ESP32

esp-dev-kits Documentation



Release master
Espressif Systems
Sep 16, 2025

Table of contents

Table of contents i

1 ESP32-DevKitC 3
1.1 ESP32-DevKitC V4 . . . 3
1.1.1 What You Need . . . 3
1.1.2 Overview . . . 3
1.1.3 Functional Description . . . 4
1.1.4 Power Supply Options . . . 4
1.1.5 Header Block . . . 5
1.1.6 Note on C15 . . . 6
1.1.7 Start Application Development . . . 6
1.1.8 Related Documents . . . 6

2 ESP32-DevKitM-1 11
2.1 ESP32-DevKitM-1 . . . 11
2.1.1 Getting Started . . . 12
2.1.2 Hardware Reference . . . 13
2.1.3 Hardware Revision Details . . . 15
2.1.4 Related Documents . . . 16

3 ESP32-PICO-KIT-1 17
3.1 ESP32-PICO-KIT-1 . . . 17
3.1.1 Overview . . . 17
3.1.2 Getting Started . . . 19
3.1.3 Contents and Packaging . . . 20
3.1.4 Hardware Reference . . . 20
3.1.5 Hardware Revision Details . . . 22
3.1.6 Related Documents . . . 22

4 ESP32-PICO-DevKitM-2 25
4.1 ESP32-PICO-DevKitM-2 . . . 25
4.1.1 Overview . . . 25
4.1.2 Getting Started . . . 27
4.1.3 Contents and Packaging . . . 28
4.1.4 Hardware Reference . . . 28
4.1.5 Hardware Revision Details . . . 30
4.1.6 Related Documents . . . 30

5 ESP32-LCDKit 33
5.1 ESP32-LCDKit . . . 33
5.1.1 Overview . . . 33
5.1.2 Block Diagram and PCB Layout . . . 34
5.1.3 Functional Modules . . . 34
5.1.4 Related Documents . . . 37

6 ESP32-Ethernet-Kit 39
6.1 ESP32-Ethernet-Kit v1.2 . . . 39

6.1.1 What You Need . . . 39
6.1.2 Overview . . . 39
6.1.3 Functionality Overview . . . 39
6.1.4 Functional Description . . . 42
6.1.5 Setup Options . . . 44
6.1.6 GPIO Allocation . . . 47
6.1.7 Start Application Development . . . 49
6.1.8 Summary of Changes from ESP32-Ethernet-Kit v1.1 . . . 49
6.1.9 Other Versions of ESP32-Ethernet-Kit . . . 49
6.1.10 Related Documents . . . 50

7 EOL (End of Life) Boards 67
7.1 ESP32-Sense-Kit . . . 67
7.1.1 ESP32-Sense-Kit . . . 67
7.2 ESP32-MeshKit-Sense . . . 78
7.2.1 ESP32-MeshKit-Sense . . . 78
7.3 ESP-WROVER-KIT . . . 86
7.3.1 ESP-WROVER-KIT v4.1 Getting Started Guide . . . 86
7.4 ESP32-PICO-KIT . . . 110
7.4.1 ESP32-PICO-KIT v4/v4.1 . . . 110

8 Related Documentation and Resources 123
8.1 Related Documentation . . . 123
8.2 Developer Zone . . . 123
8.3 Products . . . 123
8.4 Contact Us . . . 124

9 Disclaimer and Copyright Notice 125

This document provides detailed user guides and examples for ESP32 series development boards.

Note: For the full list of Espressif development boards, please go to ESP DevKits.

Chapter 1

ESP32-DevKitC

ESP32-DevKitC is a small-sized ESP32-based development board produced by Espressif. Most of the I/O pins are broken out to the pin headers on both sides for easy interfacing. Developers can either connect peripherals with jumper wires or mount ESP32-DevKitC on a breadboard.

1.1 ESP32-DevKitC V4

The older version: *ESP32-DevKitC V2*

This guide shows how to start using the ESP32-DevKitC V4 development board.

1.1.1 What You Need

- *ESP32-DevKitC V4 board*
- USB 2.0 cable (Standard-A to Micro-B)
- Computer running Windows, Linux, or macOS

You can skip the introduction sections and go directly to Section *Start Application Development*.

1.1.2 Overview

ESP32-DevKitC V4 is a small-sized ESP32-based development board produced by Espressif. Most of the I/O pins are broken out to the pin headers on both sides for easy interfacing. Developers can either connect peripherals with jumper wires or mount ESP32-DevKitC V4 on a breadboard.

To cover a wide range of user requirements, the following versions of ESP32-DevKitC V4 are available:

- different ESP32 modules
  - ESP32-WROOM-32E
  - ESP32-WROOM-32UE
  - ESP32-WROVER-E
  - ESP32-WROVER-IE
  - ESP32-WROOM-32D
  - ESP32-WROOM-32U
  - ESP32-WROOM-DA (End of Life)
  - ESP32-SOLO-1
  - ESP32-WROOM-32
- male or female pin headers

For details please refer to ESP Product Selector.

1.1.3 Functional Description

The following figure and the table below describe the key components, interfaces and controls of the ESP32-DevKitC V4 board.

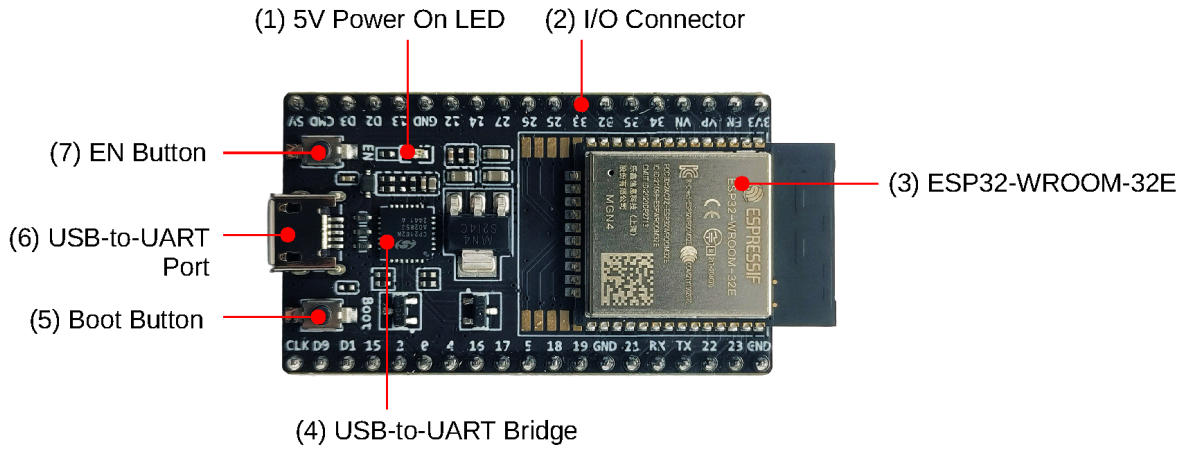


Fig. 1: ESP32-DevKitC V4 with ESP32-WROOM-32E module soldered

The key components of the board are described, starting from the 5V Power On LED, in a clockwise direction.

No.	Key Component	Description
1	5V Power On LED	Turns on when the USB or an external 5V power supply is connected to the board. For details see the schematics in *Related Documents*.
2	I/O Connector	Most of the pins on the ESP module are broken out to the pin headers on the board. You can program ESP32 to enable multiple functions such as PWM, ADC, DAC, I2C, I2S, SPI, etc.
3	ESP32-WROOM-32E	A module with ESP32 at its core. For more information, see ESP32-WROOM-32E Datasheet.
4	USB-to-UART Bridge	Single USB-to-UART bridge chip, providing transfer rates up to 3 Mbps.
5	Boot Button	Download button. Holding down **Boot** and then pressing **EN** initiates Firmware Download mode for downloading firmware through the serial port.
6	USB-to-UART Port	A Micro-USB port used for power supply to the board, as well as for communication between a computer and the ESP32-WROOM-32E module.
7	EN Button	Reset button.

1.1.4 Power Supply Options

There are three mutually exclusive ways to provide power to the board:

- Micro USB port, default power supply
- 5V and GND header pins
- 3V3 and GND header pins

Warning: The power supply must be provided using **one and only one of the options above**, otherwise the board and/or the power supply source can be damaged.

1.1.5 Header Block

The two tables below provide the **Name** and **Function** of I/O header pins on both sides of the board, as shown in *ESP32-DevKitC V4 with ESP32-WROOM-32E module soldered*.

J2

No.	Name	Type[1]	Function
1	3V3	P	3.3 V power supply
2	EN	I	CHIP_PU, Reset
3	VP	I	GPIO36, ADC1_CH0, S_VP
4	VN	I	GPIO39, ADC1_CH3, S_VN
5	IO34	I	GPIO34, ADC1_CH6, VDET_1
6	IO35	I	GPIO35, ADC1_CH7, VDET_2
7	IO32	I/O	GPIO32, ADC1_CH4, TOUCH_CH9, XTAL_32K_P
8	IO33	I/O	GPIO33, ADC1_CH5, TOUCH_CH8, XTAL_32K_N
9	IO25	I/O	GPIO25, ADC2_CH8, DAC_1
10	IO26	I/O	GPIO26, ADC2_CH9, DAC_2
11	IO27	I/O	GPIO27, ADC2_CH7, TOUCH_CH7
12	IO14	I/O	GPIO14, ADC2_CH6, TOUCH_CH6, MTMS
13	IO12	I/O	GPIO12, ADC2_CH5, TOUCH_CH5, MTDI
14	GND	G	Ground
15	IO13	I/O	GPIO13, ADC2_CH4, TOUCH_CH4, MTCK
16	D2	I/O	GPIO9, D2[2]
17	D3	I/O	GPIO10, D3[2]
18	CMD	I/O	GPIO11, CMD[2]
19	5V	P	5 V power supply

[1] P: Power supply; I: Input; O: Output.

[2] The pins D0, D1, D2, D3, CMD and CLK are used internally for communication between ESP32 and SPI flash memory. They are grouped on both sides near the USB connector. Avoid using these pins, as it may disrupt access to the SPI flash memory/SPI RAM.

J3

No.	Name	Type[?]	Function
1	GND	G	Ground
2	IO23	I/O	GPIO23
3	IO22	I/O	GPIO22
4	TX	I/O	GPIO1, U0TXD
5	RX	I/O	GPIO3, U0RXD
6	IO21	I/O	GPIO21
7	GND	G	Ground
8	IO19	I/O	GPIO19
9	IO18	I/O	GPIO18
10	IO5	I/O	GPIO5
11	IO17	I/O	GPIO17[3]
12	IO16	I/O	GPIO16[Page 6, 3]
13	IO4	I/O	GPIO4, ADC2_CH0, TOUCH_CH0
14	IO0	I/O	GPIO0, ADC2_CH1, TOUCH_CH1, Boot
15	IO2	I/O	GPIO2, ADC2_CH2, TOUCH_CH2
16	IO15	I/O	GPIO15, ADC2_CH3, TOUCH_CH3, MTDO
17	D1	I/O	GPIO8, D1[?]
18	D0	I/O	GPIO7, D0[?]
19	CLK	I/O	GPIO6, CLK[?]

Pin Layout

1.1.6 Note on C15

The component C15 may cause the following issues on earlier ESP32-DevKitC V4 boards:

- The board may boot into Download mode
- If you output clock on GPIO0, C15 may impact the signal

In case these issues occur, please remove the component. The figure below shows the location of C15 highlighted in yellow.

1.1.7 Start Application Development

Before powering up your ESP32-DevKitC V4, please make sure that the board is in good condition with no obvious signs of damage.

After that, proceed to ESP-IDF Get Started, which will quickly help you set up the development environment then flash an application example onto your board.

1.1.8 Related Documents

- ESP32 Datasheet (PDF)
- ESP32-DevKitC V4 Schematics (PDF)
- ESP32-DevKitC V4 PCB Layout (PDF)
- ESP32-DevKitC V4 Dimensions (PDF)
- ESP32-DevKitC V4 Dimensions source file (DXF) - You can view it with Autodesk Viewer online
- ESP Product Selector

For further design documentation for the board, please contact us at sales@espressif.com.

[3] The pins GPIO16 and GPIO17 are available for use only on the boards with the modules ESP32-WROOM and ESP32-SOLO-1. The boards with ESP32-WROVER modules have the pins reserved for internal use.

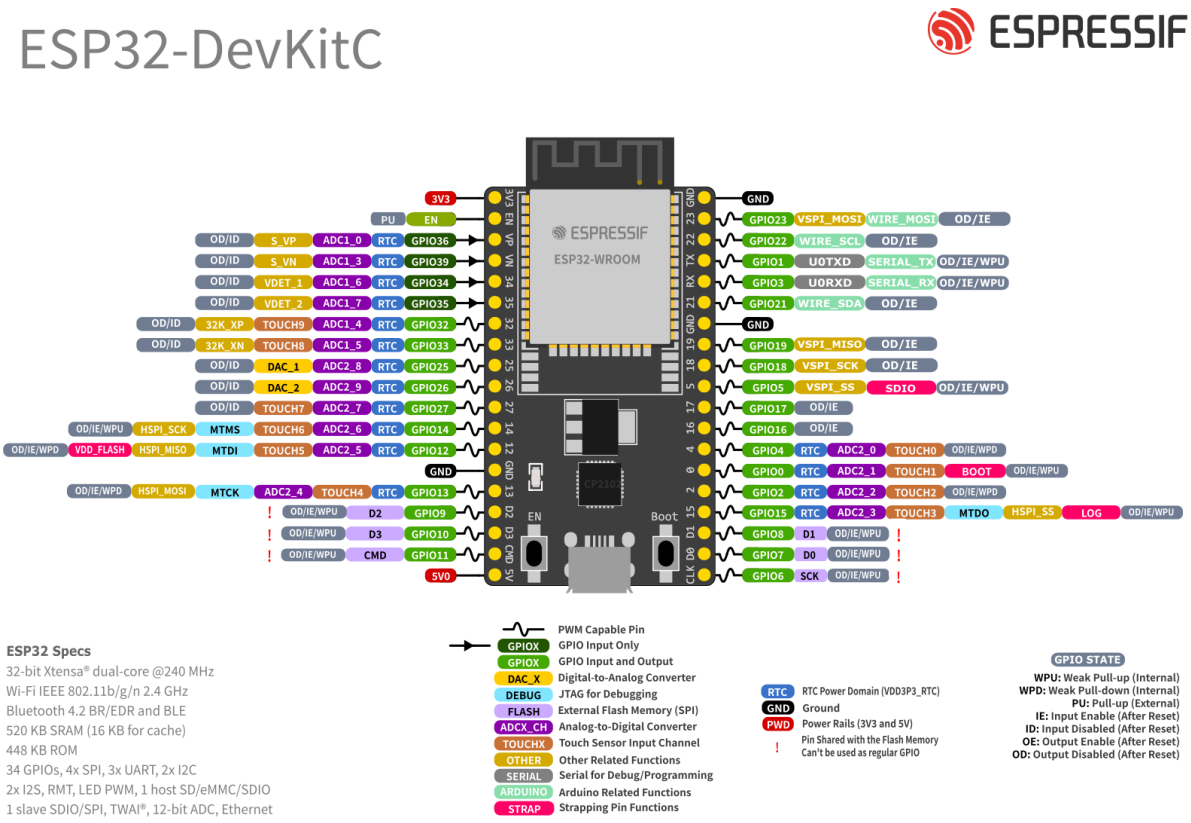


Fig. 2: ESP32-DevKitC Pin Layout (click to enlarge)

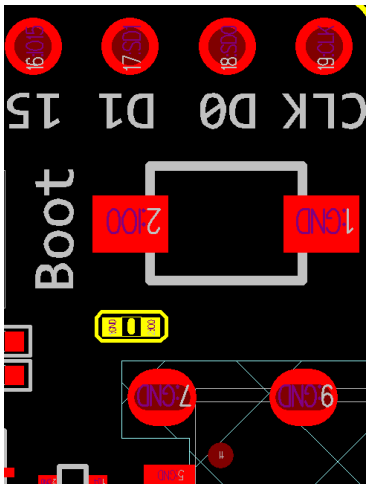


Fig. 3: Location of C15 (yellow) on ESP32-DevKitC V4 board

ESP32-DevKitC V2

New version available: *ESP32-DevKitC V4*

This guide shows how to start using the ESP32-DevKitC V2 development board.

What You Need

- *ESP32-DevKitC V2 board*
- USB 2.0 cable (Standard-A to Micro-B)
- Computer running Windows, Linux, or macOS

You can skip the introduction sections and go directly to Section *Start Application Development*.

Overview ESP32-DevKitC V2 is a small-sized ESP32-based development board produced by Espressif. Most of the I/O pins are broken out to the pin headers on both sides for easy interfacing. Developers can either connect peripherals with jumper wires or mount ESP32-DevKitC V4 on a breadboard.

Functional Description The following figure and the table below describe the key components, interfaces and controls of the ESP32-DevKitC V2 board.

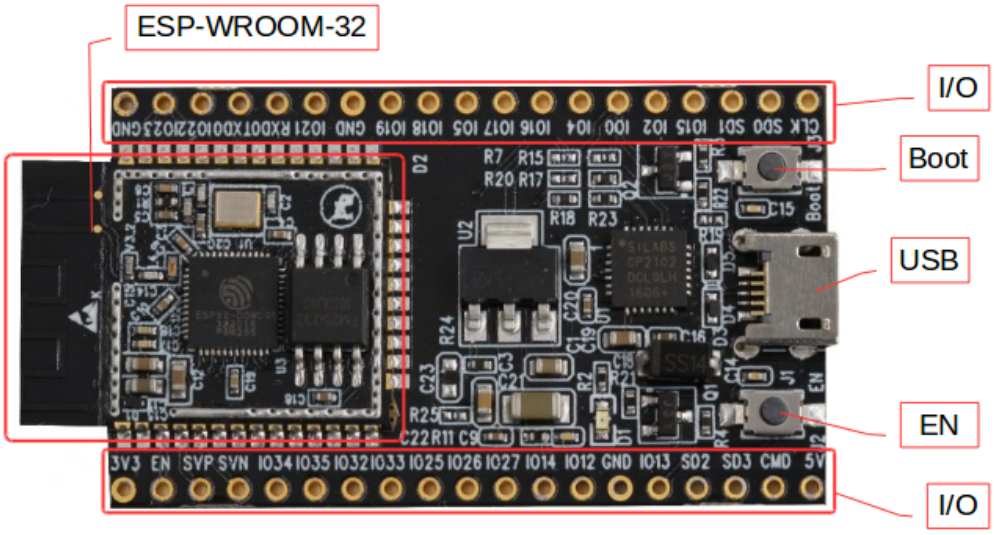


Fig. 4: ESP32-DevKitC V2 board layout

Key Component	Description
ESP32-WROOM-32	Standard module with ESP32 at its core. For more information, see ESP32-WROOM-32 Datasheet
EN	Reset button.
Boot	Download button. Holding down **Boot** and then pressing **EN** initiates Firmware Download mode for downloading firmware through the serial port.
Micro USB Port	USB interface. Power supply for the board as well as the communication interface between a computer and ESP32-WROOM-32.
I/O	Most of the pins on the ESP module are broken out to the pin headers on the board. You can program ESP32 to enable multiple functions such as PWM, ADC, DAC, I2C, I2S, SPI, etc.

Power Supply Options There are three mutually exclusive ways to provide power to the board:

- Micro USB port, default power supply
- 5V and GND header pins
- 3V3 and GND header pins

Warning: The power supply must be provided using **one and only one of the options above**, otherwise the board and/or the power supply source can be damaged.

Start Application Development Before powering up your ESP32-DevKitC V2, please make sure that the board is in good condition with no obvious signs of damage.

After that, proceed to ESP-IDF Get Started, which will quickly help you set up the development environment then flash an application example onto your board.

Related Documents

- ESP32-DevKitC schematics (PDF)
- ESP32 Datasheet (PDF)
- ESP32-WROOM-32 Datasheet (PDF)

Chapter 2

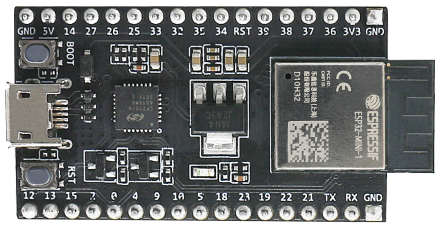
ESP32-DevKitM-1

The *ESP32-DevKitM-1* is a ESP32-MINI-1-based development board produced by Espressif. Most of the I/O pins are broken out to the pin headers on both sides for easy interfacing. Developers can either connect peripherals with jumper wires or mount ESP32-DevKitM-1 on a breadboard.

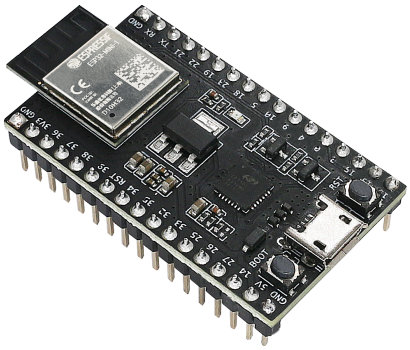
2.1 ESP32-DevKitM-1

This user guide will help you get started with ESP32-DevKitM-1 and will also provide more in-depth information.

ESP32-DevKitM-1 is an ESP32-MINI-1/1U-based development board produced by Espressif. Most of the I/O pins are broken out to the pin headers on both sides for easy interfacing. Users can either connect peripherals with jumper wires or mount ESP32-DevKitM-1 on a breadboard.



ESP32-DevKitM-1 - front



ESP32-DevKitM-1 - isometric

The document consists of the following major sections:

- *Getting started*: Provides an overview of the ESP32-DevKitM-1 and hardware/software setup instructions to get started.
- *Hardware reference*: Provides more detailed information about the ESP32-DevKitM-1' s hardware.
- *Related Documents*: Gives links to related documentation.

2.1.1 Getting Started

This section describes how to get started with ESP32-DevKitM-1. It begins with a few introductory sections about the ESP32-DevKitM-1, then Section *Start Application Development* provides instructions on how to do the initial hardware setup and then how to flash firmware onto the ESP32-DevKitM-1.

Overview

This is a small and convenient development board that features:

- ESP32-MINI-1, or ESP32-MINI-1U module
- USB-to-serial programming interface that also provides power supply for the board
- pin headers
- pushbuttons for reset and activation of Firmware Download mode
- a few other components

Contents and Packaging

Retail Orders If you order a few samples, each ESP32-DevKitM-1 comes in an individual package in either anti-static bag or any packaging depending on your retailer.

For retail orders, please Get Samples.

Wholesale Orders If you order in bulk, the boards come in large cardboard boxes.

For wholesale orders, please Contact Sales.

Description of Components

The following figure and the table below describe the key components, interfaces and controls of the ESP32-DevKitM-1 board. We take the board with a ESP32-MINI-1 module as an example in the following sections.

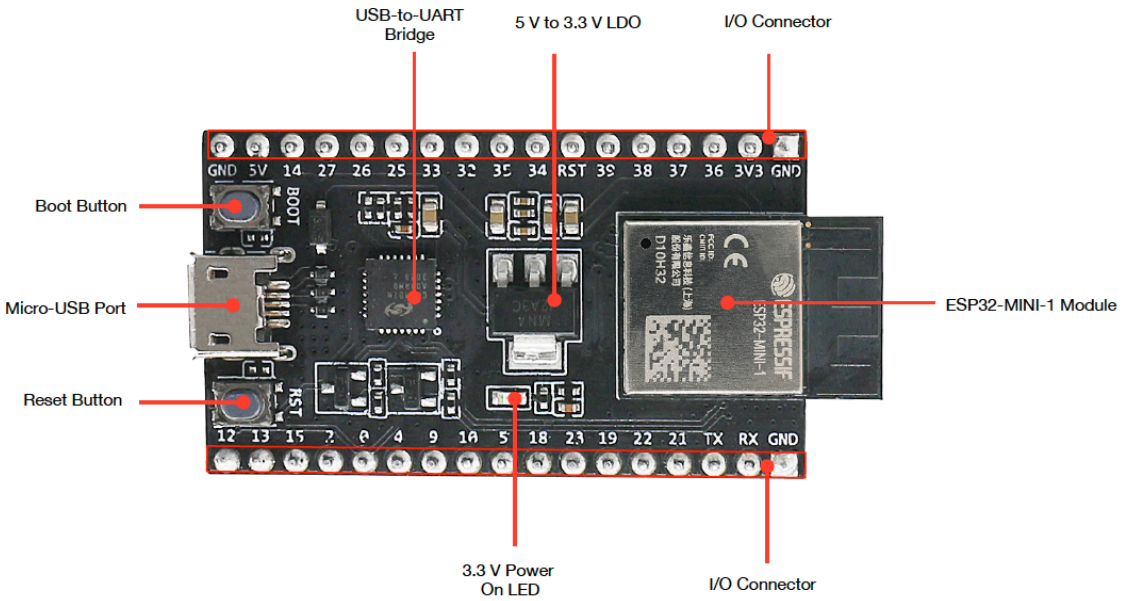


Fig. 1: ESP32-DevKitM-1 - front

Key Component	Description
On-board module	ESP32-MINI-1 module or ESP32-MINI-1U module. ESP32-MINI-1 comes with an on-board PCB antenna. ESP32-MINI-1U comes with an external antenna connector. The two modules both have a 4 MB flash in chip package. For details, please see ESP32-MINI-1 & ESP32-MINI-1U Datasheet.
5 V to 3.3 V LDO	Power regulator converts 5 V to 3.3 V.
Boot Button	Download button. Holding down **Boot** and then pressing **Reset** initiates Firmware Download mode for downloading firmware through the serial port.
Reset Button	Reset Button
Micro-USB Port	USB interface. Power supply for the board as well as the communication interface between a computer and the ESP32 chip.
USB-to-UART Bridge	Single USB-UART bridge chip provides transfer rates up to 3 Mbps.
3.3 V Power On LED	Turns on when the USB is connected to the board. For details, please see the schematics in *Related Documents*.
I/O Connector	All available GPIO pins (except for the SPI bus for flash) are broken out to the pin headers on the board. Users can program ESP32 chip to enable multiple functions.

Start Application Development

Before powering up your ESP32-DevKitM-1, please make sure that it is in good condition with no obvious signs of damage.

Required Hardware

- ESP32-DevKitM-1
- USB 2.0 cable (Standard-A to Micro-B)
- Computer running Windows, Linux, or macOS

Software Setup Please proceed to Get Started, where Section Installation will quickly help you set up the development environment and then flash an application example onto your ESP32-DevKitM-1.

Attention: ESP32-DevKitM-1 boards manufactured before December 2, 2021 have a single core module installed. To verify what module you have, please check module marking information in PCN-2021-021 . If your board has a single core module installed, please enable single core mode with CONFIG_FREERTOS_UNICORE in menuconfig before flashing your applications.

2.1.2 Hardware Reference

Block Diagram

A block diagram below shows the components of ESP32-DevKitM-1 and their interconnections.

Power Source Select

There are three mutually exclusive ways to provide power to the board:

- Micro USB port, default power supply
- 5V and GND header pins
- 3V3 and GND header pins

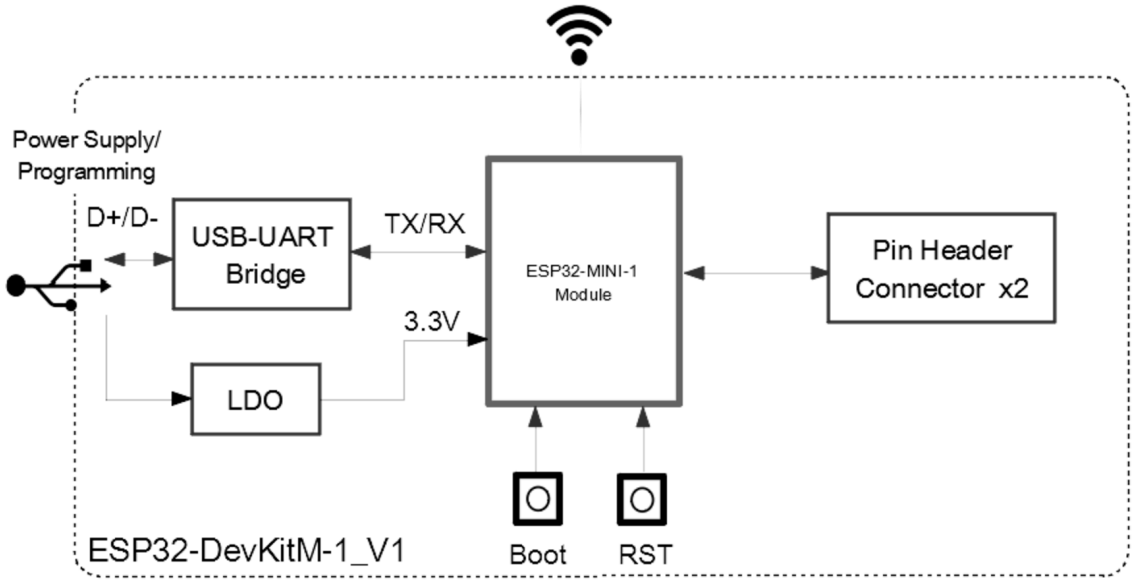


Fig. 2: ESP32-DevKitM-1

Warning:

- The power supply must be provided using **one and only one of the options above**, otherwise the board and/or the power supply source can be damaged.
- Power supply by micro USB port is recommended.

Pin Descriptions

The table below provides the Name and Function of pins on both sides of the board. For peripheral pin configurations, please refer to ESP32 Datasheet.

No.	Name	Type[Page 15, 1]	Function
1	GND	P	Ground
2	3V3	P	3.3 V power supply
3	I36	I	GPIO36, ADC1_CH0, RTC_GPIO0
4	I37	I	GPIO37, ADC1_CH1, RTC_GPIO1
5	I38	I	GPIO38, ADC1_CH2, RTC_GPIO2
6	I39	I	GPIO39, ADC1_CH3, RTC_GPIO3
7	RST	I	Reset; High: enable; Low: powers off
8	I34	I	GPIO34, ADC1_CH6, RTC_GPIO4
9	I35	I	GPIO35, ADC1_CH7, RTC_GPIO5
10	IO32	I/O	GPIO32, XTAL_32K_P (32.768 kHz crystal oscillator input), ADC1_CH4, TOUCH9, RTC_GPIO9
11	IO33	I/O	GPIO33, XTAL_32K_N (32.768 kHz crystal oscillator output), ADC1_CH5, TOUCH8, RTC_GPIO8
12	IO25	I/O	GPIO25, DAC_1, ADC2_CH8, RTC_GPIO6, EMAC_RXD0
13	IO26	I/O	GPIO26, DAC_2, ADC2_CH9, RTC_GPIO7, EMAC_RXD1
14	IO27	I/O	GPIO27, ADC2_CH7, TOUCH7, RTC_GPIO17, EMAC_RX_DV

continues on next page

Table 1 – continued from previous page

No.	Name	Type[Page 15, 1]	Function
15	IO14	I/O	GPIO14, ADC2_CH6, TOUCH6, RTC_GPIO16, MTMS, HSPICLK, HS2_CLK, SD_CLK, EMAC_TXD2
16	5V	P	5 V power supply
17	IO12	I/O	GPIO12, ADC2_CH5, TOUCH5, RTC_GPIO15, MTDI[2], HSPIQ, HS2_DATA2, SD_DATA2, EMAC_TXD3
18	IO13	I/O	GPIO13, ADC2_CH4, TOUCH4, RTC_GPIO14, MTCK, HSPID, HS2_DATA3, SD_DATA3, EMAC_RX_ER
19	IO15	I/O	GPIO15, ADC2_CH3, TOUCH3, RTC_GPIO13, MTDO[Page 15, 2], HSPICS0, HS2_CMD, SD_CMD, EMAC_RXD3
20	IO2	I/O	GPIO2[Page 15, 2], ADC2_CH2, TOUCH2, RTC_GPIO12, HSPIWP, HS2_DATA0, SD_DATA0
21	IO0	I/O	GPIO0[Page 15, 2], ADC2_CH1, TOUCH1, RTC_GPIO11, CLK_OUT1, EMAC_TX_CLK
22	IO4	I/O	GPIO4, ADC2_CH0, TOUCH0, RTC_GPIO10, HSPIHD, HS2_DATA1, SD_DATA1, EMAC_TX_ER
23	IO9	I/O	GPIO9, HS1_DATA2, U1RXD, SD_DATA2
24	IO10	I/O	GPIO10, HS1_DATA3, U1TXD, SD_DATA3
25	IO5	I/O	GPIO5[Page 15, 2], HS1_DATA6, VSPICS0, EMAC_RX_CLK
26	IO18	I/O	GPIO18, HS1_DATA7, VSPICLK
27	IO23	I/O	GPIO23, HS1_STROBE, VSPID
28	IO19	I/O	GPIO19, VSPIQ, U0CTS, EMAC_TXD0
29	IO22	I/O	GPIO22, VSPIWP, U0RTS, EMAC_TXD1
30	IO21	I/O	GPIO21, VSPIHD, EMAC_TX_EN
31	TXD0	I/O	GPIO1, U0TXD, CLK_OUT3, EMAC_RXD2
32	RXD0	I/O	GPIO3, U0RXD, CLK_OUT2

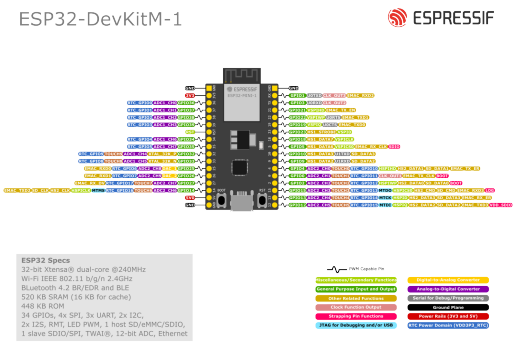


Fig. 3: ESP32-DevKitM-1 (click to enlarge)

Pin Layout

2.1.3 Hardware Revision Details

No previous versions available.

[1] P: Power supply; I: Input; O: Output.

[2] MTDI, GPIO0, GPIO2, MTDO, and GPIO5 are strapping pins. These pins are used to control several chip functions depending on binary voltage values applied to the pins during chip power-up or system reset. For description and application of the strapping pins, please refer to ESP32 Datasheet > *Boot Configurations*.

2.1.4 Related Documents

- ESP32-MINI-1 & ESP32-MINI-1U Datasheet (PDF)
- ESP32-DevKitM-1 Schematics (PDF)
- ESP32-DevKitM-1 PCB layout (PDF)
- ESP32-DevKitM-1 layout (DXF) - You can view it with Autodesk Viewer online
- ESP32 Datasheet (PDF)
- ESP Product Selector

For other design documentation for the board, please contact us at sales@espressif.com.

Chapter 3

ESP32-PICO-KIT-1

The *ESP32-PICO-KIT-1* is an ESP32-based development board produced by Espressif. ESP32-PICO-KIT-1 provides users with hardware for development of applications based on ESP32, making it easier for users to explore ESP32 functionalities.

3.1 ESP32-PICO-KIT-1

3.1.1 Overview

ESP32-PICO-KIT-1 is an ESP32-based development board produced by Espressif.

The core of this board is ESP32-PICO-V3 - a System-in-Package (SiP) module with complete Wi-Fi and Bluetooth® functionalities. Compared to other ESP32 modules, ESP32-PICO-V3 integrates the following peripheral components in one single package, which otherwise would need to be installed separately:

- 40 MHz crystal oscillator
- 4 MB flash
- Filter capacitors
- RF matching network

This setup reduces the costs of additional external components as well as the cost of assembly and testing and also increases the overall usability of the product.

The development board features a USB-to-UART Bridge circuit which allows developers to connect the board to a computer ' s USB port for flashing and debugging.

All the IO signals and system power on ESP32-PICO-V3 are led out to two rows of 18 x 0.1" header pads on both sides of the development board for easy access. For compatibility with Dupont wires, all header pads are populated with two rows of male pin headers.

Note: ESP32-PICO-KIT-1 comes with male headers by default.

ESP32-PICO-KIT-1 provides the users with hardware for development of applications based on the ESP32, making it easier for users to explore ESP32 functionalities.

This guide covers:

- *Getting Started*: Provides an overview of the ESP32-PICO-KIT-1 and software setup instructions to get started.
- *Contents and Packaging*: Provides information about packaging and contents for retail and wholesale orders.

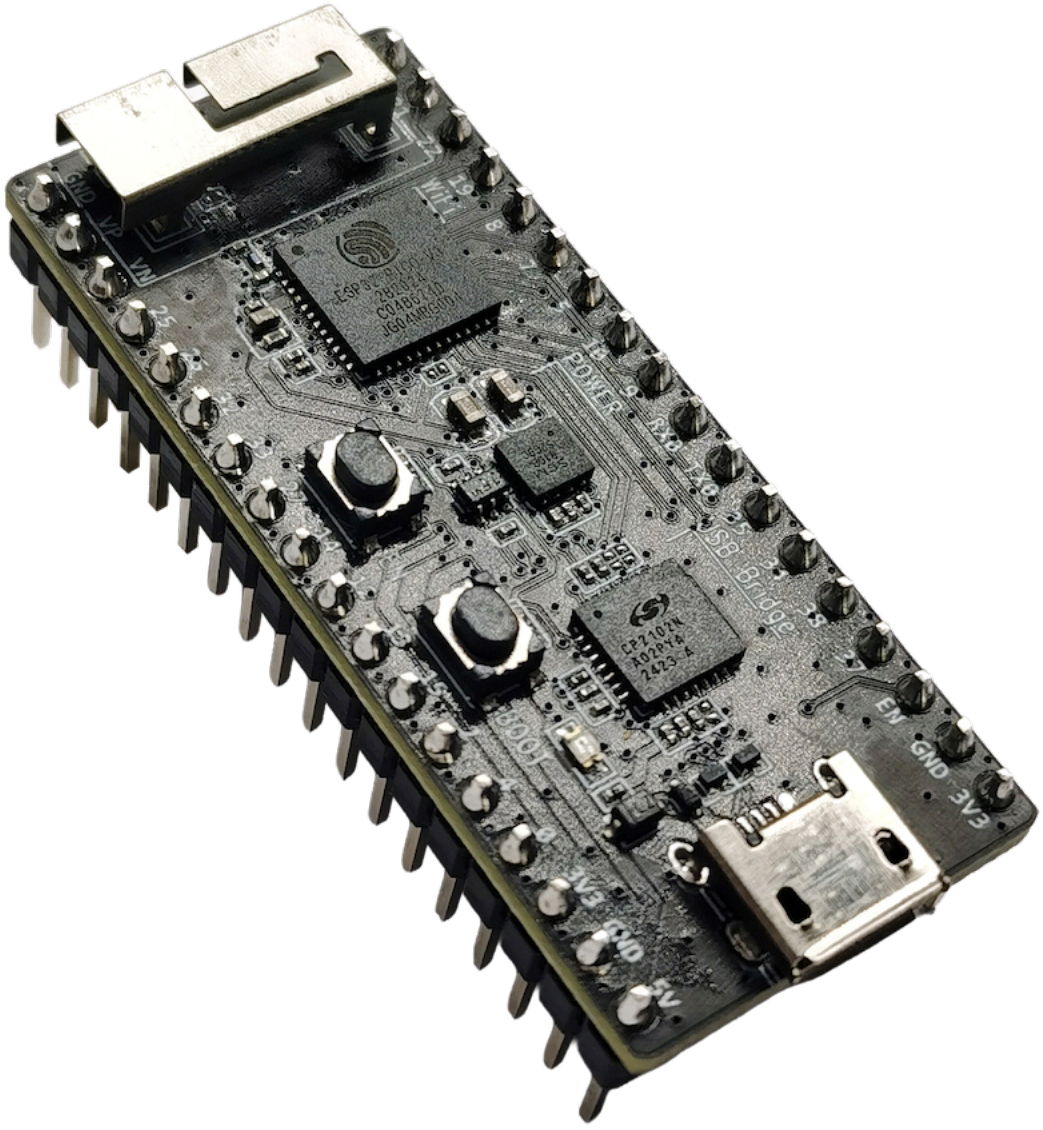


Fig. 1: ESP32-PICO-KIT-1 Overview (click to enlarge)

- *Hardware Reference*: Provides more detailed information about the ESP32-PICO-KIT-1 ' s hardware.
- *Hardware Revision Details*: Covers revision history, known issues, and links to user guides for previous versions of the ESP32-PICO-KIT-1.
- *Related Documents*: Gives links to related documentation.

3.1.2 Getting Started

This section describes how to get started with ESP32-PICO-KIT-1. It begins with a few introductory sections about ESP32-PICO-KIT-1, and then section *Start Application Development* provides instructions on how to flash firmware onto ESP32-PICO-KIT-1.

Description of Components

The following figure and the table below describe the key components, interfaces, and controls of the ESP32-PICO-KIT-1 board.

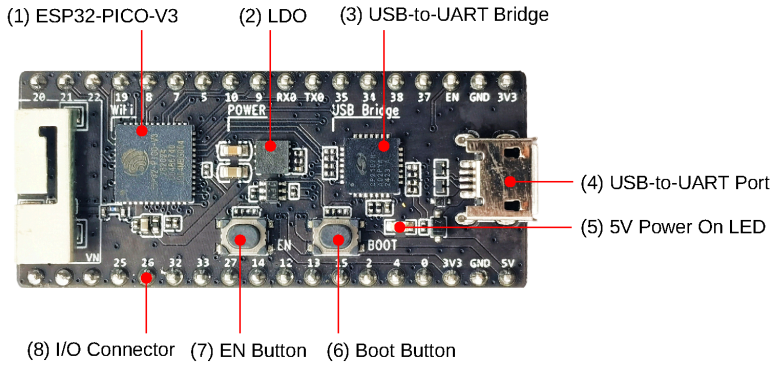


Fig. 2: ESP32-PICO-KIT-1 board layout - front (click to enlarge)

The key components of the board are described, starting from the ESP32-PICO-V3, in a clockwise direction.

No.	Key Component	Description
1	ESP32-PICO-V3	Standard ESP32-PICO-V3 module soldered to the ESP32-PICO-KIT-1 board. The complete ESP32 system on a chip (ESP32 SoC) has been integrated into the SiP module, requiring only an external antenna with LC matching network, decoupling capacitors, and a pull-up resistor for EN signals to function properly.
2	LDO	5V-to-3.3V Low dropout voltage regulator (LDO).
3	USB-to-UART Bridge	CP2102N, single USB-to-UART bridge chip, providing transfer rates up to 3 Mbps.
4	USB-to-UART Port	A Micro-USB port used for power supply to the board, as well as for communication between a computer and the board.
5	5V Power On LED	This red LED turns on when power is supplied to the board. For details, see the schematic in *Related Documents*.
6	Boot Button	Download button. Holding down **Boot** and then pressing **EN** initiates Firmware Download mode for downloading firmware through the serial port.
7	EN Button	Reset button.
8	I/O Connector	All the pins on ESP32-PICO-V3 are broken out to pin headers. You can program ESP32 to enable multiple functions, such as PWM, ADC, DAC, I2C, I2S, SPI, etc. For details, please see Section *Pin Descriptions*.

Start Application Development

Before powering up your ESP32-PICO-KIT-1, please make sure that the board is in good condition with no obvious signs of damage.

Required Hardware

- 1 x ESP32-PICO-KIT-1
- 1 x USB 2.0 cable (Standard-A to Micro-B)
- 1 x Computer running Windows, Linux, or macOS

Software Setup Please proceed to Get Started, where section Installation will quickly help you set up the development environment.

3.1.3 Contents and Packaging

Retail Orders

If you order one or several samples of the board, each ESP32-PICO-KIT-1 development board comes in an individual package.

For retail orders, please Get Samples.

Wholesale Orders

If you order in bulk, the boards come in large cardboard boxes.

For wholesale orders, please Contact Sales.

3.1.4 Hardware Reference

Block Diagram

The block diagram below shows the main components of ESP32-PICO-KIT-1 and their interconnections.

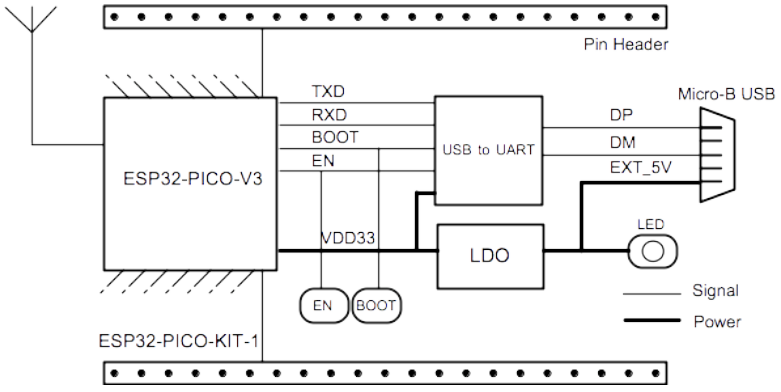


Fig. 3: ESP32-PICO-KIT-1 Block Diagram (click to enlarge)

Power Supply Options

There are three mutually exclusive ways to provide power to the board:

- Micro USB port, default power supply
- 5V/GND header pins
- 3V3/GND header pins

Warning: The power supply must be provided using **one and only one of the options above**, otherwise the board and/or the power supply source can be damaged.

Pin Descriptions

The two tables below provide the **Name** and **Function** of I/O header pins on both sides of the board, see *Description of Components*. The pin numbering and header names are the same as in the schematic given in *Related Documents*.

Header J2

No.	Name	Type	Function
1	IO20	I/O	GPIO20
2	IO21	I/O	GPIO21, VSPIHD, EMAC_TX_EN
3	IO22	I/O	GPIO22, VSPIWP, U0RTS, EMAC_TXD1
4	IO19	I/O	GPIO19, VSPIQ, U0CTS, EMAC_TXD0
5	IO8	I/O	GPIO8, SD_DATA1, HS1_DATA1, U2CTS
6	IO7	I/O	GPIO7, SD_DATA0, HS1_DATA0, U2RTS
7	IO5	I/O	GPIO5, VSPICS0, HS1_DATA6, EMAC_RX_CLK
8	IO10	I/O	GPIO10, SD_DATA3, SPIWP, HS1_DATA3, U1TXD
9	IO9	I/O	GPIO9, SD_DATA2, SPIHD, HS1_DATA2, U1RXD
10	RXD0	I/O	GPIO3, U0RXD *(See note 1)*, CLK_OUT2
11	TXD0	I/O	GPIO1, U0TXD *(See note 1)*, CLK_OUT3, EMAC_RXD2
12	IO35	I	ADC1_CH7, RTC_GPIO5
13	IO34	I	ADC1_CH6, RTC_GPIO4
14	IO38	I	GPIO38, ADC1_CH2, RTC_GPIO2
15	IO37	I	GPIO37, ADC1_CH1, RTC_GPIO1
16	EN	I	CHIP_PU
17	GND	P	Ground
18	VDD33 (3V3)	P	3.3 V power supply

Header J3

No.	Name	Type	Function
1	GND	P	Ground
2	SENSOR_VP (FSVP)	I	GPIO36, ADC1_CH0, RTC_GPIO0
3	SENSOR_VN (FSVN)	I	GPIO39, ADC1_CH3, RTC_GPIO3
4	IO25	I/O	GPIO25, DAC_1, ADC2_CH8, RTC_GPIO6, EMAC_RXD0
5	IO26	I/O	GPIO26, DAC_2, ADC2_CH9, RTC_GPIO7, EMAC_RXD1
6	IO32	I/O	32K_XP *(See note 2a)*, ADC1_CH4, TOUCH9, RTC_GPIO9
7	IO33	I/O	32K_XN *(See note 2b)*, ADC1_CH5, TOUCH8, RTC_GPIO8
8	IO27	I/O	GPIO27, ADC2_CH7, TOUCH7, RTC_GPIO17, EMAC_RX_DV
9	IO14	I/O	ADC2_CH6, TOUCH6, RTC_GPIO16, MTMS, HSPICLK, HS2_CLK, SD_CLK, EMAC_TXD2
10	IO12	I/O	ADC2_CH5, TOUCH5, RTC_GPIO15, MTDI *(See note 3)*, HSPIQ, HS2_DATA2, SD_DATA2, EMAC_TXD3
11	IO13	I/O	ADC2_CH4, TOUCH4, RTC_GPIO14, MTCK, HSPID, HS2_DATA3, SD_DATA3, EMAC_RX_ER
12	IO15	I/O	ADC2_CH3, TOUCH3, RTC_GPIO13, MTDO, HSPICS0, HS2_CMD, SD_CMD, EMAC_RXD3
13	IO2	I/O	ADC2_CH2, TOUCH2, RTC_GPIO12, HSPIWP, HS2_DATA0, SD_DATA0
14	IO4	I/O	ADC2_CH0, TOUCH0, RTC_GPIO10, HSPIHD, HS2_DATA1, SD_DATA1, EMAC_TX_ER
15	IO0	I/O	ADC2_CH1, TOUCH1, RTC_GPIO11, CLK_OUT1, EMAC_TX_CLK
16	VDD33 (3V3)	P	3.3 V power supply
17	GND	P	Ground
18	EXT_5V (5V)	P	5 V power supply

Note:

1. This pin is connected to the pin of the USB bridge chip on the board.
2. 32.768 kHz crystal oscillator: (a) input; (b) output.
3. The operating voltage of ESP32-PICO-KIT-1’ s embedded SPI flash is 3.3 V. Therefore, the strapping pin MTDI should be pulled down during the module power-on reset. If connected, please make sure that this pin is not held up on reset.

Pin Layout

3.1.5 Hardware Revision Details

No previous versions available.

3.1.6 Related Documents

- ESP32-PICO-V3 Datasheet (PDF)
- ESP Product Selector
- ESP32-PICO-KIT-1 Schematic (PDF)

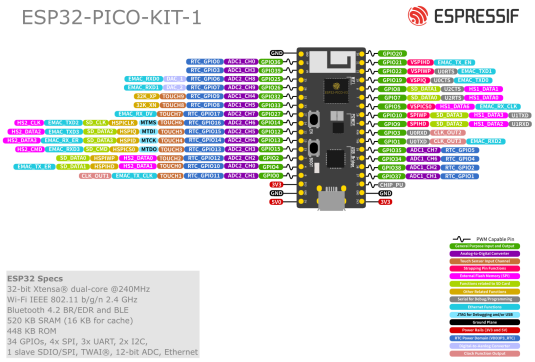


Fig. 4: ESP32-PICO-KIT-1 Pin Layout (click to enlarge)

- ESP32-PICO-KIT-1 PCB Layout (PDF)

For other design documentation for the board, please contact us at sales@espressif.com.

Chapter 4

ESP32-PICO-DevKitM-2

ESP32-PICO-DevKitM-2 is an ESP32-based development board produced by Espressif.

The core of this board is ESP32-PICO-MINI-02/02U module with complete Wi-Fi and Bluetooth® functionalities. The development board features a USB-to-UART Bridge circuit which allows developers to connect the board to a computer's USB port for flashing and debugging.

4.1 ESP32-PICO-DevKitM-2

4.1.1 Overview

ESP32-PICO-DevKitM-2 is an ESP32-based development board produced by Espressif.

The core of this board is ESP32-PICO-MINI-02/02U module with complete Wi-Fi and Bluetooth® functionalities. The development board features a USB-to-UART Bridge circuit which allows developers to connect the board to a computer's USB port for flashing and debugging.

All the IO signals and system power on ESP32-PICO-MINI-02/02U are led out to two rows of 18 x 0.1" header pads on both sides of the development board for easy access. For compatibility with Dupont wires, all header pads are populated with two rows of male pin headers.

Note: ESP32-PICO-DevKitM-2 comes with male headers by default.

ESP32-PICO-DevKitM-2 provides the users with hardware for development of applications based on the ESP32, making it easier for users to explore ESP32 functionalities.

This guide covers:

- *Getting Started*: Provides an overview of the ESP32-PICO-DevKitM-2 and software setup instructions to get started.
- *Contents and Packaging*: Provides information about packaging and contents for retail and wholesale orders.
- *Hardware Reference*: Provides more detailed information about the ESP32-PICO-DevKitM-2's hardware.
- *Hardware Revision Details*: Covers revision history, known issues, and links to user guides for previous versions (if any) of the ESP32-PICO-DevKitM-2.
- *Related Documents*: Gives links to related documentation.

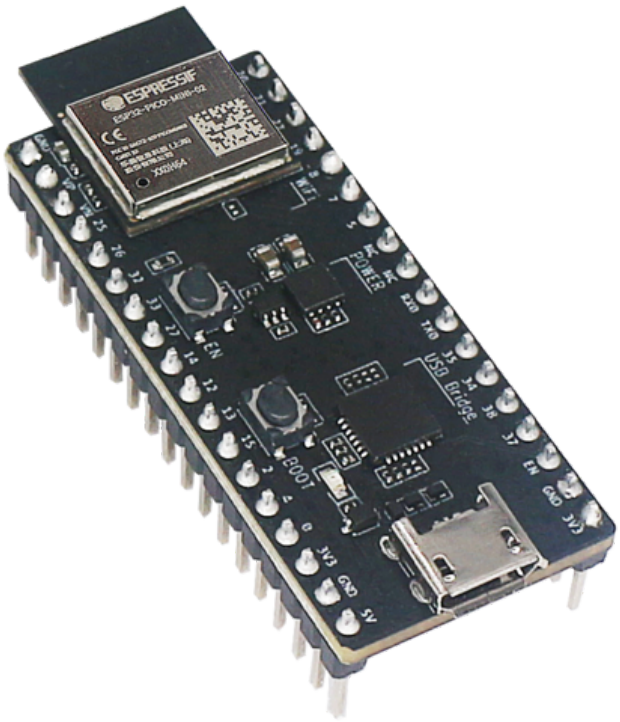


Fig. 1: ESP32-PICO-DevKitM-2 Overview (click to enlarge)

4.1.2 Getting Started

This section describes how to get started with the ESP32-PICO-DevKitM-2. It begins with a few introductory sections about the ESP32-PICO-DevKitM-2, then Section *Start Application Development* provides instructions on how to flash firmware onto the ESP32-PICO-DevKitM-2.

Description of Components

The following figure and the table below describe the key components, interfaces, and controls of the ESP32-PICO-DevKitM-2 board. We take the board with a ESP32-PICO-MINI-02 module as an example in the following sections.

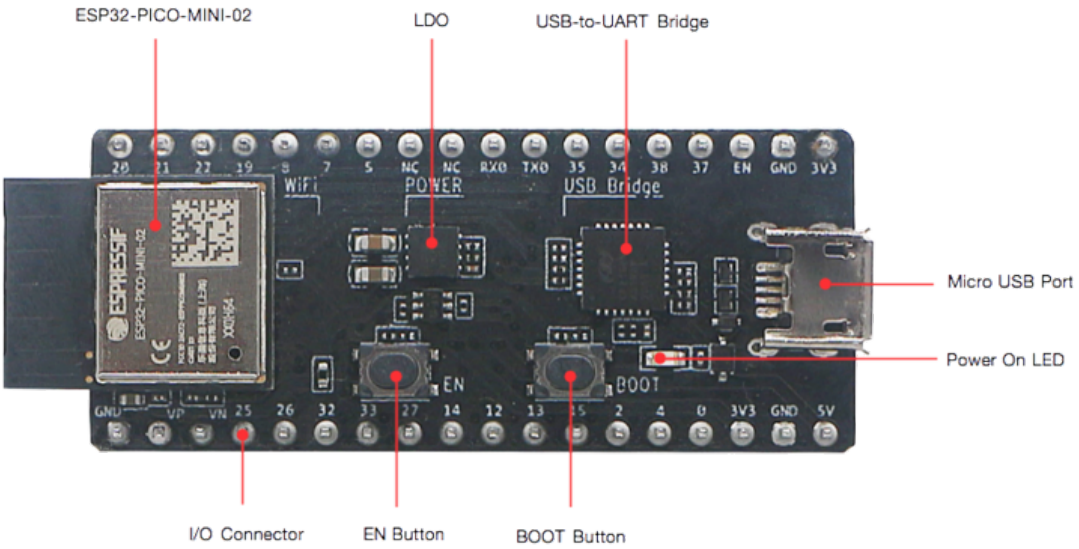


Fig. 2: ESP32-PICO-DevKitM-2 board layout - front (click to enlarge)

Below is the description of the items identified in the figure starting from the top left corner and going clockwise.

Key Component	Description
ESP32-PICO-MINI-02	Standard ESP32-PICO-MINI-02 module soldered to the ESP32-PICO-DevKitM-2 board. The complete ESP32 system on a chip (ESP32 SoC) has been integrated into the module. Users can also select the board with ESP32-PICO-MINI-02U soldered.
LDO	V-to-3.3V Low dropout voltage regulator (LDO).
USB-to-UART bridge	CP2102N, single-chip USB-UART bridge that offers up to 3 Mbps transfers rates.
Micro-B USB Port	USB interface. Power supply for the board as well as the communication interface between a computer and the board.
5V Power On LED	This red LED turns on when power is supplied to the board. For details, see the schematic in *Related Documents*.
I/O Connector	All the pins on ESP32-PICO-MINI-02 are broken out to pin headers. You can program ESP32 to enable multiple functions, such as PWM, ADC, DAC, I2C, I2S, SPI, etc. For details, please see Section *Pin Descriptions*.
BOOT Button	Download button. Holding down **Boot** and then pressing **EN** initiates Firmware Download mode for downloading firmware through the serial port.
EN Button	Reset button.

Start Application Development

Before powering up your ESP32-PICO-DevKitM-2, please make sure that the board is in good condition with no obvious signs of damage.

Required Hardware

- 1 x ESP32-PICO-DevKitM-2
- 1 x USB 2.0 A to Micro B cable
- 1 x Computer running Windows, Linux, or macOS

Software Setup Please proceed to ESP-IDF Get Started, which will quickly help you set up the development environment then flash an application example onto your board.

4.1.3 Contents and Packaging

Retail Orders

If you order one or several samples of the board, each ESP32-PICO-DevKitM-2 development board comes in an individual package.

For retail orders, please go to https://www.espressif.com/en/contact-us/get-samples.

Wholesale Orders

If you order in bulk, the boards come in large cardboard boxes.

For wholesale orders, please go to https://www.espressif.com/en/contact-us/sales-questions.

4.1.4 Hardware Reference

Block Diagram

The block diagram below shows the main components of ESP32-PICO-DevKitM-2 and their interconnections.

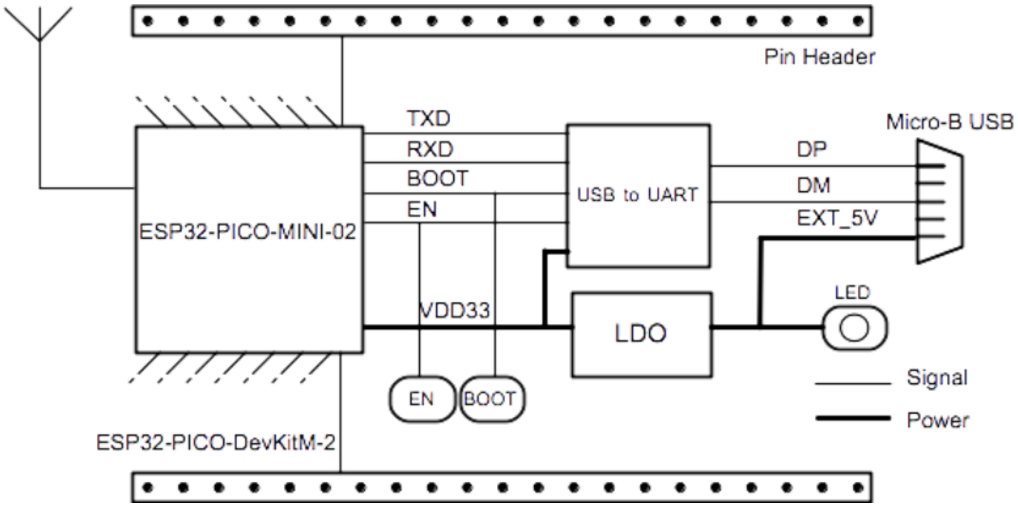


Fig. 3: ESP32-PICO-DevKitM-2 Block Diagram (click to enlarge)

Power Supply Options

There are three mutually exclusive ways to provide power to the board:

- Micro USB port, default power supply
- 5V/GND header pins
- 3V3/GND header pins

Warning: The power supply must be provided using **one and only one of the options above**, otherwise the board and/or the power supply source can be damaged.

Pin Descriptions

The two tables below provide the **Name** and **Function** of I/O header pins on both sides of the board, see *Description of Components*. The pin numbering and header names are the same as in the schematic given in *Related Documents*.

Header J2

No.	Name	Type	Function
1	IO20	I/O	GPIO20
2	IO21	I/O	GPIO21, VSPIHD, EMAC_TX_EN
3	IO22	I/O	GPIO22, VSPIWP, U0RTS, EMAC_TXD1
4	IO19	I/O	GPIO19, VSPIQ, U0CTS, EMAC_TXD0
5	IO8	I/O	GPIO8, SD_DATA1, HS1_DATA1, U2CTS
6	IO7	I/O	GPIO7, SD_DATA0, HS1_DATA0, U2RTS
7	IO5	I/O	GPIO5, VSPICS0, HS1_DATA6, EMAC_RX_CLK
8	NC	-	NC
9	NC	-	NC
10	RXD0	I/O	GPIO3, U0RXD *(See 1)*, CLK_OUT2
11	TXD0	I/O	GPIO1, U0TXD *(See 1)*, CLK_OUT3, EMAC_RXD2
12	IO35	I	ADC1_CH7, RTC_GPIO5
13	IO34	I	ADC1_CH6, RTC_GPIO4
14	IO38	I	GPIO38, ADC1_CH2, RTC_GPIO2
15	IO37	I	GPIO37, ADC1_CH1, RTC_GPIO1
16	EN	I	CHIP_PU
17	GND	P	Ground
18	VDD33 (3V3)	P	3.3 V power supply

Header J3

No.	Name	Type	Function
1	GND	P	Ground
2	SENSOR_VP (FSVP)	I	GPIO36, ADC1_CH0, RTC_GPIO0
3	SENSOR_VN (FSVN)	I	GPIO39, ADC1_CH3, RTC_GPIO3
4	IO25	I/O	GPIO25, DAC_1, ADC2_CH8, RTC_GPIO6, EMAC_RXD0
5	IO26	I/O	GPIO26, DAC_2, ADC2_CH9, RTC_GPIO7, EMAC_RXD1
6	IO32	I/O	32K_XP *(See 2a)*, ADC1_CH4, TOUCH9, RTC_GPIO9
7	IO33	I/O	32K_XN *(See 2b)*, ADC1_CH5, TOUCH8, RTC_GPIO8
8	IO27	I/O	GPIO27, ADC2_CH7, TOUCH7, RTC_GPIO17, EMAC_RX_DV
9	IO14	I/O	ADC2_CH6, TOUCH6, RTC_GPIO16, MTMS, HSPICLK, HS2_CLK, SD_CLK, EMAC_TXD2
10	IO12	I/O	ADC2_CH5, TOUCH5, RTC_GPIO15, MTDI *(See 3)*, HSPIQ, HS2_DATA2, SD_DATA2, EMAC_TXD3
11	IO13	I/O	ADC2_CH4, TOUCH4, RTC_GPIO14, MTCK, HSPID, HS2_DATA3, SD_DATA3, EMAC_RX_ER
12	IO15	I/O	ADC2_CH3, TOUCH3, RTC_GPIO13, MTDO, HSPICS0, HS2_CMD, SD_CMD, EMAC_RXD3
13	IO2	I/O	ADC2_CH2, TOUCH2, RTC_GPIO12, HSPIWP, HS2_DATA0, SD_DATA0
14	IO4	I/O	ADC2_CH0, TOUCH0, RTC_GPIO10, HSPIHD, HS2_DATA1, SD_DATA1, EMAC_TX_ER
15	IO0	I/O	ADC2_CH1, TOUCH1, RTC_GPIO11, CLK_OUT1, EMAC_TX_CLK
16	VDD33 (3V3)	P	3.3V power supply
17	GND	P	Ground
18	EXT_5V (5V)	P	5V power supply

Note:

1. This pin is connected to the pin of the USB bridge chip on the board.
2. 32.768 kHz crystal oscillator: a) input b) output
3. The operating voltage of ESP32-PICO-DevKitM-2' s embedded SPI flash is 3.3 V. Therefore, the strapping pin MTDI should be pulled down during the module power-on reset. If connected, please make sure that this pin is not held up on reset.

Pin Layout

4.1.5 Hardware Revision Details

No previous versions available.

4.1.6 Related Documents

- ESP32-PICO-MINI-02 & ESP32-PICO-MINI-1U Datasheet (PDF)
- ESP Product Selector
- ESP32-PICO-DevKitM-2 Schematic (PDF)

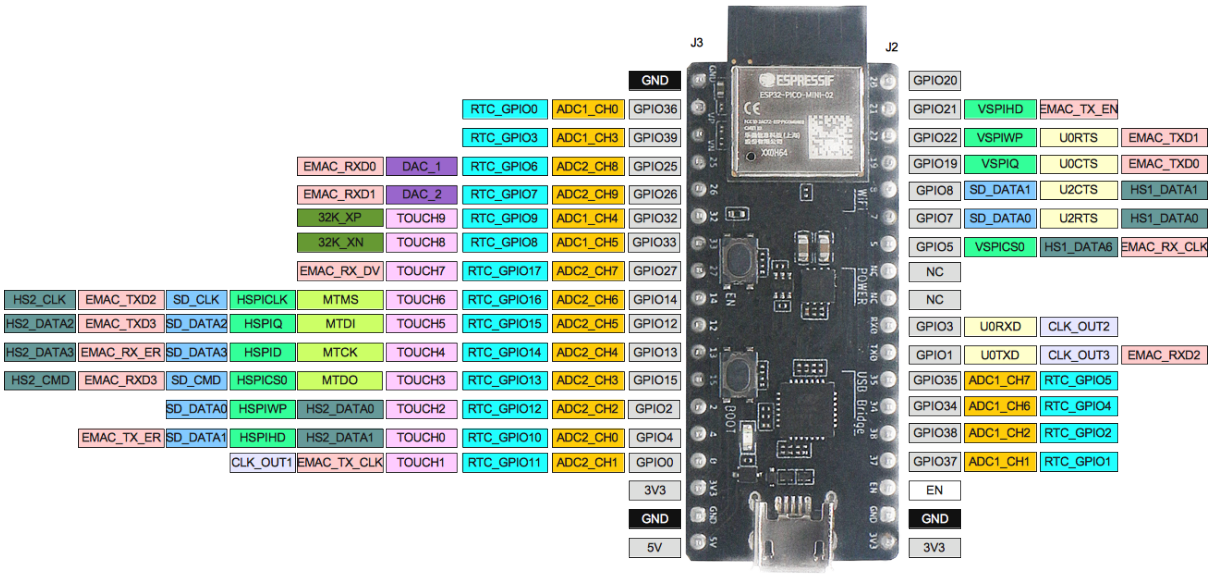


Fig. 4: ESP32-PICO-DevKitM-2 Pin Layout (click to enlarge)

- ESP32-PICO-DevKitM-2 PCB Layout (PDF)

For other design documentation for the board, please contact us at sales@espressif.com.

Chapter 5

ESP32-LCDKit

ESP32-LCDKit is an HMI (Human Machine Interface) development board with the ESP32-DevKitC at its core.

5.1 ESP32-LCDKit

5.1.1 Overview

ESP32-LCDKit is an HMI (Human Machine Interface) development board with the ESP32-DevKitC at its core. ESP32-LCDKit is integrated with such peripherals as SD-Card, DAC-Audio, and can be connected to an external display. The board is mainly used for HMI-related development and evaluation.Development board reserved screen interface type: SPI serial interface, 8-bit parallel interface, 16-bit parallel interface.

You may find HMI-related examples running with ESP32-LCDKit in HMI Example.

For more information on ESP32, please refer to ESP32 Series Datasheet.

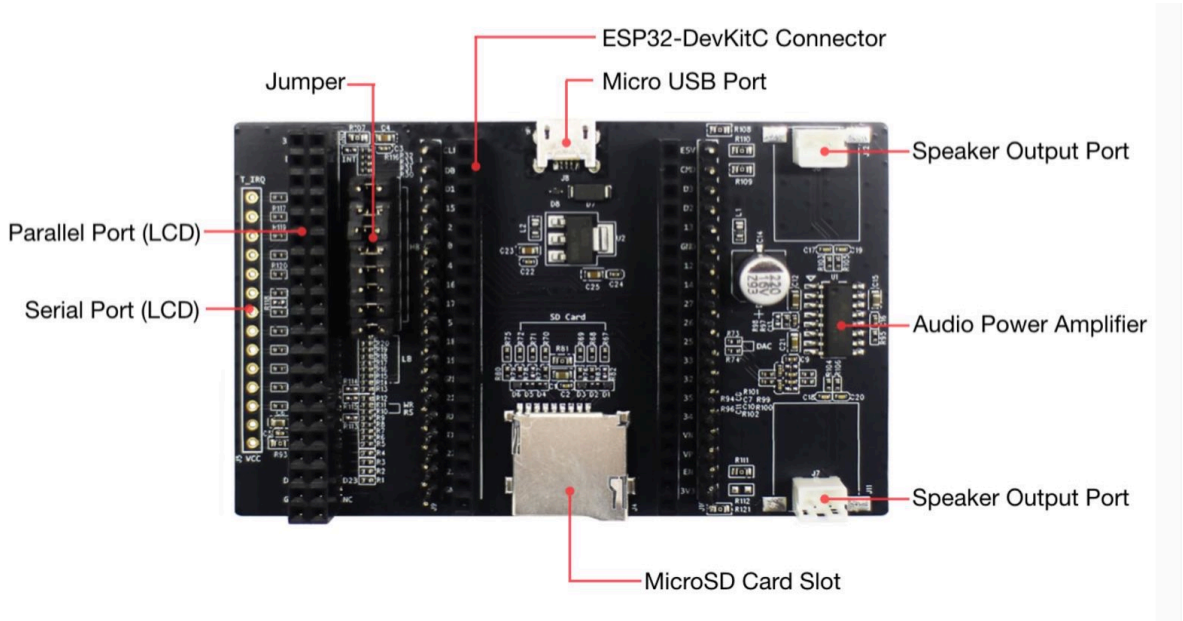


Fig. 1: ESP32-LCDKit

5.1.2 Block Diagram and PCB Layout

Block Diagram

The figure below shows the block diagram for ESP32-LCDKit.

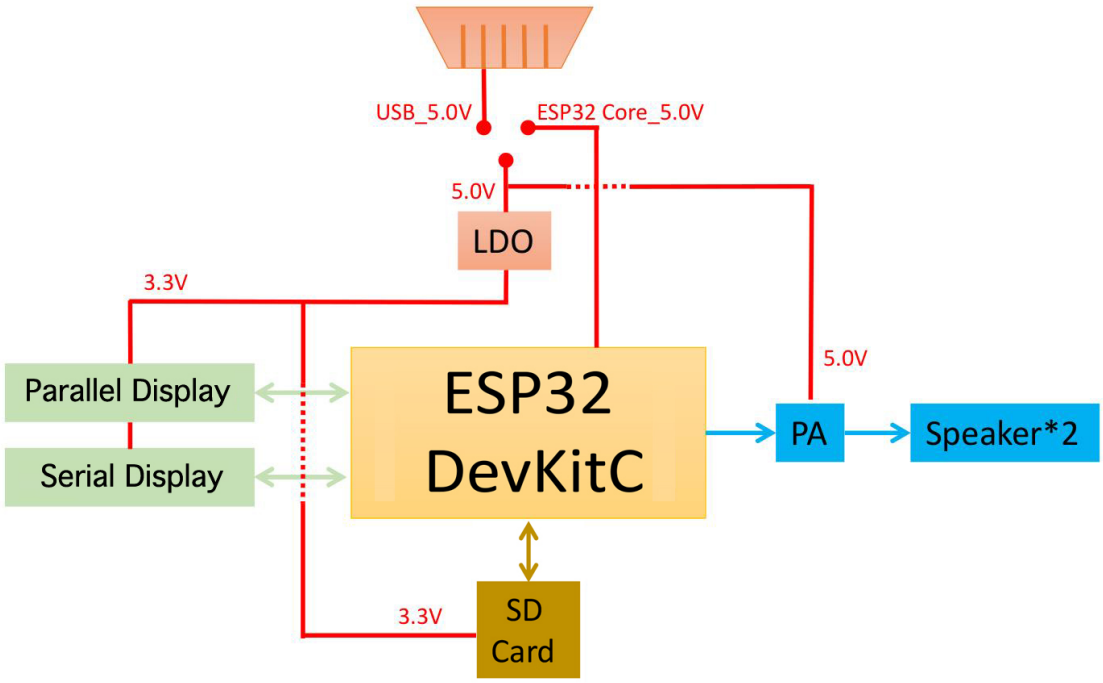


Fig. 2: ESP32-LCDKit Block Diagram

PCB Layout

The figure below shows the layout of ESP32-LCDKit PCB.

Descriptions of PCB components are shown in the following table:

Components	Description
Display connection module	Allows to connect serial or parallel LCD displays (8/16 bit)
ESP32 DevKitC connection module	Offers connection to an ESP32 DevKitC development board
SD-Card module	Provides an SD-Card slot for memory expansion
DAC-Audio module	Features an audio power amplifier and two output ports for external speakers

5.1.3 Functional Modules

This section introduces the functional modules (interfaces) of ESP32-LCDKit and their hardware schematics.

- ESP32-LCDKit Schematic
- ESP32-LCDKit PCB Layout

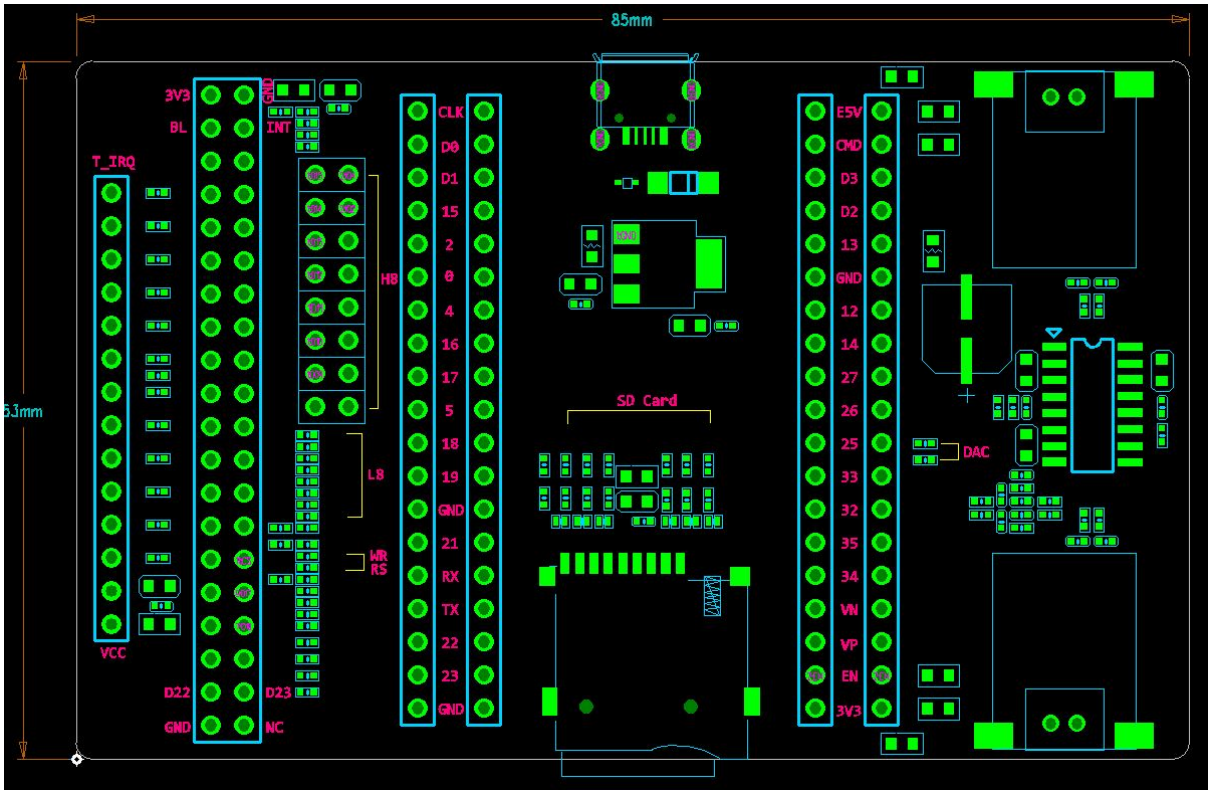


Fig. 3: ESP32-LCDKit PCB Layout

ESP32 DevKitC Connection Module

For the HMI-related development with ESP32-LCDKit, you also need the ESP32 DevKitC development board.

The figure below shows the schematics for the ESP32 DevKitC connection module.

Power Supply Management Module

The figure below shows the schematics for the USB power supply management module.

Display Connection Module

The display connection module supports the following interfaces:

- SPI serial interface
- 8-bit parallel interface
- 16-bit parallel interface

With this module, you can connect ESP32-LCDKit to an external display and interact with the pre-programmed GUI if the display has a touchscreen.

The figure below shows the schematics for this module.

SD-Card and DAC-Audio Modules

The SD-Card module provides an SD Card slot for memory expansion. The DAC-Audio module features the MIX3006 power amplifier and two output ports for connection of external speakers.

The figure below shows the schematics for the SD-Card and DAC-Audio modules.

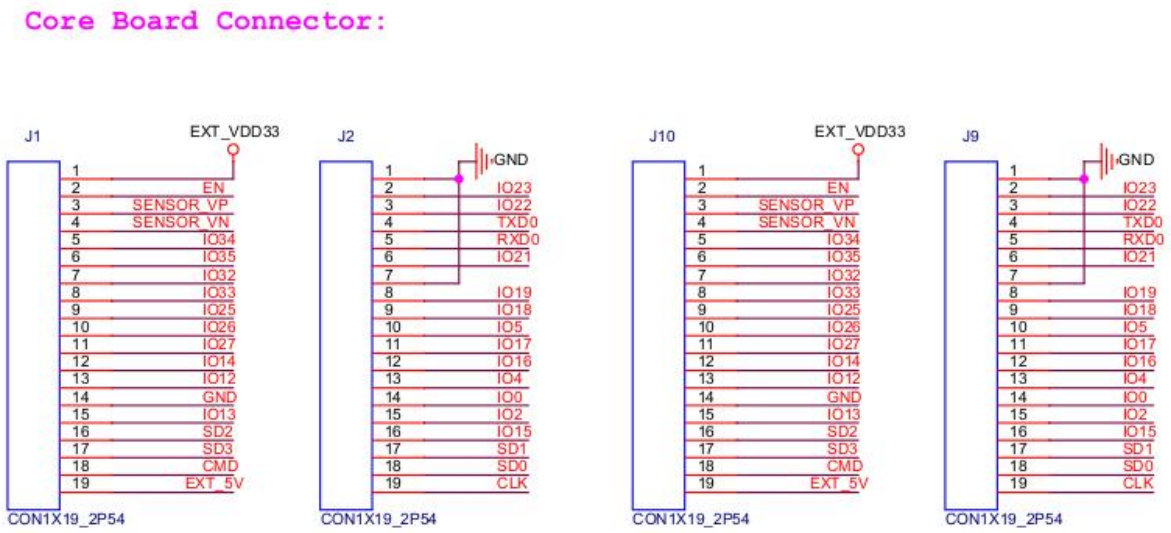


Fig. 4: ESP32 DevKitC Connection Module

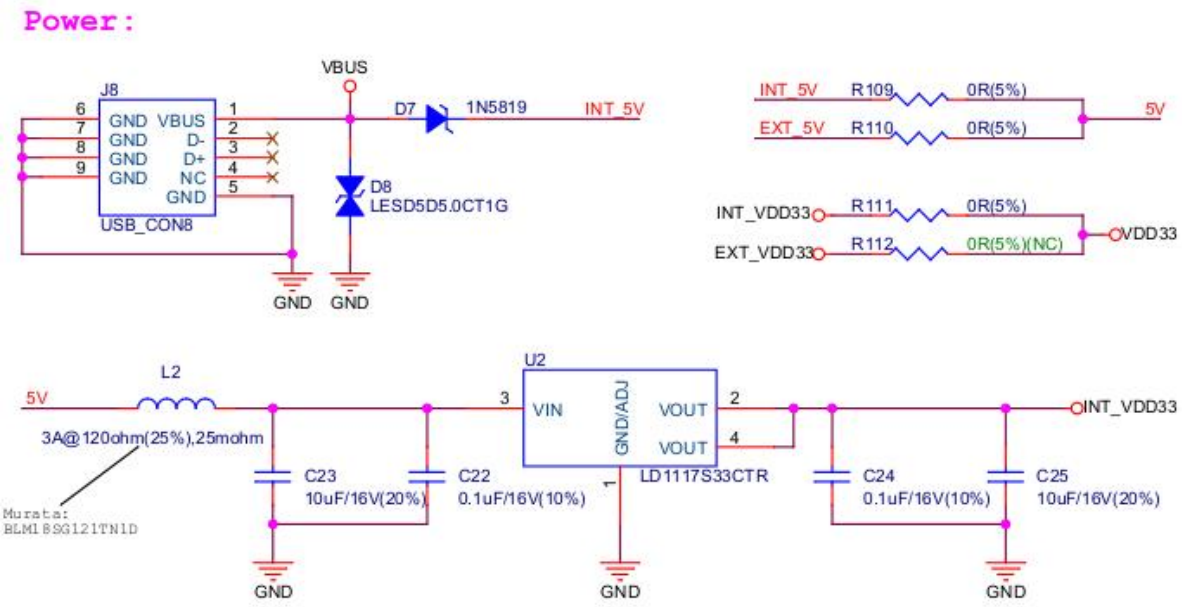


Fig. 5: ESP32-LCDKit Power Supply Module

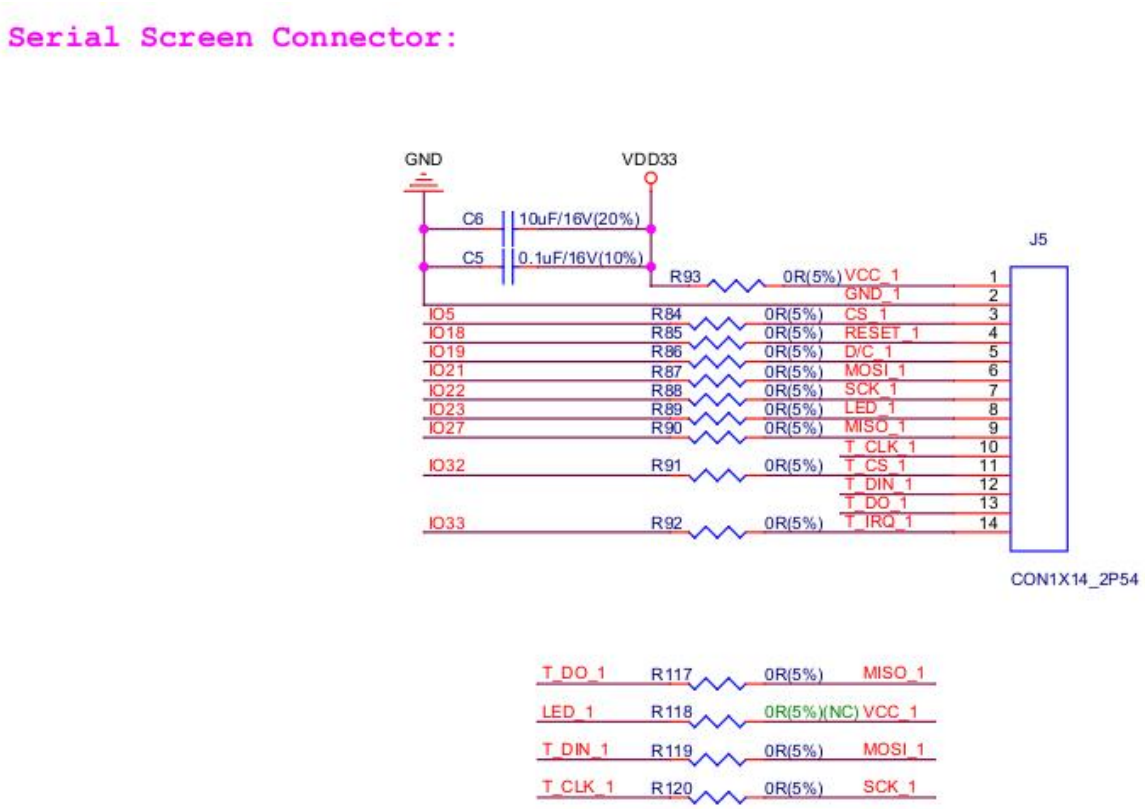


Fig. 6: ESP32-LCDKit Display Connection Module

5.1.4 Related Documents

Please download the following documents from the HTML version of esp-dev-kits Documentation.

- ESP32-LCDKit Schematic
- ESP32-LCDKit PCB Layout

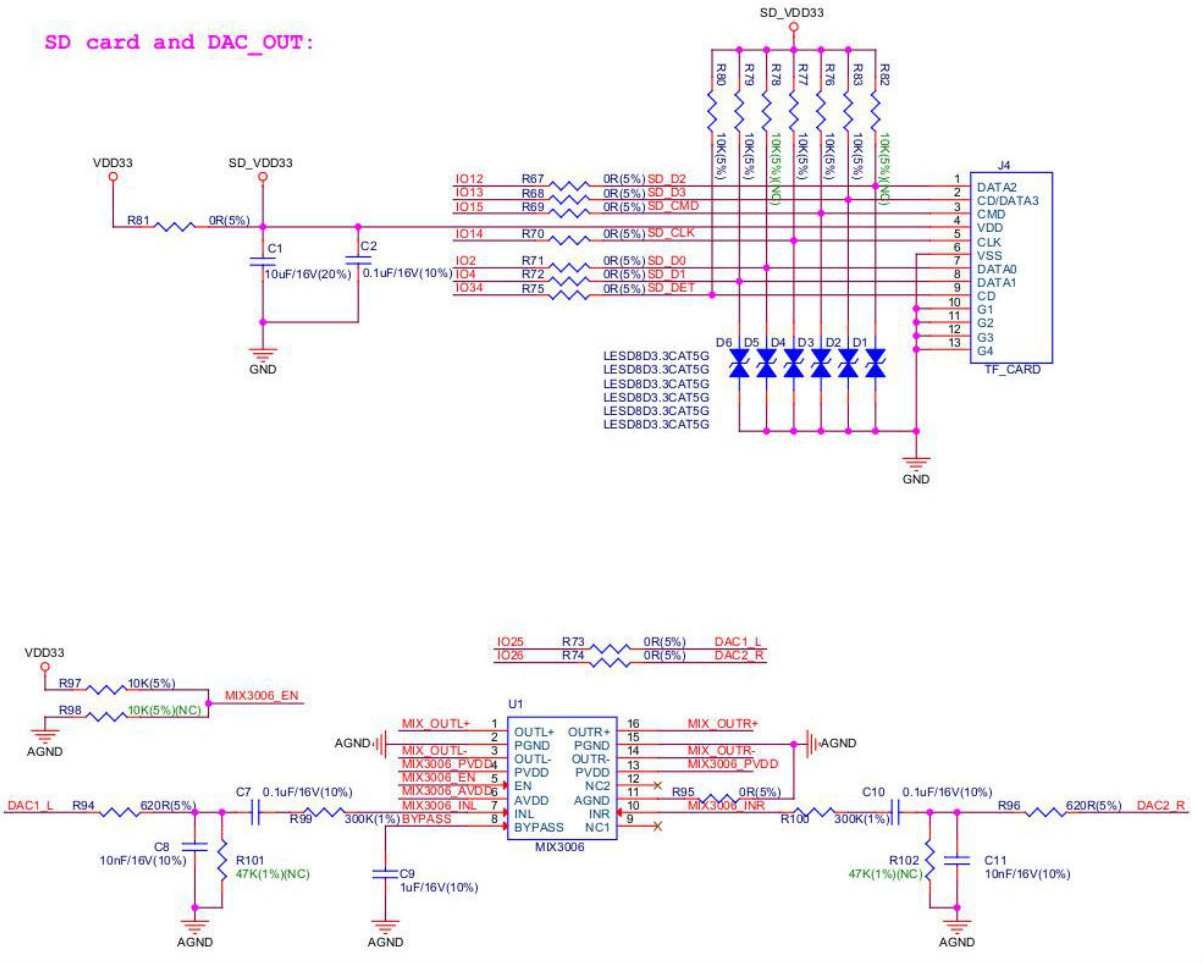


Fig. 7: SD-Card and DAC-Audio Modules

Chapter 6

ESP32-Ethernet-Kit

The *ESP32-Ethernet-Kit* is an Ethernet-to-Wi-Fi development board that enables Ethernet devices to be interconnected over Wi-Fi.

6.1 ESP32-Ethernet-Kit v1.2

This guide shows how to get started with the ESP32-Ethernet-Kit development board and also provides information about its functionality and configuration options.

The *ESP32-Ethernet-Kit* is an Ethernet-to-Wi-Fi development board that enables Ethernet devices to be interconnected over Wi-Fi. At the same time, to provide more flexible power supply options, the ESP32-Ethernet-Kit also supports power over Ethernet (PoE).

6.1.1 What You Need

- *ESP32-Ethernet-Kit v1.2 board*
- USB 2.0 cable (Standard-A to Micro-B)
- Computer running Windows, Linux, or macOS

You can skip the introduction sections and go directly to Section *Start Application Development*.

6.1.2 Overview

ESP32-Ethernet-Kit is an ESP32-based development board produced by Espressif.

It consists of two development boards, the Ethernet board A and the PoE board B. The *Ethernet board (A)* contains Bluetooth®/Wi-Fi dual-mode ESP32-WROVER-E module and IP101GRI, a Single Port 10/100 Fast Ethernet Transceiver (PHY). The *PoE board (B)* provides power over Ethernet functionality. The A board can work independently, without the board B installed.

For the application loading and monitoring, the Ethernet board (A) also features FTDI FT2232H chip - an advanced multi-interface USB bridge. This chip enables to use JTAG for direct debugging of ESP32 through the USB interface without a separate JTAG debugger.

6.1.3 Functionality Overview

The block diagram below shows the main components of ESP32-Ethernet-Kit and their interconnections.



Fig. 1: ESP32-Ethernet-Kit v1.2 Overview (click to enlarge)

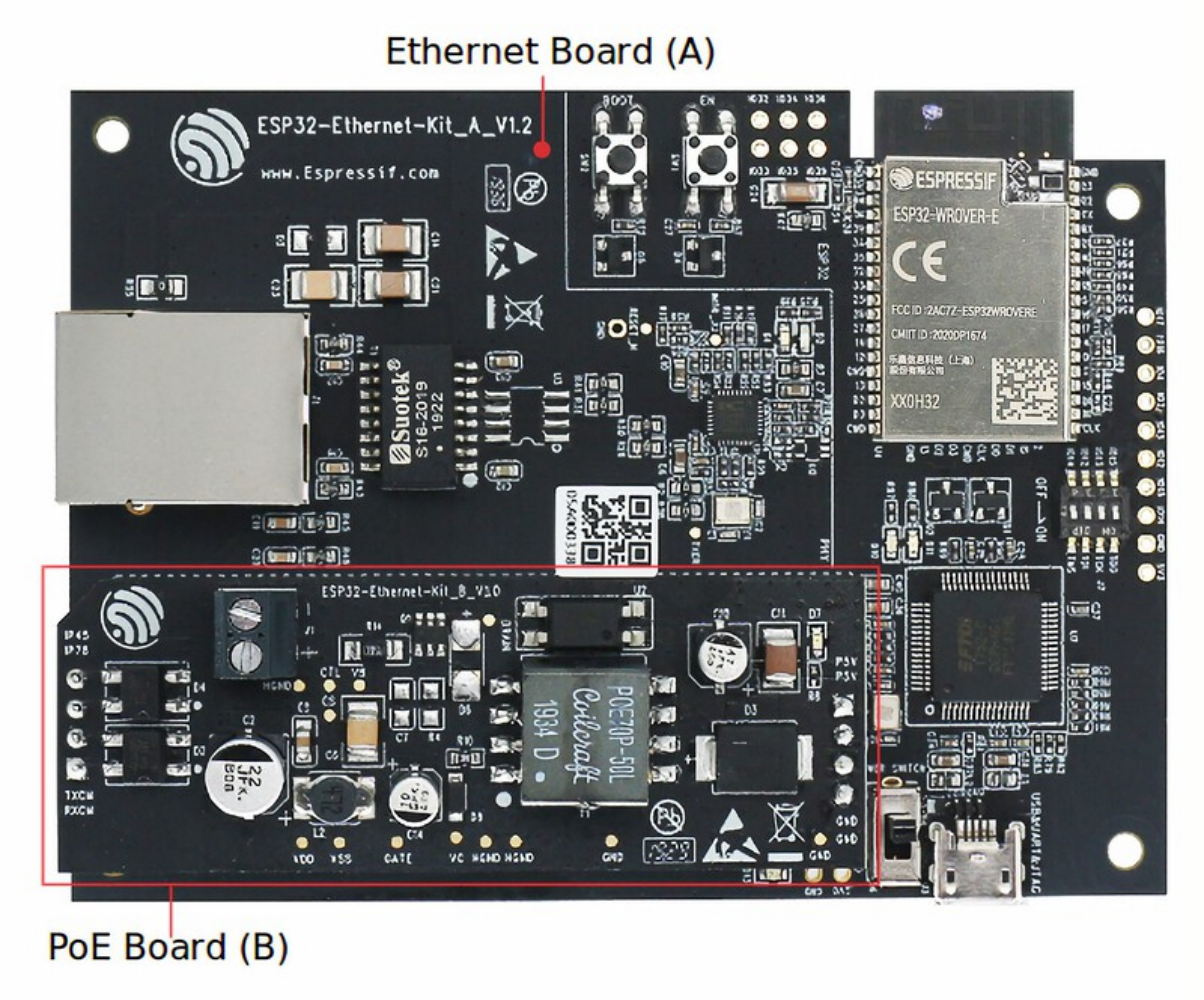


Fig. 2: ESP32-Ethernet-Kit v1.2 (click to enlarge)

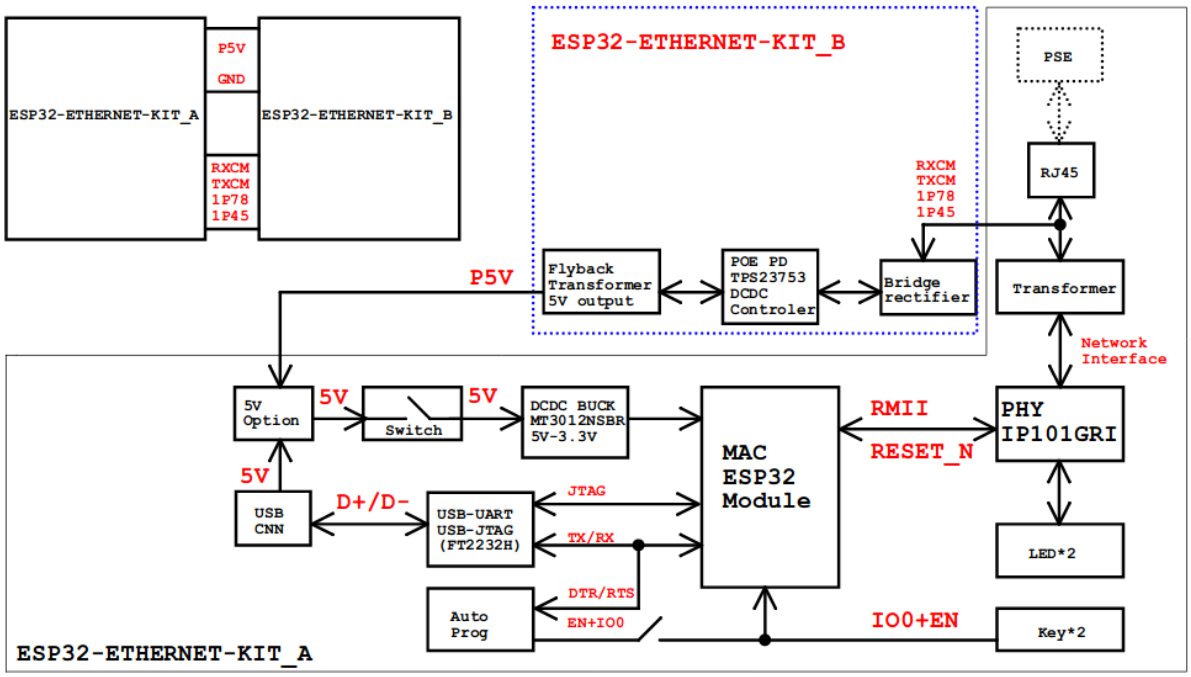


Fig. 3: ESP32-Ethernet-Kit block diagram (click to enlarge)

6.1.4 Functional Description

The following figures and tables describe the key components, interfaces, and controls of the ESP32-Ethernet-Kit.

Ethernet Board (A)

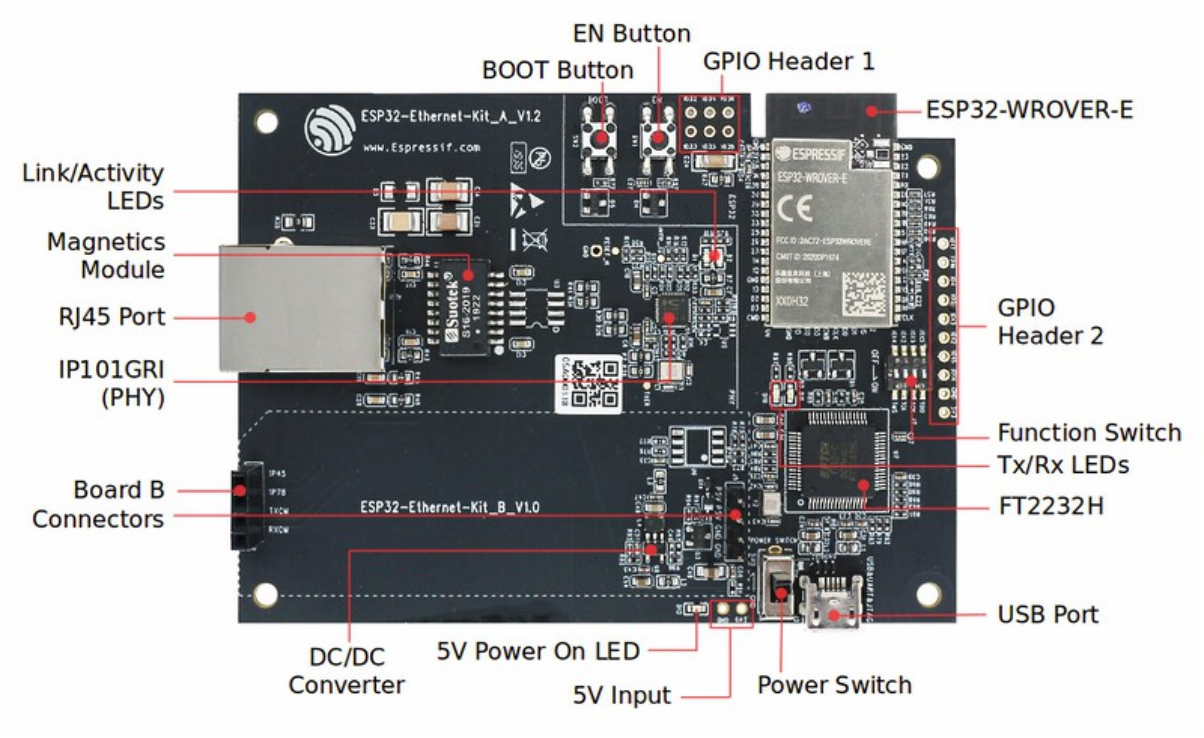


Fig. 4: ESP32-Ethernet-Kit - Ethernet board (A) layout (click to enlarge)

The table below provides description starting from the picture’ s top right corner and going clockwise.

Table 1: Table 1 Component Description

Key Component	Description
ESP32-WROVER-E	This ESP32 module features 64-Mbit PSRAM for flexible extended storage and data processing capabilities.
GPIO Header 2	Five unpopulated through-hole solder pads to provide access to selected GPIOs of ESP32. For details, see *GPIO Header 2*.
Function Switch	A 4-bit DIP switch used to configure the functionality of selected GPIOs of ESP32. For details see *Function Switch*.
Tx/Rx LEDs	Two LEDs to show the status of UART transmission.
FT2232H	The FT2232H chip serves as a multi-protocol USB-to-serial bridge which can be programmed and controlled via USB to provide communication with ESP32. FT2232H also features USB-to-JTAG interface which is available on channel A of the chip, while USB-to-serial is on channel B. The FT2232H chip enhances user-friendliness in terms of application development and debugging. See ESP32-Ethernet-Kit v1.2 Ethernet board (A) schematic.
USB Port	USB interface. Power supply for the board as well as the communication interface between a computer and the board.
Power Switch	Power On/Off Switch. Toggling the switch to **5V0** position powers the board on, toggling to **GND** position powers the board off.
5V Input	The 5 V power supply interface can be more convenient when the board is operating autonomously (not connected to a computer).
5V Power On LED	This red LED turns on when power is supplied to the board, either from USB or 5 V Input.
DC/DC Converter	Provided DC 5 V to 3.3 V conversion, output current up to 2 A.
Board B Connectors	A pair male and female header pins for mounting the *PoE board (B)*
IP101GRI (PHY)	The physical layer (PHY) connection to the Ethernet cable is implemented using the IP101GRI chip. The connection between PHY and ESP32 is done through the reduced media-independent interface (RMII), a variant of the media-independent interface (MII) standard. The PHY supports the IEEE 802.3/802.3u standard of 10/100 Mbps.
RJ45 Port	Ethernet network data transmission port.
Magnetics Module	The Magnetics are part of the Ethernet specification to protect against faults and transients, including rejection of common mode signals between the transceiver IC and the cable. The magnetics also provide galvanic isolation between the transceiver and the Ethernet device.
Link/Activity LEDs	Two LEDs (green and red) that respectively indicate the "Link" and "Activity" statuses of the PHY.
BOOT Button	Download button. Holding down **BOOT** and then pressing **EN** initiates Firmware Download mode for downloading firmware through the serial port.
EN Button	Reset button.
GPIO Header 1	This header provides six unpopulated through-hole solder pads connected to spare GPIOs of ESP32. For details, see *GPIO Header 1*.

Note: Automatic firmware download is supported. If following steps and using software described in Section *Start Application Development*, users do not need to do any operation with BOOT button or EN button.

PoE Board (B)

This board coverts power delivered over the Ethernet cable (PoE) to provide a power supply for the Ethernet board (A). The main components of the PoE board (B) are shown on the block diagram under *Functionality Overview*.

The PoE board (B) has the following features:

- Support for IEEE 802.3at
- Power output: 5 V, 1.4 A

To take advantage of the PoE functionality the **RJ45 Port** of the Ethernet board (A) should be connected with an Ethernet cable to a switch that supports PoE. When the Ethernet board (A) detects 5 V power output from the PoE board (B), the USB power will be automatically cut off.

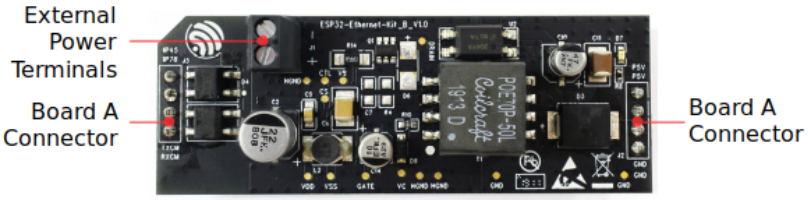


Fig. 5: ESP32-Ethernet-Kit - PoE board (B) layout (click to enlarge)

Table 2: Table PoE board (B)

Key Component	Description
Board A Connector	Four female (left) and four male (right) header pins for connecting the PoE board (B) to *Ethernet board (A)*. The pins on the left accept power coming from a PoE switch. The pins on the right deliver 5 V power supply to the Ethernet board (A).
External Power Terminals	Optional power supply (26.6 ~ 54 V) to the PoE board (B).

6.1.5 Setup Options

This section describes options to configure the ESP32-Ethernet-Kit hardware.

Function Switch

When in On position, this DIP switch is routing listed GPIOs to FT2232H to provide JTAG functionality. When in Off position, the GPIOs may be used for other purposes.

DIP SW	GPIO Pin
1	GPIO13
2	GPIO12
3	GPIO15
4	GPIO14

RMII Clock Selection

The ethernet MAC and PHY under RMII working mode need a common 50 MHz reference clock (i.e., RMII clock) that can be provided either externally, or generated from internal ESP32 APLL (not recommended).

Note: For additional information on the RMII clock selection, please refer to ESP32-Ethernet-Kit v1.2 Ethernet board (A) schematic, sheet 2, location D2.

RMII Clock Sourced Externally by PHY By default, the ESP32-Ethernet-Kit is configured to provide RMII clock for the IP101GRI PHY's 50M_CLKO output. The clock signal is generated by the frequency multiplication of 25 MHz crystal connected to the PHY. For details, please see the figure below.

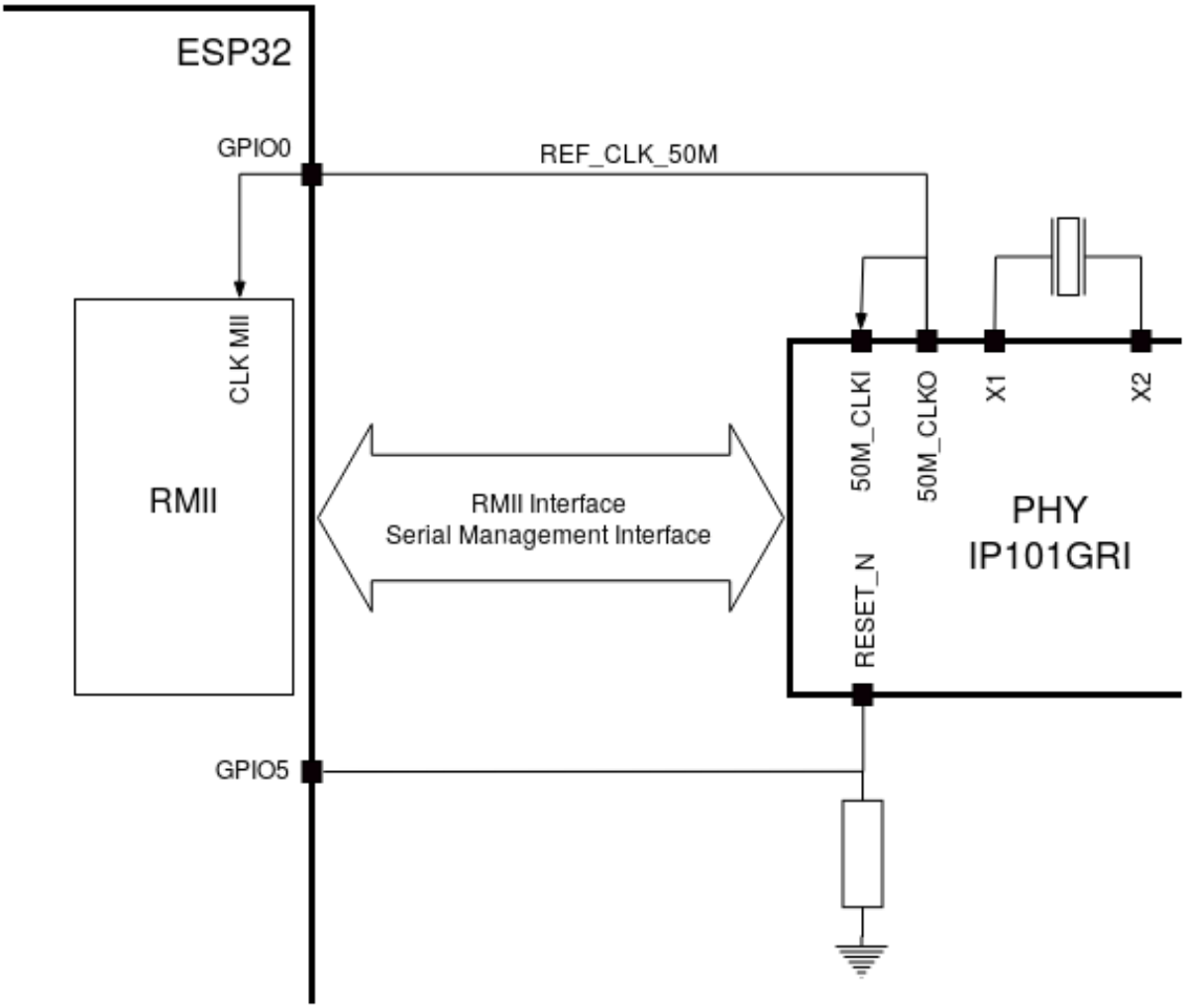


Fig. 6: RMII Clock from IP101GRI PHY (click to enlarge)

Please note that the PHY is reset on power-up by pulling the RESET_N signal down with a resistor. ESP32 should assert RESET_N high with GPIO5 to enable PHY. Only this can ensure the power-up of the system. Otherwise, ESP32 may enter download mode.

RMII Clock Sourced Internally from ESP32 ' s APLL Another option is to source the RMII Clock from internal ESP32 APLL, see figure below. The clock signal coming from GPIO0 is first inverted, to account for transmission line delay, and then supplied to the PHY.

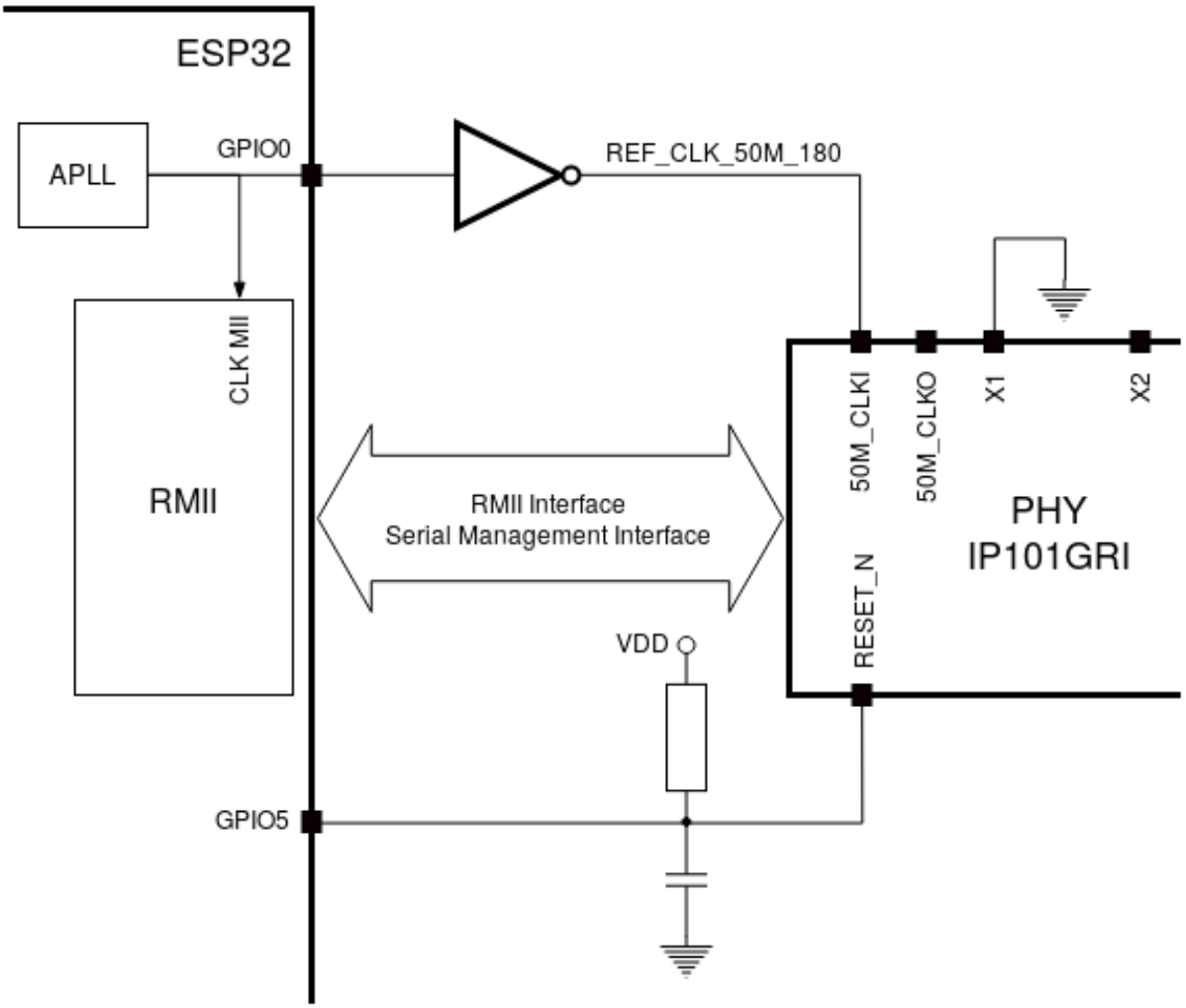


Fig. 7: RMII Clock from ESP Internal APLL (click to enlarge)

To implement this option, users need to remove or add some RC components on the board. For details please refer to ESP32-Ethernet-Kit v1.2 Ethernet board (A) schematic, sheet 2, location U2.

Note: Please note that you need to have *RMII Clock Sourced Externally by PHY* or by an external clock source in the following cases:

- If Wi-Fi and Ethernet are used simultaneously, the RMII clock cannot be generated by the internal APLL clock, as it would result in clock instability.
- APLL is already used for other purposes (e.g., I2S peripheral).

6.1.6 GPIO Allocation

This section describes the allocation of ESP32 GPIOs to specific interfaces or functions of the ESP32-Ethernet-Kit.

IP101GRI (PHY) Interface

The allocation of the ESP32 (MAC) pins to IP101GRI (PHY) is shown in the table below. Implementation of ESP32-Ethernet-Kit defaults to Reduced Media-Independent Interface (RMII).

No.	ESP32 Pin (MAC)	IP101GRI (PHY)
RMII Interface		
1	GPIO21	TX_EN
2	GPIO19	TXD[0]
3	GPIO22	TXD[1]
4	GPIO25	RXD[0]
5	GPIO26	RXD[1]
6	GPIO27	CRS_DV
7	GPIO0	REF_CLK
Serial Management Interface		
8	GPIO23	MDC
9	GPIO18	MDIO
PHY Reset		
10	GPIO5	Reset_N

Note:

1. The allocation of all pins under the ESP32's *RMII Interface* is fixed and cannot be changed either through IO MUX or GPIO Matrix.
2. For REF_CLK, GPIO0 supports both input and output modes, while GPIO16 and GPIO17 support only output mode. However, GPIO16 and GPIO17 are not broken out to the ESP32-WROVER-E module and therefore not available for use. If you need to use these pins, please replace the module with one that does not include PSRAM memory and exposes GPIO16 and GPIO17.

GPIO Header 1

This header exposes some GPIOs that are not used elsewhere on the ESP32-Ethernet-Kit.

No.	ESP32 Pin
1	GPIO32
2	GPIO33
3	GPIO34
4	GPIO35
5	GPIO36
6	GPIO39

GPIO Header 2

This header contains GPIOs that may be used for other purposes depending on scenarios described in column **Notes**.

No.	ESP32 Pin	Notes
1	GPIO17	See note 1
2	GPIO16	See note 1
3	GPIO4	
4	GPIO2	
5	GPIO13	See note 2
6	GPIO12	See note 2
7	GPIO15	See note 2
8	GPIO14	See note 2
9	GND	Ground
10	3V3	3.3 V power supply

Note:

1. The ESP32 pins GPIO16 and GPIO17 are not broken out to the ESP32-WROVER-E module and therefore not available for use. If you need to use these pins, please replace the module with one that does not include PSRAM memory and exposes GPIO16 and GPIO17.
2. Functionality depends on the settings of the *Function Switch*.

GPIO Allocation Summary

ESP32-WROVER-E	IP101GRI	UART	JTAG	GPIO	Notes
S_VP				IO36	
S_VN				IO39	
IO34				IO34	
IO35				IO35	
IO32				IO32	
IO33				IO33	
IO25	RXD[0]				
IO26	RXD[1]				
IO27	CRS_DV				
IO14			TMS	IO14	
IO12			TDI	IO12	
IO13			TCK	IO13	
IO15			TDO	IO15	
IO2				IO2	
IO0	REF_CLK				See note 1
IO4				IO4	
IO16				IO16 (NC)	See note 2
IO17				IO17 (NC)	See note 2
IO5	Reset_N				See note 1
IO18	MDIO				
IO19	TXD[0]				
IO21	TX_EN				
RXD0		RXD			
TXD0		TXD			
IO22	TXD[1]				
IO23	MDC				

Note:

1. To prevent the power-on state of the GPIO0 from being affected by the clock output on the PHY side, the RESET_N signal to PHY defaults to low, turning the clock output off. After power-on you can control RESET_N with GPIO5 to turn the clock output on. See also *RMII Clock Sourced Externally by PHY*. For PHYs that cannot turn off the clock output through RESET_N, it is recommended to use a crystal module that can be disabled/enabled externally. Similarly like when using RESET_N, the oscillator module should be disabled by default and turned on by ESP32 after power-up. For a reference design please see ESP32-Ethernet-Kit v1.2 Ethernet board (A) schematic.
2. The ESP32 pins GPIO16 and GPIO17 are not broken out to the ESP32-WROVER-E module and therefore not available for use. If you need to use these pins, please replace the module with one that does not include PSRAM memory and exposes GPIO16 and GPIO17.

6.1.7 Start Application Development

Before powering up your ESP32-Ethernet-Kit, please make sure that the board is in good condition with no obvious signs of damage.

Initial Setup

1. Set the **Function Switch** on the *Ethernet board (A)* to its default position by turning all the switches to **ON**.
2. To simplify flashing and testing of the application, do not input extra signals to the board headers.
3. The *PoE board (B)* can now be plugged in, but do not connect external power to it.
4. Connect the *Ethernet board (A)* to the PC with a USB cable.
5. Turn the **Power Switch** from GND to 5V0 position, the **5V Power On LED** should light up.

Now to Development

Proceed to Get Started, where Section Installation will quickly help you set up the development environment and then flash an example project onto your board.

Move on to the next section only if you have successfully completed all the above steps.

Configure and Load the Ethernet Example

After setting up the development environment and testing the board, you can configure and flash the ethernet/basic example. This example has been created for testing Ethernet functionality. It supports different PHY, including **IP101GRI** installed on *ESP32-Ethernet-Kit v1.2 board*.

6.1.8 Summary of Changes from ESP32-Ethernet-Kit v1.1

- Correct the placement of GPIO pin number marking on the board' s silkscreen besides the DIP switch.
- Values of C1, C2, C42, and C43 are updated to 20 pF. For more information, please check ESP32-Ethernet-Kit v1.2 Ethernet board (A) schematic.
- Replace ESP32-WROVER-B with ESP32-WROVER-E.

6.1.9 Other Versions of ESP32-Ethernet-Kit

- *ESP32-Ethernet-Kit v1.0*
- *ESP32-Ethernet-Kit v1.1*

6.1.10 Related Documents

- ESP32-Ethernet-Kit v1.2 Ethernet Board (A) Schematic (PDF)
- ESP32-Ethernet-Kit PoE Board (B) Schematic (PDF)
- ESP32-Ethernet-Kit v1.2 Ethernet Board (A) PCB Layout (PDF)
- ESP32-Ethernet-Kit PoE Board (B) PCB Layout (PDF)
- ESP32 Datasheet (PDF)
- ESP32-WROVER-E Datasheet (PDF)
- JTAG Debugging

For other design documentation for the board, please contact us at sales@espressif.com.

ESP32-Ethernet-Kit v1.0

This guide shows how to get started with the ESP32-Ethernet-Kit development board and also provides information about its functionality and configuration options.

The *ESP32-Ethernet-Kit* is an Ethernet-to-Wi-Fi development board that enables Ethernet devices to be interconnected over Wi-Fi. At the same time, to provide more flexible power supply options, the ESP32-Ethernet-Kit also supports power over Ethernet (PoE).

What You Need

- *ESP32-Ethernet-Kit v1.0 board*
- USB 2.0 cable (Standard-A to Micro-B)
- Computer running Windows, Linux, or macOS

You can skip the introduction sections and go directly to Section *Start Application Development*.

Overview ESP32-Ethernet-Kit is an ESP32-based development board produced by Espressif.

It consists of two development boards, the Ethernet board A and the PoE board B. The *Ethernet board (A)* contains Bluetooth®/Wi-Fi dual-mode ESP32-WROVER-B module and IP101GRI, a Single Port 10/100 Fast Ethernet Transceiver (PHY). The *PoE board (B)* provides power over Ethernet functionality. The A board can work independently, without the board B installed.

For the application loading and monitoring the Ethernet board (A) also features FTDI FT2232H chip - an advanced multi-interface USB bridge. This chip enables to use JTAG for direct debugging of ESP32 through the USB interface without a separate JTAG debugger.

Functionality Overview The block diagram below shows the main components of ESP32-Ethernet-Kit and their interconnections.

Functional Description The following two figures and tables describe the key components, interfaces, and controls of the ESP32-Ethernet-Kit.

Ethernet Board (A) The table below provides description starting from the picture's top right corner and going clockwise.

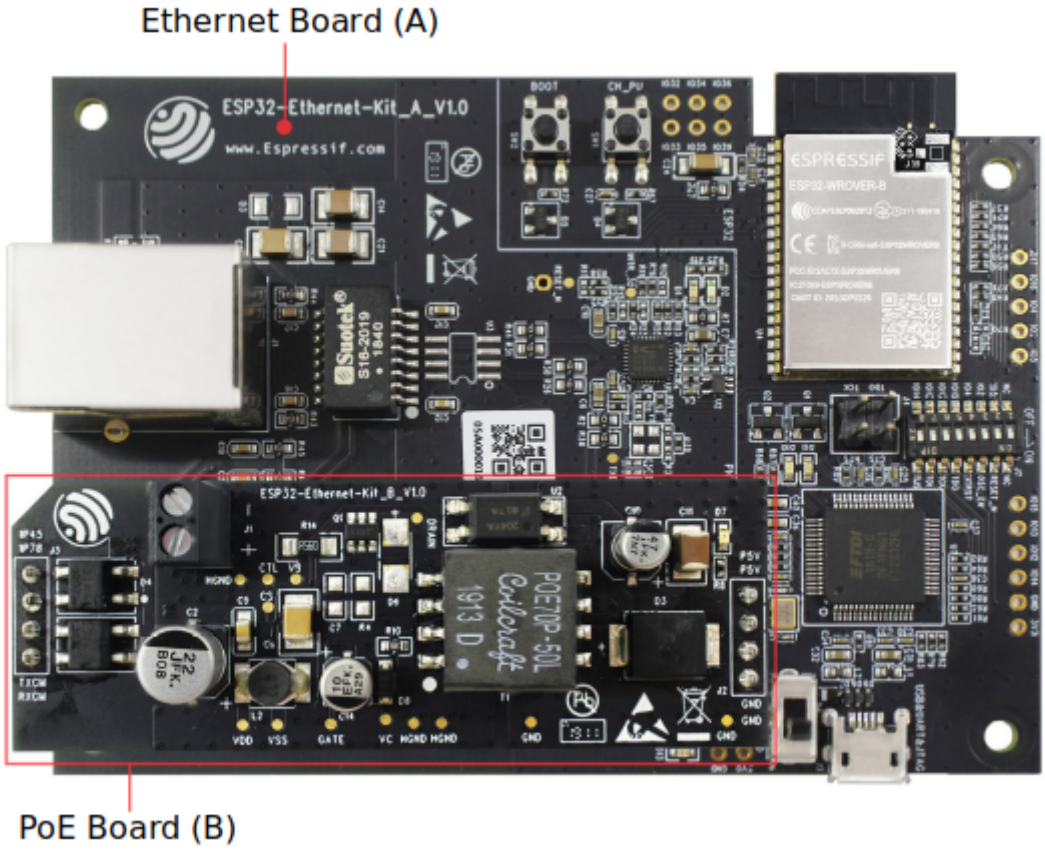


Fig. 8: ESP32-Ethernet-Kit v1.0

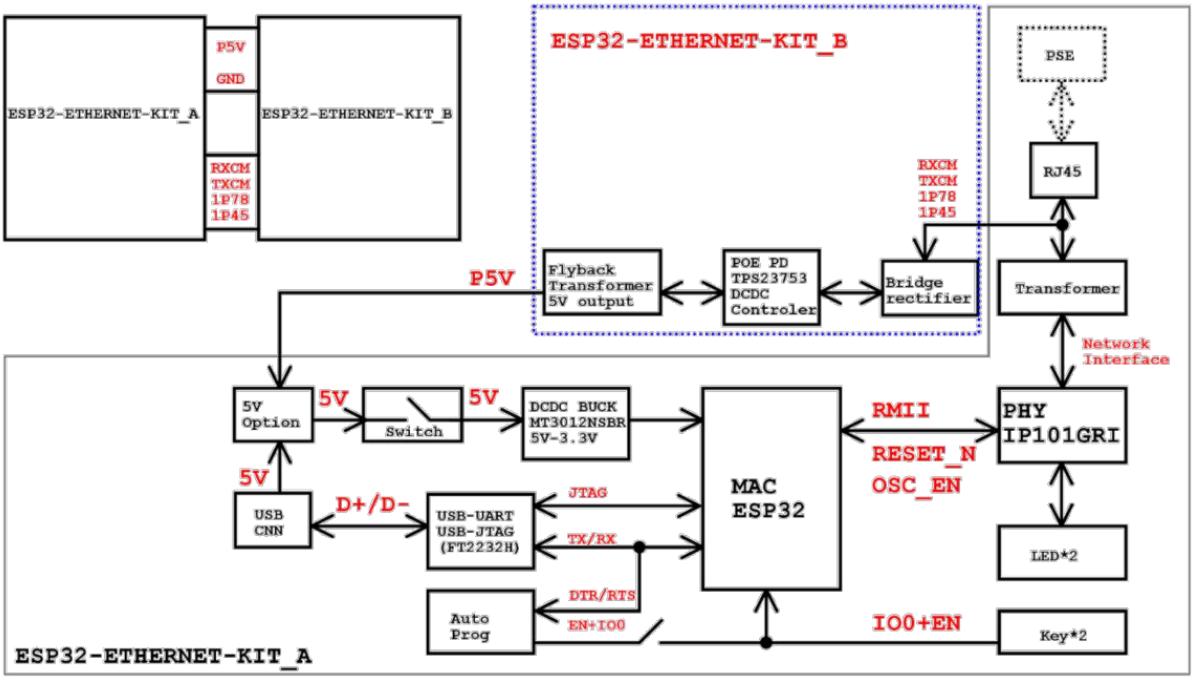


Fig. 9: ESP32-Ethernet-Kit block diagram (click to enlarge)

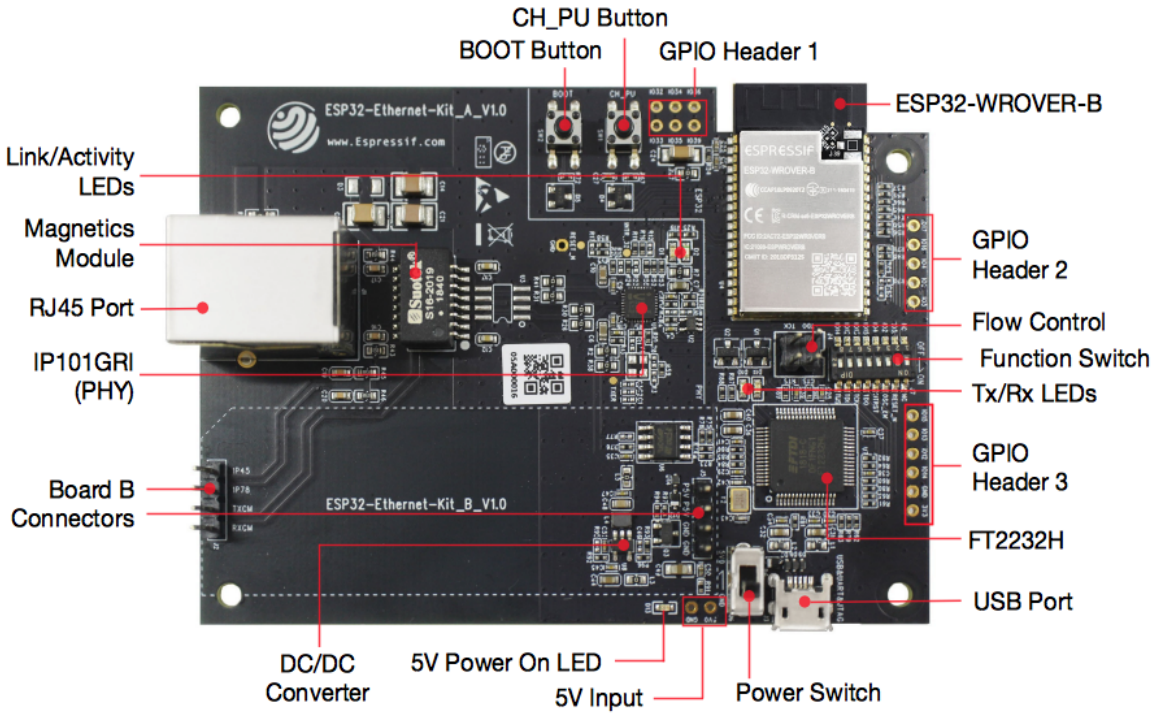


Fig. 10: ESP32-Ethernet-Kit - Ethernet board (A) layout (click to enlarge)

Key Component	Description
ESP32-WROVER-B	This ESP32 module features 64-Mbit PSRAM for flexible extended storage and data processing capabilities.
GPIO Header 2	Five unpopulated through-hole solder pads to provide access to selected GPIOs of ESP32. For details, see *GPIO Header 2*.
Flow Control	A jumper header with access to the board signals. For details, see *Flow Control*.
Function Switch	A DIP switch used to configure the functionality of selected GPIOs of ESP32. For details, see *Function Switch*.
Tx/Rx LEDs	Two LEDs to show the status of UART transmission.
GPIO Header 3	Provides access to some GPIOs of ESP32 that can be used depending on the position of the *Function Switch*.
FT2232H	The FT2232H chip serves as a multi-protocol USB-to-serial bridge which can be programmed and controlled via USB to provide communication with ESP32. FT2232H also features USB-to-JTAG interface which is available on channel A of the chip, while USB-to-serial is on channel B. The FT2232H chip enhances user-friendliness in terms of application development and debugging. See ESP32-Ethernet-Kit v1.0 Ethernet board (A) schematic.
USB Port	USB interface. Power supply for the board as well as the communication interface between a computer and the board.
Power Switch	Power On/Off Switch. Toggling toward the **Boot** button powers the board on, toggling away from **Boot** powers the board off.
5V Input	The 5V power supply interface can be more convenient when the board is operating autonomously (not connected to a computer).
5V Power On LED	This red LED turns on when power is supplied to the board, either from USB or 5 V Input.

PoE Board (B) This board coverts power delivered over the Ethernet cable (PoE) to provide a power supply for the Ethernet board (A). The main components of the PoE board (B) are shown on the block diagram under *Functionality Overview*.

The PoE board (B) has the following features:

- Support for IEEE 802.3at
- Power output: 5 V, 1.4 A

To take advantage of the PoE functionality the **RJ45 Port** of the Ethernet board (A) should be connected with an Ethernet cable to a switch that supports PoE. When the Ethernet board (A) detects 5 V power output from the PoE board (B), the USB power will be automatically cut off.

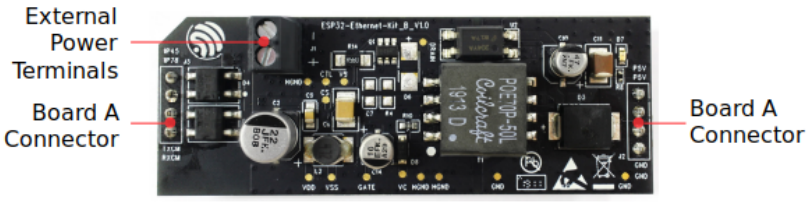


Fig. 11: ESP32-Ethernet-Kit - PoE board (B) layout (click to enlarge)

Key Component	Description
Board A Connector	Four female header pins for mounting this board onto *Ethernet board (A)*.
External Power Terminals	Optional power supply to the PoE board (B).

Setup Options This section describes options to configure the ESP32-Ethernet-Kit hardware.

Function Switch The functions for specific GPIO pins can be selected with the **Function Switch**.

DIP SW	GPIO Pin	Pin Functionality if DIP SW is ON
1	GPIO14	Connected to FT2232H to provide JTAG functionality
2	GPIO12	Connected to FT2232H to provide JTAG functionality
3	GPIO13	Connected to FT2232H to provide JTAG functionality
4	GPIO15	Connected to FT2232H to provide JTAG functionality
5	GPIO4	Connected to FT2232H to provide JTAG functionality
6	GPIO2	Connected to on-board 25 MHz oscillator
7	GPIO5	Connected to RESET_N input of IP101GRI
8	n/a	

You can make a certain GPIO pin available for other purposes by putting its DIP SW to the Off position.

Flow Control This is a 2 x 2 jumper pin header intended for the UART flow control.

No.	Signal	Notes
1	MTDO	GPIO13, see also *Function Switch*
2	MTCK	GPIO15, see also *Function Switch*
3	RTS	RTS signal of FT2232H
4	CTS	CTS signal of FT2232H

GPIO Allocation This section describes allocation of ESP32 GPIOs to specific interfaces or functions of the ESP32-Ethernet-Kit.

IP101GRI (PHY) Interface The allocation of the ESP32 (MAC) pins to IP101GRI (PHY) is shown in the table below. Implementation of ESP32-Ethernet-Kit defaults to Reduced Media-Independent Interface (RMII).

No.	ESP32 Pin (MAC)	IP101GRI (PHY)
RMII Interface		
1	GPIO21	TX_EN
2	GPIO19	TXD[0]
3	GPIO22	TXD[1]
4	GPIO25	RXD[0]
5	GPIO26	RXD[1]
6	GPIO27	CRS_DV
7	GPIO0	REF_CLK
Serial Management Interface		
8	GPIO23	MDC
9	GPIO18	MDIO
PHY Reset		
10	GPIO5	Reset_N

Note: Except for REF_CLK, the allocation of all pins under the *RMII Interface* is fixed and cannot be changed either through IO MUX or GPIO Matrix.

GPIO Header 1 This header exposes some GPIOs that are not used elsewhere on the ESP32-Ethernet-Kit.

No.	ESP32 Pin
1	GPIO32
2	GPIO33
3	GPIO34
4	GPIO35
5	GPIO36
6	GPIO39

GPIO Header 2 This header contains the GPIOs with specific MII functionality (except GPIO2), as opposed to Reduced Media-Independent Interface (RMII) functionality implemented on ESP32-Ethernet-Kit board by default, see *IP101GRI (PHY) Interface*. Depending on the situation, if MMI is used, specific Ethernet applications might require this functionality.

No.	ESP32 Pin	MII Function	Notes
1	GPIO17	EMAC_CLK_180	See note 1
2	GPIO16	EMAC_CLK_OUT	See note 1
3	GPIO4	EMAC_TX_ER	
4	GPIO2	n/a	See note 2
5	GPIO5	EMAC_RX_CLK	See note 2

Note:

1. The ESP32 pins GPIO16 and GPIO17 are not broken out to the ESP32-WROVER-B module and therefore not available for use. If you need to use these pins, please solder a module without SPIRAM memory inside, e.g., the ESP32-WROOM-32D or ESP32-SOLO-1.
2. Functionality depends on the settings of the *Function Switch*.

GPIO Header 3 The functionality of GPIOs connected to this header depends on the settings of the *Function Switch*.

No.	ESP32 Pin
1	GPIO15
2	GPIO13
3	GPIO12
4	GPIO14
5	GND
6	3V3

GPIO Allocation Summary

ESP32-WROVER-B	IP101GRI	UART	JTAG	GPIO	Notes
S_VP				IO36	
S_VN				IO39	
IO34				IO34	
IO35				IO35	
IO32				IO32	
IO33				IO33	
IO25	RXD[0]				
IO26	RXD[1]				
IO27	CRS_DV				
IO14			TMS	IO14	
IO12			TDI	IO12	
IO13		RTS	TCK	IO13	
IO15		CTS	TDO	IO15	
IO2				IO2	See note 1 and 3 below
IO0	REF_CLK				See note 2 and 3 below
IO4			nTRST	IO4	
IO16				IO16 (NC)	See note 4 below
IO17				IO17 (NC)	See note 4 below
IO5	Reset_N			IO5	
IO18	MDIO				
IO19	TXD[0]				
IO21	TX_EN				
RXD0		RXD			
TXD0		TXD			
IO22	TXD[1]				
IO23	MDC				

Note:

1. GPIO2 is used to enable external oscillator of the PHY.
2. GPIO0 is a source of 50 MHz reference clock for the PHY. The clock signal is first inverted, to account for transmission line delay, and then supplied to the PHY.
3. To prevent affecting the power-on state of GPIO0 by the clock output on the PHY side, the PHY external oscillator is enabled using GPIO2 after ESP32 is powered up.
4. The ESP32 pins GPIO16 and GPIO17 are not broken out to the ESP32-WROVER-B module and therefore not available for use. If you need to use these pins, please solder a module without SPIRAM memory inside, e.g., the ESP32-WROOM-32D or ESP32-SOLO-1.

Start Application Development Before powering up your ESP32-Ethernet-Kit, please make sure that the board is in good condition with no obvious signs of damage.

Initial Setup

1. Set the **Function Switch** on the *Ethernet board (A)* to its default position by turning all the switches to **ON**.
2. To simplify flashing and testing the application, do not install any jumpers and do not connect any signals to the board headers.
3. The *PoE board (B)* can now be plugged in, but do not connect external power to it.
4. Connect the *Ethernet board (A)* to the PC with a USB cable.
5. Turn the **Power Switch** from GND to 5V0 position, the **5V Power On LED** should light up.

Now to Development Proceed to Get Started, where Section Installation will quickly help you set up the development environment and then flash an example project onto your board.

Move on to the next section only if you have successfully completed all the above steps.

Configure and Load the Ethernet Example After setting up the development environment and testing the board, you can configure and flash the ethernet/basic example. This example has been created for testing Ethernet functionality. It supports different PHY, including **IP101GRI** installed on *ESP32-Ethernet-Kit v1.0 board*.

Related Documents

- ESP32-Ethernet-Kit v1.0 Ethernet board (A) schematic (PDF)
- ESP32-Ethernet-Kit v1.0 PoE board (B) schematic (PDF)
- ESP32 Datasheet (PDF)
- ESP32-WROVER-B Datasheet (PDF)
- JTAG Debugging

For other design documentation for the board, please contact us at sales@espressif.com.

ESP32-Ethernet-Kit v1.1

This guide shows how to get started with the ESP32-Ethernet-Kit development board and also provides information about its functionality and configuration options.

The *ESP32-Ethernet-Kit* is an Ethernet-to-Wi-Fi development board that enables Ethernet devices to be interconnected over Wi-Fi. At the same time, to provide more flexible power supply options, the ESP32-Ethernet-Kit also supports power over Ethernet (PoE).

What You Need

- *ESP32-Ethernet-Kit v1.1 board*
- USB 2.0 cable (Standard-A to Micro-B)
- Computer running Windows, Linux, or macOS

You can skip the introduction sections and go directly to Section *Start Application Development*.

Overview ESP32-Ethernet-Kit is an ESP32-based development board produced by Espressif.

It consists of two development boards, the Ethernet board A and the PoE board B. The *Ethernet board (A)* contains Bluetooth®/Wi-Fi dual-mode ESP32-WROVER-B module and IP101GRI, a Single Port 10/100 Fast Ethernet Transceiver (PHY). The *PoE board (B)* provides power over Ethernet functionality. The A board can work independently, without the board B installed.

For the application loading and monitoring, the Ethernet board (A) also features FTDI FT2232H chip - an advanced multi-interface USB bridge. This chip enables to use JTAG for direct debugging of ESP32 through the USB interface without a separate JTAG debugger.

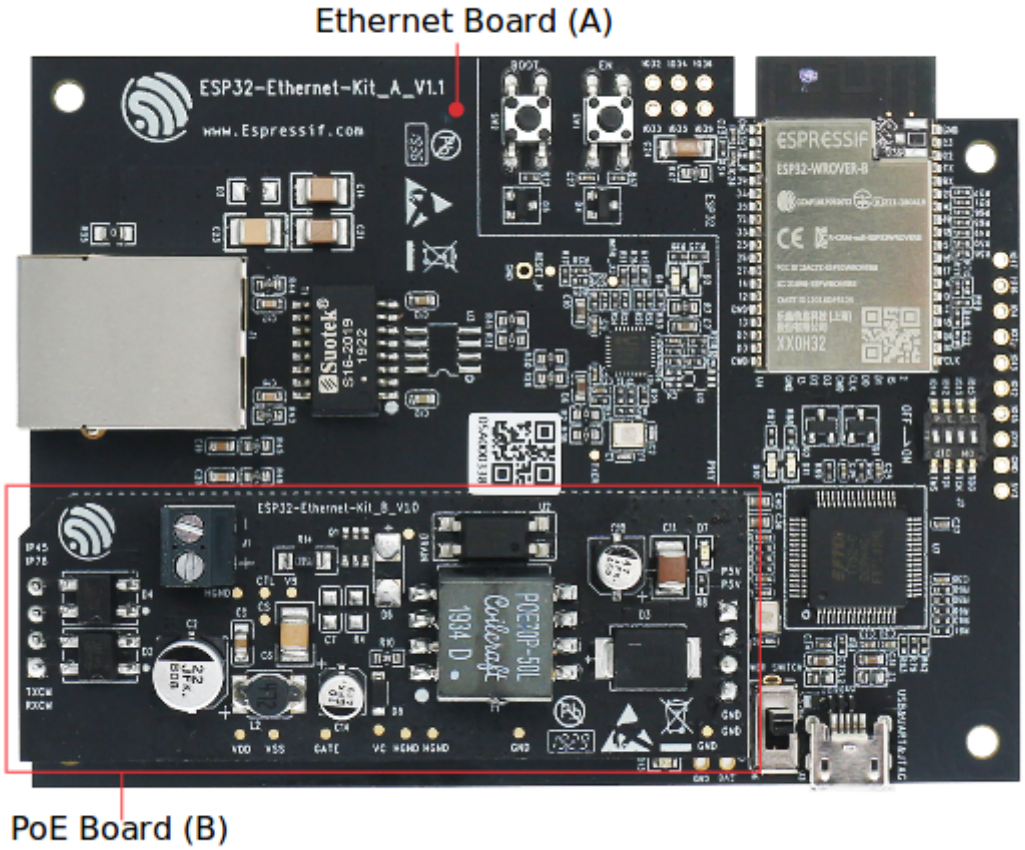


Fig. 12: ESP32-Ethernet-Kit v1.1

Functionality Overview The block diagram below shows the main components of ESP32-Ethernet-Kit and their interconnections.

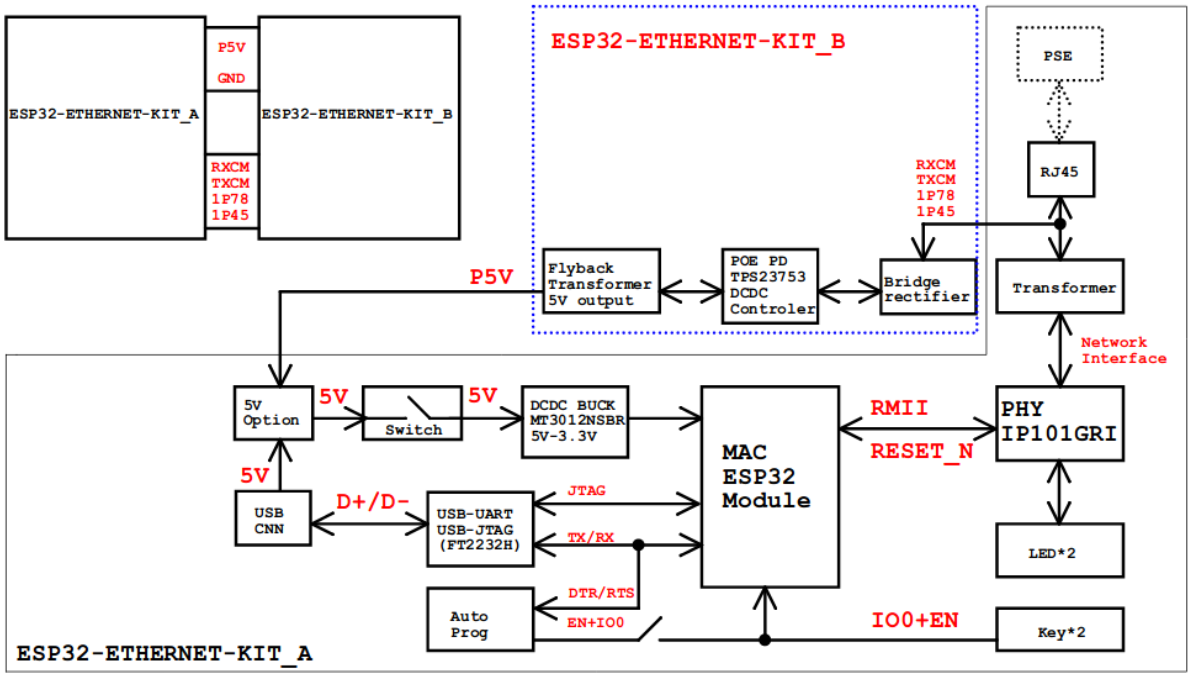


Fig. 13: ESP32-Ethernet-Kit block diagram (click to enlarge)

Functional Description The following figures and tables describe the key components, interfaces, and controls of the ESP32-Ethernet-Kit.

Ethernet Board (A) The table below provides description starting from the picture's top right corner and going clockwise.

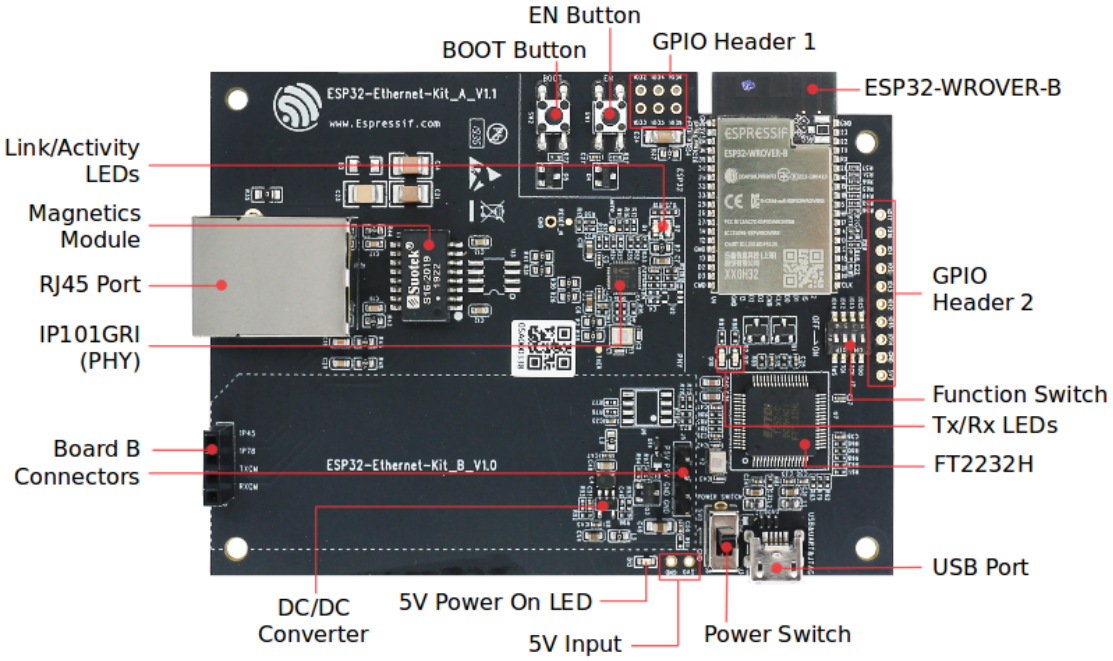


Fig. 14: ESP32-Ethernet-Kit - Ethernet board (A) layout (click to enlarge)

Table 3: Table 1 Component Description

Key Component	Description
ESP32-WROVER-B	This ESP32 module features 64-Mbit PSRAM for flexible extended storage and data processing capabilities.
GPIO Header 2	Five unpopulated through-hole solder pads to provide access to selected GPIOs of ESP32. For details, see *GPIO Header 2*.
Function Switch	A 4-bit DIP switch used to configure the functionality of selected GPIOs of ESP32. Please note that placement of GPIO pin number marking on the board' s silkscreen besides the DIP switch is incorrect. For details and correct pin allocation see *Function Switch*.
Tx/Rx LEDs	Two LEDs to show the status of UART transmission.
FT2232H	The FT2232H chip serves as a multi-protocol USB-to-serial bridge which can be programmed and controlled via USB to provide communication with ESP32. FT2232H also features USB-to-JTAG interface which is available on channel A of the chip, while USB-to-serial is on channel B. The FT2232H chip enhances user-friendliness in terms of application development and debugging. See ESP32-Ethernet-Kit v1.1 Ethernet board (A) schematic.
USB Port	USB interface. Power supply for the board as well as the communication interface between a computer and the board.
Power Switch	Power On/Off Switch. Toggling the switch to **5V0** position powers the board on, toggling to **GND** position powers the board off.
5V Input	The 5 V power supply interface can be more convenient when the board is operating autonomously (not connected to a computer).
5V Power On LED	This red LED turns on when power is supplied to the board, either from USB or 5 V Input.
DC/DC Converter	Provided DC 5 V to 3.3 V conversion, output current up to 2 A.
Board B Connectors	A pair male and female header pins for mounting the *PoE board (B)*.
IP101GRI (PHY)	The physical layer (PHY) connection to the Ethernet cable is implemented using the IP101GRI chip. The connection between PHY and ESP32 is done through the reduced media-independent interface (RMII), a variant of the media-independent interface (MII) standard. The PHY supports the IEEE 802.3/802.3u standard of 10/100 Mbps.
RJ45 Port	Ethernet network data transmission port.
Magnetics Module	The Magnetics are part of the Ethernet specification to protect against faults and transients, including rejection of common mode signals between the transceiver IC and the cable. The magnetics also provide galvanic isolation between the transceiver and the Ethernet device.

PoE Board (B) This board coverts power delivered over the Ethernet cable (PoE) to provide a power supply for the Ethernet board (A). The main components of the PoE board (B) are shown on the block diagram under *Functionality Overview*.

The PoE board (B) has the following features:

- Support for IEEE 802.3at
- Power output: 5 V, 1.4 A

To take advantage of the PoE functionality the **RJ45 Port** of the Ethernet board (A) should be connected with an Ethernet cable to a switch that supports PoE. When the Ethernet board (A) detects 5 V power output from the PoE board (B), the USB power will be automatically cut off.

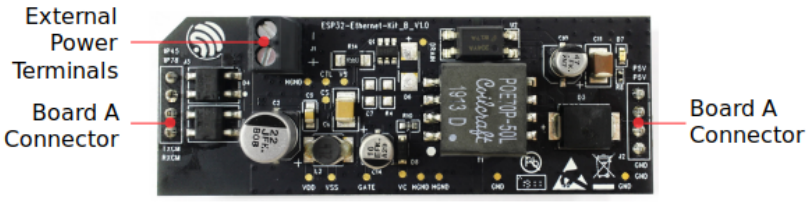


Fig. 15: ESP32-Ethernet-Kit - PoE board (B) layout (click to enlarge)

Table 4: Table PoE board (B)

Key Component	Description
Board A Connector	Four female (left) and four male (right) header pins for connecting the PoE board (B) to *Ethernet board (A)*. The pins on the left accept power coming from a PoE switch. The pins on the right deliver 5 V power supply to the Ethernet board (A).
External Power Terminals	Optional power supply (26.6 ~ 54 V) to the PoE board (B).

Setup Options This section describes options to configure the ESP32-Ethernet-Kit hardware.

Function Switch When in On position, this DIP switch is routing listed GPIOs to FT2232H to provide JTAG functionality. When in Off position, the GPIOs may be used for other purposes.

DIP SW	GPIO Pin
1	GPIO13
2	GPIO12
3	GPIO15
4	GPIO14

Note: Placement of GPIO pin number marking on the board' s silkscreen besides the DIP switch is incorrect. Please use instead the pin order as in the table above.

RMII Clock Selection The ethernet MAC and PHY under RMII working mode need a common 50 MHz reference clock (i.e., RMII clock) that can be provided either externally, or generated from internal ESP32 APLL.

Note: For additional information on the RMII clock selection, please refer to ESP32-Ethernet-Kit v1.1 Ethernet board (A) schematic, sheet 2, location D2.

RMII Clock Sourced Externally by PHY By default, the ESP32-Ethernet-Kit is configured to provide RMII clock for the IP101GRI PHY's 50M_CLKO output. The clock signal is generated by the frequency multiplication of 25 MHz crystal connected to the PHY. For details, please see the figure below.

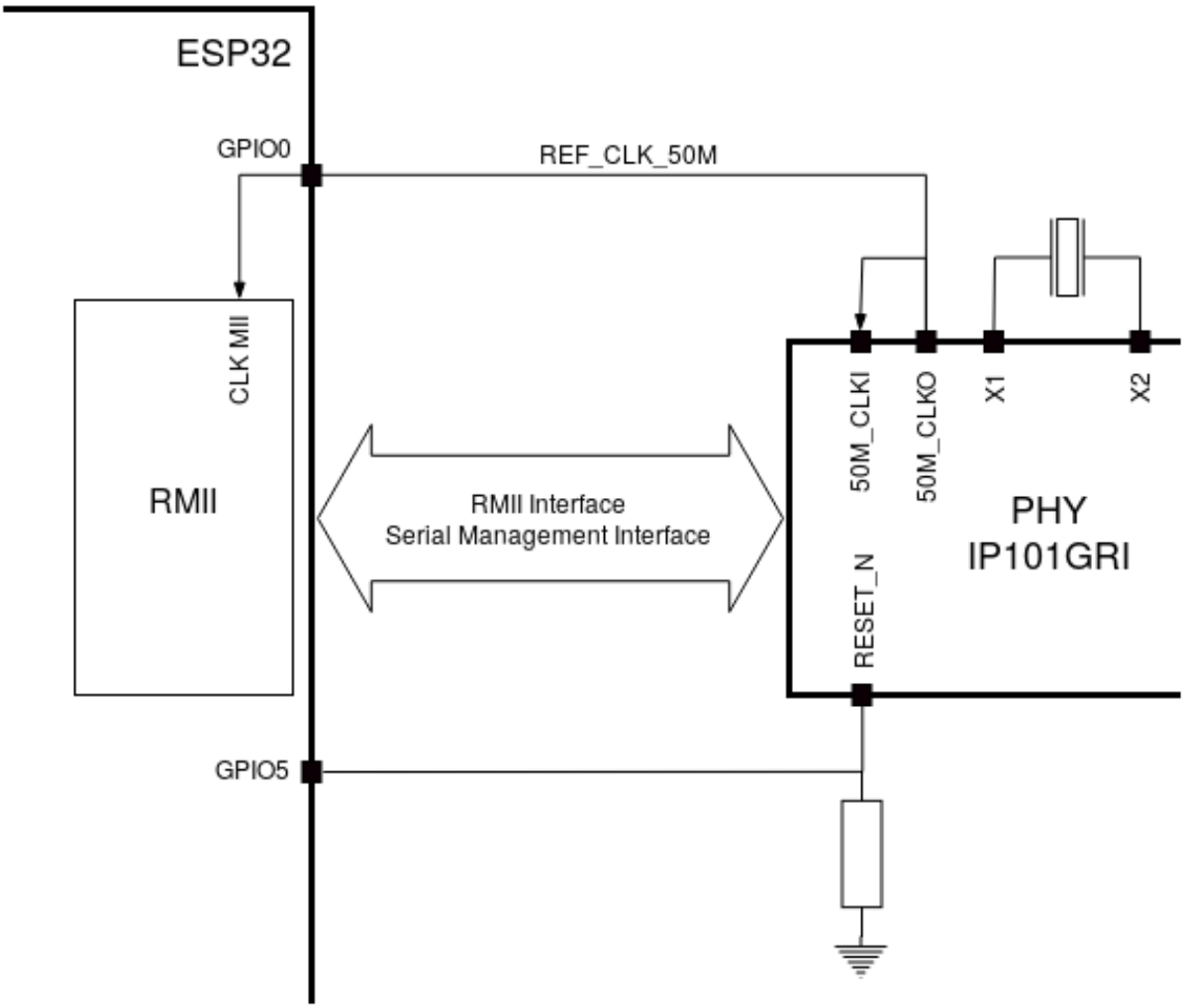


Fig. 16: RMII Clock from IP101GRI PHY (click to enlarge)

Please note that the PHY is reset on power-up by pulling the RESET_N signal down with a resistor. ESP32 should assert RESET_N high with GPIO5 to enable PHY. Only this can ensure the power-up of the system. Otherwise, ESP32 may enter download mode.

RMII Clock Sourced Internally from ESP32's APLL Another option is to source the RMII Clock from internal ESP32 APLL, see figure below. The clock signal coming from GPIO0 is first inverted, to account for transmission line delay, and then supplied to the PHY.

To implement this option, users need to remove or add some RC components on the board. For details please refer to ESP32-Ethernet-Kit v1.1 Ethernet board (A) schematic, sheet 2, location U2.

Note: Please note that you need to have *RMII Clock Sourced Externally by PHY* or by an external clock source in the following cases:

- If Wi-Fi and Ethernet are used simultaneously, the RMII clock cannot be generated by the internal APLL clock, as it would result in clock instability.
- APLL is already used for other purposes (e.g., I2S peripheral).

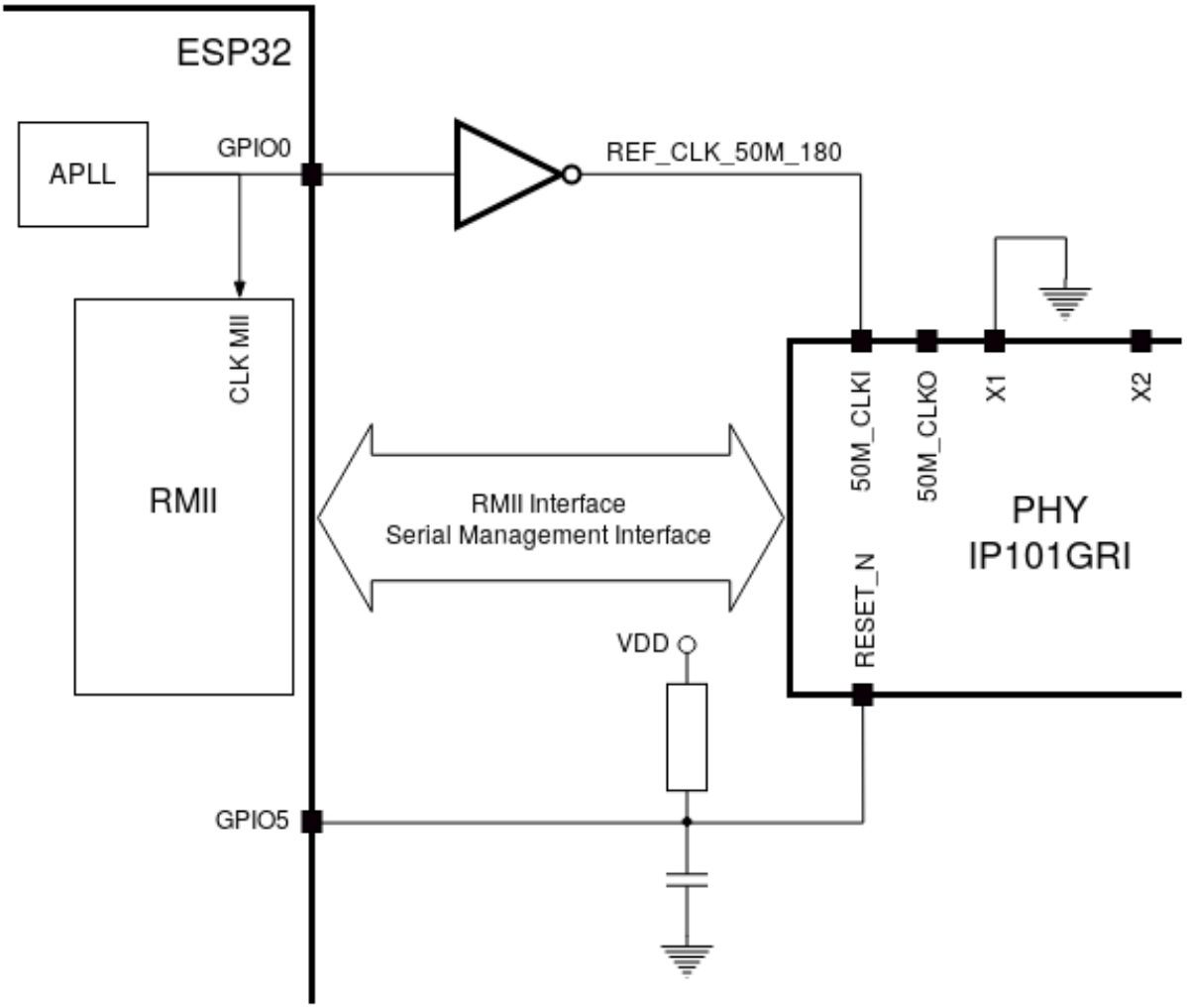


Fig. 17: RMII Clock from ESP Internal APLL (click to enlarge)

GPIO Allocation This section describes the allocation of ESP32 GPIOs to specific interfaces or functions of the ESP32-Ethernet-Kit.

IP101GRI (PHY) Interface The allocation of the ESP32 (MAC) pins to IP101GRI (PHY) is shown in the table below. Implementation of ESP32-Ethernet-Kit defaults to Reduced Media-Independent Interface (RMII).

No.	ESP32 Pin (MAC)	IP101GRI (PHY)
RMII Interface		
1	GPIO21	TX_EN
2	GPIO19	TXD[0]
3	GPIO22	TXD[1]
4	GPIO25	RXD[0]
5	GPIO26	RXD[1]
6	GPIO27	CRS_DV
7	GPIO0	REF_CLK
Serial Management Interface		
8	GPIO23	MDC
9	GPIO18	MDIO
PHY Reset		
10	GPIO5	Reset_N

Note: Except for REF_CLK, the allocation of all pins under the ESP32's *RMII Interface* is fixed and cannot be changed either through IO MUX or GPIO Matrix.

GPIO Header 1 This header exposes some GPIOs that are not used elsewhere on the ESP32-Ethernet-Kit.

No.	ESP32 Pin
1	GPIO32
2	GPIO33
3	GPIO34
4	GPIO35
5	GPIO36
6	GPIO39

GPIO Header 2 This header contains GPIOs that may be used for other purposes depending on scenarios described in column **Notes**.

No.	ESP32 Pin	Notes
1	GPIO17	See note 1
2	GPIO16	See note 1
3	GPIO4	
4	GPIO2	
5	GPIO13	See note 2
6	GPIO12	See note 2
7	GPIO15	See note 2
8	GPIO14	See note 2
9	GND	Ground
10	3V3	3.3 V power supply

Note:

1. The ESP32 pins GPIO16 and GPIO17 are not broken out to the ESP32-WROVER-B module and therefore not available for use. If you need to use these pins, please solder a module without PSRAM memory inside, e.g., the ESP32-WROOM-32D or ESP32-SOLO-1.
2. Functionality depends on the settings of the *Function Switch*.

GPIO Allocation Summary

ESP32-WROVER-B	IP101GRI	UART	JTAG	GPIO	Notes
S_VP				IO36	
S_VN				IO39	
IO34				IO34	
IO35				IO35	
IO32				IO32	
IO33				IO33	
IO25	RXD[0]				
IO26	RXD[1]				
IO27	CRS_DV				
IO14			TMS	IO14	
IO12			TDI	IO12	
IO13		RTS	TCK	IO13	
IO15		CTS	TDO	IO15	
IO2				IO2	
IO0	REF_CLK				See note 1
IO4				IO4	
IO16				IO16 (NC)	See note 2
IO17				IO17 (NC)	See note 2
IO5	Reset_N				See note 1
IO18	MDIO				
IO19	TXD[0]				
IO21	TX_EN				
RXD0		RXD			
TXD0		TXD			
IO22	TXD[1]				
IO23	MDC				

Note:

1. To prevent the power-on state of the GPIO0 from being affected by the clock output on the PHY side, the RESET_N signal to PHY defaults to low, turning the clock output off. After power-on you can control RESET_N with GPIO5 to turn the clock output on. See also *RMII Clock Sourced Externally by PHY*. For PHYs that cannot turn off the clock output through RESET_N, it is recommended to use a crystal module that can be disabled/enabled externally. Similarly like when using RESET_N, the oscillator module should be disabled by default and turned on by ESP32 after power-up. For a reference design please see ESP32-Ethernet-Kit v1.1 Ethernet board (A) schematic.
2. The ESP32 pins GPIO16 and GPIO17 are not broken out to the ESP32-WROVER-B module and therefore not available for use. If you need to use these pins, please solder a module without PSRAM memory inside, e.g., the ESP32-WROOM-32D or ESP32-SOLO-1.

Start Application Development Before powering up your ESP32-Ethernet-Kit, please make sure that the board is in good condition with no obvious signs of damage.

Initial Setup

1. Set the **Function Switch** on the *Ethernet board (A)* to its default position by turning all the switches to **ON**.
2. To simplify flashing and testing of the application, do not input extra signals to the board headers.
3. The *PoE board (B)* can now be plugged in, but do not connect external power to it.
4. Connect the *Ethernet board (A)* to the PC with a USB cable.
5. Turn the **Power Switch** from GND to 5V0 position, the **5V Power On LED** should light up.

Now to Development Proceed to Get Started, where Section Installation will quickly help you set up the development environment and then flash an example project onto your board.

Move on to the next section only if you have successfully completed all the above steps.

Configure and Load the Ethernet Example After setting up the development environment and testing the board, you can configure and flash the ethernet/basic example. This example has been created for testing Ethernet functionality. It supports different PHY, including **IP101GRI** installed on *ESP32-Ethernet-Kit v1.1 board*.

Summary of Changes from ESP32-Ethernet-Kit v1.0

- The original inverted clock provided to the PHY by ESP32 using GPIO0 has been replaced by a clock generated on PHY side. The PHY's clock is connected to the ESP32 with same GPIO0. The GPIO2 which was originally used to control the active crystal oscillator on the PHY side, can now be used for other purposes.
- On power up, the ESP32 boot strapping pin GPIO0 may be affected by clock generated on the PHY side. To resolve this issue the PHY's Reset-N signal is pulled low using resistor R17 and effectively turning off the PHY's clock output. The Reset-N signal can be then pulled high by ESP32 using GPIO5.
- Removed FT2232H chip's external SPI Flash U6.
- Removed flow control jumper header J4.
- Removed nTRST JTAG signal. The corresponding GPIO4 can now be used for other purposes.
- Pull-up resistor R68 on the GPIO15 line is moved to the MTDO side of JTAG.
- To make the A and B board connections more foolproof (reduce chances of plugging in the B board in reverse orientation), the original two 4-pin male rows on board A were changed to one 4-pin female row and one 4-pin male row. Corresponding male and female 4-pins rows were installed on board B.

Other Versions of ESP32-Ethernet-Kit

- *ESP32-Ethernet-Kit v1.0*

Related Documents

- ESP32-Ethernet-Kit v1.1 Ethernet board (A) schematic (PDF)
- ESP32-Ethernet-Kit v1.0 PoE board (B) schematic (PDF)
- ESP32 Datasheet (PDF)
- ESP32-WROVER-B Datasheet (PDF)
- JTAG Debugging

For other design documentation for the board, please contact us at sales@espressif.com.

Chapter 7

EOL (End of Life) Boards

This section contains user guides for the ESP32 end-of-life development boards and is provided for reference only. While these boards may still be available on the market or used in legacy systems, they no longer receive updates, bug fixes, or official support. It is recommended to switch to newer development boards for better performance and more features.

7.1 ESP32-Sense-Kit

The ESP32 touch sensor development kit, ESP32-Sense-Kit, is used for evaluating and developing ESP32 touch sensor system.

7.1.1 ESP32-Sense-Kit

Overview

The ESP32 touch sensor development kit, ESP32-Sense-Kit, is used for evaluating and developing ESP32 touch sensor system. ESP32-Sense-Kit consists of one motherboard and multiple daughterboards. The motherboard contains a display unit, a main control unit and a debug unit. The daughterboards have touch electrodes in different combinations or shapes, such as linear slider, wheel slider, matrix buttons and spring buttons, depending on the application scenarios. Users can design and add their own daughterboards for special usage cases.

The following image shows the whole ESP32-Sense-Kit.

Preparation

- **Install overlay**
  If plastic is used for the overlay, the recommended thickness is 3 mm or less. Because air reduces touch sensitivity, any air gaps between the daughterboard and overlay must be eliminated. You can use double-sided adhesive tape to fill in the air gap. For the daughterboard with metal springs, 7 mm stud bolts should be used to install the overlay.
- **Install daughterboard**
  Use a connector to connect motherboard with daughterboard. You can use four 7 mm plastic stud bolts to have the daughterboard steadily parallel to the motherboard, as shown in the image below:

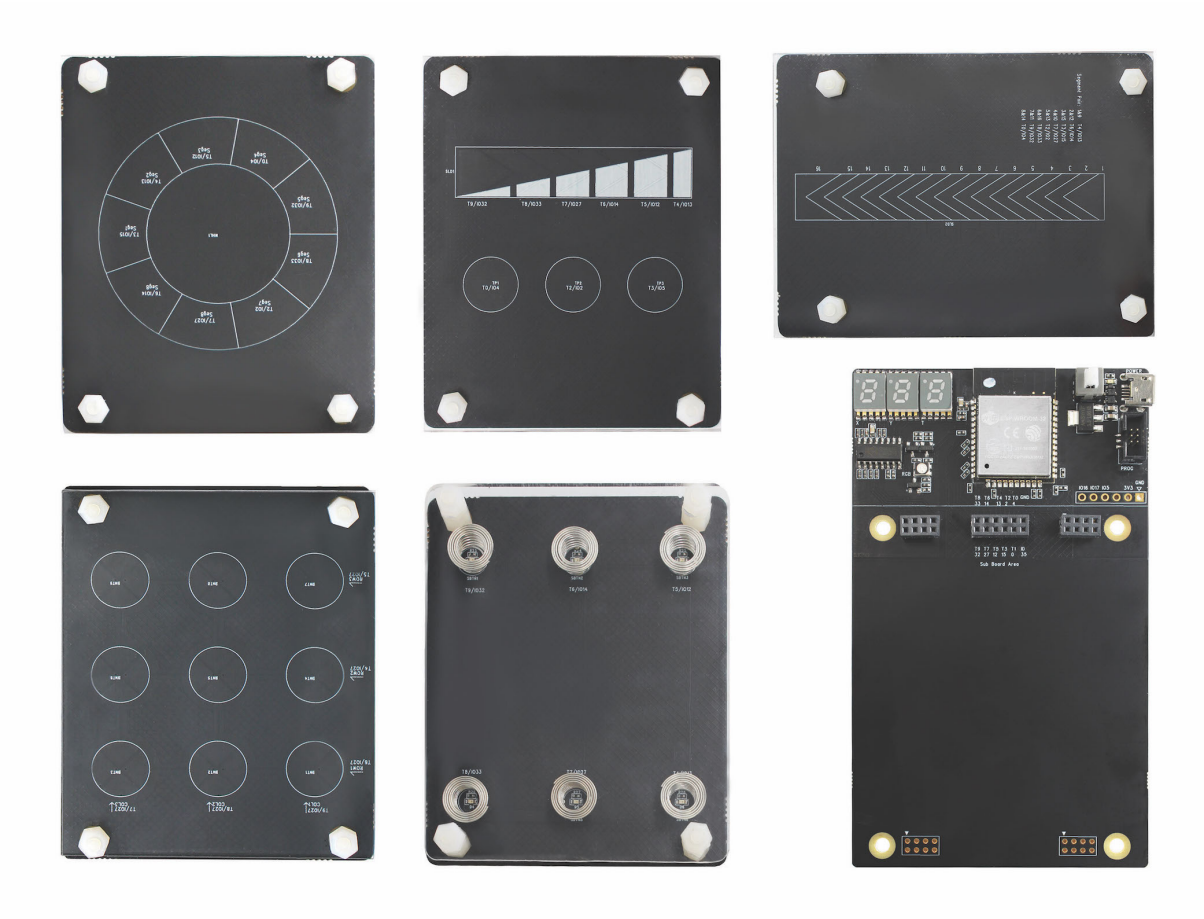


Fig. 1: ESP32-Sense-Kit

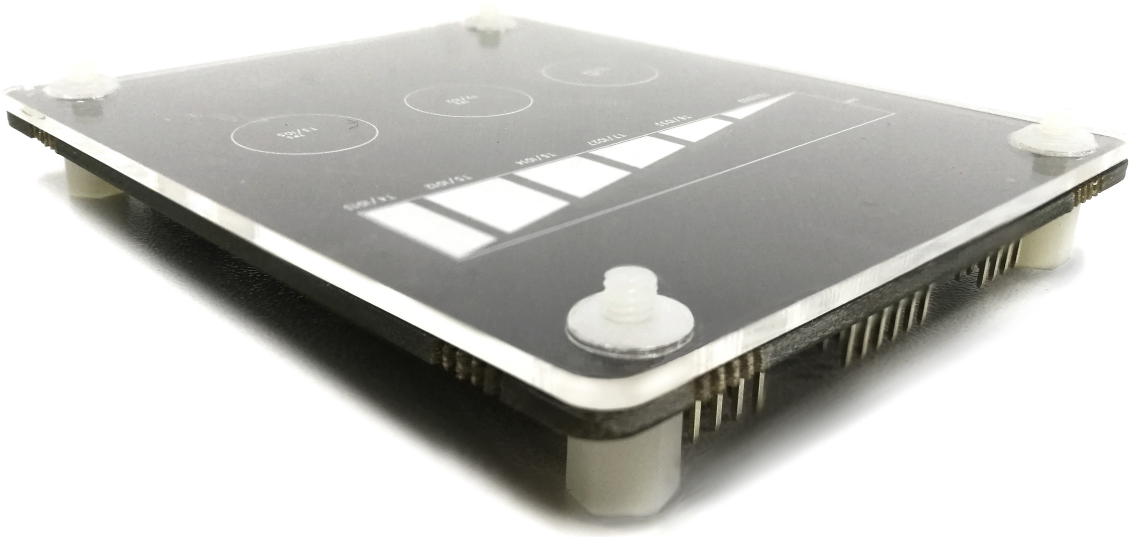


Fig. 2: Install Overlay

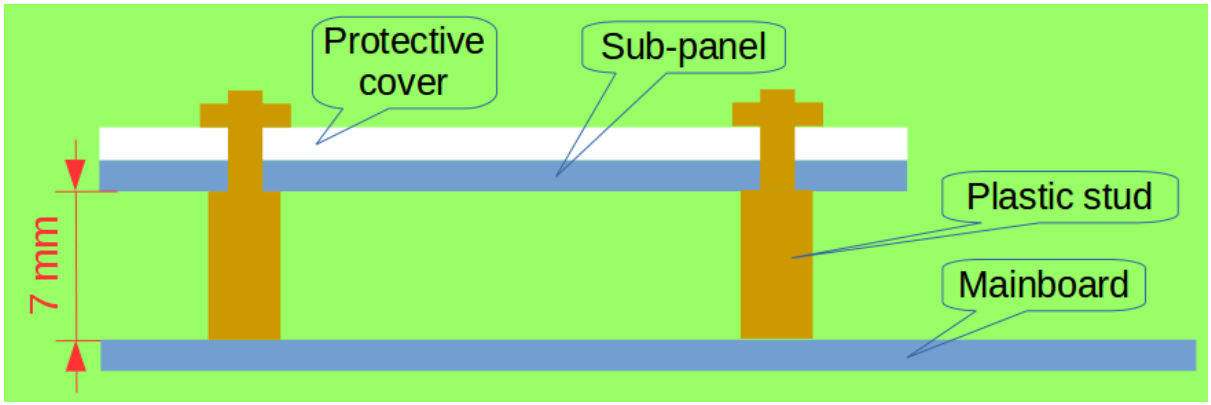


Fig. 3: Install Daughterboard

- **Set ESP-Prog debugger**
  ESP-Prog is used as the program download tool and power supply. ESP-Prog has two sets of jumpers: IO0 jumper and power supply jumper. Choose 5 V power supply for the latter. IO0 can be used both for selecting boot mode (download mode or working mode) and as a touch pin. As a result, it should be disconnected if used as a touch pin in working mode. The image below shows the settings for ESP-Prog.

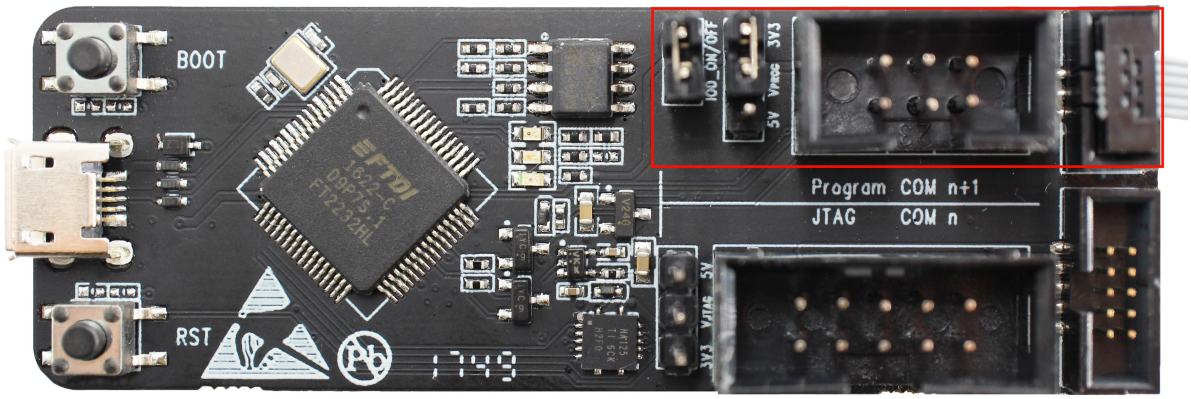


Fig. 4: Set ESP-Prog Debuggers

- **Connect ESP-Prog with motherboard**
  The ESP-Prog has a Jtag interface and a Program interface. Connect ESP-Prog and the motherboard through the Program interface.
- **Download programs**
  Run `make menuconfig` to configure the config settings for ESP32-Sense Project, as the screenshot below shows. Run `make flash` to download programs into the development board.
- **Replace daughterboard**
  ESP32 will detect the divided voltage of the voltage divider on the daughterboard when it is powered on to identify different daughterboards. Re-power on the development board after replacing the daughterboard.

Hardware Resources

Motherboard

- **Function Block Diagram**
  The image below shows the function block diagram of the motherboard.
- **Motherboard Components**

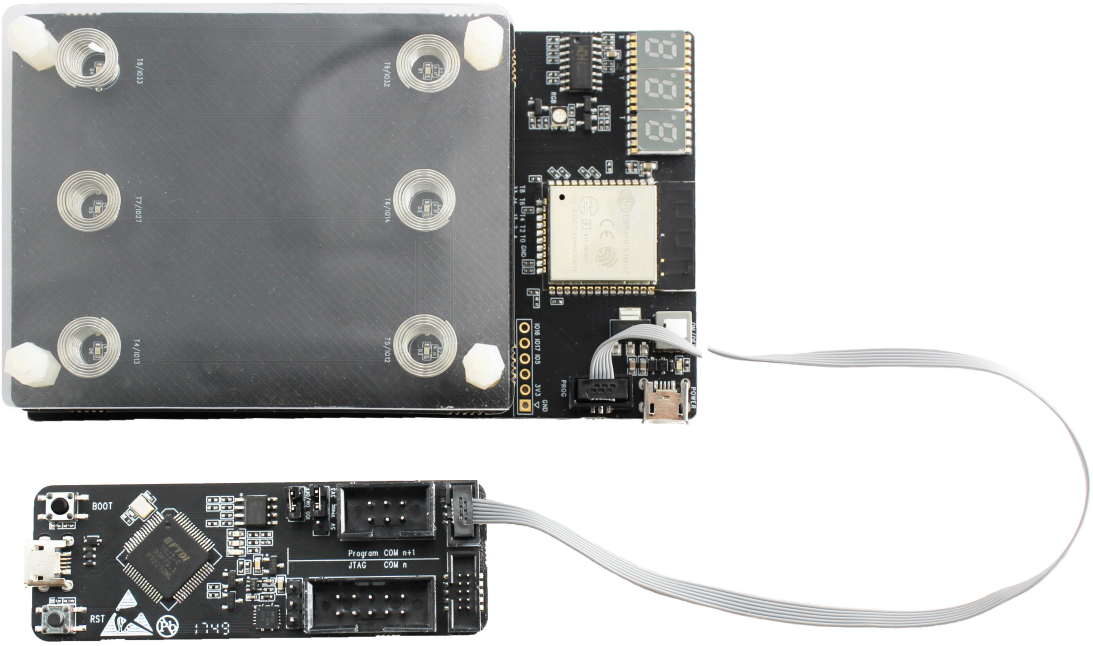
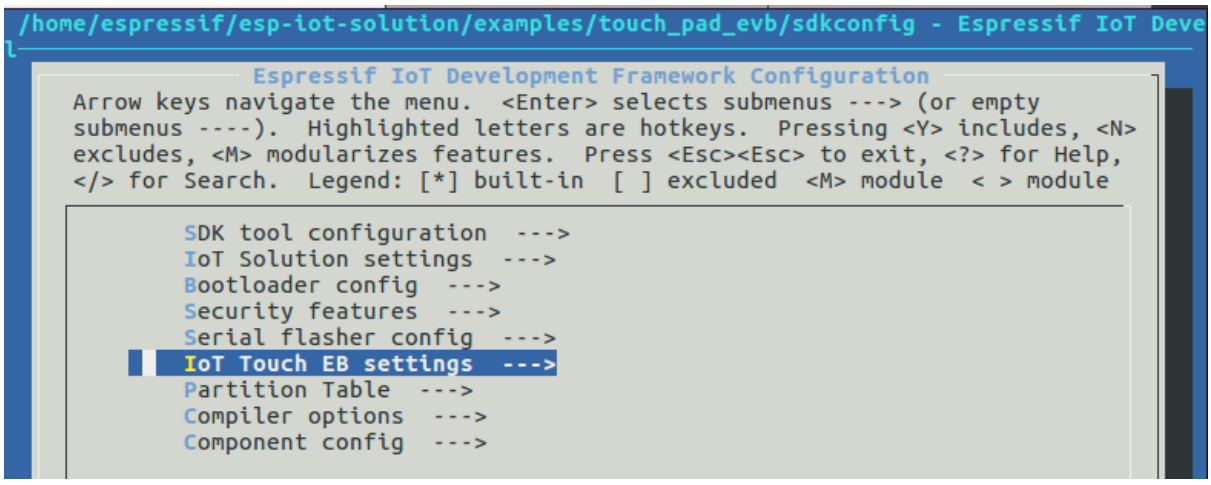


Fig. 5: Connect ESP-Prog with Motherboard



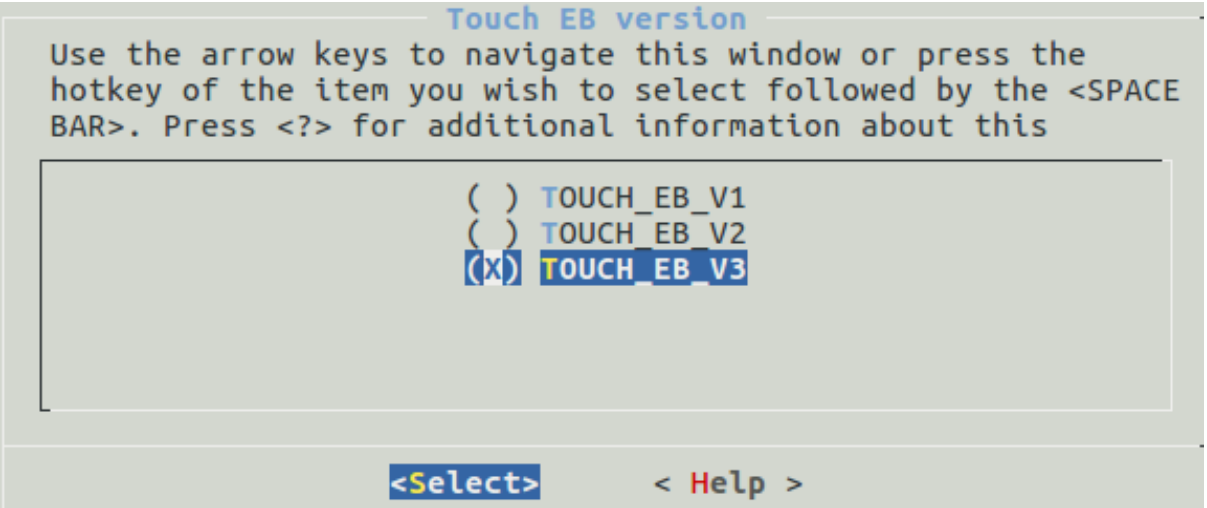


Fig. 6: Download Programs

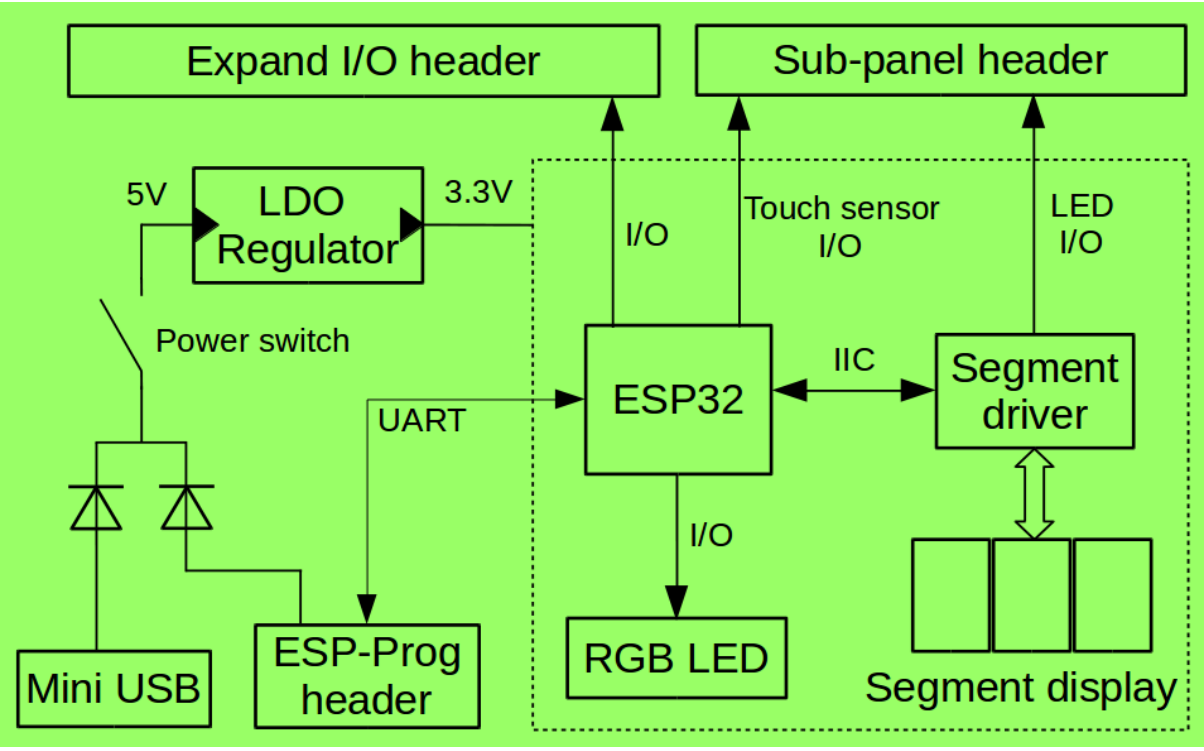


Fig. 7: Function Block Diagram

The display unit includes three segment displays and an RGB circuit. The debug unit includes the ESP-Prog debugger interface. The main control unit includes the ESP32 module. The mini USB is the power supply.

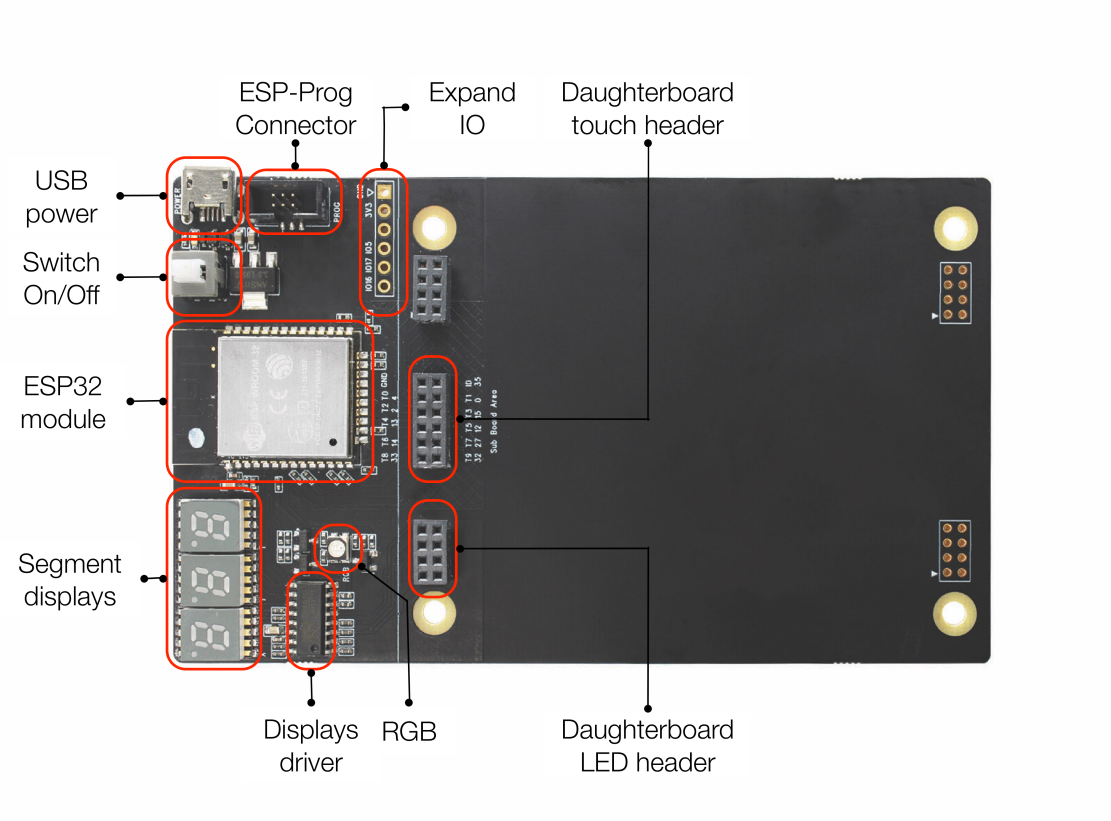


Fig. 8: Motherboard Components

- **Power Management System**
  The mini USB and ESP-Prog can both be the power supply for ESP32-Sense Kit. They do not interfere with each other thanks to the protection diode. The mini USB can only serve as the power supply, while ESP-Prog also supports automatic firmware downloading. The figure below shows the schematics of the power management system.
- **Display Unit**
  The display unit on the motherboard can intuitively feedback touch event. The three 7-segment displays show the location of the pad that is being touched and the duration of a touch event. The segment displays are driven by CH455G chip, and controlled through I2C interface. The RGB LED reflects the colors when a touch event occurs. When a finger moves on the slider, the RGB LED will show the change of colors.
  The figure below shows the schematics of the display unit:

Daughterboard

- **Divided resistance**
  The touch electrodes are arranged in different combinations depending on the application scenario. Each daughterboard has a voltage divider that has a unique value. The program running on motherboard reads the divider value through ADC and thus each daughterboard can be identified. The voltage divider is shown below:

Daughterboard	Divided resistance (Kohm)	ADC reading (Min)	ADC reading (Max)
Spring button	0	0	250
Linear slider	4.7	805	1305
Matrix button	10	1400	1900
Duplex slider	19.1	1916	2416
Wheel slider	47	2471	2971

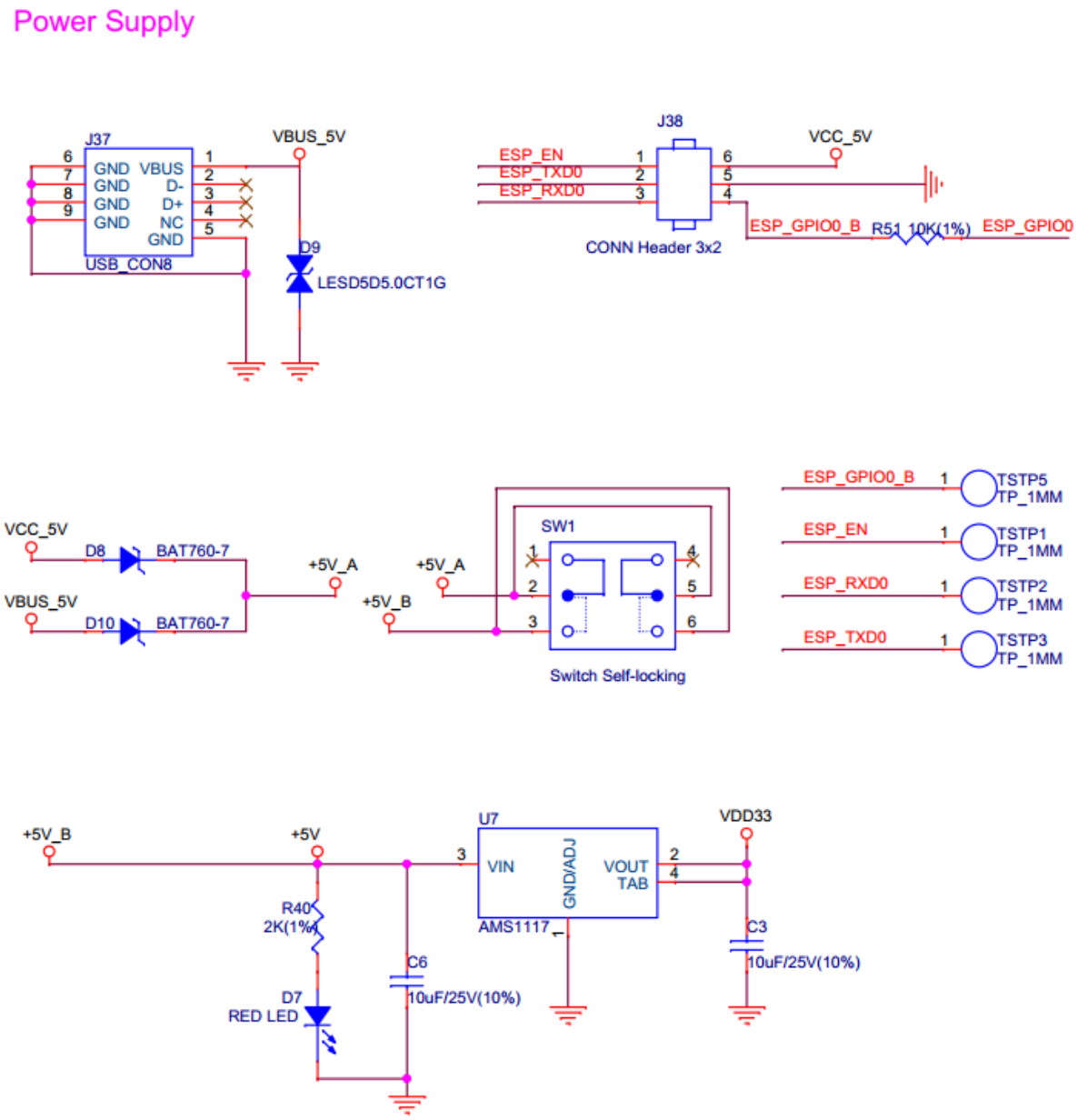


Fig. 9: Power Management System

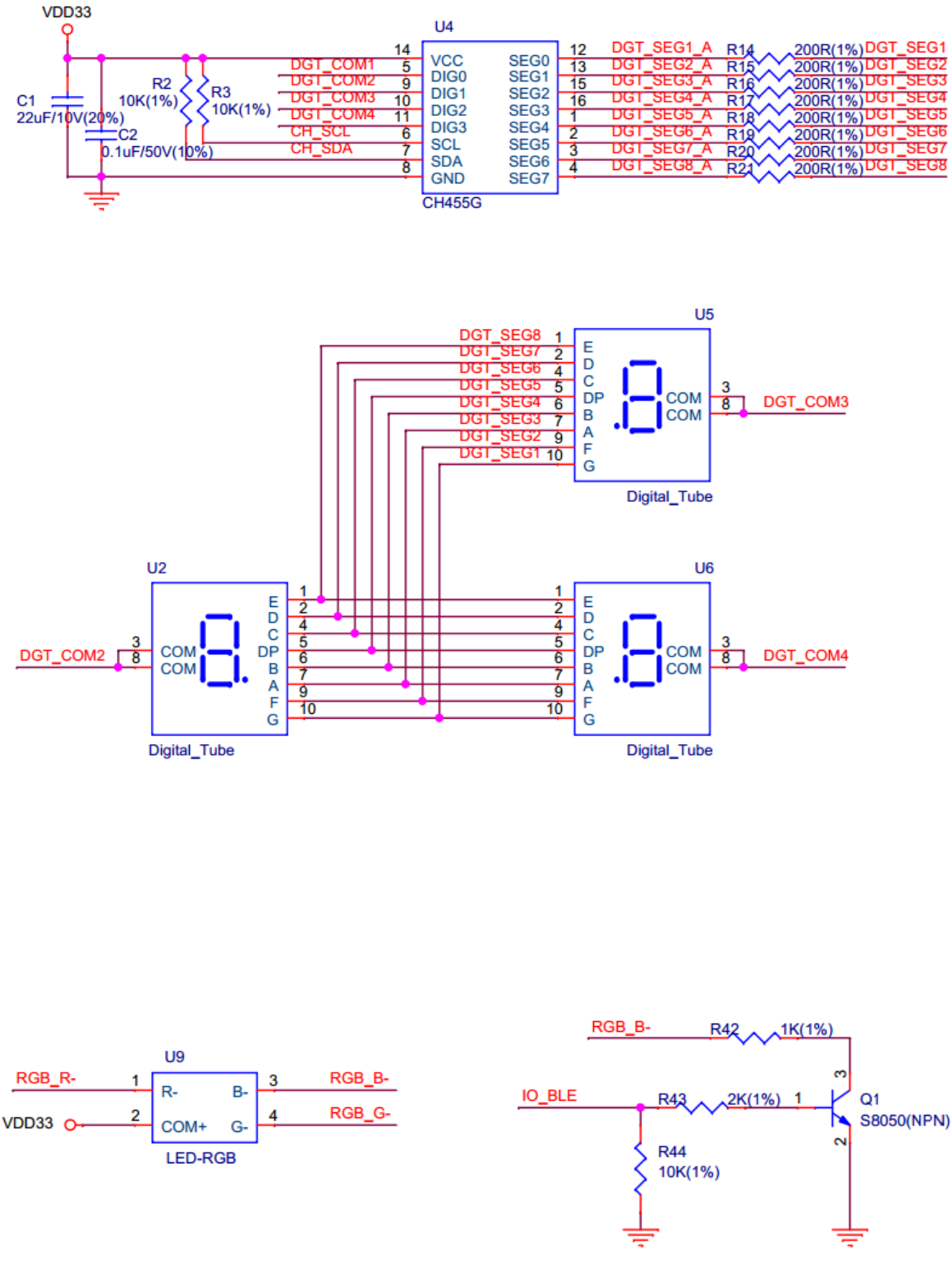


Fig. 10: Diaplay Unit

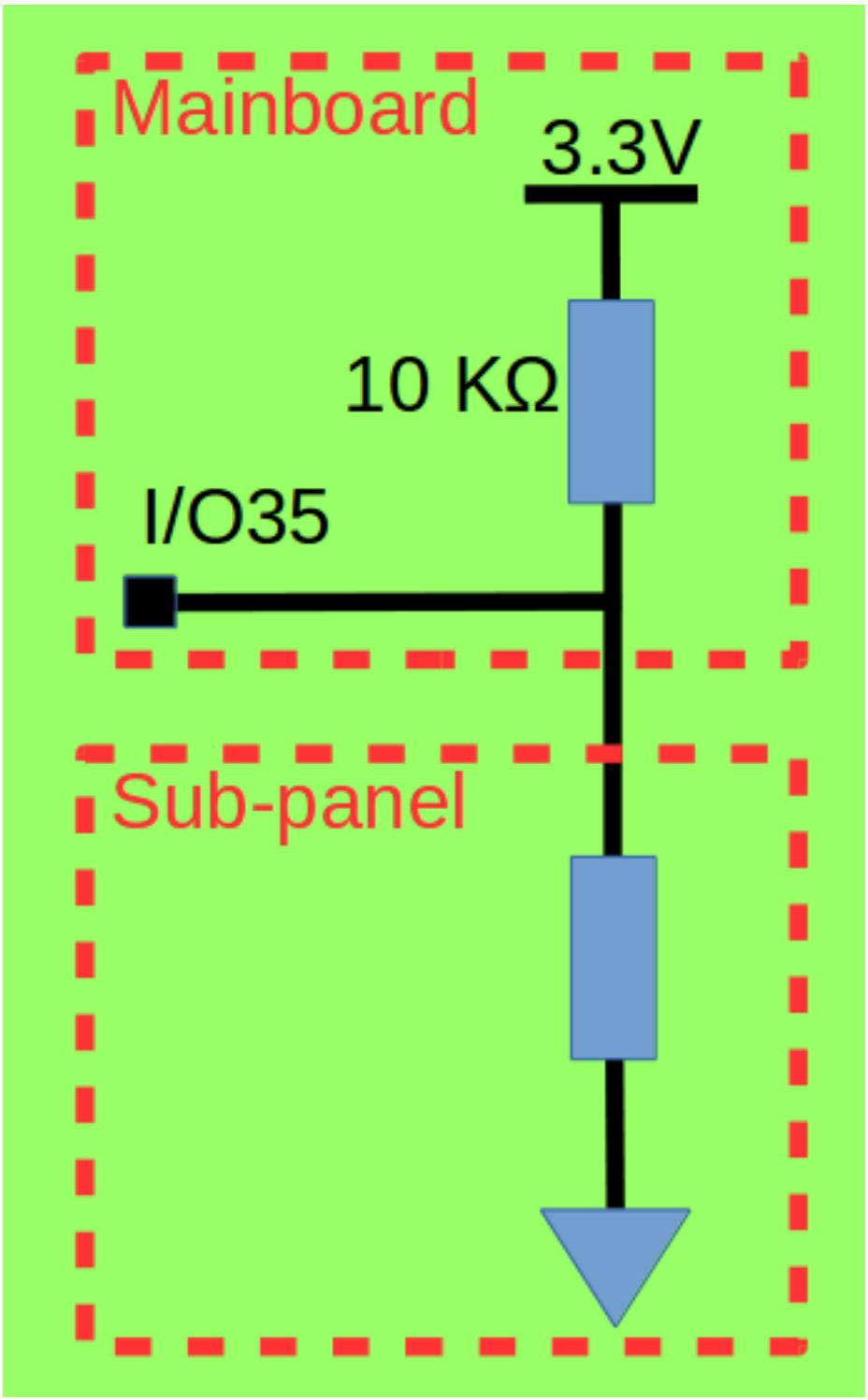


Fig. 11: Voltage Divider

The divided resistance on the motherboard is 10 KΩ. The table below lists the divided resistance on each daughterboard.

Application Programs

ESP32-Sense Project within ESP32 IoT Solution repository contains the application programs for ESP32-Sense Kit. The directory structure is shown below:

```
.
├── main
│   ├── evb_adc.c               //Identifies different daughterboards through ADC.␣
↪Sets unique ADC threshold for each daughterboard.
│   ├── evb.h                   //Configures settings for motherboard, including␣
↪touch threshold, ADC I/O, IIC I/O, etc.
│   ├── evb_led.cpp              //Initialization program of RGB LED.
│   ├── evb_seg_led.c           //Driver for digital tube.
│   ├── evb_touch_button.cpp    //Driver for touch button.
│   ├── evb_touch_wheel.cpp     //Driver for wheel slider.
│   ├── evb_touch_matrix.cpp    //Driver for matrix button.
│   ├── evb_touch_seq_slide.cpp //Driver for duplex slider.
│   ├── evb_touch_slide.cpp     //Driver for linear slider.
│   ├── evb_touch_spring.cpp    //Driver for spring button.
│   ├── Kconfig.projbuild
│   └── main.cpp                //Entry point.
├── Makefile
└── sdkconfig.defaults
```

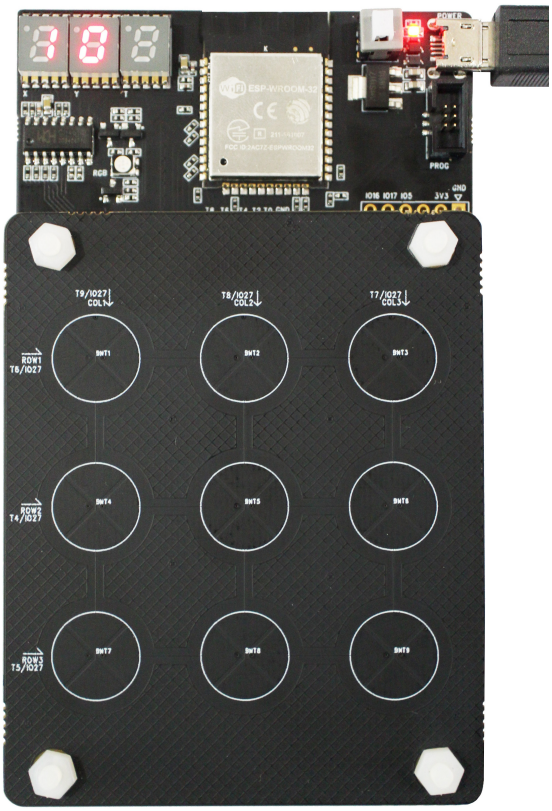
Configure Settings When using overlays of different thicknesses or materials, users need to reset the change rate of touch readings on each channel, that is, the sensitivity. This parameter is calculated from the pulse count value. The calculation formula is: (Non-touch value - Touch value) / Non-touch value, where "Non-touch value" refers to the pulse count value when there is no touch event, and "Touch value" refers to the pulse count value when a touch event occurs. Users need to take a measurement and obtain these two values. When the system is initialized, the touch threshold is automatically calculated from the change rate of touch readings. The touch threshold is directly proportional to the change rate.

When the change rate is set, users can write it into `evb.h` file.

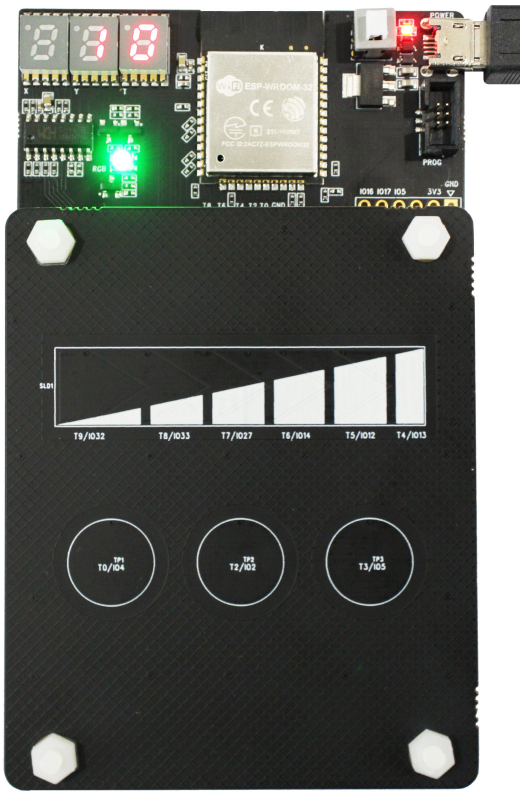
Demo



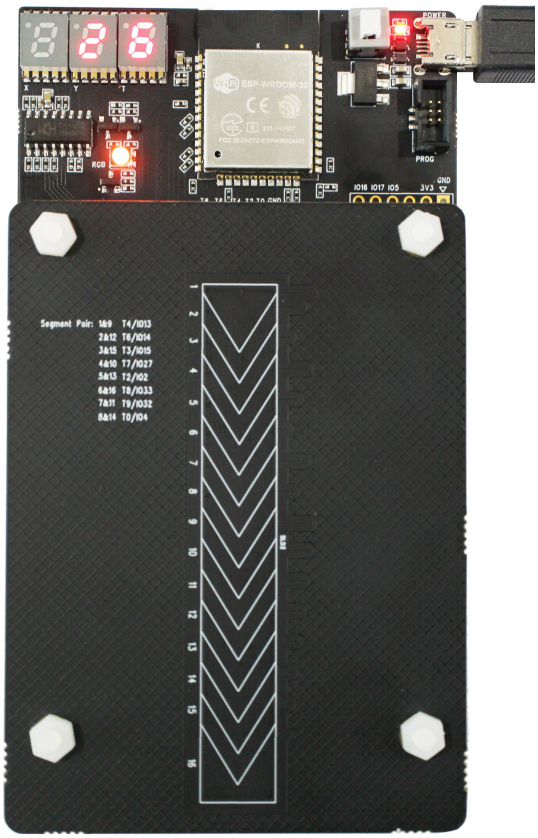
Spring Button



Matrix Button



Linear Slider



Duplex Slider



Submit Document Feedback

Related Resources

Please download the following documents from the HTML version of esp-dev-kits Documentation.

- **Schematic**
- ESP32-Sense-Kit Mainboard Schematic
- ESP32-Sense-Kit Subboard Schematic
- **PCB Layout**
- ESP32-Sense-Kit Mainboard PCB Layout
- ESP32-Sense-Kit Subboard PCB Layout
- **Set up Software Environment**
  - ESP-IDF is the SDK for ESP32. You can refer to Get Started for how to set up the ESP32 software environment.
  - ESP-Prog is the debugger for ESP32 that features download and debugging functions.
- **ESP32 IoT Solution**
  - ESP32 IoT Solution project is based on ESP-IDF and contains multiple projects.
  - ESP32-Sense Project contains the programs for ESP32-Sense Kit that can be downloaded to the development board to enable touch sensor function.
- **Hardware Manuals**
  - Please click ESP32-Sense Kit Reference Design to download the hardware resources including schematics, PCB reference design, BOM and other files.
- **Useful References**
  - Espressif website.
  - ESP32 programming guide: It hosts extensive documentation for ESP-IDF ranging from hardware guides to API reference.
  - ESP32 touch sensor design: It is the reference design manual for the ESP32 touch sensor system.
- **Technical Support**
  - If you need technical support regarding ESP32-Sense-Kit, please submit a new issue referring to the ESP32-Sense Project.
- **How to buy**
  - WeChat Account: espressif_systems.
  - Purchase consulting.

7.2 ESP32-MeshKit-Sense

ESP32-MeshKit-Sense is a development board with an ESP32 module at its core. It features peripherals, such as a temperature and humidity sensor, an ambient light sensor, etc. The board can be interfaced with screens. The board is mainly used to detect the current consumption of ESP32 modules in a normal operation state or in sleep mode, when connected to different peripherals.

7.2.1 ESP32-MeshKit-Sense

Overview

ESP32-MeshKit-Sense is a development board with an ESP32 module at its core. It features peripherals, such as a temperature and humidity sensor, an ambient light sensor, etc. The board can be interfaced with screens. The board is mainly used to detect the current consumption of ESP32 modules in a normal operation state or in sleep mode, when connected to different peripherals.

For more information on ESP32, please refer to ESP32 Datasheet.

Block Diagram and PCB Layout

Block Diagram The figure below shows the block diagram of ESP32.

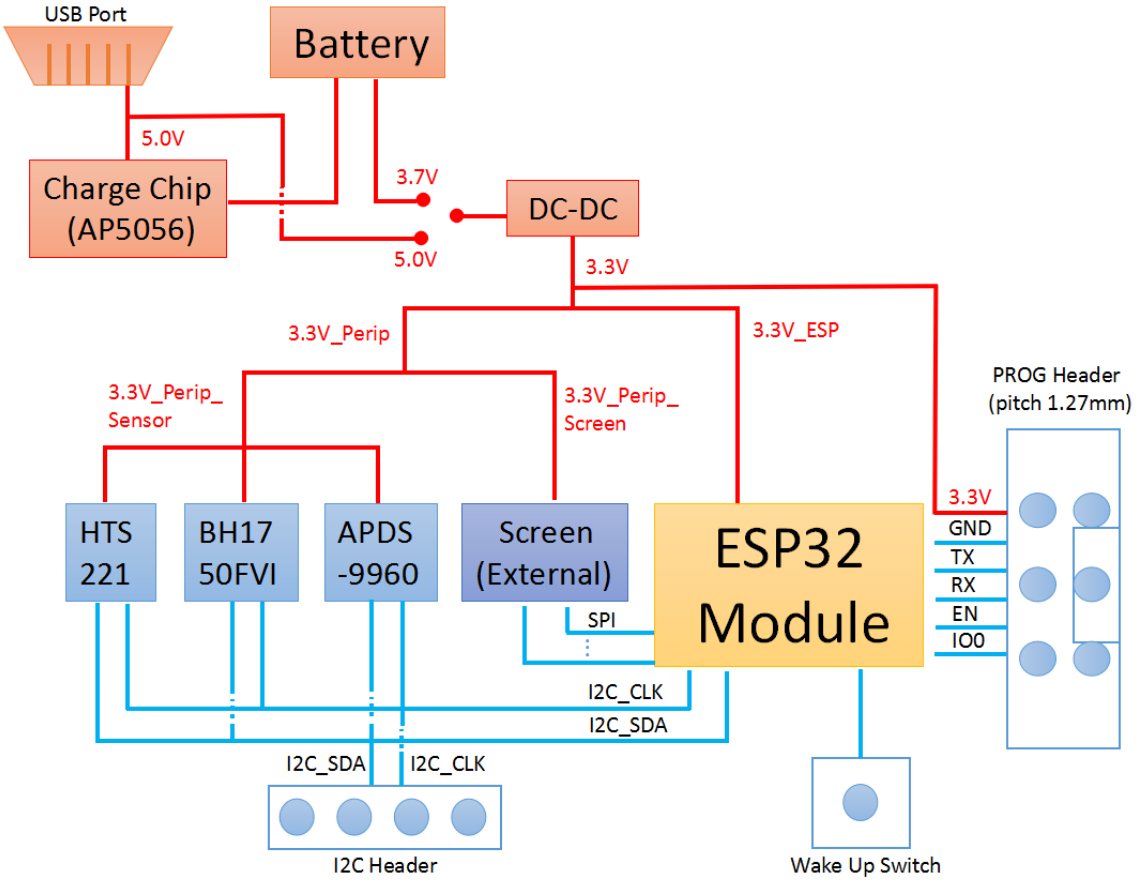


Fig. 12: ESP32 Block Diagram

PCB Layout The figure below shows the layout of ESP32-MeshKit-Sense PCB.

Functional Descriptions of PCB Layout are shown in the following table:

PCB Elements	Description
EXT5V	5 V input from USB
CH5V	input from the electrical charging chip
CHVBA	output from the electrical charging chip
VBA	Connects to the positive electrode of the battery
SUFVCC	When the switch is toggled to the ” ON” position, it is connected to the power input. When the switch is toggled to the ” OFF” position, the power supply is disconnected.
DCVCC	Input from power management chip DC-DC
3.3V	3.3 V power output from power supply management chip
3.3V_PER	3.3 V power supply for all peripherals
3.3V_ESP	3.3 V power supply for all ESP32 modules
3.3V_SEN	3.3 V power supply for the three on-board sensors
3.3V_SCR	3.3 V power supply for the off-board screen
Charge	Battery charging indicator, D5 is a red light, indicating that charging is undergoing; D6 is a green light, indicating that charging is complete.
Sensor	Power indicator, indicating that 3.3V_Perip_Sensor is enabled
Screen	Power indicator, indicates that 3.3V_Perip_Screen is enabled
WiFi / IO15	Signal indicator, indicating that Wi-Fi connection is working properly
Network / IO4	Signal indicator, indicating the board is properly connected to the server

Functional Modules

This chapter mainly introduces each functional module (interface) and the hardware schematics for them.

Power Supply Management Module

Power Supply Management Module The development board can be powered by battery and the AP5056 power supply management chip can be used to charge the battery. The AP5056 is a complete constant current constant voltage linear charger for single cell lithium-ion batteries. It has 4.2 V of preset charge voltage and 1 A of programmable charge current.

When both the USB power supply and the battery power supply are available, the system selection of power supply will be: VBUS is high, Q4 is in cut-off state, VBAT (battery power) is automatically cut off from the system power supply, and the USB supplies power for the system.

The figure below shows the schematics for USB/BAT power supply management.

Power Supply Management for Peripherals First of all, the input from the USB or BAT is converted by the power management chip into a 3.3 V voltage to power the circuit. The power management chip on the board is ETA3425, which has an output voltage of 3.3 V and a maximum output current of 600 mA.

The figure below shows the schematics for peripheral power supply.

The main VDD33 circuit has two branches:

- ESP32_VDD33, used to power the ESP32 module module
- VDD33_PeriP, used to power all peripherals.

The connection between them can be controlled via the pin header and jumper cap. The figure below shows the schematics for ESP32_VDD33.

The VDD33_PeriP branch circuit also has two sub-branches

- VDD33_PeriP_Screen, dedicated power supply for the external screen
- VDD33_PeriP_Sensor, power supply for the three sensors

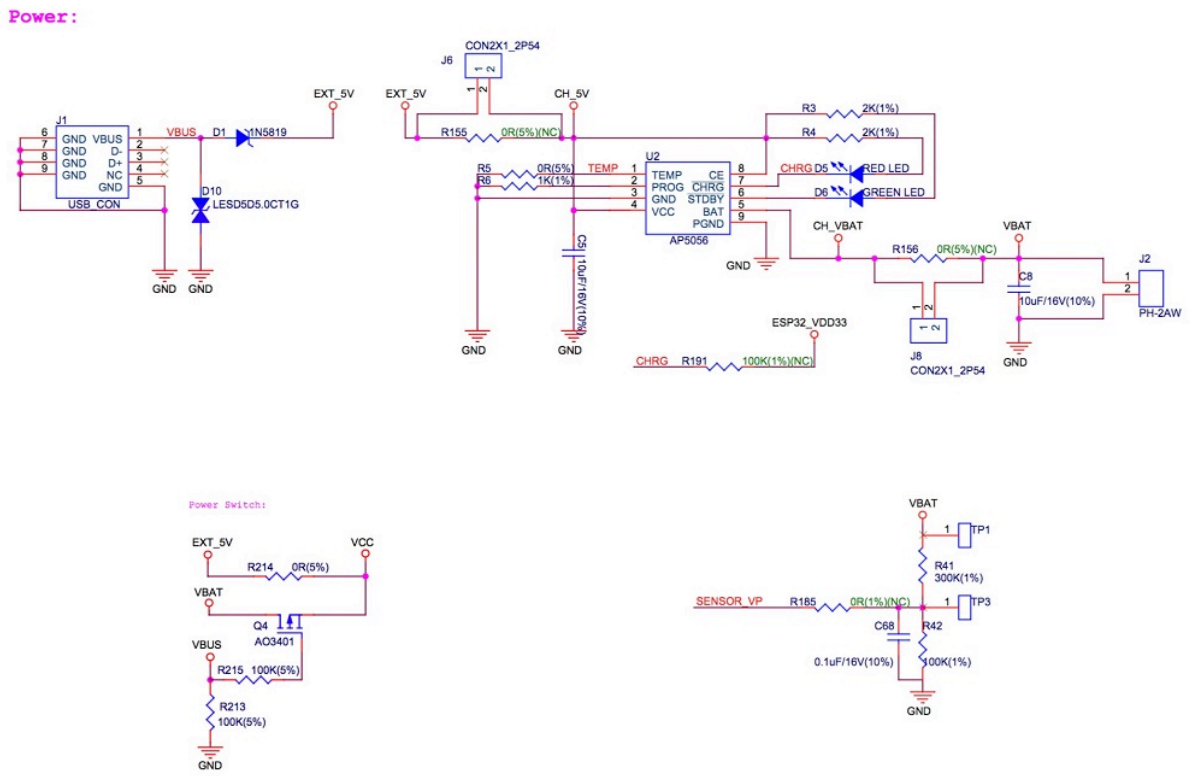
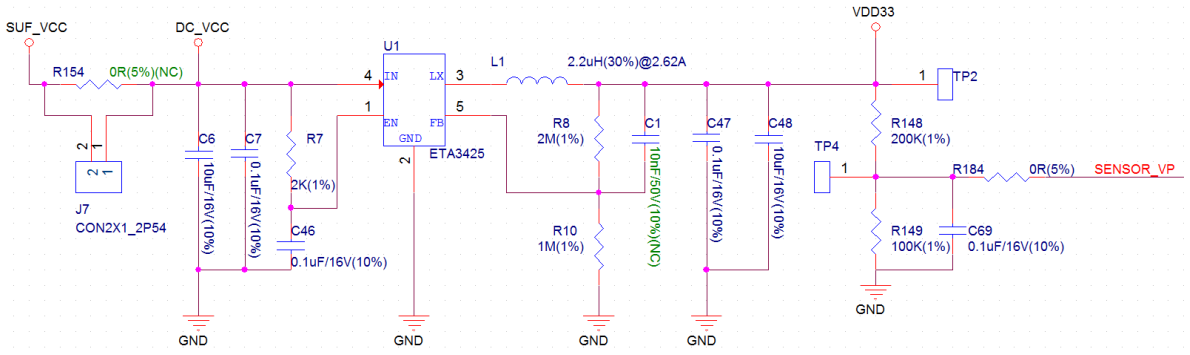


Fig. 13: USB/BAT Power Supply Management Schematics




```
Notes:
1.Vout=1.13*(1+R1/R2)=3.39V;
  R2 is recommended to be 1M ohm for low standby current.
```

Fig. 14: Peripheral Power Supply Schematics

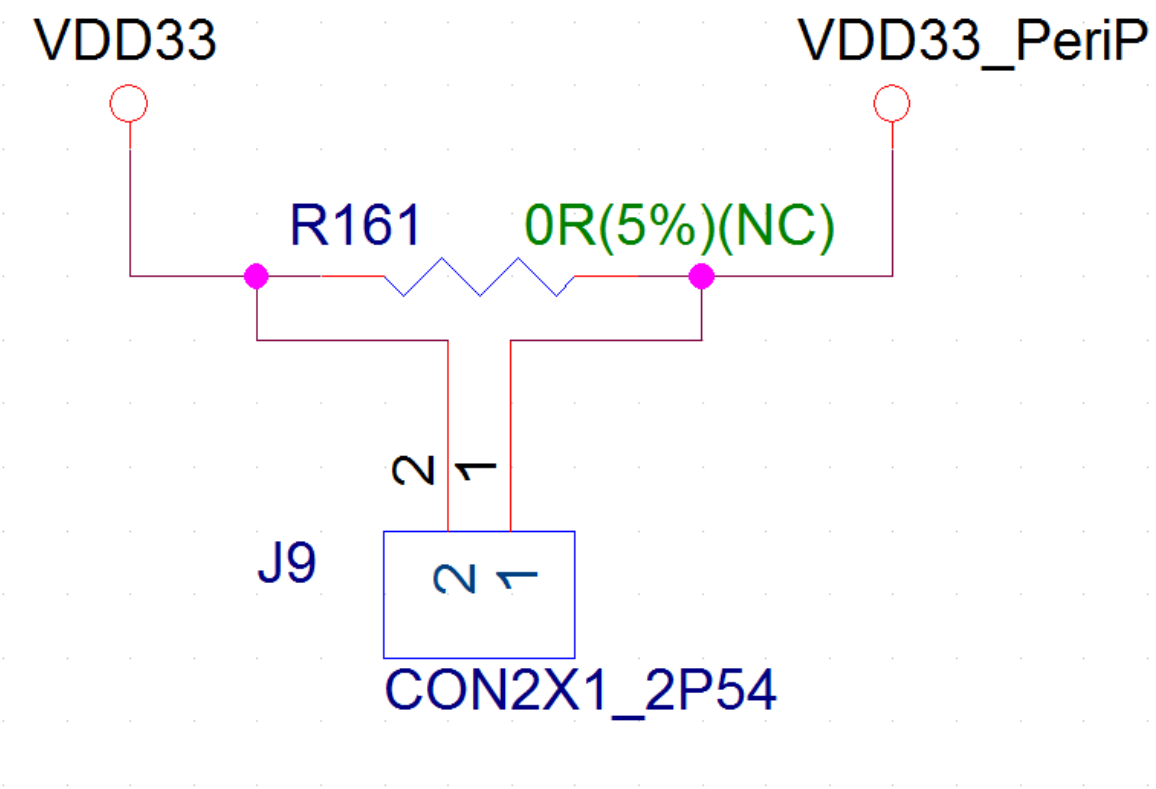


Fig. 15: ESP32_VDD33 Schematics

The connection of the two can be controlled by the module GPIO+MOS. The figure below shows the schematics for VDD33_PeriP.

Boot & UART The development board is integrated with a PROG Header, which can be connected to a ESP-PROG development board via a cable. Users can then connect the micro USB of the ESP-PROG development board to a PC for ESP32-MeshKit-Sense firmware download and debugging.

The figure below shows the schematics for Boot & UART Circuit.

Module for Wakeup from Sleep The board has a button connected to the pin IO34, which is a pin in the RTC domain. When the chip is in sleep, pressing the button will wake up ESP32.

The figure below shows the schematics for wakeup-from-sleep module.

External Screens The development board is integrated with a screen connector that can connect different external screens to the board via cables.

The figure below shows the schematics for external screens.

Sensors

Temperature and Humidity Sensor The HTS221 is an ultra-compact sensor for relative humidity and temperature. A 3.3 V power supply and I2C interface on the board are dedicated to HTS221.

The figure below shows the schematics for the temperature and humidity sensor.

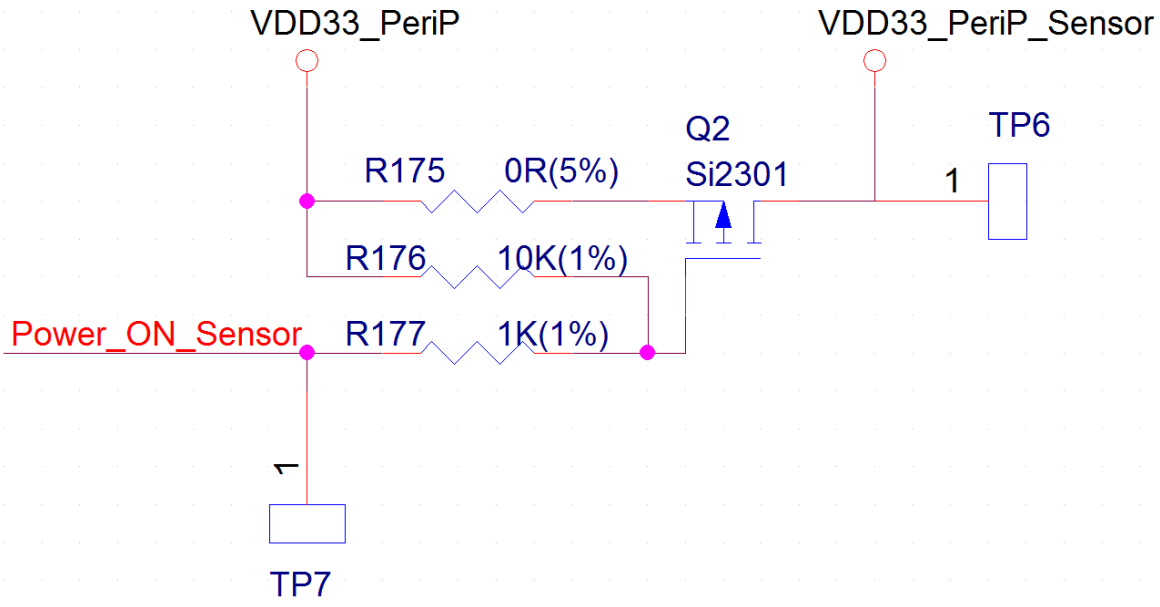


Fig. 16: VDD33_PeriP Schematics

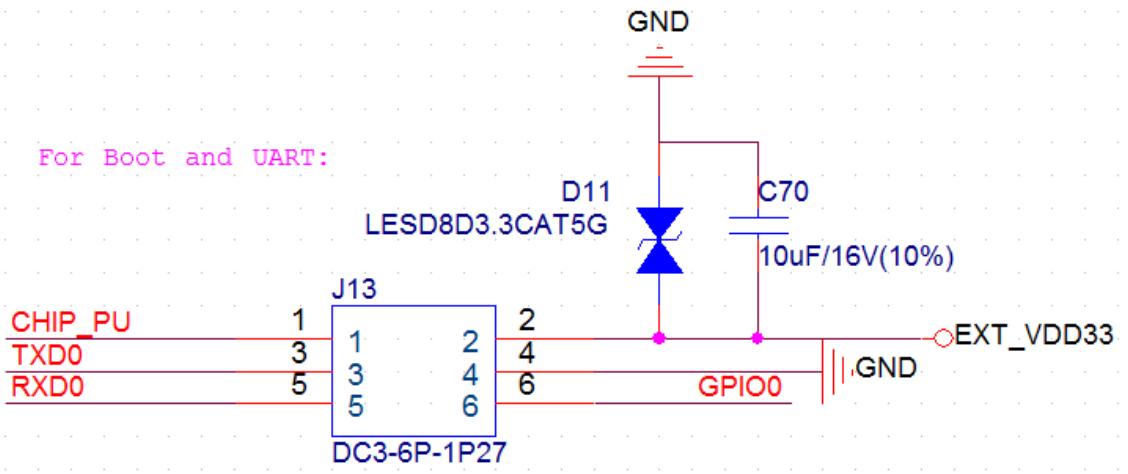


Fig. 17: Boot & UART Circuit

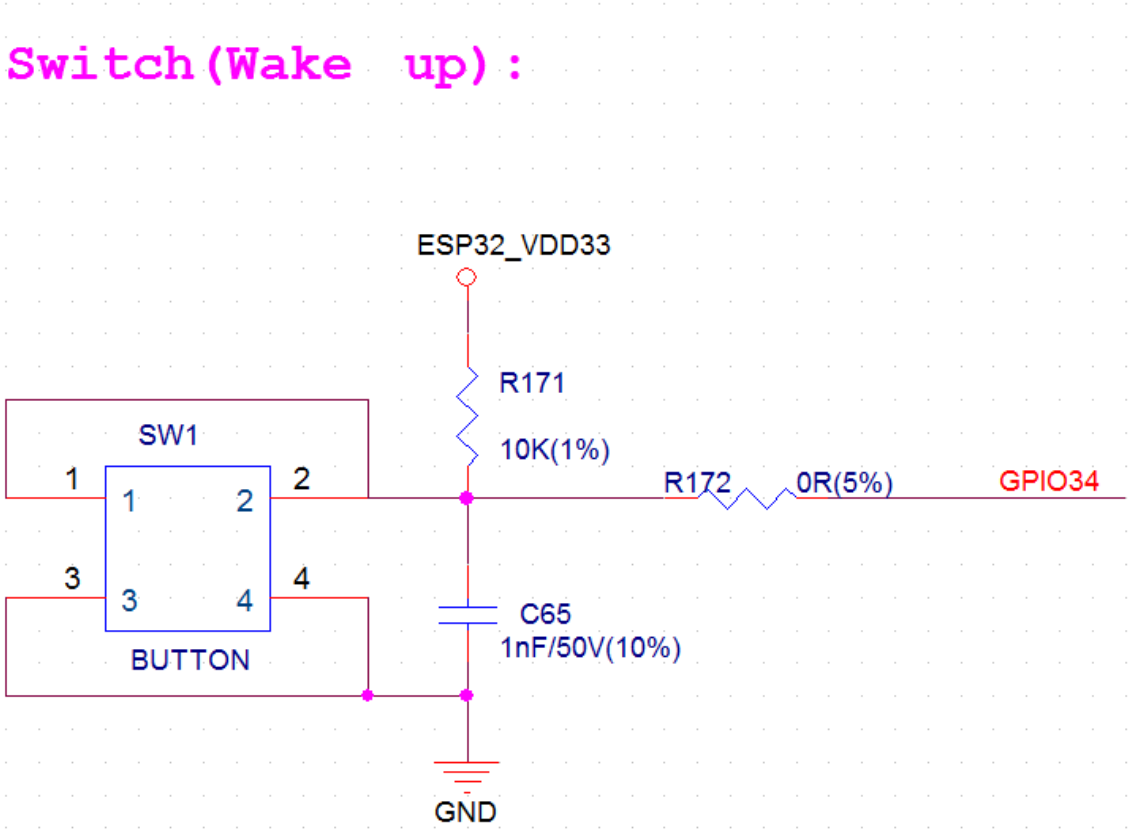


Fig. 18: Wake-from-Sleep Module Schematics

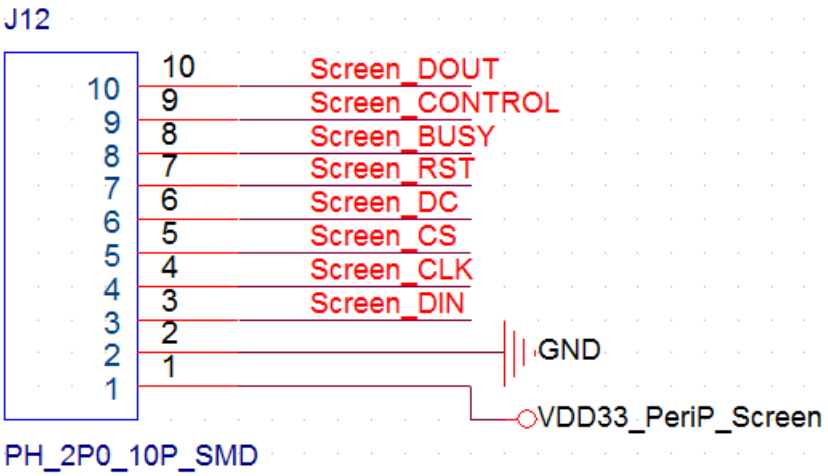


Fig. 19: Schematics for External Screens

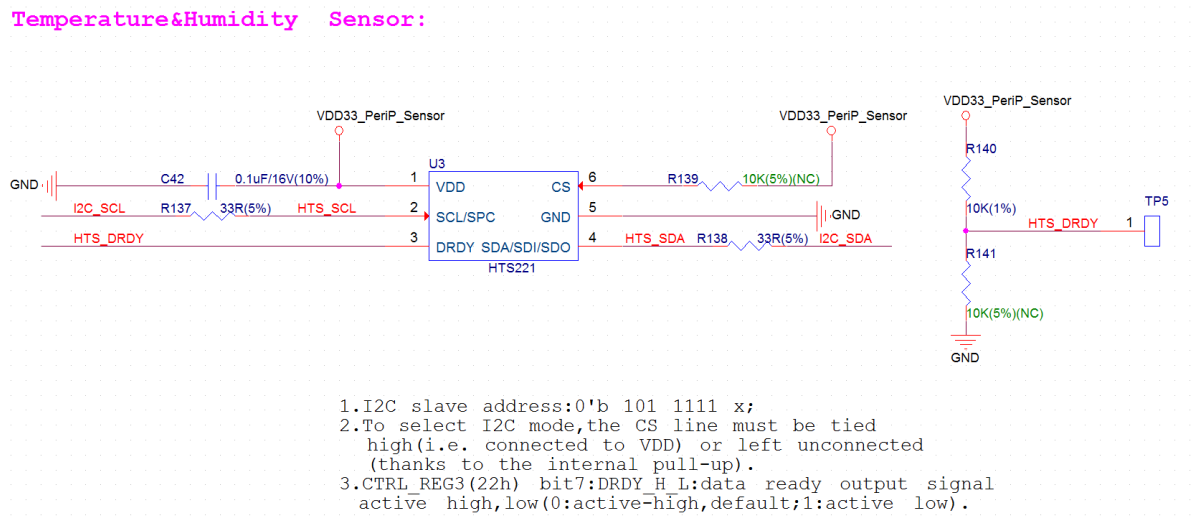


Fig. 20: Temperature and Humidity Sensor Schematics

Ambient Light Sensor The BH1750FVI is a digital ambient light sensor. A 3.3 V power supply and I2C interface on the board are dedicated to HTS221.

The figure below shows the schematics for the ambient light sensor.

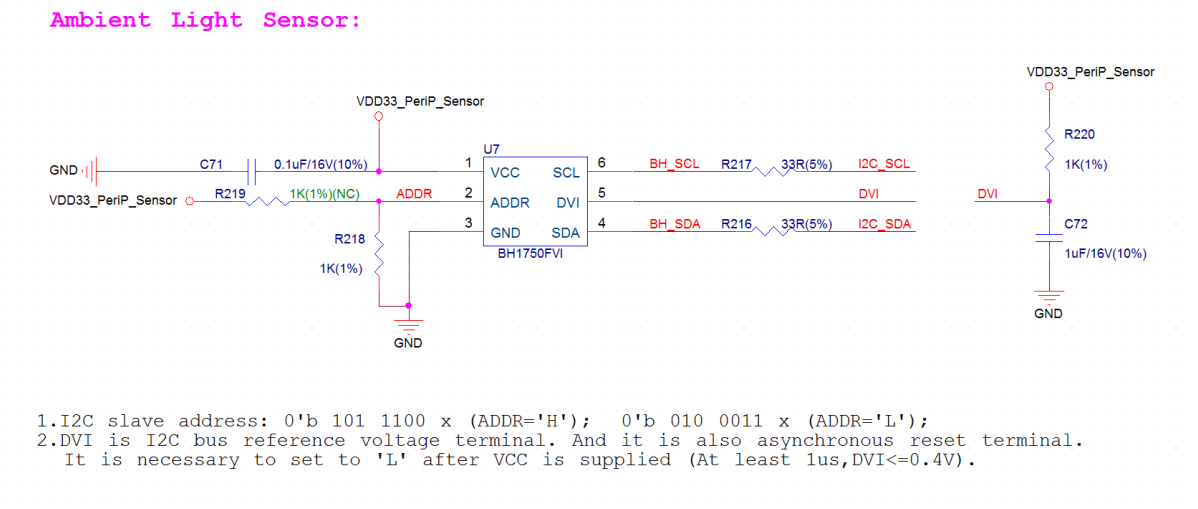


Fig. 21: Ambient Light Sensor Schematics

Ambient Brightness Sensor The APDS-9960 is a ambient brightness sensor featuring advanced gesture detection, proximity detection, digital Ambient Light Sense (ALS) and Color Sense (RGBC). It also incorporates an IR LED driver. The development board uses 3.3V power supply and I2C interface. It should be noted that this device is not surface-mounted by default.

The figure below shows the schematics for the ambient brightness sensor.

Example

See esp-mdf/examples/development_kit/sense.

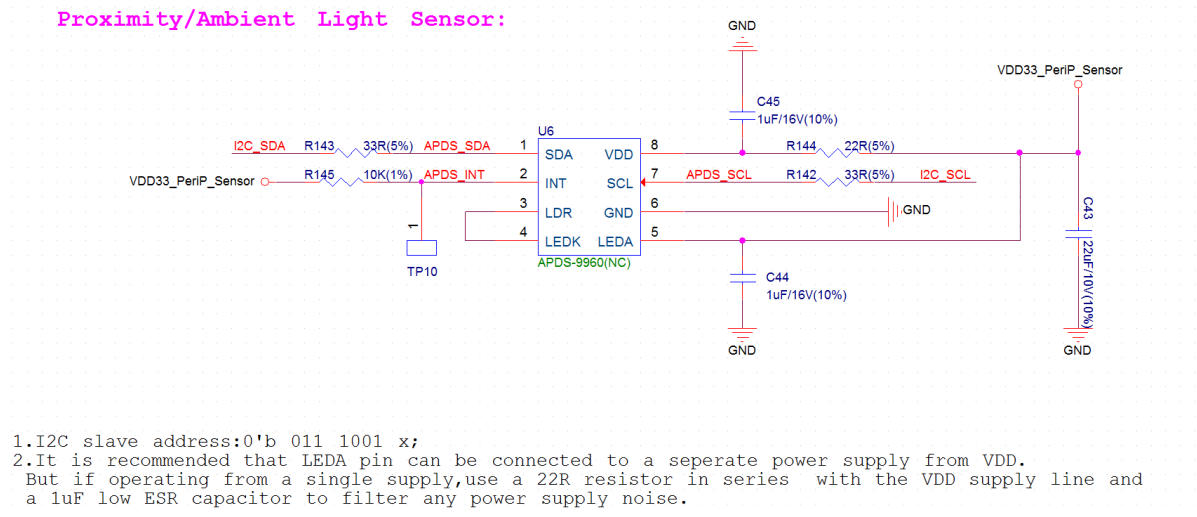


Fig. 22: Ambient Brightness Sensor Schematics

Related Documents

Please download the following documents from the HTML version of esp-dev-kits Documentation.

- ESP32-MeshKit-Sense Schematic
- ESP32-MeshKit-Sense PCB Layout

7.3 ESP-WROVER-KIT

ESP-WROVER-KIT is an ESP32-based development board produced by Espressif. ESP-WROVER-KIT features the ESP32-WROVER-E module, LCD screen, and microSD card slot.

7.3.1 ESP-WROVER-KIT v4.1 Getting Started Guide

The older version:

ESP-WROVER-KIT v2 Getting Started Guide

ESP-WROVER-KIT v3 Getting Started Guide

This guide shows how to get started with the ESP-WROVER-KIT v4.1 development board and also provides information about its functionality and configuration options.

What You Need

- *ESP-WROVER-KIT v4.1 board*
- USB 2.0 cable (A to Micro-B)
- Computer running Windows, Linux, or macOS

You can skip the introduction sections and go directly to Section *Start Application Development*.

Overview

ESP-WROVER-KIT is an ESP32-based development board produced by Espressif.

ESP-WROVER-KIT features the following integrated components:

- ESP32-WROVER-E module
- LCD screen
- microSD card slot

Another distinguishing feature is the embedded FTDI FT2232HL chip, an advanced multi-interface USB bridge. This chip enables to use JTAG for direct debugging of ESP32 through the USB interface without a separate JTAG debugger. ESP-WROVER-KIT makes development convenient, easy, and cost-effective.

Most of the ESP32 I/O pins are broken out to the board's pin headers for easy access.

Note: ESP32's GPIO16 and GPIO17 are used as chip select and clock signals for PSRAM. By default, the two GPIOs are not broken out to the board's pin headers in order to ensure reliable performance.

Functionality Overview

The block diagram below shows the main components of ESP-WROVER-KIT and their interconnections.

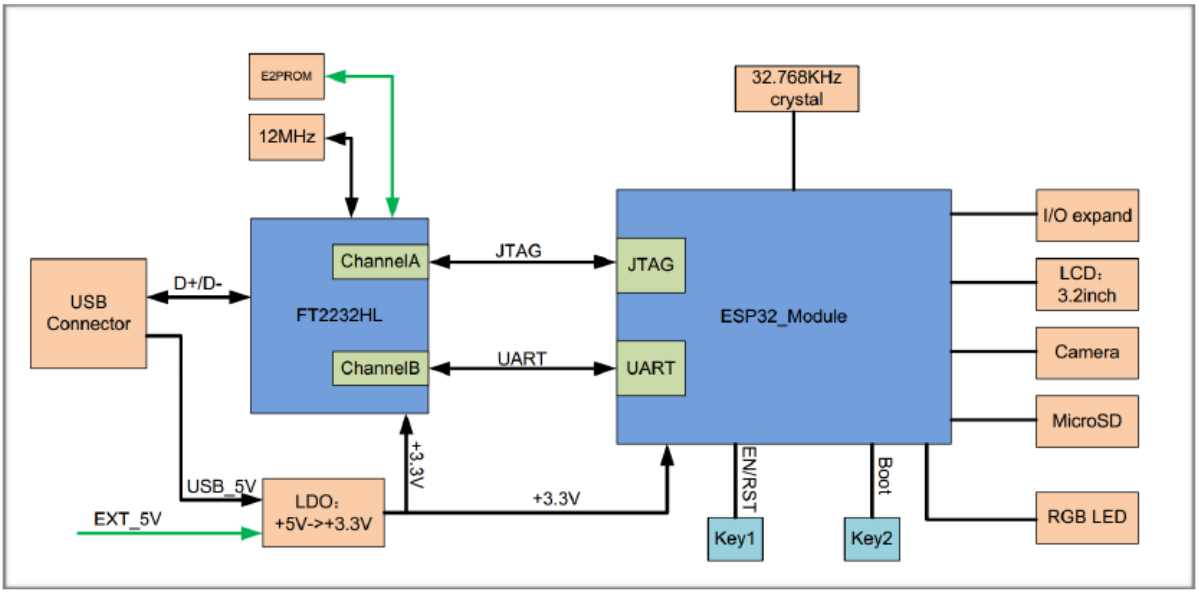


Fig. 23: ESP-WROVER-KIT block diagram

Functional Description

The following two figures and the table below describe the key components, interfaces, and controls of the ESP-WROVER-KIT board.

The table below provides description in the following manner:

- Starting from the first picture's top right corner and going clockwise
- Then moving on to the second picture

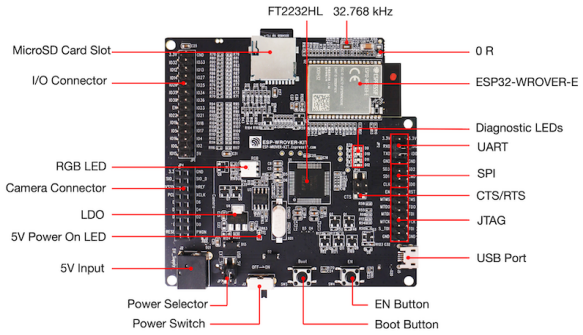


Fig. 24: ESP-WROVER-KIT board layout - front

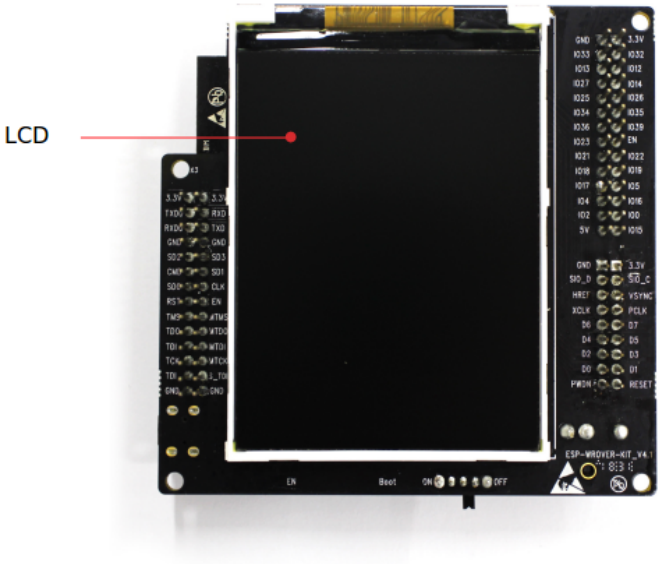
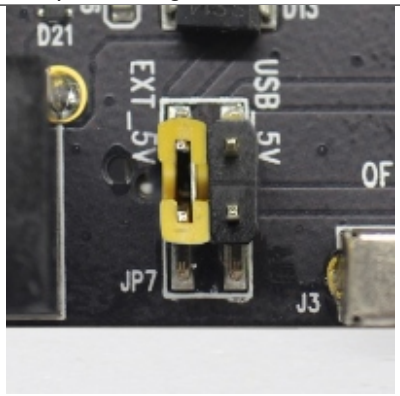
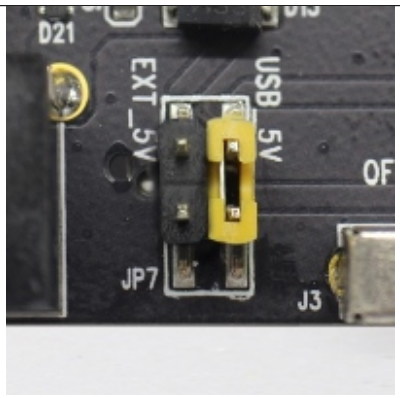
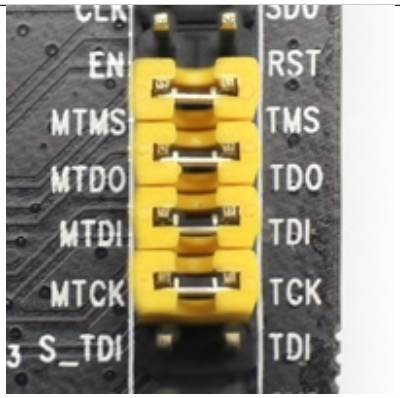
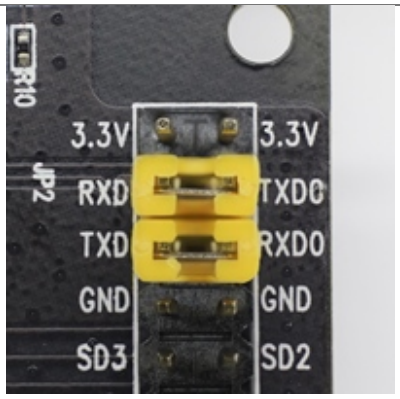
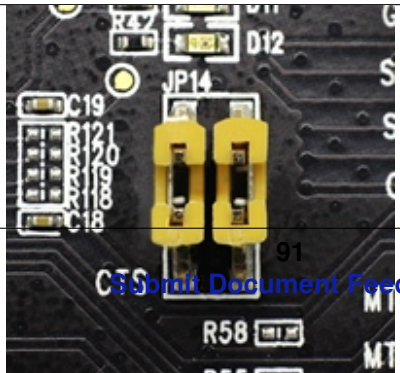


Fig. 25: ESP-WROVER-KIT board layout - back

Key Component	Description
FT2232HL	The FT2232HL chip serves as a multi-protocol USB-to-serial bridge which can be programmed and controlled via USB to provide communication with ESP32. FT2232HL also features USB-to-JTAG interface which is available on channel A of the chip, while USB-to-serial is on channel B. The FT2232HL chip enhances user-friendliness in terms of application development and debugging. See ESP-WROVER-KIT v4.1 schematic.
32.768 kHz	External precision 32.768 kHz crystal oscillator serves as a clock with low-power consumption while the chip is in Deep-sleep mode.
0R	Zero-ohm resistor intended as a placeholder for a current shunt, can be desoldered or replaced with a current shunt to facilitate the measurement of ESP32' s current consumption in different modes.
ESP32-WROVER-E module	This ESP32 module features 64-Mbit PSRAM for flexible extended storage and data processing capabilities.
Diagnostic LEDs	Four red LEDs connected to the GPIO pins of FT2232HL. Intended for future use.
UART	Serial port. The serial TX/RX signals of FT2232HL and ESP32 are broken out to the inward and outward sides of JP2 respectively. By default, these pairs of pins are connected with jumpers. To use ESP32' s serial interface, remove the jumpers and connect another external serial device to the respective pins.
SPI	By default, ESP32 uses its SPI interface to access flash and PSRAM memory inside the module. Use these pins to connect ESP32 to another SPI device. In this case, an extra chip select (CS) signal is needed. Please note that the voltage of this interface is 3.3 V.
CTS/RTS	Serial port flow control signals: the pins are not connected to the circuitry by default. To enable them, short the respective pins of JP14 with jumpers.
JTAG	JTAG interface. JTAG signals of FT2232HL and ESP32 are broken out to the inward and outward sides of JP2 respectively. By default, these pairs of pins are disconnected. To enable JTAG, short the respective pins with jumpers as shown in Section *Setup Options*.
USB Port	USB interface. Power supply for the board as well as the communication interface between a computer and the board.
EN Button	Reset button.
BOOT Button	Download button. Holding down **Boot** and then pressing **EN** initiates Firmware Download mode for downloading firmware through the serial port.
Power Switch	Power On/Off Switch. Toggling toward the **Boot** button powers the board on, toggling away from **Boot** powers the board off.
Power Selector	Power supply selector interface. The board can be powered either via USB or via the 5V Input interface. Select the power source with a jumper. For more details, see Section *Setup Options*, jumper header JP7.
5V Input	5V power supply interface for a standard coaxial power connector, 5.5 x 2.1 mm, center positive. This interface can be more convenient when the board is operating autonomously (not connected to a computer).
5V Power On LED	This red LED turns on when power is supplied to the board, either from **USB** or **5V Input**.
LDO	NCP1117(1A). 5V-to-3.3V LDO. NCP1117 can provide a maximum current of 1A. The LDO on the board has a fixed output voltage, but the user can install an LDO with adjustable output voltage. For details, please refer to ESP-WROVER-KIT v4.1 schematic.
Camera Connector	Camera interface, a standard OV7670 camera module.
RGB LED	Red, green and blue (RGB) light emitting diodes (LEDs), can be controlled by pulse width modulation (PWM).
I/O Connector	All the pins on the ESP32 module are broken out to pin headers. You can program ESP32 to enable multiple functions, such as PWM, ADC, DAC, I2C, I2S, SPI, etc.
microSD Card Slot	Useful for developing applications that access microSD card for data storage and retrieval.
LCD	Support for mounting and interfacing a 3.2" SPI (standard 4-wire Serial Peripheral Interface) LCD, as shown in figure *ESP-WROVER-KIT board layout - back*.

Setup Options

There are three jumper blocks available to set up the board functionality. The most frequently required options are listed in the table below.

Header	Jumper Setting	Description of Functionality
JP7		Power ESP-WROVER-KIT via an external power supply
JP7		Power ESP-WROVER-KIT via USB
JP2		Enable JTAG functionality
JP2		Enable UART communication
		

Allocation of ESP32 Pins

Some pins or terminals of ESP32 are allocated for use with the onboard or external hardware. If that hardware is not used, e.g., nothing is plugged into the Camera (JP4) header, then these GPIOs can be used for other purposes.

Some of the pins, such as GPIO0 or GPIO2, have multiple functions and some of them are shared among onboard and external peripheral devices. Certain combinations of peripherals cannot work together. For example, it is not possible to do JTAG debugging of an application that is using SD card, because several pins are shared by JTAG and the SD card slot.

In other cases, peripherals can coexist under certain conditions. This is applicable to, for example, LCD screen and SD card that share only a single pin GPIO21. This pin is used to provide D/C (Data/Control) signal for the LCD as well as the Card Detect signal read from the SD card slot. If the card detect functionality is not essential, then it may be disabled by removing R167, so both LCD and SD may operate together.

For more details on which pins are shared among which peripherals, please refer to the table in the next section.

Main I/O Connector/JP1 The JP1 connector consists of 14x2 male pins whose functions are shown in the middle two "I/O" columns of the table below. The two "Shared With" columns on both sides describe where else on the board a certain GPIO is used.

Shared With	I/O	I/O	Shared With
n/a	3.3V	GND	n/a
NC/XTAL	IO32	IO33	NC/XTAL
JTAG, microSD	IO12	IO13	JTAG, microSD
JTAG, microSD	IO14	IO27	Camera
Camera	IO26	IO25	Camera, LCD
Camera	IO35	IO34	Camera
Camera	IO39	IO36	Camera
JTAG	EN	IO23	Camera, LCD
Camera, LCD	IO22	IO21	Camera, LCD, microSD
Camera, LCD	IO19	IO18	Camera, LCD
Camera, LCD	IO5	IO17	PSRAM
PSRAM	IO16	IO4	LED, Camera, microSD
Camera, LED, Boot	IO0	IO2	LED, microSD
JTAG, microSD	IO15	5V	

Legend:

- NC/XTAL - *32.768 kHz Oscillator*
- JTAG - *JTAG/JP2*
- Boot - Boot button/SW2
- Camera - *Camera/JP4*
- LED - *RGB LED*
- microSD - *microSD Card/J4*
- LCD - *LCD/U5*
- PSRAM - ESP32-WROVER-E's PSRAM

32.768 kHz Oscillator

.	ESP32 Pin
1	GPIO32
2	GPIO33

Note: Since GPIO32 and GPIO33 are connected to the oscillator by default, they are not connected to the JP1 I/O connector to maintain signal integrity. This allocation may be changed from the oscillator to JP1 by desoldering the zero-ohm resistors from positions R11 or R23 and re-soldering them to positions R12 or R24.

SPI Flash/JP2

.	ESP32 Pin
1	CLK/GPIO6
2	SD0/GPIO7
3	SD1/GPIO8
4	SD2/GPIO9
5	SD3/GPIO10
6	CMD/GPIO11

Note: SPI Flash pins are used to access the internal flash memory. Therefore, they are not available to connect external SPI devices. Those pins are exposed for monitoring or for advanced usage only.

Important: The module's flash bus is connected to the jumper block JP2 through zero-ohm resistors R140 ~ R145. If the flash memory needs to operate at the frequency of 80 MHz, for reasons such as improving the integrity of bus signals, you can desolder these resistors to disconnect the module's flash bus from the pin header JP2.

JTAG/JP2

.	ESP32 Pin	JTAG Signal
1	EN	TRST_N
2	MTMS/GPIO14	TMS
3	MTDO/GPIO15	TDO
4	MTDI/GPIO12	TDI
5	MTCK/GPIO13	TCK

Camera/JP4

.	ESP32 Pin	Camera Signal
1	n/a	3.3V
2	n/a	Ground
3	GPIO27	SIO_C/SCCB Clock
4	GPIO26	SIO_D/SCCB Data
5	GPIO25	VSYNC/Vertical Sync
6	GPIO23	HREF/Horizontal Reference
7	GPIO22	PCLK/Pixel Clock
8	GPIO21	XCLK/System Clock
9	GPIO35	D7/Pixel Data Bit 7
10	GPIO34	D6/Pixel Data Bit 6
11	GPIO39	D5/Pixel Data Bit 5
12	GPIO36	D4/Pixel Data Bit 4
13	GPIO19	D3/Pixel Data Bit 3
14	GPIO18	D2/Pixel Data Bit 2
15	GPIO5	D1/Pixel Data Bit 1
16	GPIO4	D0/Pixel Data Bit 0
17	GPIO0	RESET/Camera Reset
18	n/a	PWDN/Camera Power Down

- Signals D0 .. D7 denote camera data bus

RGB LED

.	ESP32 Pin	RGB LED
1	GPIO0	Red
2	GPIO2	Green
3	GPIO4	Blue

microSD Card

.	ESP32 Pin	microSD Signal
1	MTDI/GPIO12	DATA2
2	MTCK/GPIO13	CD/DATA3
3	MTDO/GPIO15	CMD
4	MTMS/GPIO14	CLK
5	GPIO2	DATA0
6	GPIO4	DATA1
7	GPIO21	Card Detect

LCD/U5

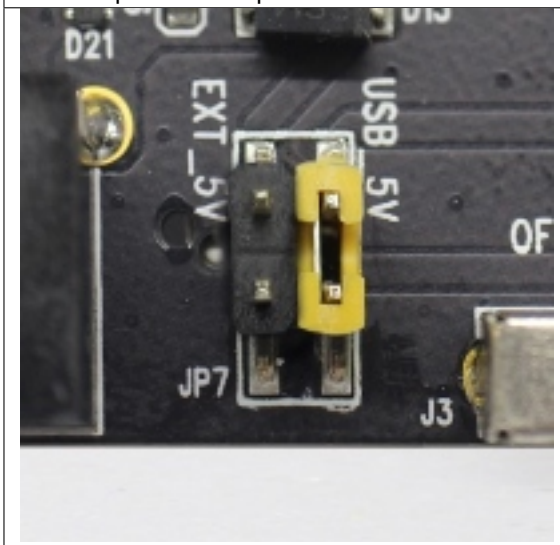
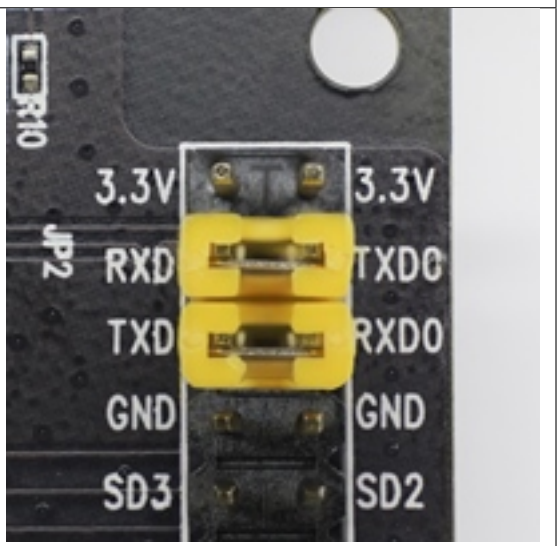
.	ESP32 Pin	LCD Signal
1	GPIO18	RESET
2	GPIO19	SCL
3	GPIO21	D/C
4	GPIO22	CS
5	GPIO23	SDA
6	GPIO25	SDO
7	GPIO5	Backlight

Start Application Development

Before powering up your ESP-WROVER-KIT, please make sure that the board is in good condition with no obvious signs of damage.

Initial Setup Please set only the following jumpers shown in the pictures below:

- Select USB as the power source using the jumper block JP7.
- Enable UART communication using the jumper block JP2.

Power up from USB port	Enable UART communication
	

Do not install any other jumpers.

Turn the **Power Switch** to ON, and the **5 V Power On LED** should light up.

Now to Development After that, proceed to ESP-IDF Get Started, which will quickly help you set up the development environment then flash an application example onto your board.

A Board Support Package can be found in ESP Component Registry.

The application examples that use some hardware specific to your ESP-WROVER-KIT can be found below.

- On-board LCD example
- SD card slot example
- Camera connector example

Related Documents

- ESP-WROVER-KIT v4.1 schematic (PDF)
- ESP-WROVER-KIT v4.1 layout (DXF) may be opened online with Autodesk Viewer
- ESP32 Datasheet (PDF)
- ESP32-WROVER-E Datasheet (PDF)

ESP-WROVER-KIT v2 Getting Started Guide

New version available: *ESP-WROVER-KIT v4.1 Getting Started Guide*

This guide shows how to get started with the ESP-WROVER-KIT v2 development board and also provides information about its functionality and configuration options.

What You Need

- ESP-WROVER-KIT v2 board
- USB 2.0 cable (A to Micro-B)
- Computer running Windows, Linux, or macOS

You can skip the introduction sections and go directly to Section *Start Application Development*.

Overview ESP-WROVER-KIT is an ESP32-based development board produced by Espressif. This board features an integrated LCD screen and microSD card slot.

ESP-WROVER-KIT comes with the following ESP32 modules:

- ESP32-WROOM-32
- ESP32-WROVER series

Its another distinguishing feature is the embedded FTDI FT2232HL chip - an advanced multi-interface USB bridge. This chip enables to use JTAG for direct debugging of ESP32 through the USB interface without a separate JTAG debugger. ESP-WROVER-KIT makes development convenient, easy, and cost-effective.

Most of the ESP32 I/O pins are broken out to the board's pin headers for easy access.

Note: The version with the ESP32-WROVER module uses ESP32's GPIO16 and GPIO17 as chip select and clock signals for PSRAM. By default, the two GPIOs are not broken out to the board's pin headers in order to ensure reliable performance.

Functionality Overview The block diagram below shows the main components of ESP-WROVER-KIT and their interconnections.

Functional Description The following two figures and the table below describe the key components, interfaces, and controls of the ESP-WROVER-KIT board.

The table below provides description in the following manner:

- Starting from the first picture's top right corner and going clockwise
- Then moving on to the second picture

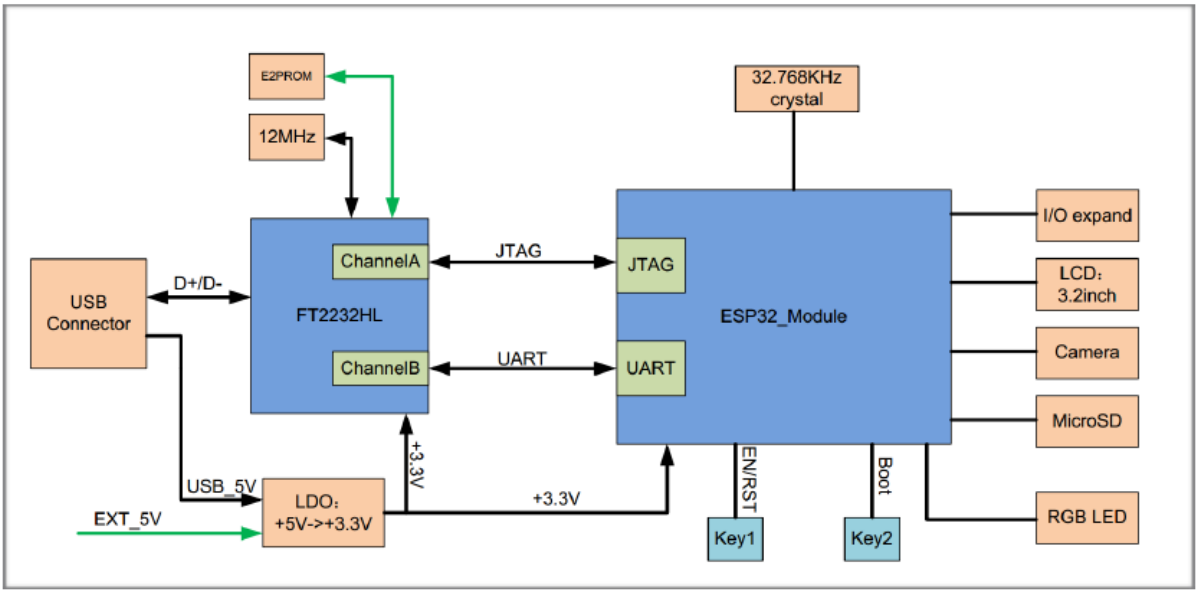


Fig. 26: ESP-WROVER-KIT block diagram

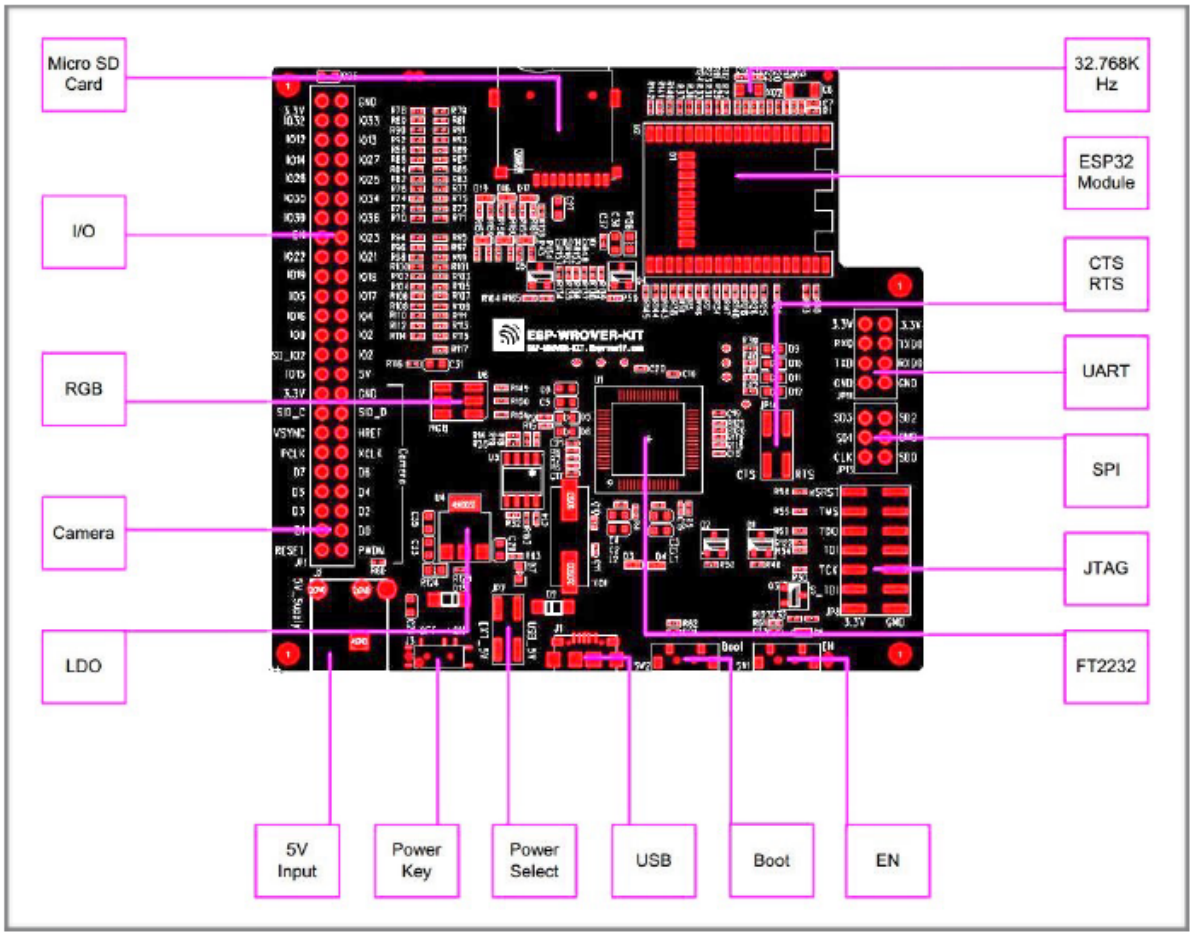


Fig. 27: ESP-WROVER-KIT board layout - front

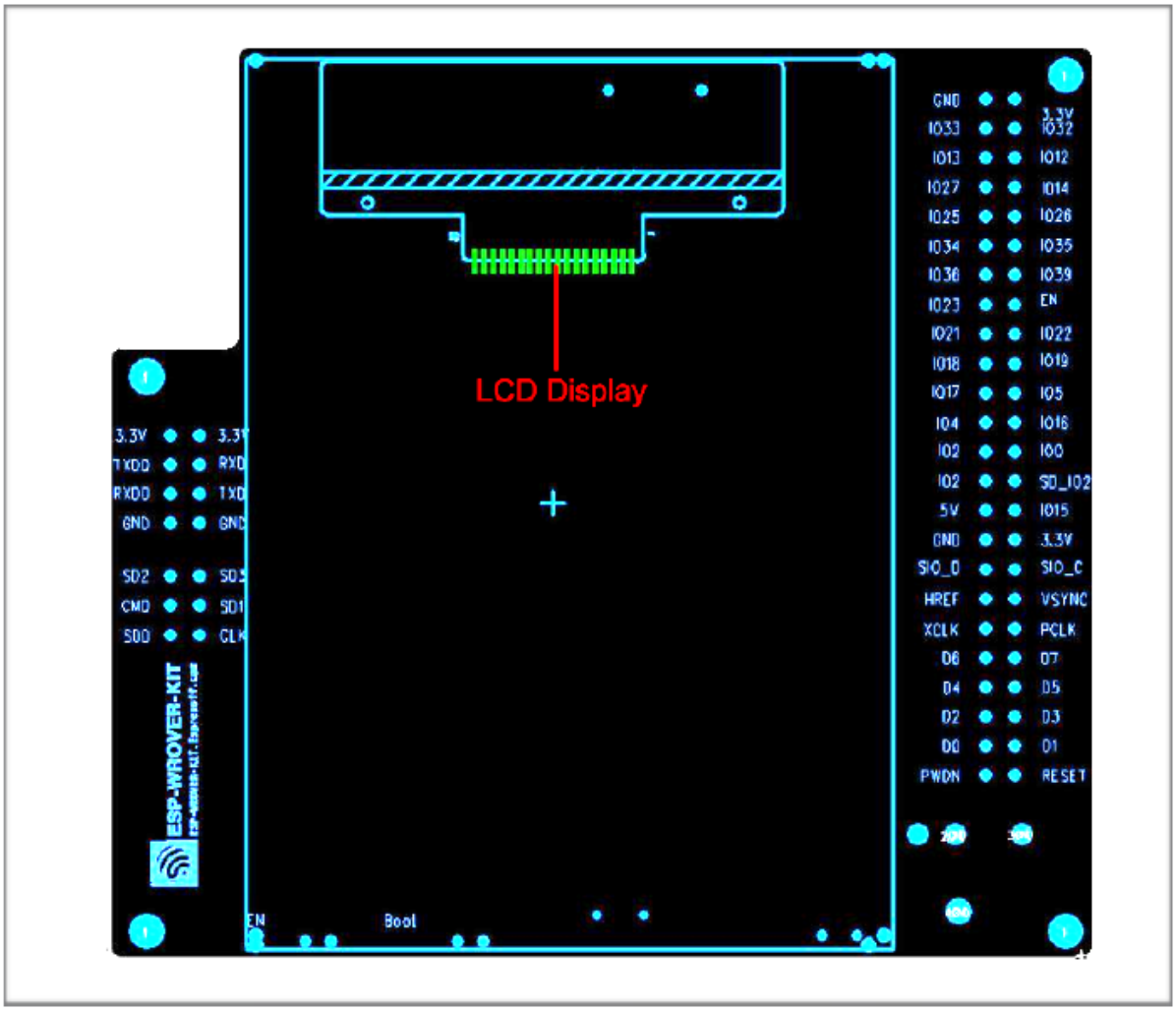
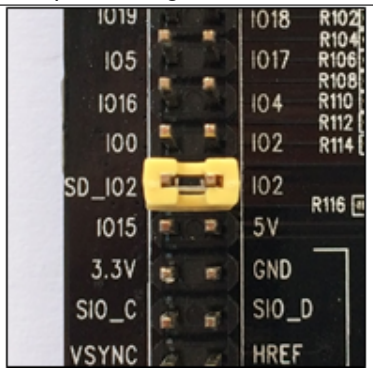
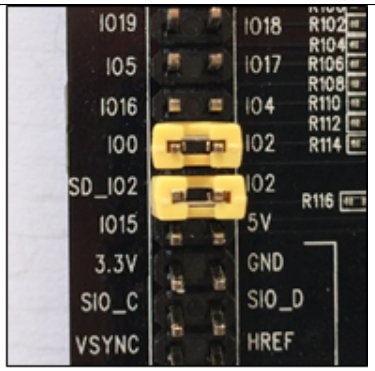
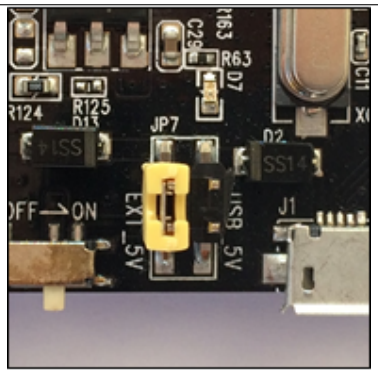
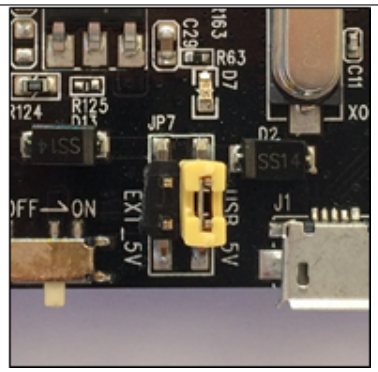
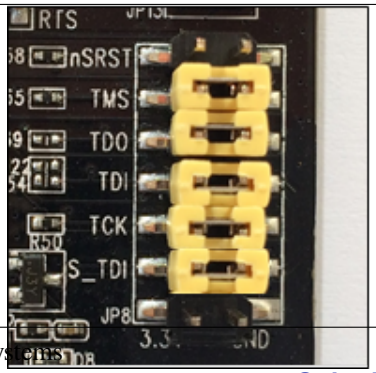
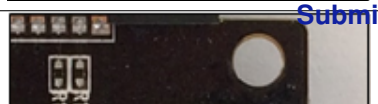


Fig. 28: ESP-WROVER-KIT board layout - back

Key Component	Description
32.768 kHz	External precision 32.768 kHz crystal oscillator serves as a clock with low-power consumption while the chip is in Deep-sleep mode.
ESP32 Module	Either ESP32-WROOM-32 or ESP32-WROVER with an integrated ESP32. The ESP32-WROVER module features all the functions of ESP32-WROOM-32 and integrates an external 32-MBit PSRAM for flexible extended storage and data processing capabilities.
CTS/RTS	Serial port flow control signals: the pins are not connected to the circuitry by default. To enable them, short the respective pins of JP14 with jumpers.
UART	Serial port. The serial TX/RX signals of FT2232 and ESP32 are broken out to the inward and outward sides of JP11 respectively. By default, these pairs of pins are connected with jumpers. To use ESP32 ' s serial interface, remove the jumpers and connect another external serial device to the respective pins.
SPI	By default, ESP32 uses its SPI interface to access flash and PSRAM memory inside the module. Use these pins to connect ESP32 to another SPI device. In this case, an extra chip select (CS) signal is needed. Please note that the interface voltage for the version with ESP32-WROVER is 1.8V, while that for the version with ESP32-WROOM-32 is 3.3 V.
JTAG	JTAG interface. JTAG signals of FT2232 and ESP32 are broken out to the inward and outward sides of JP8 respectively. By default, these pairs of pins are disconnected. To enable JTAG, short the respective pins with jumpers as shown in Section *Setup Options*.
FT2232	The FT2232 chip serves as a multi-protocol USB-to-serial bridge which can be programmed and controlled via USB to provide communication with ESP32. FT2232 features USB-to-UART and USB-to-JTAG functionalities.
EN	Reset button.
Boot	Download button. Holding down **Boot** and then pressing **EN** initiates Firmware Download mode for downloading firmware through the serial port.
USB	USB interface. Power supply for the board as well as the communication interface between a computer and the board.
Power Select	Power supply selector interface. The board can be powered either via USB or via the 5 V Input interface. Select the power source with a jumper. For more details, see Section *Setup Options*, jumper header JP7.
Power Key	Power On/Off Switch. Toggling toward **USB** powers the board on, toggling away from **USB** powers the board off.
5V Input	The 5 V power supply interface can be more convenient when the board is operating autonomously (not connected to a computer).
LDO	NCP1117(1 A). 5V-to-3.3V LDO. NCP1117 can provide a maximum current of 1 A. The LDO on the board has a fixed output voltage. Although, the user can install an LDO with adjustable output voltage. For details, please refer to ESP-WROVER-KIT v2 schematic.
Camera	Camera interface, a standard OV7670 camera module.
RGB	Red, green and blue (RGB) light emitting diodes (LEDs), can be controlled by pulse width modulation (PWM).
I/O	All the pins on the ESP32 module are broken out to pin headers. You can program ESP32 to enable multiple functions, such as PWM, ADC, DAC, I2C, I2S, SPI, etc.
microSD Card	microSD card slot for data storage: when ESP32 enters the download mode, GPIO2 cannot be held high. However, a pull-up resistor is required on GPIO2 to enable the microSD Card. By default, GPIO2 and the pull-up resistor R153 are disconnected. To enable the SD Card, use jumpers on JP1 as shown in Section *Setup Options*.
LCD	Support for mounting and interfacing a 3.2" SPI (standard 4-wire Serial Peripheral Interface) LCD, as shown on figure *ESP-WROVER-KIT board layout - back*.

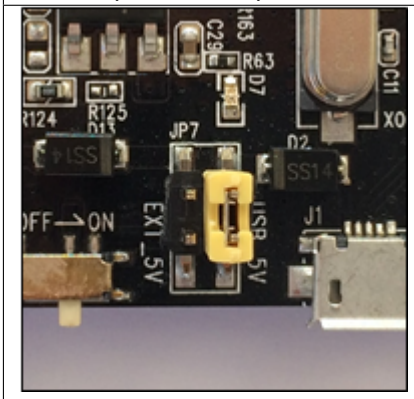
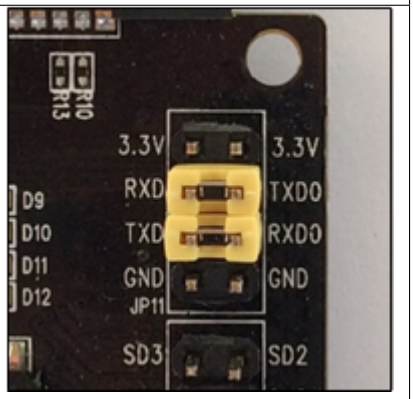
Setup Options There are five jumper blocks available to set up the board functionality. The most frequently required options are listed in the table below.

Header	Jumper Setting	Description of Functionality
JP1		Enable pull up for the microSD Card
JP1		Assert GPIO2 low during each download (by jumping it to GPIO0)
JP7		Power ESP-WROVER-KIT via an external power supply
JP7		Power ESP-WROVER-KIT via USB
JP8		Enable JTAG functionality
JP8		

Start Application Development Before powering up your ESP-WROVER-KIT, please make sure that the board is in good condition with no obvious signs of damage.

Initial Setup Please set only the following jumpers shown in the pictures below:

- Select USB as the power source using the jumper block JP7.
- Enable UART communication using the jumper block JP11.

Power up from USB port	Enable UART communication
	

Do not install any other jumpers.

Turn the **Power Switch** to ON, the **5V Power On LED** should light up.

Now to Development After that, proceed to ESP-IDF Get Started, which will quickly help you set up the development environment then flash an application example onto your board.

Related Documents

- ESP-WROVER-KIT v2 schematic (PDF)
- ESP32 Datasheet (PDF)
- ESP32-WROOM-32 Datasheet (PDF)

ESP-WROVER-KIT v3 Getting Started Guide

New version available: *ESP-WROVER-KIT v4.1 Getting Started Guide*

This guide shows how to get started with the ESP-WROVER-KIT v3 development board and also provides information about its functionality and configuration options.

What You Need

- *ESP-WROVER-KIT v3 board*
- USB 2.0 cable (A to Micro-B)
- Computer running Windows, Linux, or macOS

You can skip the introduction sections and go directly to Section *Start Application Development*.

Overview ESP-WROVER-KIT is an ESP32-based development board produced by Espressif. This board features an integrated LCD screen and microSD card slot.

ESP-WROVER-KIT comes with the following ESP32 modules:

- ESP32-WROOM-32
- ESP32-WROVER series

Its another distinguishing feature is the embedded FTDI FT2232HL chip - an advanced multi-interface USB bridge. This chip enables to use JTAG for direct debugging of ESP32 through the USB interface without a separate JTAG debugger. ESP-WROVER-KIT makes development convenient, easy, and cost-effective.

Most of the ESP32 I/O pins are broken out to the board's pin headers for easy access.

Note: The version with the ESP32-WROVER module uses ESP32's GPIO16 and GPIO17 as chip select and clock signals for PSRAM. By default, the two GPIOs are not broken out to the board's pin headers in order to ensure reliable performance.

Functionality Overview The block diagram below shows the main components of ESP-WROVER-KIT and their interconnections.

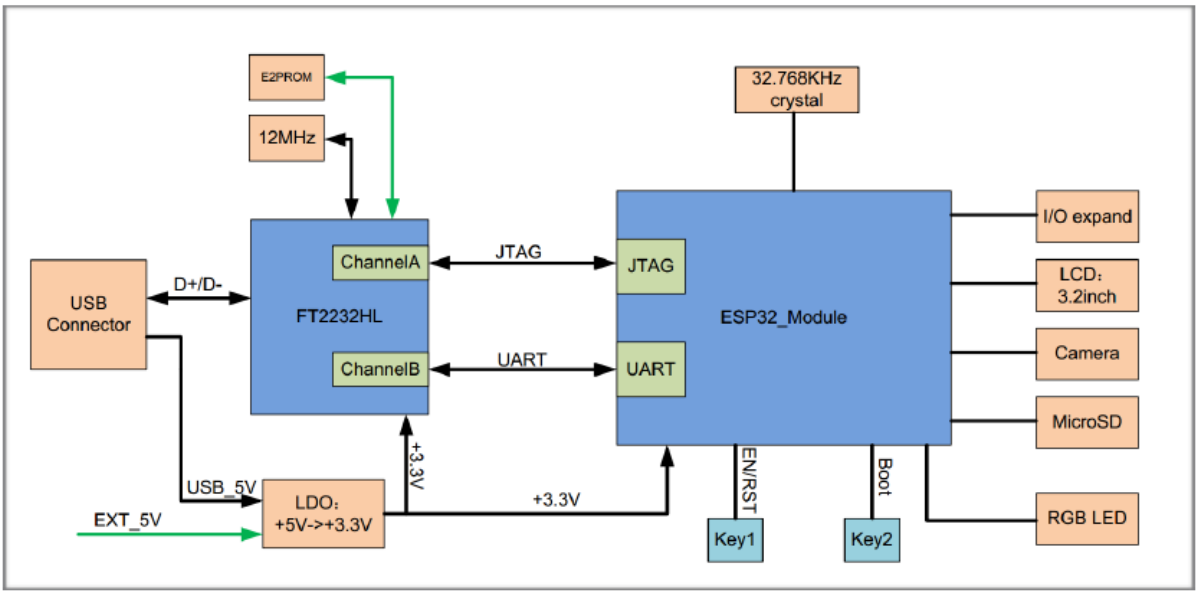


Fig. 29: ESP-WROVER-KIT block diagram

Functional Description The following two figures and the table below describe the key components, interfaces, and controls of the ESP-WROVER-KIT board.

The table below provides description in the following manner:

- Starting from the first picture's top right corner and going clockwise
- Then moving on to the second picture

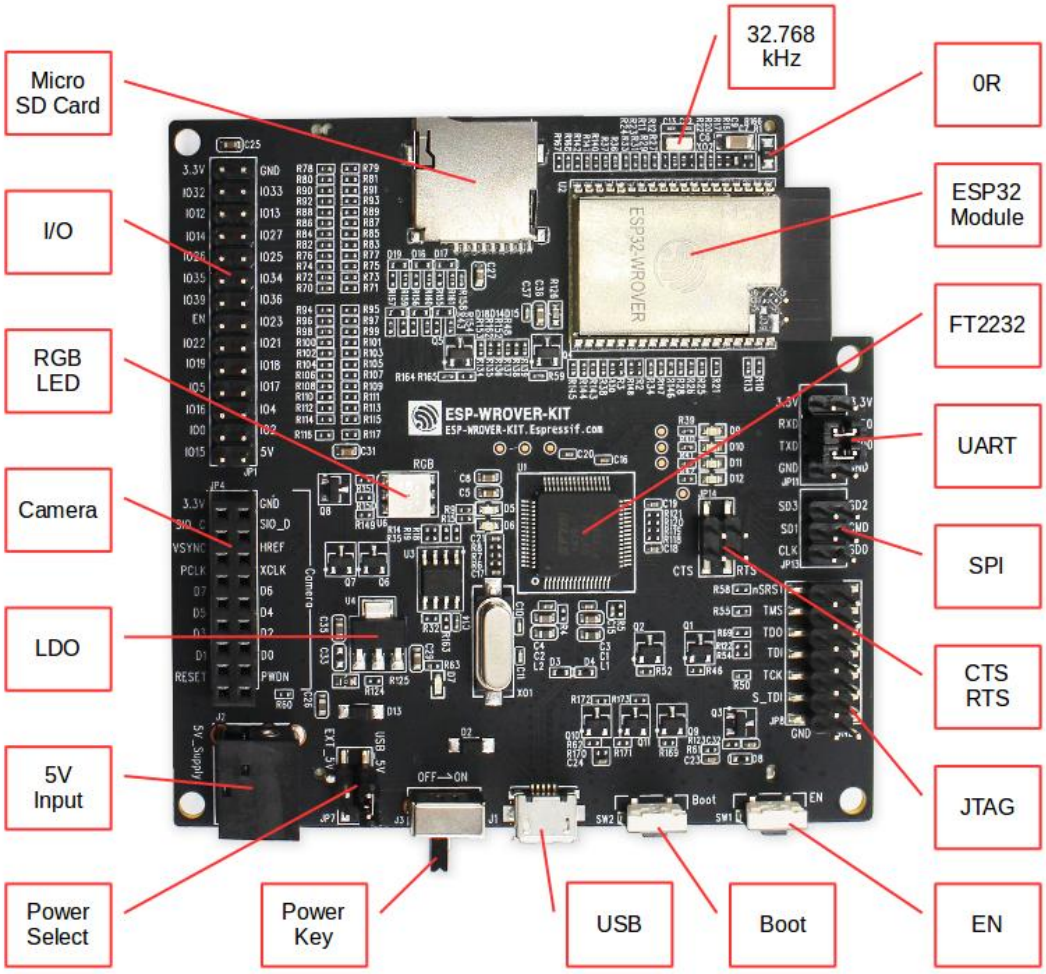


Fig. 30: ESP-WROVER-KIT board layout - front

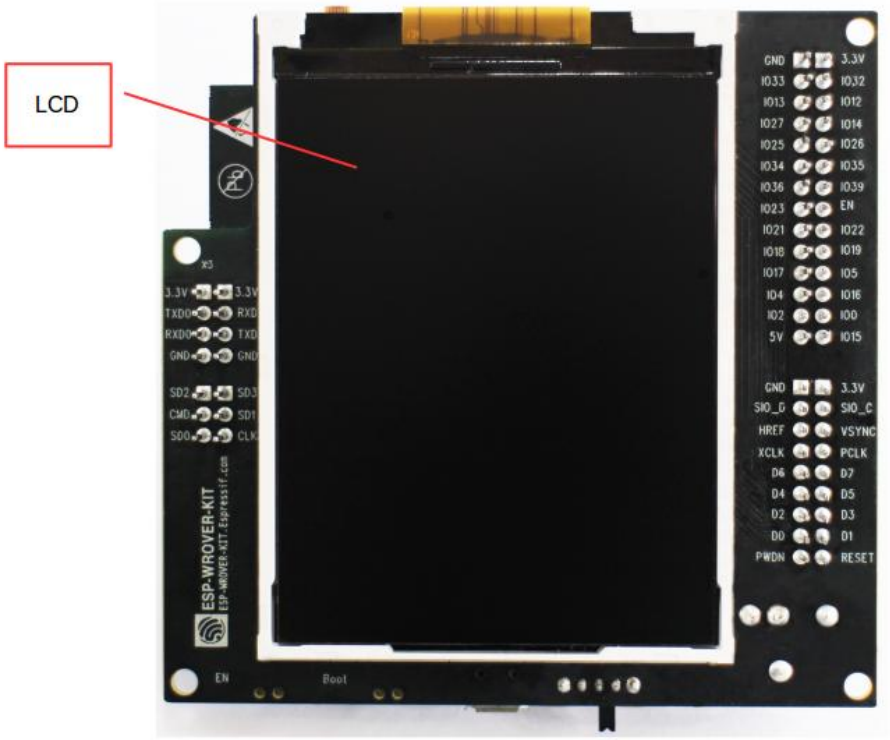
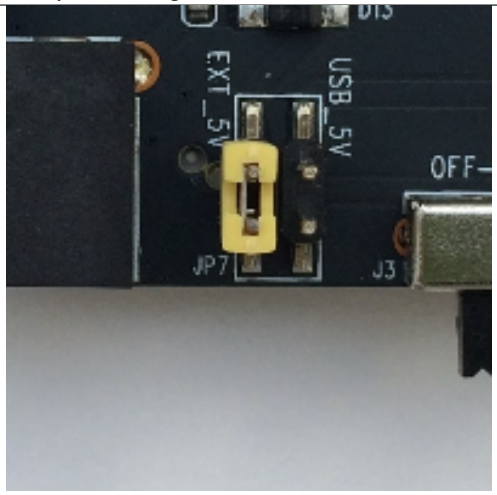
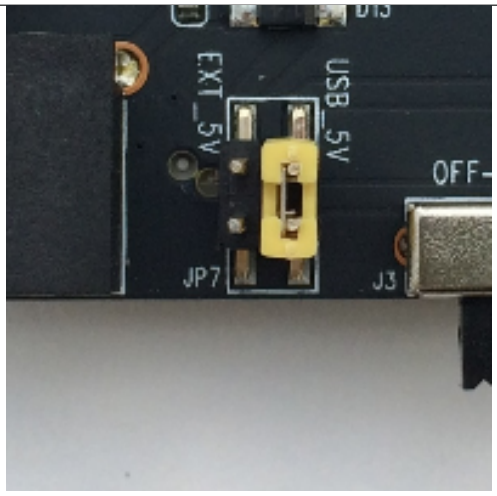
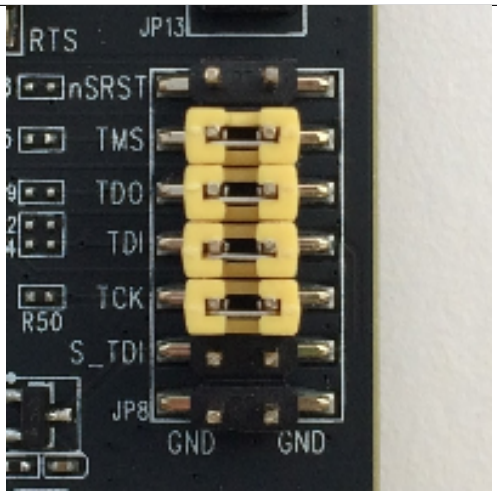
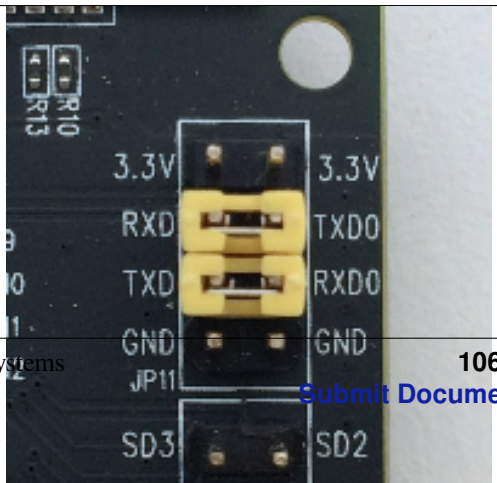


Fig. 31: ESP-WROVER-KIT board layout - back

Key Component	Description
32.768 kHz	External precision 32.768 kHz crystal oscillator serves as a clock with low-power consumption while the chip is in Deep-sleep mode.
0R	Zero-ohm resistor intended as a placeholder for a current shunt, can be desoldered or replaced with a current shunt to facilitate the measurement of ESP32's current consumption in different modes.
ESP32 Module	Either ESP32-WROOM-32 or ESP32-WROVER with an integrated ESP32. The ESP32-WROVER module features all the functions of ESP32-WROOM-32 and integrates an external 32-MBit PSRAM for flexible extended storage and data processing capabilities.
FT2232	The FT2232 chip serves as a multi-protocol USB-to-serial bridge which can be programmed and controlled via USB to provide communication with ESP32. FT2232 also features USB-to-JTAG interface which is available on channel A of the chip, while USB-to-serial is on channel B. The FT2232 chip enhances user-friendliness in terms of application development and debugging. See ESP-WROVER-KIT v3 schematic.
UART	Serial port. The serial TX/RX signals of FT2232 and ESP32 are broken out to the inward and outward sides of JP11 respectively. By default, these pairs of pins are connected with jumpers. To use ESP32's serial interface, remove the jumpers and connect another external serial device to the respective pins.
SPI	By default, ESP32 uses its SPI interface to access flash and PSRAM memory inside the module. Use these pins to connect ESP32 to another SPI device. In this case, an extra chip select (CS) signal is needed. Please note that the interface voltage for the version with ESP32-WROVER is 1.8V, while that for the version with ESP32-WROOM-32 is 3.3V.
CTS/RTS	Serial port flow control signals: the pins are not connected to the circuitry by default. To enable them, short the respective pins of JP14 with jumpers.
JTAG	JTAG interface. JTAG signals of FT2232 and ESP32 are broken out to the inward and outward sides of JP8 respectively. By default, these pairs of pins are disconnected. To enable JTAG, short the respective pins with jumpers as shown in Section *Setup Options*.
EN	Reset button.
Boot	Download button. Holding down **Boot** and then pressing **EN** initiates Firmware Download mode for downloading firmware through the serial port.
USB	USB interface. Power supply for the board as well as the communication interface between a computer and the board.
Power Key	Power On/Off Switch. Toggling toward **USB** powers the board on, toggling away from **USB** powers the board off.
Power Select	Power supply selector interface. The board can be powered either via USB or via the 5V Input interface. Select the power source with a jumper. For more details, see Section *Setup Options*, jumper header JP7.
5V Input	The 5 V power supply interface can be more convenient when the board is operating autonomously (not connected to a computer).
LDO	NCP1117(1A). 5V-to-3.3V LDO. NCP1117 can provide a maximum current of 1A. The LDO on the board has a fixed output voltage. Although, the user can install an LDO with adjustable output voltage. For details, please refer to ESP-WROVER-KIT v3 schematic.
Camera	Camera interface, a standard OV7670 camera module.
RGB LED	Red, green and blue (RGB) light emitting diodes (LEDs), can be controlled by pulse width modulation (PWM).
I/O	All the pins on the ESP32 module are broken out to pin headers. You can program ESP32 to enable multiple functions, such as PWM, ADC, DAC, I2C, I2S, SPI, etc.
microSD Card Slot	Useful for developing applications that access microSD card for data storage and retrieval.
LCD	Support for mounting and interfacing a 3.2" SPI (standard 4-wire Serial Peripheral Interface) LCD, as shown on figure *ESP-WROVER-KIT board layout - back*.

Setup Options There are five jumper blocks available to set up the board functionality. The most frequently required options are listed in the table below.

Header	Jumper Setting	Description of Functionality
JP7		Power ESP-WROVER-KIT via an external power supply
JP7		Power ESP-WROVER-KIT via USB
JP8		Enable JTAG functionality
JP11		Enable UART communication

Allocation of ESP32 Pins Some pins/terminals of ESP32 are allocated for use with the onboard or external hardware. If that hardware is not used, e.g., nothing is plugged into the Camera (JP4) header, then these GPIOs can be used for other purposes.

Some of the pins, such as GPIO0 or GPIO2, have multiple functions and some of them are shared among onboard and external peripheral devices. Certain combinations of peripherals cannot work together. For example, it is not possible to do JTAG debugging of an application that is using SD card, because several pins are shared by JTAG and the SD card slot.

In other cases, peripherals can coexist under certain conditions. This is applicable to, for example, LCD screen and SD card that share only a single pin GPIO21. This pin is used to provide D/C (Data/Control) signal for the LCD as well as the CD (Card Detect) signal read from the SD card slot. If the card detect functionality is not essential, then it may be disabled by removing R167, so both LCD and SD may operate together.

For more details on which pins are shared among which peripherals, please refer to the table in the next section.

Main I/O Connector/JP1 The JP1 connector consists of 14x2 male pins whose functions are shown in the middle two "I/O" columns of the table below. The two "Shared With" columns on both sides describe where else on the board a certain GPIO is used.

Shared With	I/O	I/O	Shared With
n/a	3.3V	GND	n/a
NC/XTAL	IO32	IO33	NC/XTAL
JTAG, microSD	IO12	IO13	JTAG, microSD
JTAG, microSD	IO14	IO27	Camera
Camera	IO26	IO25	Camera, LCD
Camera	IO35	IO34	Camera
Camera	IO39	IO36	Camera
JTAG	EN	IO23	Camera, LCD
Camera, LCD	IO22	IO21	Camera, LCD, microSD
Camera, LCD	IO19	IO18	Camera, LCD
Camera, LCD	IO5	IO17	PSRAM
PSRAM	IO16	IO4	LED, Camera, microSD
Camera, LED, Boot	IO0	IO2	LED, microSD
JTAG, microSD	IO15	5V	

Legend:

- NC/XTAL - *32.768 kHz Oscillator*
- JTAG - *JTAG / JP8*
- Boot - Boot button / SW2
- Camera - *Camera / JP4*
- LED - *RGB LED*
- microSD - *microSD Card / J4*
- LCD - *LCD / U5*
- PSRAM - only in case ESP32-WROVER is installed

32.768 kHz Oscillator

.	ESP32 Pin
1	GPIO32
2	GPIO33

Note: Since GPIO32 and GPIO33 are connected to the oscillator by default, they are not connected to the JP1 I/O connector to maintain signal integrity. This allocation may be changed from the oscillator to JP1 by desoldering the zero-ohm resistors from positions R11/R23 and re-soldering them to positions R12/R24.

SPI Flash/JP13

.	ESP32 Pin
1	CLK/GPIO6
2	SD0/GPIO7
3	SD1/GPIO8
4	SD2/GPIO9
5	SD3/GPIO10
6	CMD/GPIO11

Important: The module's flash bus is connected to the jumper block JP13 through zero-ohm resistors R140 ~ R145. If the flash memory needs to operate at the frequency of 80 MHz, for reasons such as improving the integrity of bus signals, you can desolder these resistors to disconnect the module's flash bus from the pin header JP13.

JTAG/JP8

.	ESP32 Pin	JTAG Signal
1	EN	TRST_N
2	MTMS/GPIO14	TMS
3	MTDO/GPIO15	TDO
4	MTDI/GPIO12	TDI
5	MTCK/GPIO13	TCK

Camera/JP4

.	ESP32 Pin	Camera Signal
1	n/a	3.3V
2	n/a	Ground
3	GPIO27	SIO_C/SCCB Clock
4	GPIO26	SIO_D/SCCB Data
5	GPIO25	VSYNC/Vertical Sync
6	GPIO23	HREF/Horizontal Reference
7	GPIO22	PCLK/Pixel Clock
8	GPIO21	XCLK/System Clock
9	GPIO35	D7/Pixel Data Bit 7
10	GPIO34	D6/Pixel Data Bit 6
11	GPIO39	D5/Pixel Data Bit 5
12	GPIO36	D4/Pixel Data Bit 4
13	GPIO19	D3/Pixel Data Bit 3
14	GPIO18	D2/Pixel Data Bit 2
15	GPIO5	D1/Pixel Data Bit 1
16	GPIO4	D0/Pixel Data Bit 0
17	GPIO0	RESET/Camera Reset
18	n/a	PWDN/Camera Power Down

- Signals D0 .. D7 denote camera data bus

RGB LED

.	ESP32 Pin	RGB LED
1	GPIO0	Red
2	GPIO2	Green
3	GPIO4	Blue

microSD Card

.	ESP32 Pin	microSD Signal
1	MTDI/GPIO12	DATA2
2	MTCK/GPIO13	CD/DATA3
3	MTDO/GPIO15	CMD
4	MTMS/GPIO14	CLK
5	GPIO2	DATA0
6	GPIO4	DATA1
7	GPIO21	CD

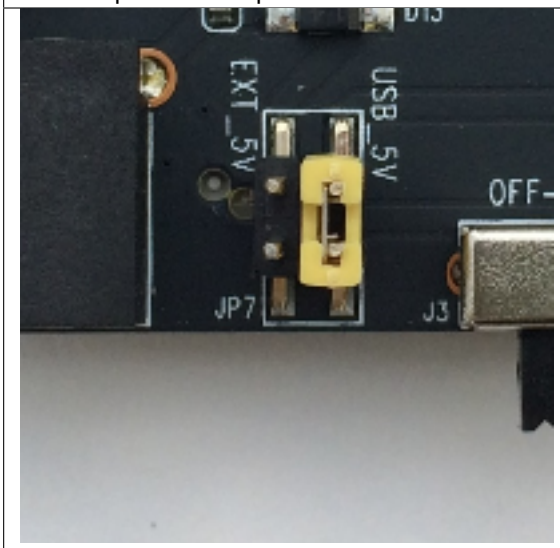
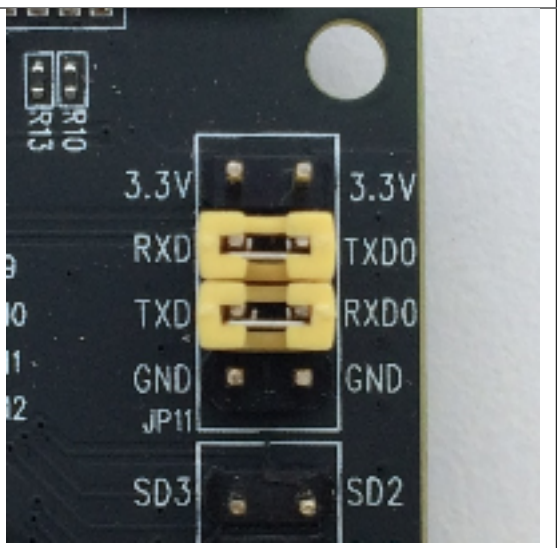
LCD/U5

.	ESP32 Pin	LCD Signal
1	GPIO18	RESET
2	GPIO19	SCL
3	GPIO21	D/C
4	GPIO22	CS
5	GPIO23	SDA
6	GPIO25	SDO
7	GPIO5	Backlight

Start Application Development Before powering up your ESP-WROVER-KIT, please make sure that the board is in good condition with no obvious signs of damage.

Initial Setup Please set only the following jumpers shown in the pictures below:

- Select USB as the power source using the jumper block JP7.
- Enable UART communication using the jumper block JP11.

Power up from USB port	Enable UART communication
	

Do not install any other jumpers.

Turn the **Power Switch** to ON, the **5V Power On LED** should light up.

Now to Development After that, proceed to ESP-IDF Get Started, which will quickly help you set up the development environment then flash an application example onto your board.

Related Documents

- ESP-WROVER-KIT v3 schematic (PDF)
- ESP32 Datasheet (PDF)
- ESP32-WROOM-32 Datasheet (PDF)

7.4 ESP32-PICO-KIT

ESP32-PICO-KIT is an ESP32-based mini development board produced by Espressif. The core of this board is ESP32-PICO-D4 - a System-in-Package (SiP) module with complete Wi-Fi and Bluetooth® functionalities.

7.4.1 ESP32-PICO-KIT v4/v4.1

This guide shows how to get started with the ESP32-PICO-KIT v4/v4.1 mini development board. For the description of other ESP32-PICO-KIT versions, please check *ESP32-PICO-KIT v3*.

This particular description covers ESP32-PICO-KIT v4 and v4.1. The difference is the upgraded USB-UART bridge from CP2102 in v4 with up to 1 Mbps transfer rates to CP2102N in v4.1 with up to 3 Mbps transfer rates.

What You Need

- *ESP32-PICO-KIT mini development board*
- USB 2.0 A to Micro B cable
- Computer running Windows, Linux, or macOS

You can skip the introduction sections and go directly to Section *Start Application Development*.

Overview

ESP32-PICO-KIT is an ESP32-based mini development board produced by Espressif.

The core of this board is ESP32-PICO-D4 - a System-in-Package (SiP) module with complete Wi-Fi and Bluetooth® functionalities. Compared to other ESP32 modules, ESP32-PICO-D4 integrates the following peripheral components in one single package, which otherwise would need to be installed separately:

- 40 MHz crystal oscillator
- 4 MB flash
- Filter capacitors
- RF matching links

This setup reduces the costs of additional external components as well as the cost of assembly and testing and also increases the overall usability of the product.

The development board features a USB-UART Bridge circuit which allows developers to connect the board to a computer's USB port for flashing and debugging.

All the IO signals and system power on ESP32-PICO-D4 are led out to two rows of 20 x 0.1" header pads on both sides of the development board for easy access. For compatibility with Dupont wires, 2 x 17 header pads are populated with two rows of male pin headers. The remaining 2 x 3 header pads beside the antenna are not populated. These pads may be populated later by the user if required.

Note:

1. There are two versions of ESP32-PICO-KIT boards, respectively with male headers and female headers. In this guide, the male header version is taken as an example.

2. The 2 x 3 pads not populated with pin headers are connected to the flash memory embedded in the ESP32-PICO-D4 SiP module. For more details, see module's datasheet in *Related Documents*.

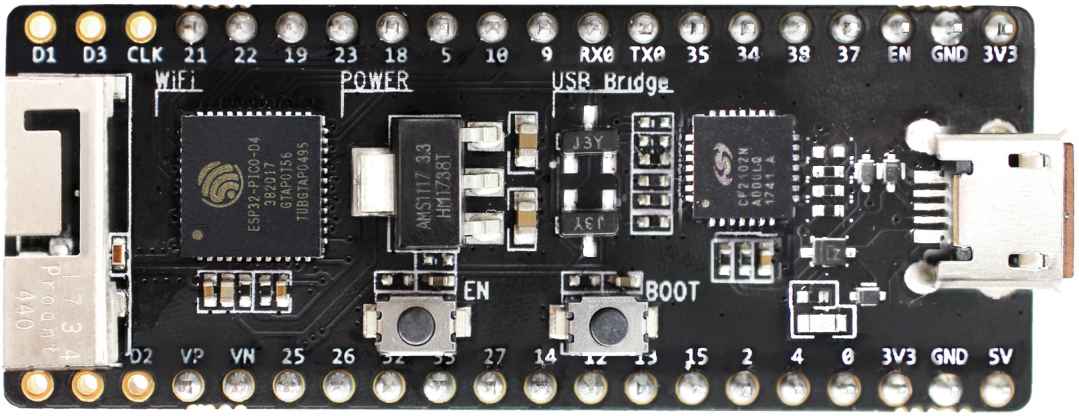


Fig. 32: ESP32-Pico-Kit (click to enlarge)

Functionality Overview

The block diagram below shows the main components of ESP32-PICO-KIT and their interconnections.

Functional Description

The following figure and the table below describe the key components, interfaces, and controls of the ESP32-PICO-KIT board.

Below is the description of the items identified in the figure starting from the top left corner and going clockwise.

Key Component	Description
ESP32-PICO-D4	Standard ESP32-PICO-D4 module soldered to the ESP32-PICO-KIT board. The complete ESP32 system on a chip (ESP32 SoC) has been integrated into the SiP module, requiring only an external antenna with LC matching network, decoupling capacitors, and a pull-up resistor for EN signals to function properly.
LDO	5V-to-3.3V Low dropout voltage regulator (LDO).
USB-UART bridge	Single-chip USB-UART bridge: CP2102 in v4 provides up to 1 Mbps transfer rates and CP2102N in v4.1 offers up to 3 Mbps transfers rates.
Micro USB Port	USB interface. Power supply for the board as well as the communication interface between a computer and the board.
5V Power On LED	This red LED turns on when power is supplied to the board. For details, see the schematics in *Related Documents*.
I/O	All the pins on ESP32-PICO-D4 are broken out to pin headers. You can program ESP32 to enable multiple functions, such as PWM, ADC, DAC, I2C, I2S, SPI, etc. For details, please see Section *Pin Descriptions*.
BOOT Button	Download button. Holding down **Boot** and then pressing **EN** initiates Firmware Download mode for downloading firmware through the serial port.
EN Button	Reset button.

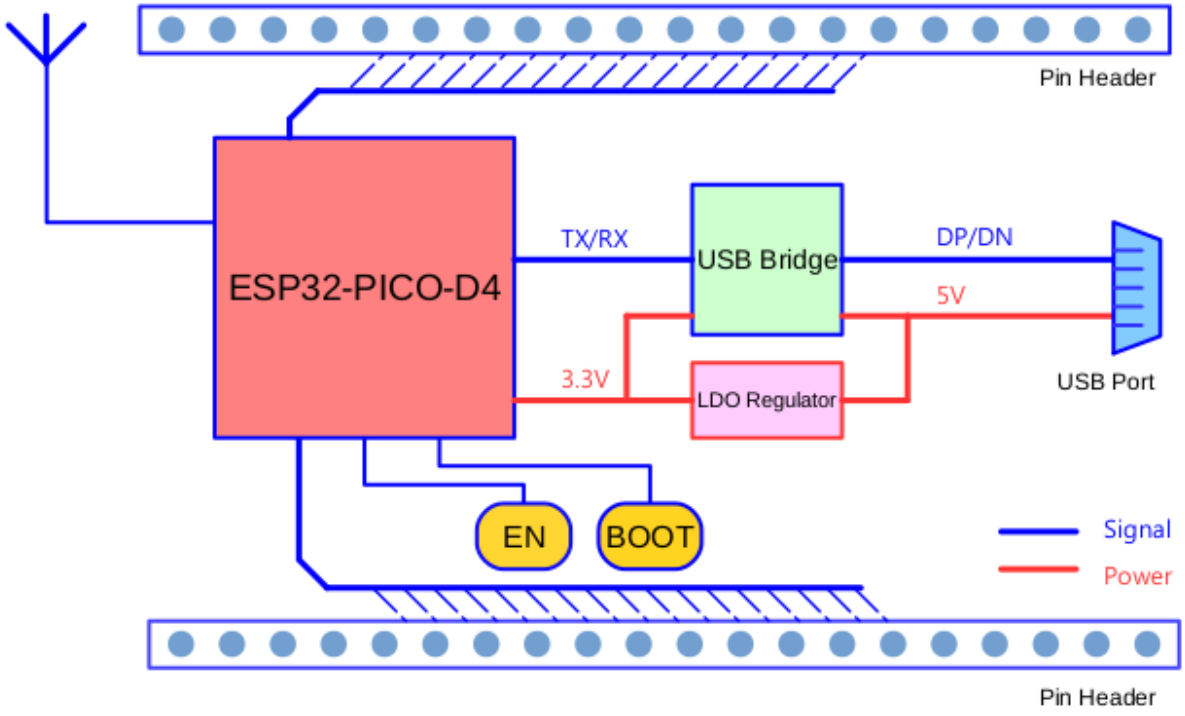


Fig. 33: ESP32-PICO-KIT block diagram

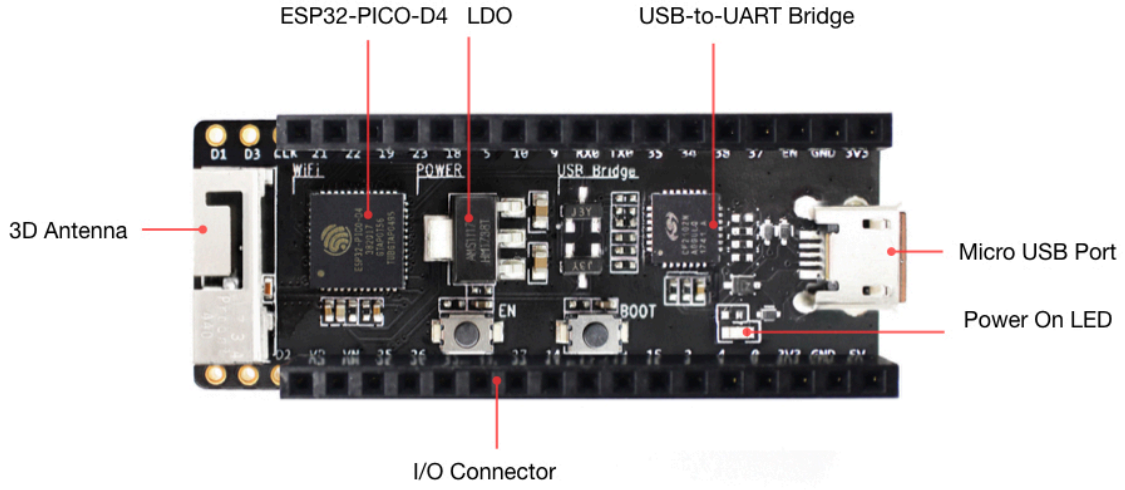


Fig. 34: ESP32-PICO-KIT board layout (with female headers)

Power Supply Options

There are three mutually exclusive ways to provide power to the board:

- Micro USB port, default power supply
- 5V / GND header pins
- 3V3 / GND header pins

> **Warning:** The power supply must be provided using **one and only one of the options above**, otherwise the board and/or the power supply source can be damaged.

Pin Descriptions

The two tables below provide the **Name** and **Function** of I/O header pins on both sides of the board, see *ESP32-PICO-KIT board layout (with female headers)*. The pin numbering and header names are the same as in the schematic given in *Related Documents*.

Header J2

No.	Name	Type	Function
1	FLASH_SD1 (FSD1)	I/O	GPIO8, SD_DATA1, SPID, HS1_DATA1 *(See 1)* , U2CTS
2	FLASH_SD3 (FSD3)	I/O	GPIO7, SD_DATA0, SPIQ, HS1_DATA0 *(See 1)* , U2RTS
3	FLASH_CLK (FCLK)	I/O	GPIO6, SD_CLK, SPICLK, HS1_CLK *(See 1)* , U1CTS
4	IO21	I/O	GPIO21, VSPIHD, EMAC_TX_EN
5	IO22	I/O	GPIO22, VSPIWP, U0RTS, EMAC_TXD1
6	IO19	I/O	GPIO19, VSPIQ, U0CTS, EMAC_TXD0
7	IO23	I/O	GPIO23, VSPID, HS1_STROBE
8	IO18	I/O	GPIO18, VSPICLK, HS1_DATA7
9	IO5	I/O	GPIO5, VSPICS0, HS1_DATA6, EMAC_RX_CLK
10	IO10	I/O	GPIO10, SD_DATA3, SPIWP, HS1_DATA3, U1TXD
11	IO9	I/O	GPIO9, SD_DATA2, SPIHD, HS1_DATA2, U1RXD
12	RXD0	I/O	GPIO3, U0RXD *(See 3)* , CLK_OUT2
13	TXD0	I/O	

Header J3

No.	Name	Type	Function
1	FLASH_CS (FCS)	I/O	GPIO16, HS1_DATA4 *(See 1)* , U2RXD, EMAC_CLK_OUT
2	FLASH_SD0 (FSD0)	I/O	GPIO17, HS1_DATA5 *(See 1)* , U2TXD, EMAC_CLK_OUT_180
3	FLASH_SD2 (FSD2)	I/O	GPIO11, SD_CMD, SPICS0, HS1_CMD *(See 1)* , U1RTS
4	SENSOR_VP (FSVP)	I	GPIO36, ADC1_CH0, RTC_GPIO0
5	SENSOR_VN (FSVN)	I	GPIO39, ADC1_CH3, RTC_GPIO3
6	IO25	I/O	GPIO25, DAC_1, ADC2_CH8, RTC_GPIO6, EMAC_RXD0
7	IO26	I/O	GPIO26, DAC_2, ADC2_CH9, RTC_GPIO7, EMAC_RXD1
8	IO32	I/O	32K_XP *(See 2a)* , ADC1_CH4, TOUCH9, RTC_GPIO9
9	IO33	I/O	32K_XN *(See 2b)* , ADC1_CH5, TOUCH8, RTC_GPIO8
10	IO27	I/O	GPIO27, ADC2_CH7, TOUCH7, RTC_GPIO17 EMAC_RX_DV
11	IO14	I/O	ADC2_CH6, TOUCH6, RTC_GPIO16, MTMS, HSPICLK,

Note:

1. This pin is connected to the flash pin of ESP32-PICO-D4.
2. 32.768 kHz crystal oscillator: (a) input; (b) output.
3. This pin is connected to the pin of the USB bridge chip on the board.
4. The operating voltage of ESP32-PICO-KIT' s embedded SPI flash is 3.3 V. Therefore, the strapping pin MTDI should hold bit zero during the module power-on reset. If connected, please make sure that this pin is not held up on reset.

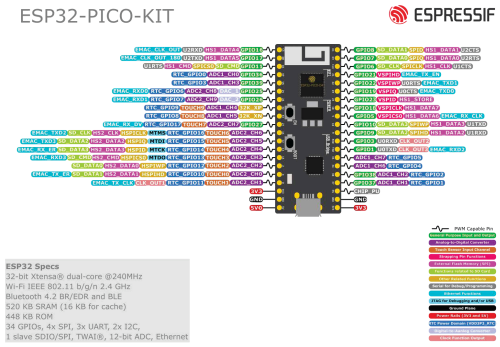


Fig. 35: ESP32-PICO-KIT Pin Layout (click to enlarge)

Pin Layout

Start Application Development

Before powering up your ESP32-PICO-KIT, please make sure that the board is in good condition with no obvious signs of damage.

After that, proceed to Get Started, where Section Installation will quickly help you set up the development environment and then flash an example project onto your board.

Board Dimensions

The dimensions are 52 x 20.3 x 10 mm (2.1" x 0.8" x 0.4").

For the board physical construction details, please refer to its Reference Design listed below.

Related Documents

- ESP32-PICO-KIT v4 schematic (PDF)
- ESP32-PICO-KIT v4.1 schematic (PDF)
- ESP32-PICO-KIT Reference Design containing OrCAD schematic, PCB layout, gerbers and BOM
- ESP32-PICO-D4 Datasheet (PDF)

ESP32-PICO-KIT v3

This guide shows how to get started with the ESP32-PICO-KIT v3 mini development board. For the description of other ESP32-PICO-KIT versions, please check *ESP32-PICO-KIT v4/v4.1*.

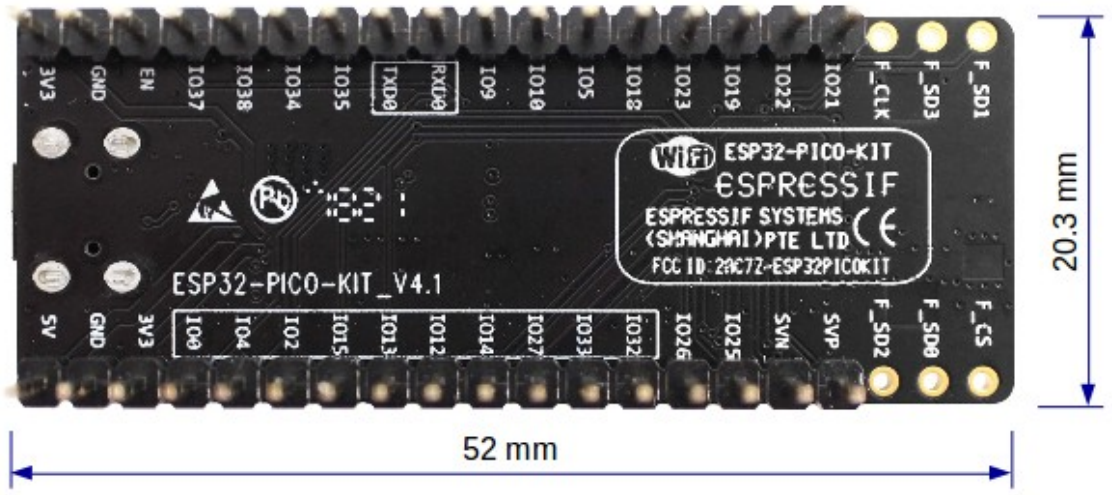


Fig. 36: ESP32-PICO-KIT dimensions - back (with male headers)

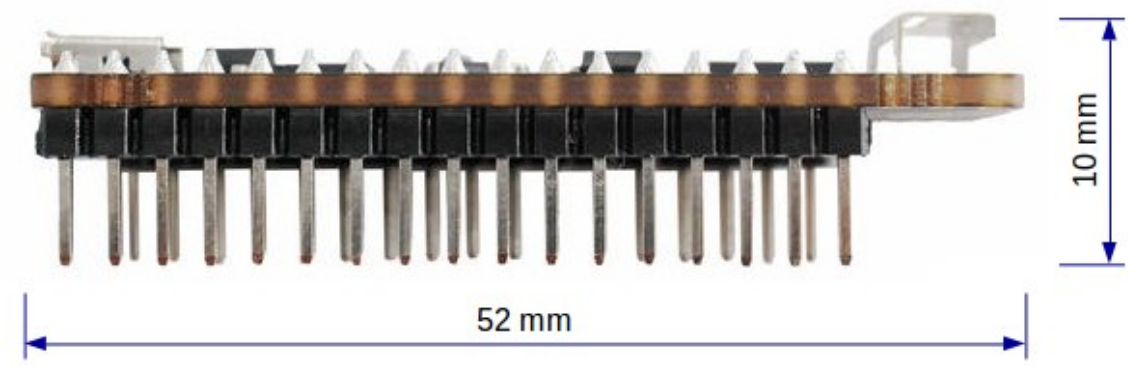


Fig. 37: ESP32-PICO-KIT dimensions - side (with male headers)

What You Need

- ESP32-PICO-KIT v3 mini development board
- USB 2.0 A to Micro B cable
- Computer running Windows, Linux, or macOS

You can skip the introduction sections and go directly to Section *Start Application Development*.

Overview ESP32-PICO-KIT v3 is an ESP32-based mini development board produced by Espressif. The core of this board is ESP32-PICO-D4 - a System-in-Package (SiP) module.

The development board features a USB-UART Bridge circuit, which allows developers to connect the board to a computer's USB port for flashing and debugging.

All the IO signals and system power on ESP32-PICO-D4 are led out to two rows of 20 x 0.1" header pads on both sides of the development board for easy access.

Functional Description The following figure and the table below describe the key components, interfaces, and controls of the ESP32-PICO-KIT v3 board.

Below is the description of the items identified in the figure starting from the top left corner and going clockwise.

Key Component	Description
ESP32-PICO-D4	Standard ESP32-PICO-D4 module soldered to the ESP32-PICO-KIT V3 board. The complete ESP32 system on a chip (ESP32 SoC) has been integrated into the SiP module, requiring only an external antenna with LC matching network, decoupling capacitors, and a pull-up resistor for EN signals to function properly.
LDO	5V-to-3.3V Low dropout voltage regulator (LDO).
USB-UART bridge	Single-chip USB-UART bridge provides up to 1 Mbps transfers rates.
Micro USB Port	USB interface. Power supply for the board as well as the communication interface between a computer and the board.
Power On LED	This red LED turns on when power is supplied to the board.
I/O	All the pins on ESP32-PICO-D4 are broken out to pin headers. You can program ESP32 to enable multiple functions, such as PWM, ADC, DAC, I2C, I2S, SPI, etc.
BOOT Button	Download button. Holding down **Boot** and then pressing **EN** initiates Firmware Download mode for downloading firmware through the serial port.
EN Button	Reset button.

Start Application Development Before powering up your ESP32-PICO-KIT v3, please make sure that the board is in good condition with no obvious signs of damage.

After that, proceed to Get Started, where Section Installation will quickly help you set up the development environment and then flash an example project onto your board.

Related Documents

- ESP32-PICO-KIT v3 schematic (PDF)
- ESP32-PICO-D4 Datasheet (PDF)

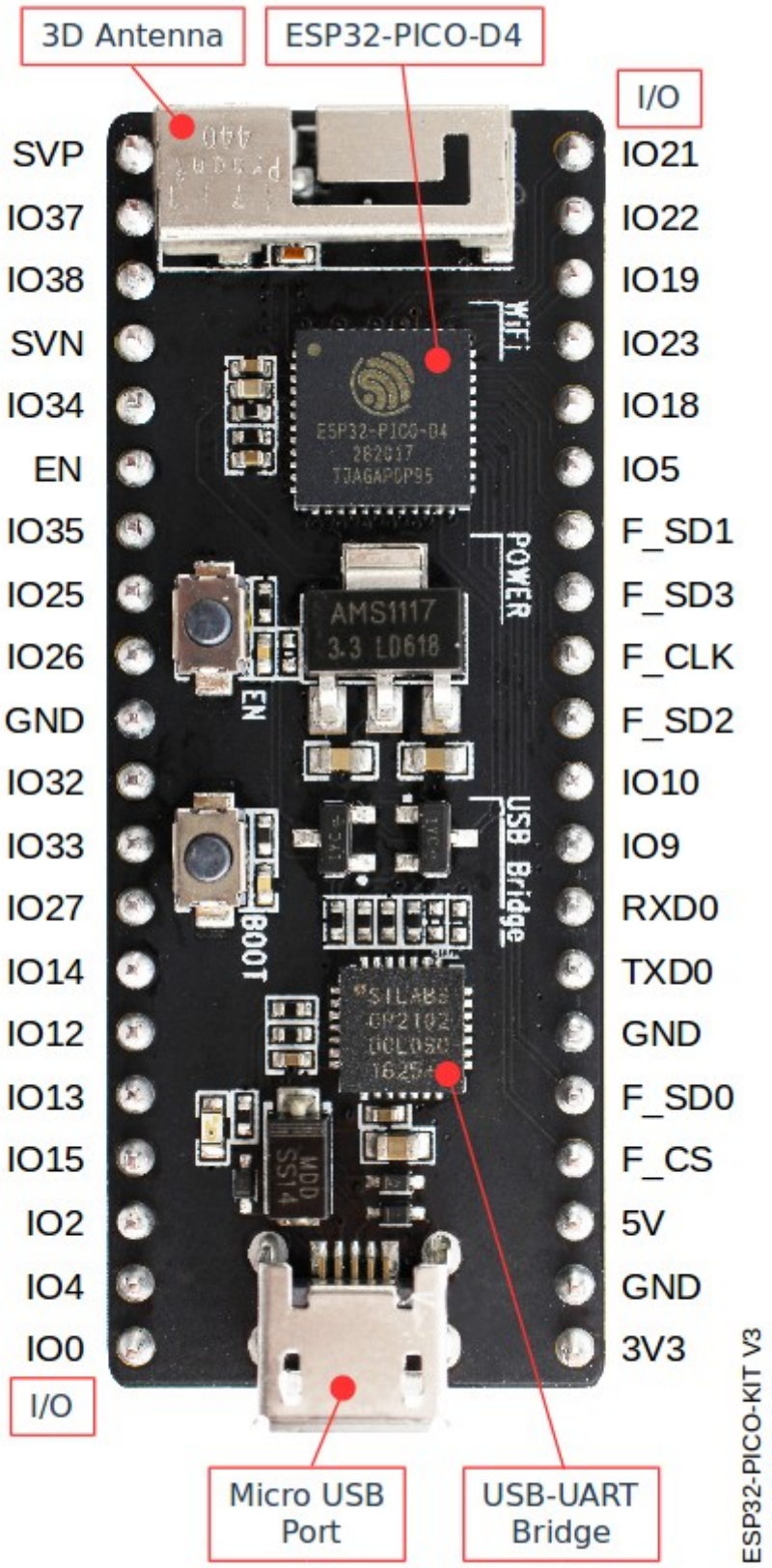
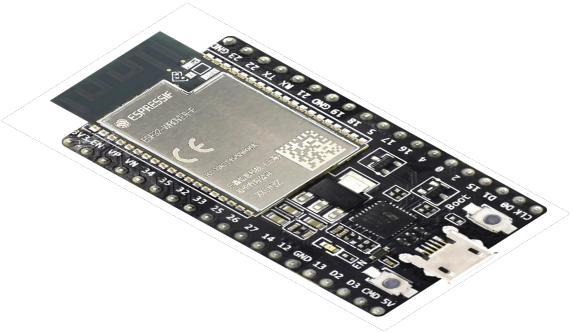
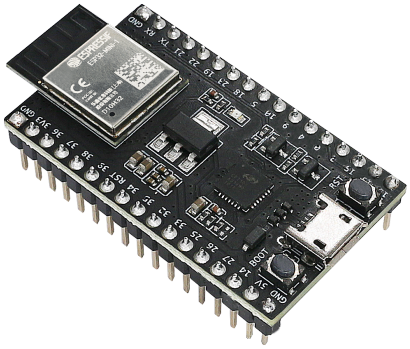
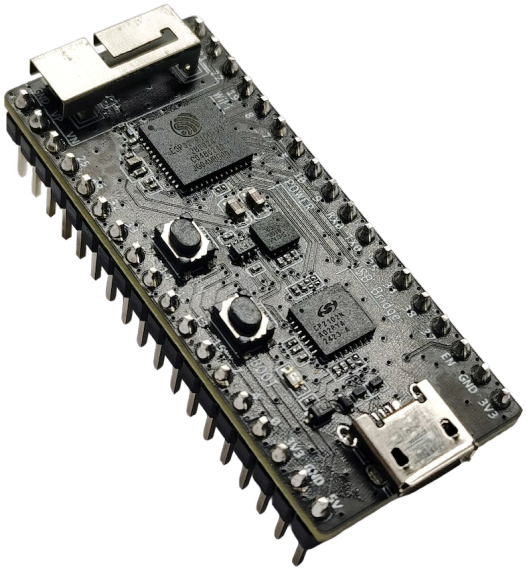
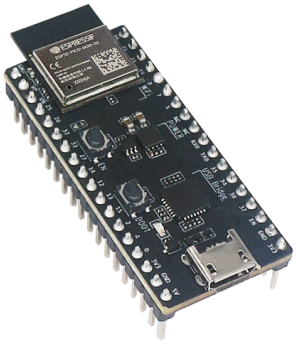
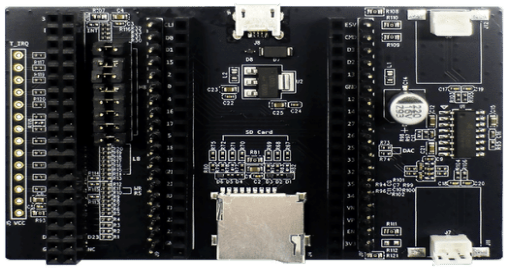


Fig. 38: ESP32-PICO-KIT v3 board layout (click to enlarge)

ESP32 Development Boards	
	
ESP32-DevKitC	ESP32-DevKitM-1
	
ESP32-PICO-KIT-1	ESP32-PICO-DevKitM-2
	
ESP32-LCDKit	ESP32-Ethernet-Kit

Chapter 8

Related Documentation and Resources

8.1 Related Documentation

- ESP32 Datasheet –Specifications of the ESP32 hardware.
- ESP32 Technical Reference Manual –Detailed information on how to use the ESP32 memory and peripherals.
- ESP32 Hardware Design Guidelines –Guidelines on how to integrate the ESP32 into your hardware product.
- ESP32 Product/Process Change Notifications (PCN)
  https://espressif.com/en/support/documents/pcns?keys=ESP32
- ESP32 Advisories –Information on security, bugs, compatibility, component reliability.
  https://espressif.com/en/support/documents/advisories?keys=ESP32
- Certificates
  https://espressif.com/en/support/documents/certificates
- Documentation Updates and Update Notification Subscription
  https://espressif.com/en/support/download/documents

8.2 Developer Zone

- ESP-IDF Programming Guide for ESP32 –Extensive documentation for the ESP-IDF development framework.
- ESP-IoT-Solution Programming Guide - Extensive documentation for the ESP-IoT-Solution development framework.
- ESP-FAQ - A summary document of frequently asked questions released by Espressif.
- ESP-IDF and other development frameworks on GitHub.
  https://github.com/espressif
- ESP32 BBS Forum –Engineer-to-Engineer (E2E) Community for Espressif products where you can post questions, share knowledge, explore ideas, and help solve problems with fellow engineers.
  https://esp32.com/
- The ESP Journal –Best Practices, Articles, and Notes from Espressif folks.
  https://blog.espressif.com/
- See the tabs SDKs and Demos, Apps, Tools, AT Firmware.
  https://espressif.com/en/support/download/sdks-demos

8.3 Products

- ESP32 Series SoCs –Browse through all ESP32 SoCs.
  https://espressif.com/en/products/socs?id=ESP32

- ESP32 Series Modules –Browse through all ESP32-based modules.
  https://espressif.com/en/products/modules?id=ESP32
- ESP32 Series DevKits –Browse through all ESP32-based devkits.
  https://espressif.com/en/products/devkits?id=ESP32
- ESP Product Selector –Find an Espressif hardware product suitable for your needs by comparing or applying filters.
  https://products.espressif.com/#/product-selector

8.4 Contact Us

- See the tabs Sales Questions, Technical Enquiries, Circuit Schematic & PCB Design Review, Get Samples (Online stores), Become Our Supplier, Comments & Suggestions.
  https://espressif.com/en/contact-us/sales-questions

Chapter 9

Disclaimer and Copyright Notice

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

All third party’ s information in this document is provided as is with no warranties to its authenticity and accuracy.

No warranty is provided to this document for its merchantability, non-infringement, fitness for any particular purpose, nor does 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.