your computer with a USB cable.
The installed library already provides many examples to allow a quick insight into the module.
You can find the examples in your Arduino IDE under "File -> Examples -> ESP32".

The fastest and easiest way to test your NodeMCU ESP32 is to get the device number. Either copy the code below or use the example "GetChipID":

```
uint64_t chipid;

void setup() {
     Serial.begin(115200);
}

void loop() {
     chipid=ESP.getEfuseMac();//The chip ID is essentially its MAC address
(length: 6 bytes).
     Serial.printf("ESP32 Chip ID = %04X",(uint16_t)(chipid>>32));//print
High 2 bytes
     Serial.printf("%08X\n",(uint32_t)chipid);//print Low 4bytes.

     delay(3000);

}
```

To upload, click the upload button from the Arduino IDE and hold down the "BOOT"-button on the SBC NodeMCU ESP32. The upload is completed until the writing has reached 100% and you will be asked to reboot (hard resetting via RTS pin ...) with the "EN" key.

You can show the output of the example in the serial monitor.

5. SUPPORT

We also support you after your purchase. If there are any questions left or if you encounter any prob lems please feel free to contact us by mail, phone or via our ticket support system on our website.

E-Mail: service@joy-it.net

Ticket-System: http://support.joy-it.net

Telefon: +49 (0)2845 98469 – 66 (11- 18 Uhr)

Visit our website for more informations:

www.joy-it.net

6. OTHER INFORMATIONS

Our take-back obligations under the Electrical and Electronic Equipment Act (ElektroG)

Symbol on electrical and electronic equipment: 

This crossed-out dustbin means that electrical and electronic equipment does **not** belong in household waste. You must return the old devices to a collection point. Before disposal, you must separate old batteries and accumulators which are not enclosed in the old device from it.

Return options:
As an end user, you can dispose of your old device (which essentially fulfils the same function as the new one purchased from us) free of charge when purchasing a new device.
Small appliances where no external dimensions are larger than 25 cm can be delivered in normal household quantities, irrespective of the purchase of a new appliance.

Possibility of return at our company during opening hours:
Simac GmbH, Pascalstr. 8, 47506 Neukirchen-Vluyn, Germany

Possibility of return in your area:
We will send you a parcel stamp with which you can return the device to us free of charge.

Please contact us by e-mail at Service@joy-it.net or by phone.

Information about packaging:
If you do not have suitable packaging material or do not wish to use your own, please contact us and we will send you suitable packaging. Please inform us if your device contains LiIon batteries in this case. an additional warning label must be attached.