

# ESP32-Ethernet-Kit V1.2 Getting Started Guide

[\[中文\]](#)

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).



ESP32-Ethernet-Kit V1.2 Overview (click to enlarge)

## What You Need

- [ESP32-Ethernet-Kit V1.2 board](#)
- USB 2.0 A to Micro B Cable

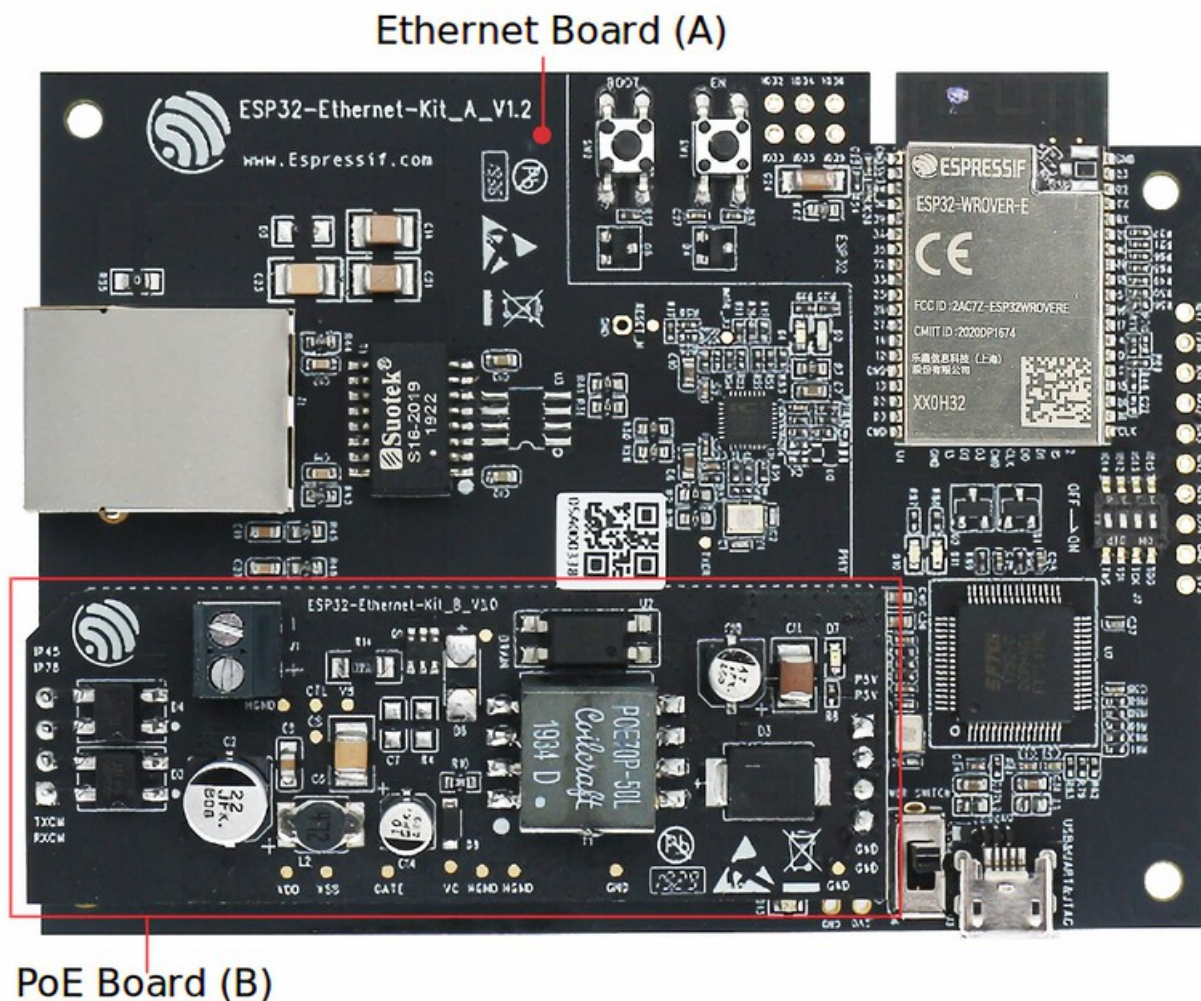
- Computer running Windows, Linux, or macOS

You can skip the introduction sections and go directly to Section [Start Application Development](#).

## Overview

ESP32-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.

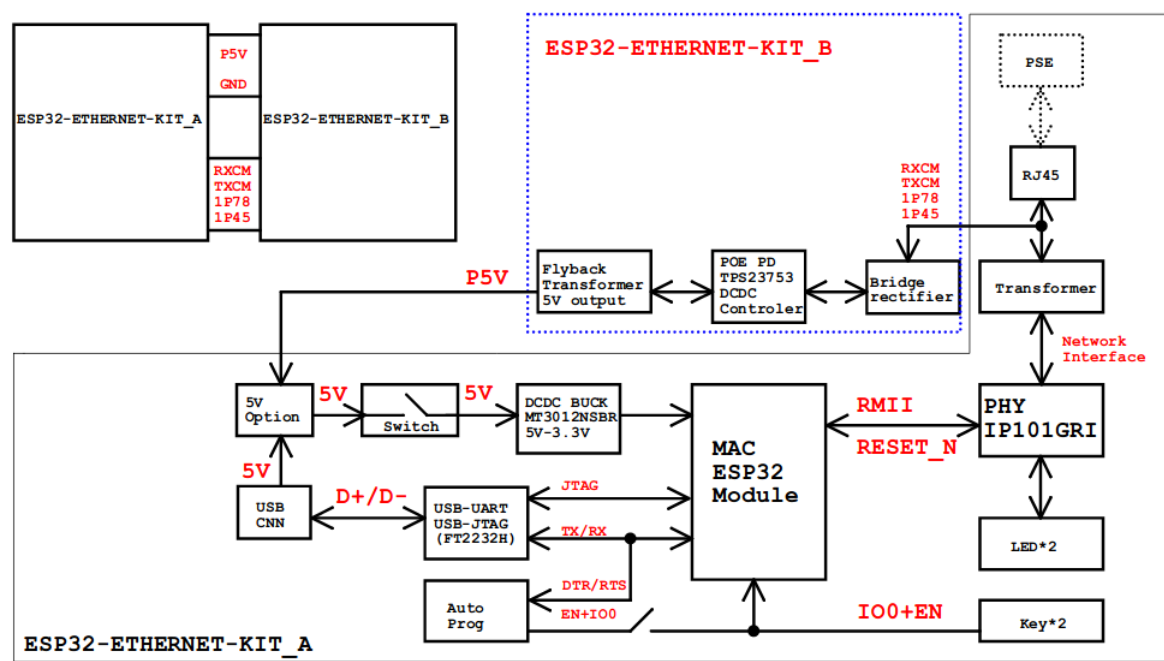


ESP32-Ethernet-Kit V1.2 (click to enlarge)

For the application loading and monitoring, the Ethernet board (A) also features 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.

# Functionality Overview

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



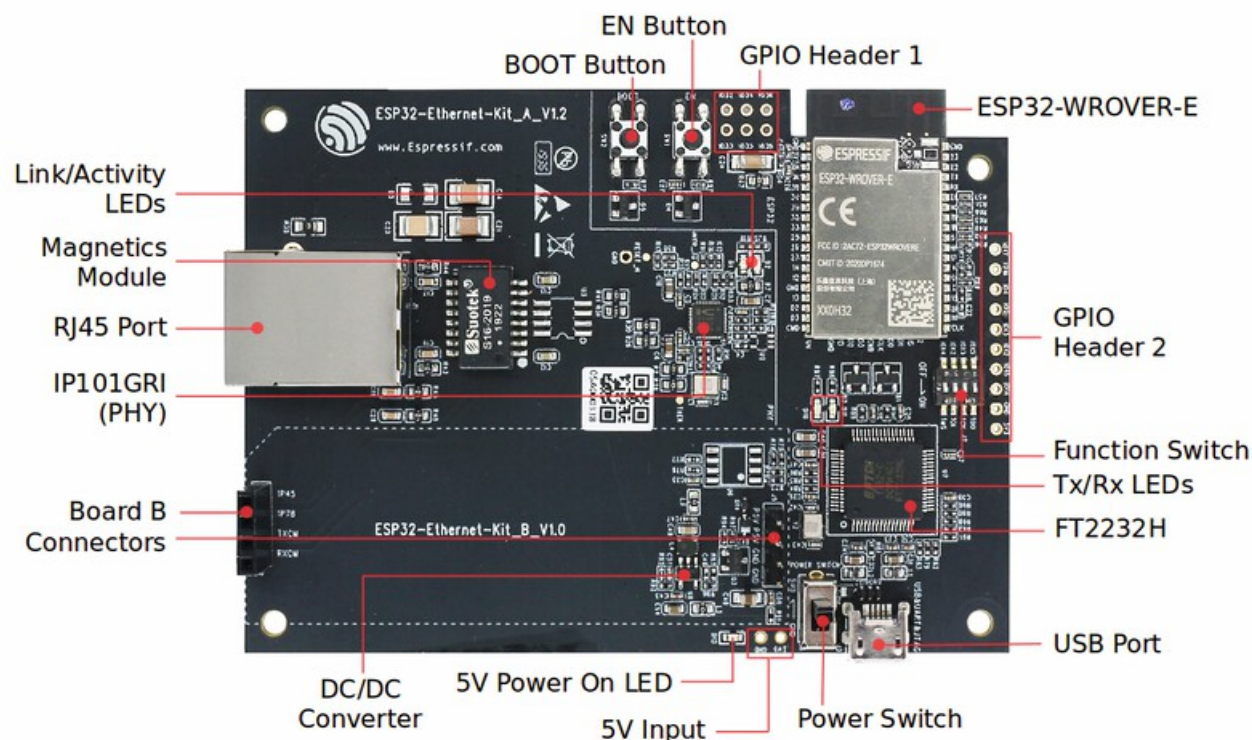
ESP32-Ethernet-Kit block diagram (click to enlarge)

# Functional Description

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



# Ethernet Board (A)



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.

Key Component	Description
ESP32-WROVER-E	This ESP32 module features 64-Mbit PSRAM for flexible extended storage and
GPIO Header 2	Five unpopulated through-hole solder pads to provide access to selected GPIOs
Function Switch	A 4-bit DIP switch used to configure the functionality of selected GPIOs of ESP
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
USB Port	USB interface. Power supply for the board as well as the communication interfa
Power Switch	Power On/Off Switch. Toggling the switch to <b>5V0</b> position powers the board on
5V Input	The 5 V power supply interface can be more convenient when the board is oper
5V Power On LED	This red LED turns on when power is supplied to the board, either from USB or
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 <a href="#">PoE board (B)</a>

Key Component	Description
IP101GRI (PHY)	The physical layer (PHY) connection to the Ethernet cable is implemented using
RJ45 Port	Ethernet network data transmission port.
Magnetics Module	The Magnetics are part of the Ethernet specification to protect against faults an
Link/Activity LEDs	Two LEDs (green and red) that respectively indicate the “Link” and “Activity” sta
BOOT Button	Download button. Holding down <b>BOOT</b> and then pressing <b>EN</b> initiates Firmwar
EN Button	Reset button.
GPIO Header 1	This header provides six unpopulated through-hole solder pads connected to sp

### **Note**

Automatic firmware download is supported. If following steps and using software described in Section [Start Application Development](#), users don't 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.



ESP32-Ethernet-Kit - PoE board (B) layout (click to enlarge)

Key Component	Description
Board A Connector	Four female (left) and four male (right) header pins for connecting the PoE I
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

### RMII Clock Selection

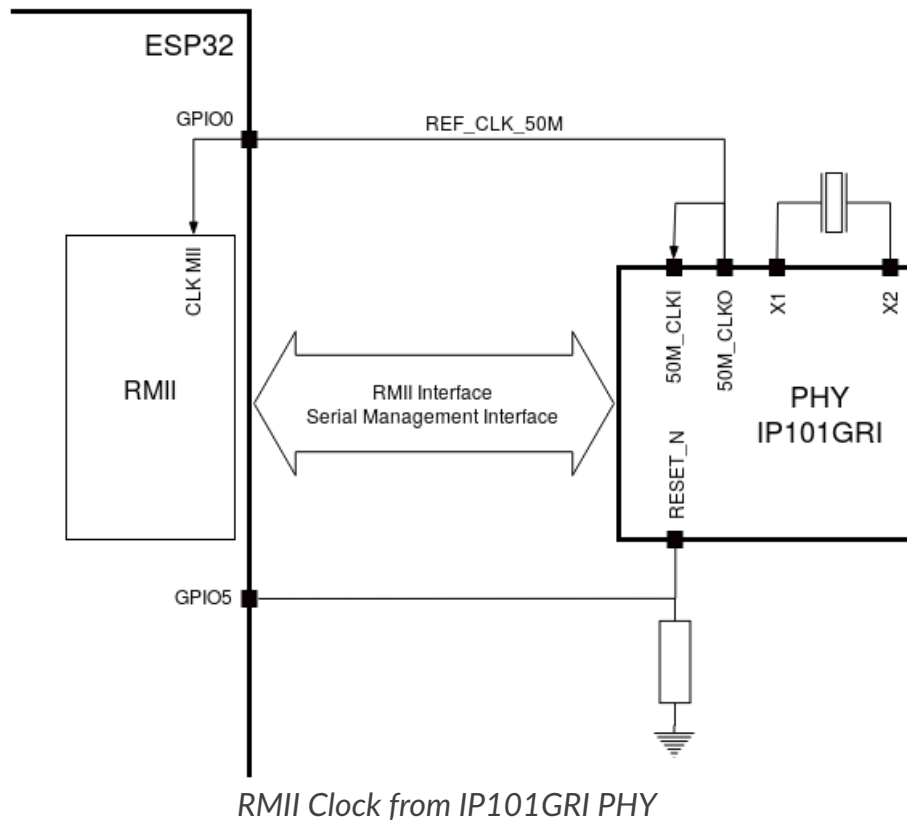
The ethernet MAC and PHY under RMII working mode need a common 50 MHz reference clock (i.e. RMII clock) that can be provided either externally, or generated from internal ESP32 APLL (not recommended).

#### 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

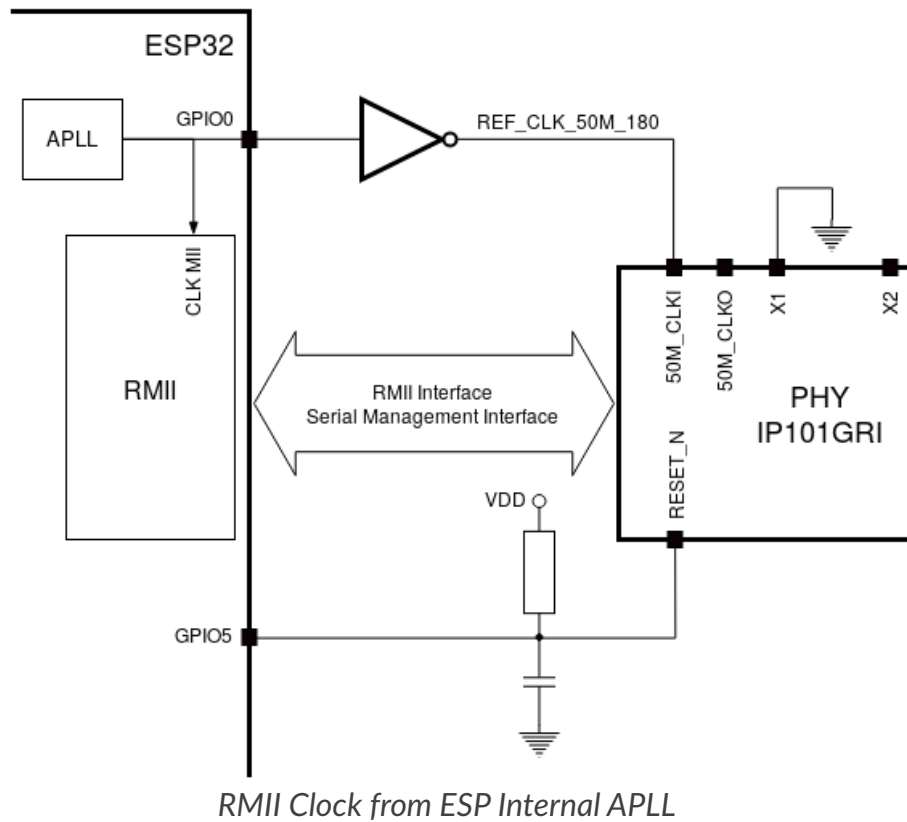
By default, the ESP32-Ethernet-Kit is configured to provide RMII clock for the IP101GRI PHY's 50M\_CLKO output. The clock signal is generated by the frequency multiplication of 25 MHz crystal connected to the PHY. For details, please see the figure below.



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 system. Otherwise ESP32 may enter download mode (when the clock signal of REF\_CLK\_50M is at a high logic level during the GPIO0 power-up sampling phase).

### **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 V1.2 Ethernet board \(A\) schematic](#), sheet 2, location D2. Please note that if the APLL is already used for other purposes (e.g. I2S peripheral), then you have no choice but use an external RMII clock.

## 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)
<i>RMII Interface</i>		
1	GPIO21	TX_EN
2	GPIO19	TXD[0]
3	GPIO22	TXD[1]
4	GPIO25	RXD[0]



No.	ESP32 Pin (MAC)	IP101GRI (PHY)
5	GPIO26	RXD[1]
6	GPIO27	CRS_DV
7	GPIO0	REF_CLK
<i>Serial Management Interface</i>		
8	GPIO23	MDC
9	GPIO18	MDIO
<i>PHY Reset</i>		
10	GPIO5	Reset_N

### Note

The allocation of all pins under the ESP32's *RMII Interface* is fixed and cannot be changed either through IO MUX or GPIO Matrix. REF\_CLK can only be selected from GPIO0, GPIO16 or GPIO17 and it can not be changed through 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 "Comments".

No.	ESP32 Pin	Comments
1	GPIO17	See note 1

No.	ESP32 Pin	Comments
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 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-E	IP101GRI	UART	JTAG	GPIO	Comments
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	

ESP32-WROVER-E	IP101GRI	UART	JTAG	GPIO	Comments
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 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 V1.2](#) ([click to enlarge](#)).

## 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.

## Other Versions of ESP32-Ethernet-Kit

- [ESP32-Ethernet-Kit V1.0 Getting Started Guide](#)
- [ESP32-Ethernet-Kit V1.1 Getting Started Guide](#)

## 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](#)
- [Hardware Reference](#)

For other design documentation for the board, please contact us at [sales@espressif.com](mailto:sales@espressif.com).

[Provide feedback about this document](#)