ESP32-DevKit

Getting Started Guide

Version 1.0

About This Guide

This user guide introduces the basic features, interfaces and download operations of ESP32-DevKit.

The document is structured as follows.

Chapter	Title	Content
Chapter 1	Overview	Introduction to the basic features of the ESP32-DevKit.
Chapter 2	Interface Description	Introduction to hardware interface options available on the ESP32-DevKit.
Chapter 3	Download Process	Introduction to the power-on mode of programming.

Release Notes

Date	Version	Release notes
2016.09	V1.0	Initial release.

Table of Contents

1. Overview

The ESP32-DevKit is an ESP-WROOM-32-based development board provided by Espressif. It supports TOUCH, LCD display technology and I/O extension. Based on this minimal system development board, developers and users can initiate further application development. The dimension is 104.3 mm X 84.6 mm with the layout shown in Figure 1-1:

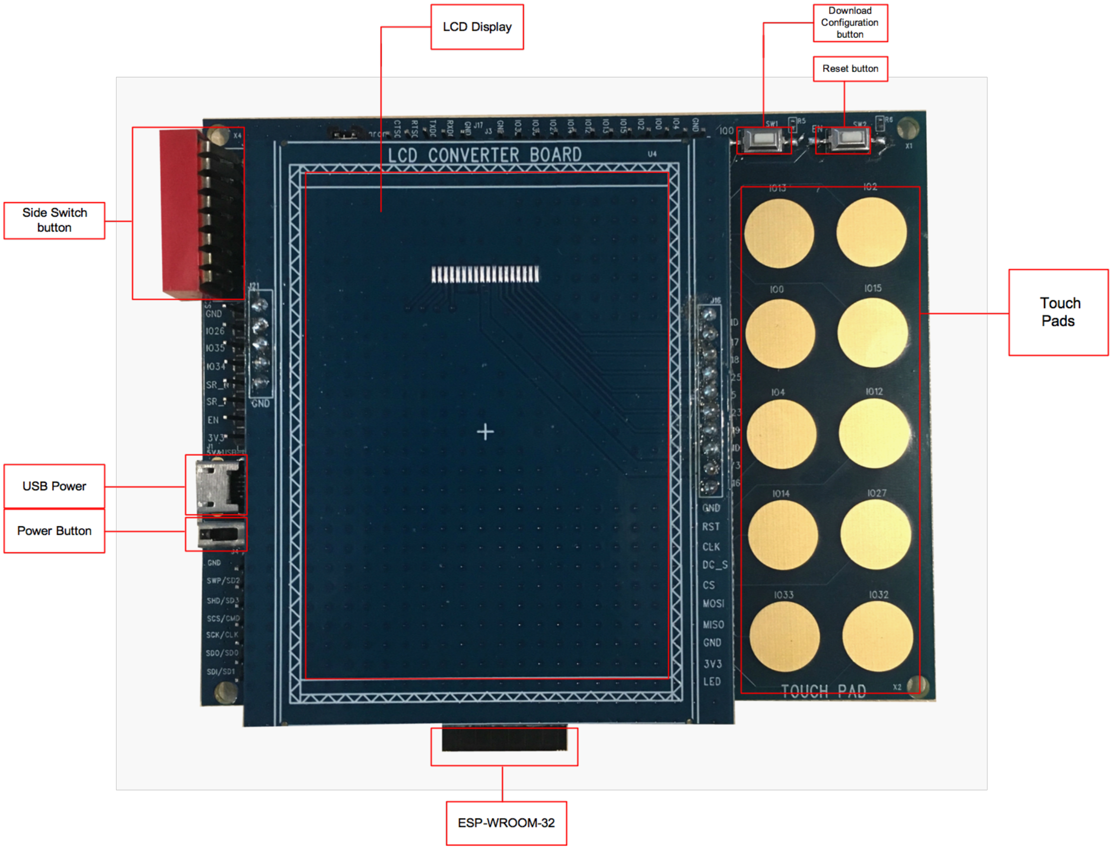


Figure 1-1. the Layout of ESP32-DevKit

This development board supports mounting and interfacing 3.2" SPI (standard 4-wire Serial Peripheral Interface) TFT LCD display. Developers can purchase an appropriate LCD display module either by themselves or from Espressif. The development board consists of two parts: an LCD loading board and a baseboard. The layout of the baseboard is shown in Figure 1-2:

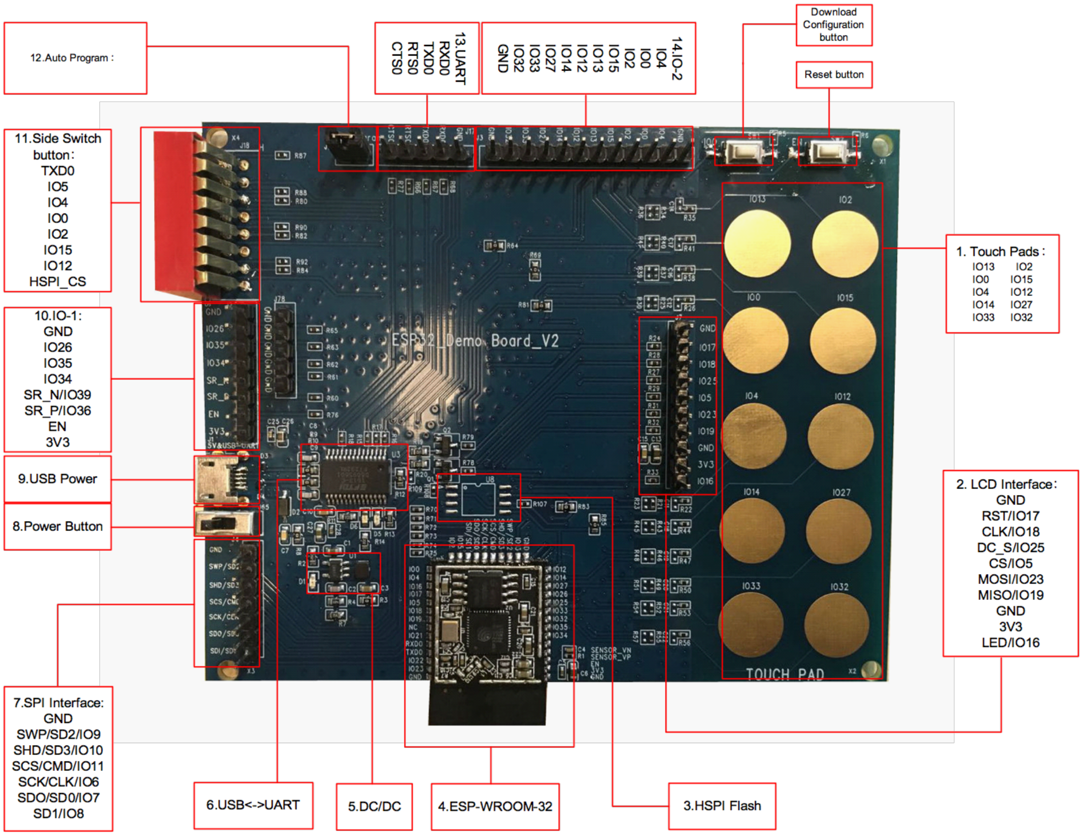


Figure 1-2. the Layout of the Baseboard

2. Interface Description

Here are some details of the hardware interface options provided by the ESP32-DevKit:

1. Touch Pads: The ESP32-DevKit uses 10 Touch Pads instead of mechanical buttons. Developers can customize the touch pad operation in firmware.
2. LCD Interface: Supports a 3.2” 4-wire SPI LCD screen.
3. HSPI Flash: The ESP-WROOM-32 at the heart of this board already carries a flash memory for program (and data) storage. The development board also supports memory extension using another flash (U8), which is not soldered on by default. Developers can select and add a suitable flash memory chip to the empty footprint. ESP32 supports most of the flash types in the market.
4. ESP-WROOM-32: This is a high performance module based on ESP32. The module integrates some important discrete components and flash memory that stores user firmware.
5. DC-DC Power Module: USB port (5V power supply) is used to power the board. Developers and users can switch 5V to 3.3V by using a DC-DC converter. The power section has a power indicator (Red LED is normally on). R3 can be removed to directly measure the current consumption of the circuit.
6. USB <-> UART: The USB to UART converter chip makes development and debugging more convenient on systems that only offer USB ports and no external serial port interface.
7. SPI Interface: SPI bus Interface can connect to various compatible external devices with an independent GPIO acting as chip select signal.
8. Power Button: Put switch to the right side to enable 3.3V power output. Put switch to the left side (the default state) to disable power output.
9. USB Power: Port for 5V power supply and communication with PC.
10. I/O Interface 1: Expanded I/O interfaces, some of which can only be used for input. For more details, please refer to *ESP32 Pin List*.
11. Side Switch Button: Select power-on mode and enable HSPI chip select signal.
12. Automatic Download: Enable the function of automatic download (shorted by jumper cap by default).
13. UART: UART interfaces.
14. I/O Interface 2: Expanded I/O interfaces, some of which can only be used for input. For more details, please refer to *ESP32 Pin List*.

3. Download Process

You will need the hardwares mentioned below.

- 1 × ESP32-DevKit
- 1 × PC (with Windows OS as an example in this document)
- 1 × USB cable

3.1. Create Serial Communication

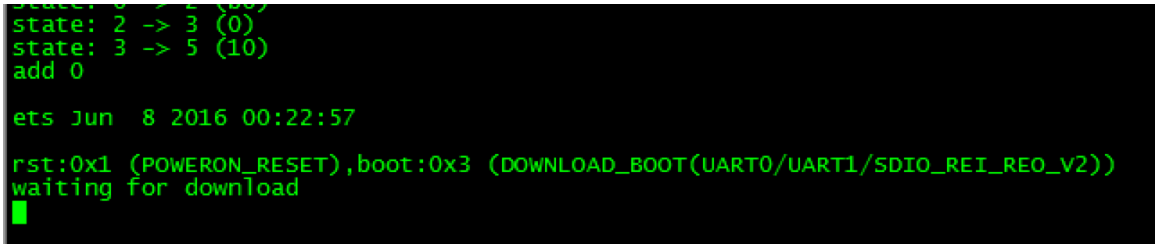
Connect the ESP32-DevKit to the PC using the USB cable. Check from the Windows Device Manager and confirm the COM port of the chip.

3.2. Set Download Mode and SPI Boot Mode

At the top right of Figure 1-1, there are two buttons on ESP32-DevKit: Reset button for SPI boot mode and Download Configuration button for download mode.

- There are two ways to enable download mode:

1. Press and hold the Download Configuration button, press Reset button at the same time, system would then enter download mode as it shows below:



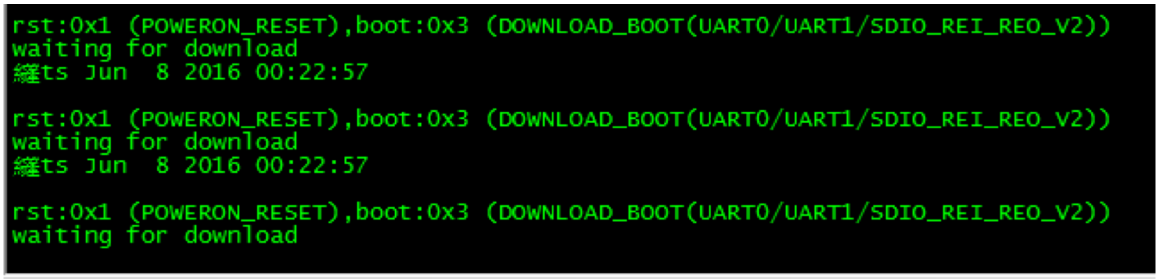
```
state: 0 -> 2 (b0)
state: 2 -> 3 (0)
state: 3 -> 5 (10)
add 0

ets Jun  8 2016 00:22:57

rst:0x1 (POWERON_RESET),boot:0x3 (DOWNLOAD_BOOT(UART0/UART1/SDIO_REI_REO_V2))
waiting for download
█
```

Figure 3-1. Download Mode Prints 1

2. Power off the development board and turn the fourth side switch button (IO0) to ON. After powering on the development board, system would automatically enter download mode as it shows below:



```
rst:0x1 (POWERON_RESET),boot:0x3 (DOWNLOAD_BOOT(UART0/UART1/SDIO_REI_REO_V2))
waiting for download
縒ts Jun  8 2016 00:22:57

rst:0x1 (POWERON_RESET),boot:0x3 (DOWNLOAD_BOOT(UART0/UART1/SDIO_REI_REO_V2))
waiting for download
縒ts Jun  8 2016 00:22:57

rst:0x1 (POWERON_RESET),boot:0x3 (DOWNLOAD_BOOT(UART0/UART1/SDIO_REI_REO_V2))
waiting for download
```

Figure 3-2. Download Mode Prints 2

- SPI boot mode:

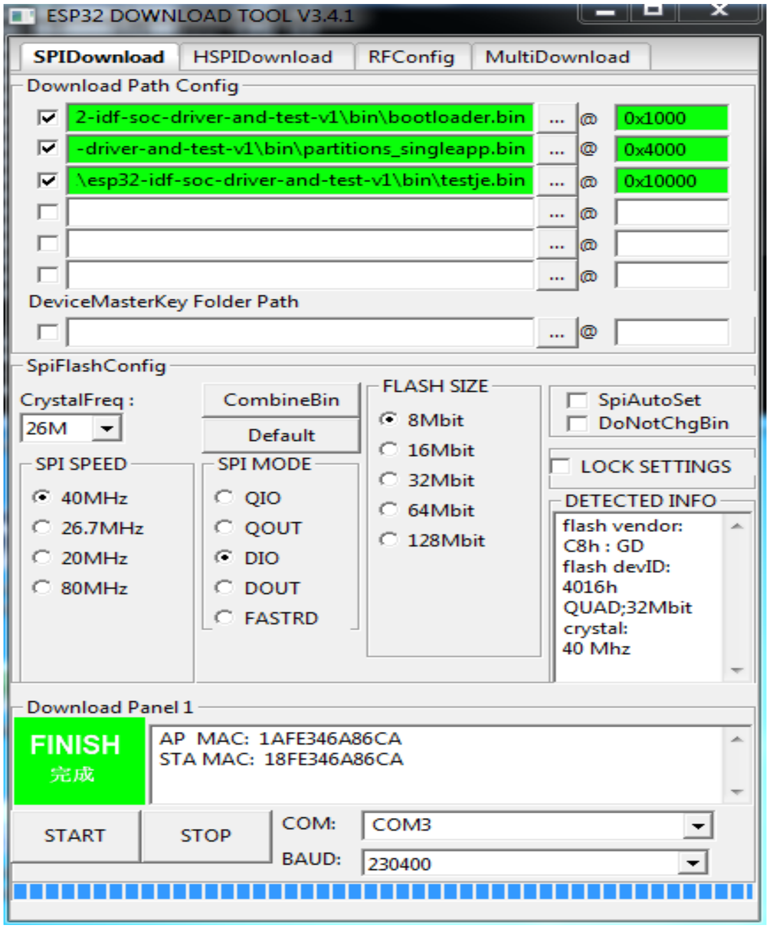
Press the Reset button and the development board will enter the SPI boot mode. If download is completed successfully, the system will have prints as Figure 3-4 shows.

3.3. Download Methods

Use ESP32 DOWNLOAD TOOL

Please download ESP32 DOWNLOAD TOOL V3.4.1 from Espressif website.

Open the ESP32 DOWNLOAD TOOL. Select ***bootloader.bin***, ***paritions_singeapp.bin***, and ***testje.bin***, then enter the addresses 0x1000, 0x4000, and 0x10000 respectively. Press "*START*" and wait for the prompt that indicates the download result.



Download Path Config

☑ 2-idf-soc-driver-and-test-v1\bin\bootloader.bin ... @ 0x1000
☑ -driver-and-test-v1\bin\partitions_singleapp.bin ... @ 0x4000
☑ \esp32-idf-soc-driver-and-test-v1\bin\testje.bin ... @ 0x10000
□ ... @
□ ... @
□ ... @

DeviceMasterKey Folder Path
□ ... @

SpiFlashConfig

CrystalFreq : 26M

SPI SPEED: ● 40MHz ○ 26.7MHz ○ 20MHz ○ 80MHz

SPI MODE: ○ QIO ○ QOUT ● DIO ○ DOUT ○ FASTRD

FLASH SIZE: ● 8Mbit ○ 16Mbit ○ 32Mbit ○ 64Mbit ○ 128Mbit

□ SpiAutoSet
□ DoNotChgBin
□ LOCK SETTINGS

DETECTED INFO
flash vendor:
C8h : GD
flash devID:
4016h
QUAD;32Mbit
crystal:
40 Mhz

Download Panel 1

FINISH
完成

```
AP  MAC:  1AFE346A86CA
STA MAC:  18FE346A86CA
```

COM: COM3
BAUD: 230400

Figure 3-3. ESP32 DOWNLOAD TOOL Interface

Use Python Scripts

Get Python scripts from ***esp-idf/components/esptool_py/esptool/esptool.py*** in the ESP-IDF.

The download process will require the system to have a command terminal. For the Windows System, open the "Windows PowerShell" or "Windows Command Terminal".

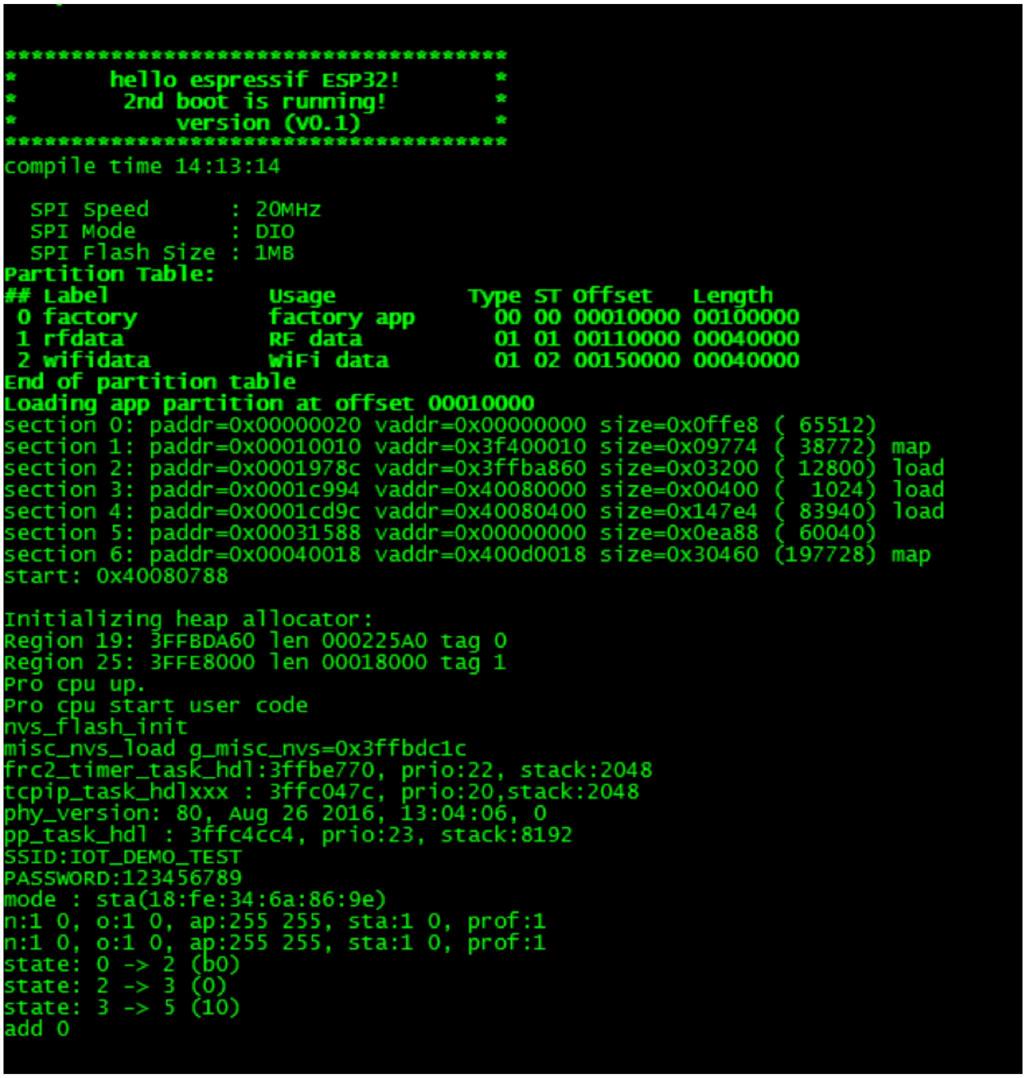
On the terminal, change the current path to be where the download Python scripts are stored, and then type in the following commands:

```
python esptool.py -b 115200 -p COM3 write_flash -ff 40m -fm qio   -ih
0x0 -il 0x00  0x1000 bootloader.bin 0x4000 partitions_singleapp.bin
0x10000 testje.bin
```

The parameters highlighted in blue represent the baud rate, serial port, flash frequency, and the flash mode, respectively. These parameters can be changed as needed.

3.4. Check the Prints

Open the serial tool and press the Reset button. The system is expected to enter the SPI boot mode, and have the following prints:



```
**************************************
*        hello espressif ESP32!      *
*         2nd boot is running!       *
*            version (V0.1)          *
**************************************
compile time 14:13:14

  SPI Speed      : 20MHz
  SPI Mode       : DIO
  SPI Flash Size : 1MB
Partition Table:
## Label            Usage          Type ST Offset   Length
 0 factory          factory app     00 00 00010000 00100000
 1 rfdata           RF data         01 01 00110000 00040000
 2 wifidata         WiFi data       01 02 00150000 00040000
End of partition table
Loading app partition at offset 00010000
section 0: paddr=0x00000020 vaddr=0x00000000 size=0x0ffe8 ( 65512)
section 1: paddr=0x00010010 vaddr=0x3f400010 size=0x09774 ( 38772) map
section 2: paddr=0x0001978c vaddr=0x3ffba860 size=0x03200 ( 12800) load
section 3: paddr=0x0001c994 vaddr=0x40080000 size=0x00400 (  1024) load
section 4: paddr=0x0001cd9c vaddr=0x40080400 size=0x147e4 ( 83940) load
section 5: paddr=0x00031588 vaddr=0x00000000 size=0x0ea88 ( 60040)
section 6: paddr=0x00040018 vaddr=0x400d0018 size=0x30460 (197728) map
start: 0x40080788

Initializing heap allocator:
Region 19: 3FFBDA60 len 000225A0 tag 0
Region 25: 3FFE8000 len 00018000 tag 1
Pro cpu up.
Pro cpu start user code
nvs_flash_init
misc_nvs_load g_misc_nvs=0x3ffbdc1c
frc2_timer_task_hdl:3ffbe770, prio:22, stack:2048
tcpip_task_hdlxxx : 3ffc047c, prio:20,stack:2048
phy_version: 80, Aug 26 2016, 13:04:06, 0
pp_task_hdl : 3ffc4cc4, prio:23, stack:8192
SSID:IOT_DEMO_TEST
PASSWORD:123456789
mode : sta(18:fe:34:6a:86:9e)
n:1 0, o:1 0, ap:255 255, sta:1 0, prof:1
n:1 0, o:1 0, ap:255 255, sta:1 0, prof:1
state: 0 -> 2 (b0)
state: 2 -> 3 (0)
state: 3 -> 5 (10)
add 0
```

Figure 3-4. SPI Boot Mode Prints

Espressif IOT Team

www.espressif.com

Disclaimer and Copyright Notice

Information in this document, including URL references, is subject to change without notice.

THIS DOCUMENT IS PROVIDED AS IS WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE.

All liability, including liability for infringement of any proprietary rights, relating to use of information in this document is disclaimed. No licenses express or implied, by estoppel or otherwise, to any intellectual property rights are granted herein.

The Wi-Fi Alliance Member logo is a trademark of the Wi-Fi Alliance. The Bluetooth logo is a registered trademark of Bluetooth SIG.

All trade names, trademarks and registered trademarks mentioned in this document are property of their respective owners, and are hereby acknowledged.