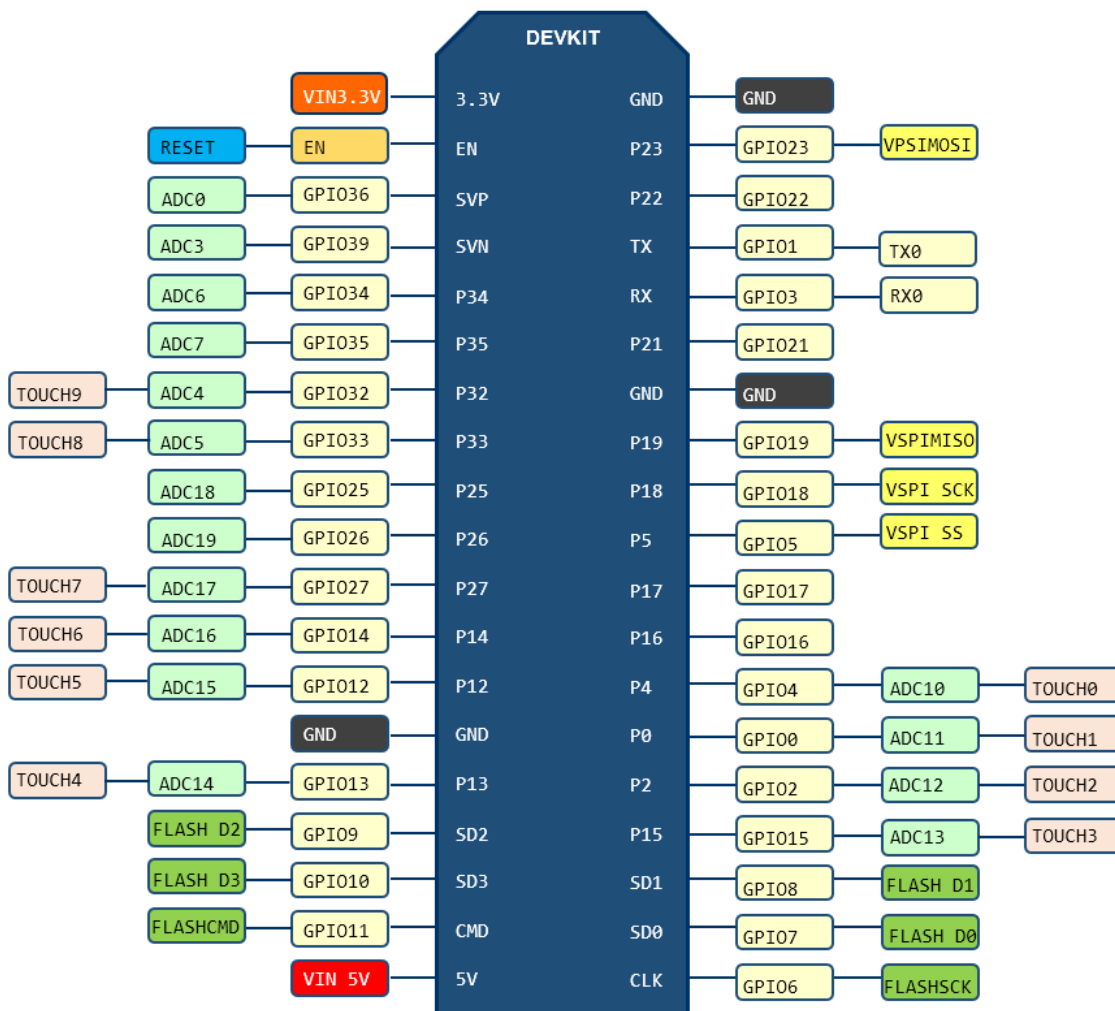


Overview

NodeMCU-32S is a core development board designed by Essence based on the ESP32 module. The development board continues the classic design of NodeMCU, leading most of the I/O to the headers on both sides, and developers can connect peripherals according to their needs. When using the breadboard for development and debugging, the standard headers on both sides can make the operation easier and more convenient.

Pin Diagram

PIN DEFINITION



Pin function definition

No.	Pin Name	Functional Description
1	3.3V	Module power supply pin
2	EN	Chip Enabled Pin, Active High
3	SVP	GPIO36, ADC1_CH0, RTC_GPIO0
4	SVN	GPIO39, ADC1_CH3, RTC_GPIO3
5	P34	GPIO34, ADC1_CH6, RTC_GPIO4
6	P35	GPIO35, ADC1_CH7, RTC_GPIO5
7	P32	GPIO32, XTAL_32K_P (32.768kHz Crystal input), ADC1_CH4, TOUCH9, RTC_GPIO9
8	P33	GPIO33, XTAL_32K_N (32.768kHz Crystal output), ADC1_CH5, TOUCH8, RTC_GPIO8

9	P25	GPIO25, DAC_1, ADC2_CH8, RTC_GPIO6, EMAC_RXD0
10	P26	GPIO26, DAC_2, ADC2_CH9, RTC_GPIO7, EMAC_RX_DV
11	P27	GPIO27, ADC2_CH7, TOUCH7, RTC_GPIO17, EMAC_RX_DV
12	P14	GPIO14, ADC2_CH6, TOUCH6, RTC_GPIO16, MTMS,

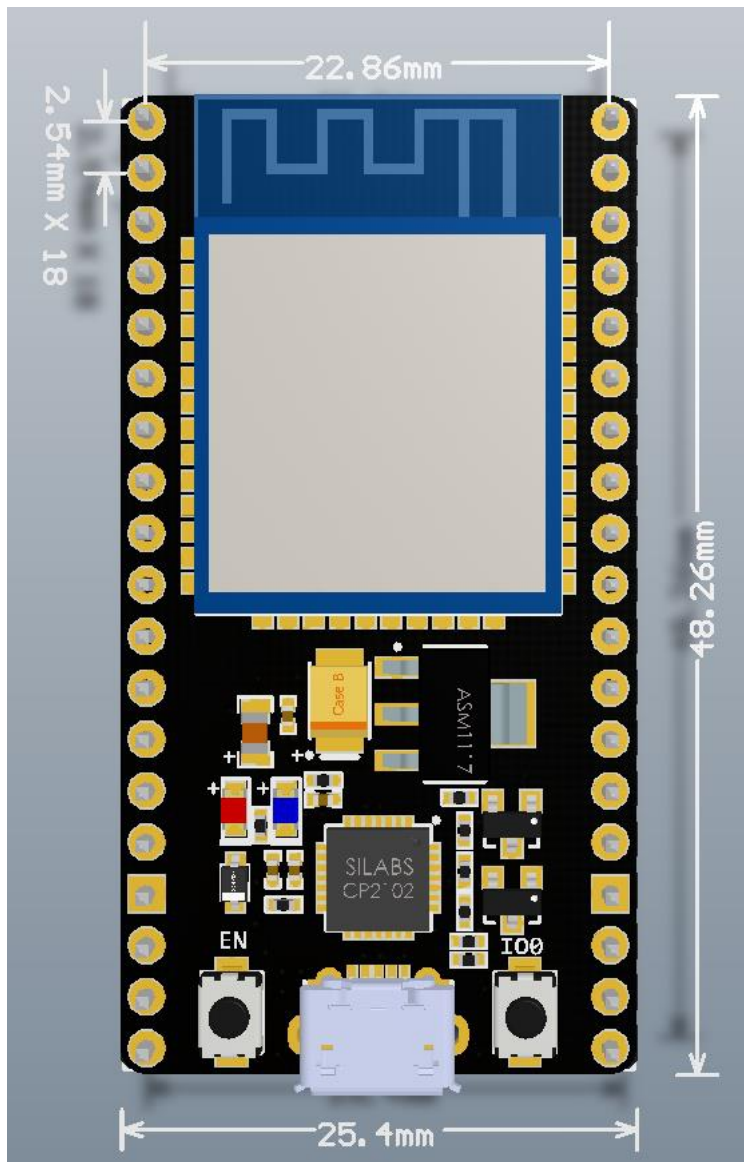
		HSPICLK, HS2_CLK, SD_CLK, EMAC_TXD2
13	P12	GPIO12, ADC2_CH5, TOUCH5, RTC_GPIO15, MTDI, HSPIQ, HS2_DATA2, SD_DATA2, EMAC_TXD3
14	GND	GND
15	P13	GPIO13, ADC2_CH4, TOUCH4, RTC_GPIO14, MTCK, HSPID, HS2_DATA3, SD_DATA3, EMAC_RX_ER

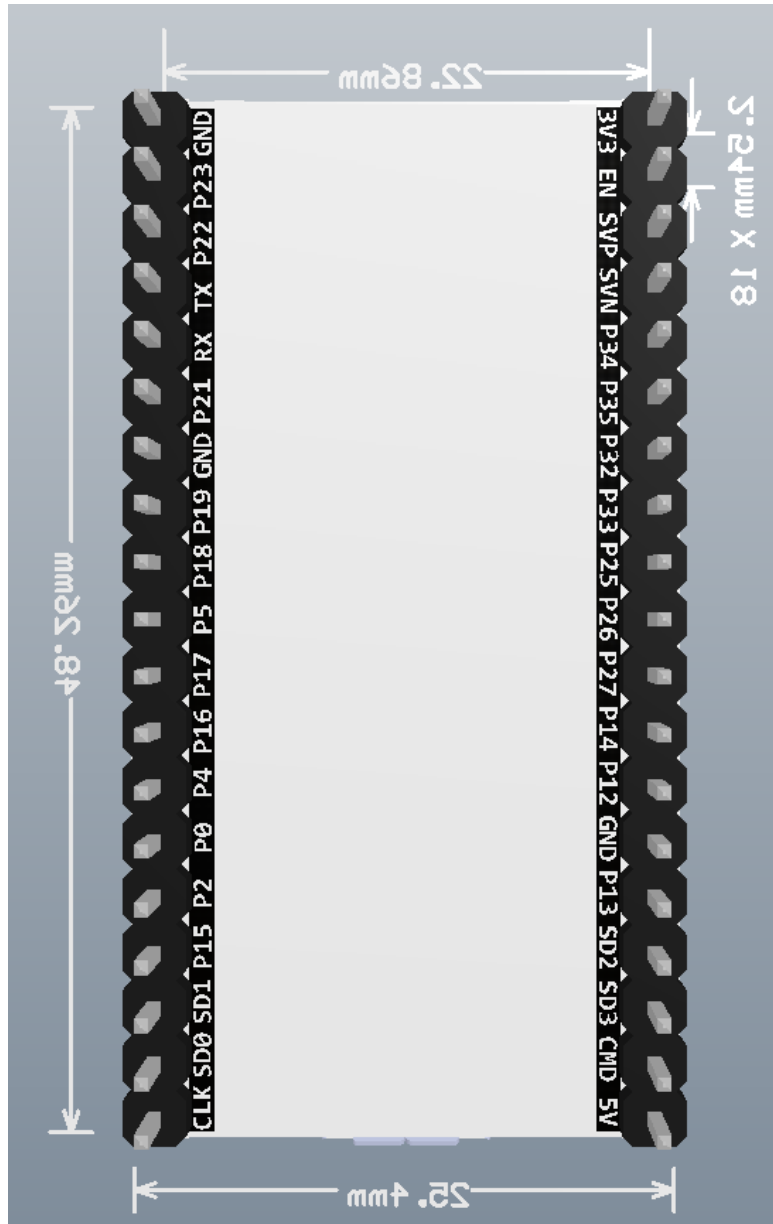
16	SD2	GPIO9, SD_DATA2, SPIHD, HS1_DATA2, U1RXD
17	SD3	GPIO10, SD_DATA3, SPIWP, HS1_DATA3, U1TXD
18	CMD	GPIO11, SD_CMD, SPICS0, HS1_CMD, U1RTS
19	5V	Module power supply pin
20	CLK	GPIO6, SD_CLK, SPICLK, HS1_CLK, U1CTS
21	SD0	GPIO7, SD_DATA0, SPIQ, HS1_DATA0, U2RTS
22	SD1	GPIO8, SD_DATA1, SPID, HS1_DATA1, U2CTS
23	P15	GPIO15, ADC2_CH3, TOUCH3, MTDO, HSPICS0, RTC_GPIO13, HS2_CMD, SD_CMD, EMAC_RXD3
24	P2	GPIO2, ADC2_CH2, TOUCH2, RTC_GPIO12, HSPIWP, HS2_DATA0, SD_DATA0

25	P0	GPIO0, ADC2_CH1, TOUCH1, CLK_OUT1, RTC_GPIO11, EMAC_TX_CLK; Download mode:
		external pull low, running mode: floating or external pull high
26	P4	GPIO4, ADC2_CH0, TOUCH0, RTC_GPIO10, HSPIHD, HS2_DATA1, SD_DATA1, EMAC_TX_ER
27	P16	GPIO16, HS1_DATA4, U2RXD, EMAC_CLK_OUT
28	P17	GPIO17, HS1_DATA5, U2TXD, EMAC_CLK_OUT_180
29	P5	GPIO5, VSPICS0, HS1_DATA6, EMAC_RX_CLK
30	P18	GPIO18, VSPICLK, HS1_DATA7
31	P19	GPIO19, VSPIQ, U0CTS, EMAC_TXD0
32	GND	GND

33	P21	GPIO21, VSPIHD, EMAC_TX_EN
34	RX	GPIO3, U0RXD, CLK_OUT2
35	TX	GPIO1, U0TXD, CLK_OUT3, EMAC_RXD2
36	P22	GPIO22, VSPIWP, U0RTS, EMAC_TXD1
37	P23	GPIO23, VSPID, HS1_STROBE
38	GND	GND

Dimension drawing





Features

- 802.11b/g/n (802.11n, speed up to 150Mbps)
- WIFI Frequency Range 2.4GHz ~ 2.5GHz

- Clock frequency adjustment range from 80 MHz to 240 MHz, support for
RTOS
- Built-in 2-channel 12-bit high-precision ADC with up to 18 channels
- Support UART/GPIO/ADC/DAC/SDIO/SD card/PWM/I2C/I2S interface
- Support multiple sleep modes, ESP32 chip sleep current is less than 5 μ A / Embedded Lwip protocol stack
- Supports STA/AP/STA + AP operation mode
- Supports remote firmware upgrade (FOTA)
- General AT commands can be used quickly
- Support secondary development, integrated Windows, Linux development environment

Product Specification

Description of main parameters

Module model	ESP-WROOM-32s
Size	25.4*48.26*3mm (\pm 0.2mm)
Certification	FCC/CE-RED/IC/TELEC/KCC/SRRC/NCC/ BQB/ RoHS/REACH
SPI Flash	32Mbit(default)
Support interface	UART/GPIO/ADC/DAC/SDIO/SD card /PWM/I2C/I2S

Integrated crystal oscillator	40MHz Crystal oscillator
IO Port	38
Antenna	Onboard antenna
Power Supply	Voltage 3.0V ~ 3.6V, Typical 3.3V, Current >500mA
Operating Temperature	-40 °C ~ 85 °C
Storage Environment	-40 °C ~ 120 °C

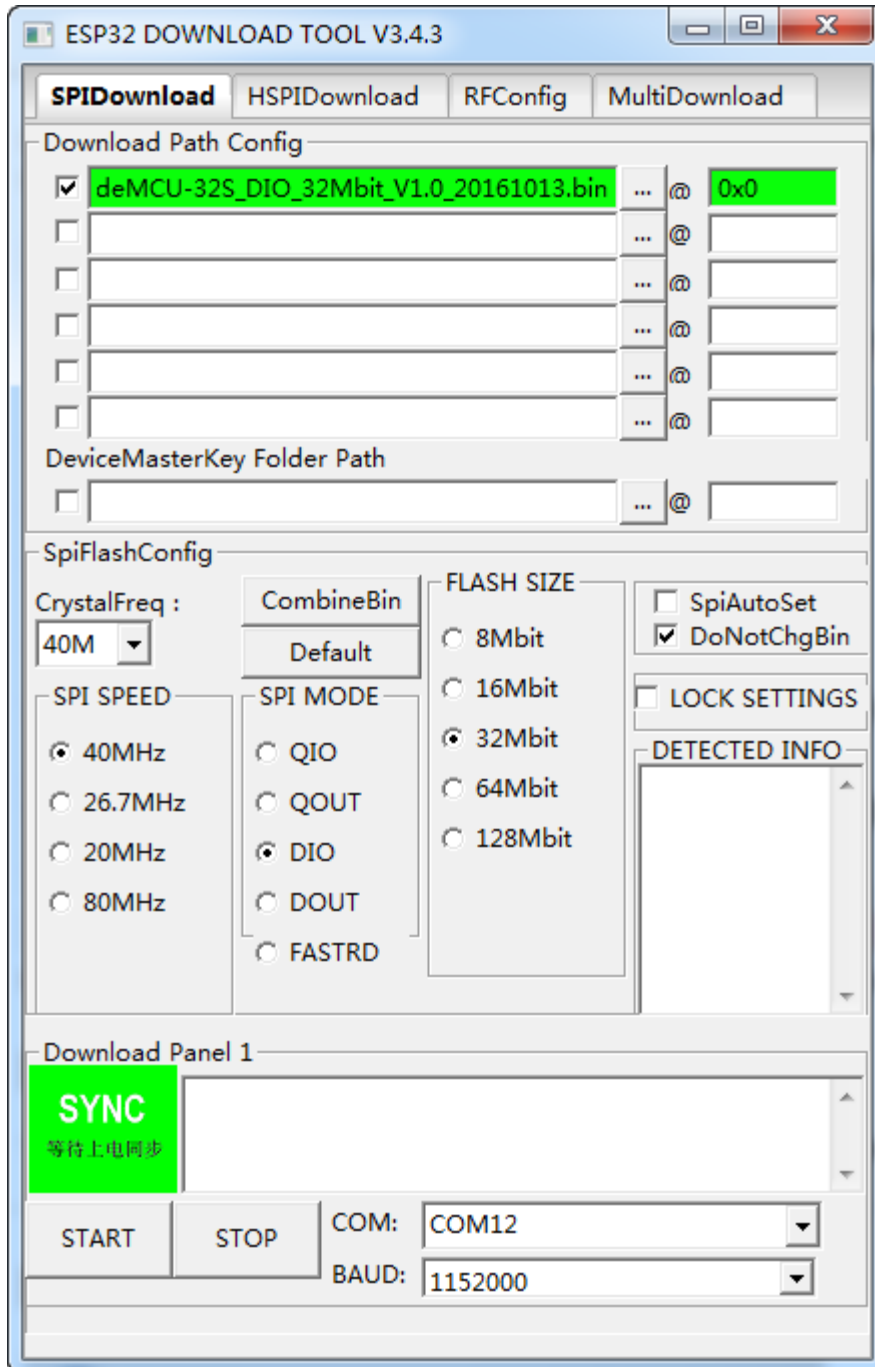
How to use

1. Connect the cable

Use the Micro USB cable to connect the PC and the Nodemcu core development board, and confirm the COM port of the chip from the Windows Device Manager.

2. Download the factory default firmware

Download the factory firmware [nodemcu-32s_dio_32mbit_20170705.zip](#)



3. Verify the download result

After the download is complete, use the serial port tool to open the port, view the data output, and print as follows:

```
I (202) heap_alloc_caps: Initializing heap allocator:
I (203) heap_alloc_caps: Region 19: 3FFC0D24 len 0001F2DC tag 0
I (203) heap_alloc_caps: Region 25: 3FFE8000 len 00018000 tag 1
I (210) cpu_start: Pro cpu up.
I (214) cpu_start: Starting app cpu, entry point is 0x400808e4
I (0) cpu_start: App cpu up.
I (226) cpu_start: Pro cpu start user code
rtc w112 Sep 26 2016 22:32:10
XTAL 40M
I (261) cpu_start: Starting scheduler on PRO CPU.
I (40) cpu_start: Starting scheduler on APP CPU.

Mounting flash filesystem...
Formatting file system. Please wait...
mount res: -10025, -10025
Formatting: size 0x70000, addr 0x190000
mount res: 0, 0
frc2_timer_task_hdl:3ffc6214, prio:22, stack:2048
tcpip_task_hdlxxx : 3ffc6d78, prio:18, stack:2048
Task task_lua started.
phy_version: 123, Sep 13 2016, 20:01:58, 0
pp_task_hdl : 3ffc9ff0, prio:23, stack:8192
NodeMCU ESP32 build unspecified powered by Lua 5.1.4 on IDF 1.0.0(970dec6)
lua: cannot open init.lua
> Heap size::184592.
```

When the program is executed for the first time, the file system will be formatted automatically, which will take a long time, please be patient.

Use the serial port tool to send `print("hello NodeMCU")` (note the line feed `\r\n`)

The program results are shown as follows:

```
> print("hello world!")
hello world!
>
```