

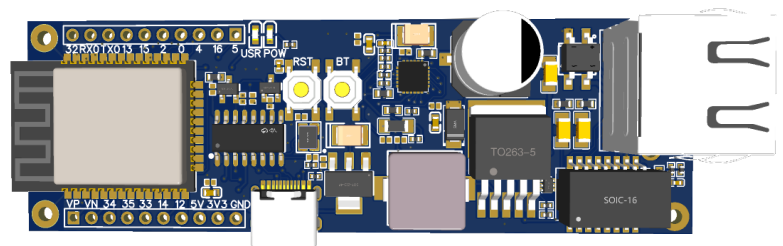
## ESP32 POWER OVER ETHERNET ACTIVE BOARD DATASHEET

### Introduction

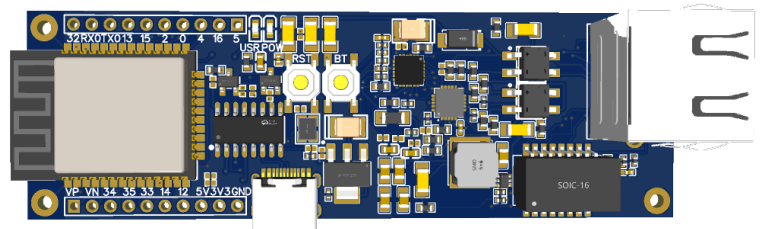
ESP32-Stick is a line of low-cost development boards for ESP32 with Ethernet support. The boards use LAN8720A chip for ethernet and CH340G USB-UART converter for communication with PC and programming. ESP32-Stick-PoE-A provides active Power-Over-Ethernet support, ESP32-Stick-PoE-P provides passive Power-Over-Ethernet support(5-57V). Programming is performed via USB connector.

### Features

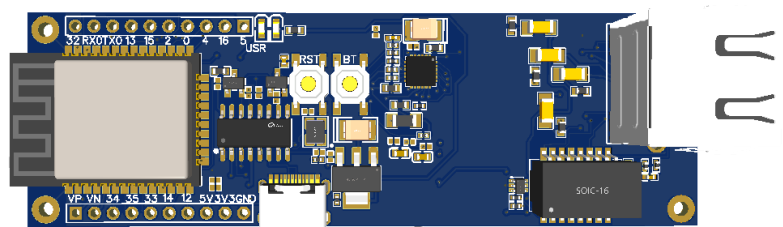
- ESP32-Wroom module.
- USB-C connector.
- 17 GPIO pins are available.
- UART pins are available.
- LAN8720A chip for Ethernet.
- Si3404 for POE(POE-A)
- LM2596HV for POE(POE-P)
- CH340G USB-UART converter.
- USER-Led (GPIO2).
- Reset button and User button(GPIO0).
- Accepts power through:
  - USB
  - External source (1.8V-5V)
  - Active POE(POE-A)
  - Passive POE(POE-P)



Esp32-Stick-PoE-P

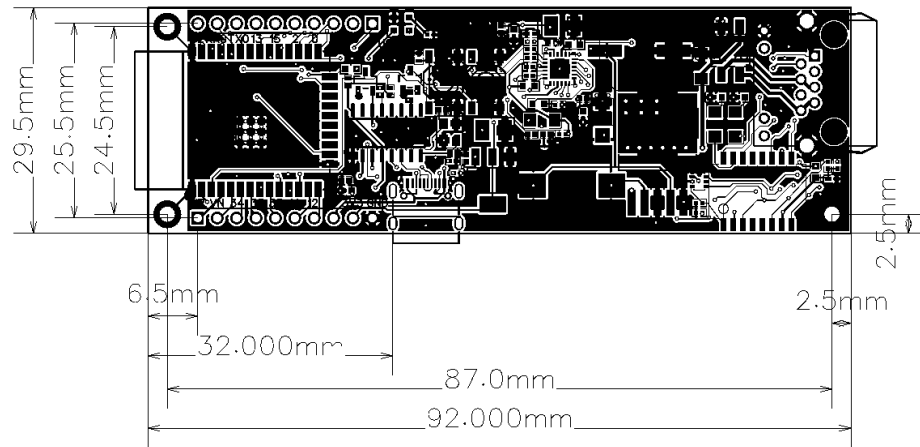


Esp32-Stick-PoE-A

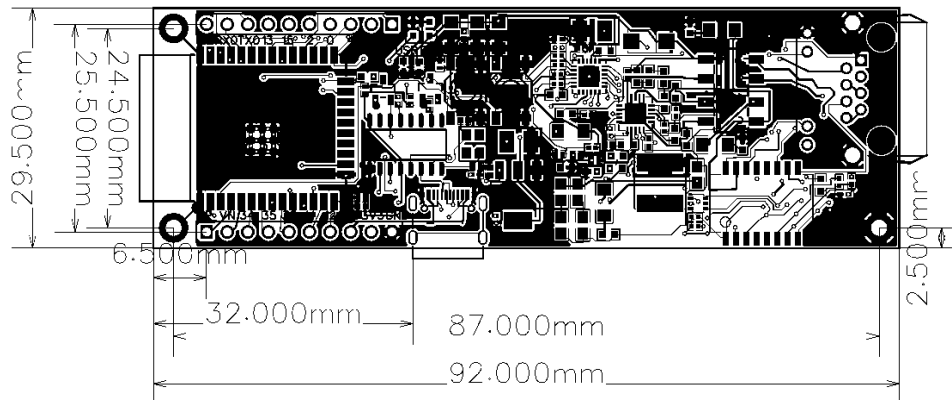


Esp32-Stick-Eth

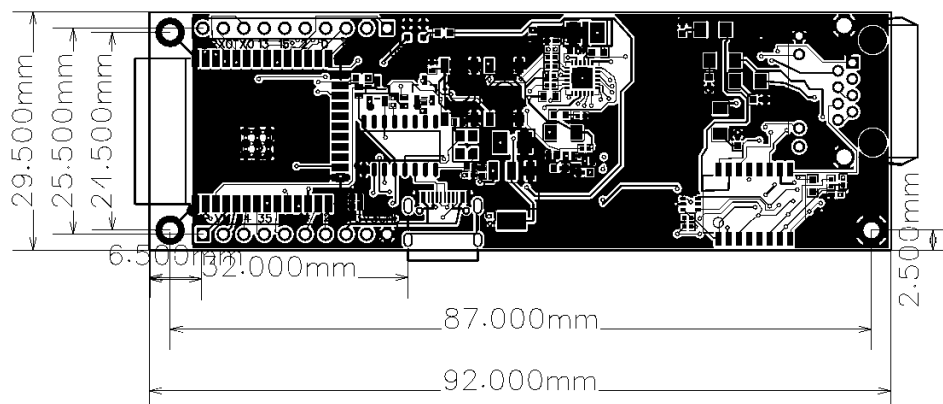
## Dimensions



**Picture 1: Esp32-Stick-PoE-P Dimensions**

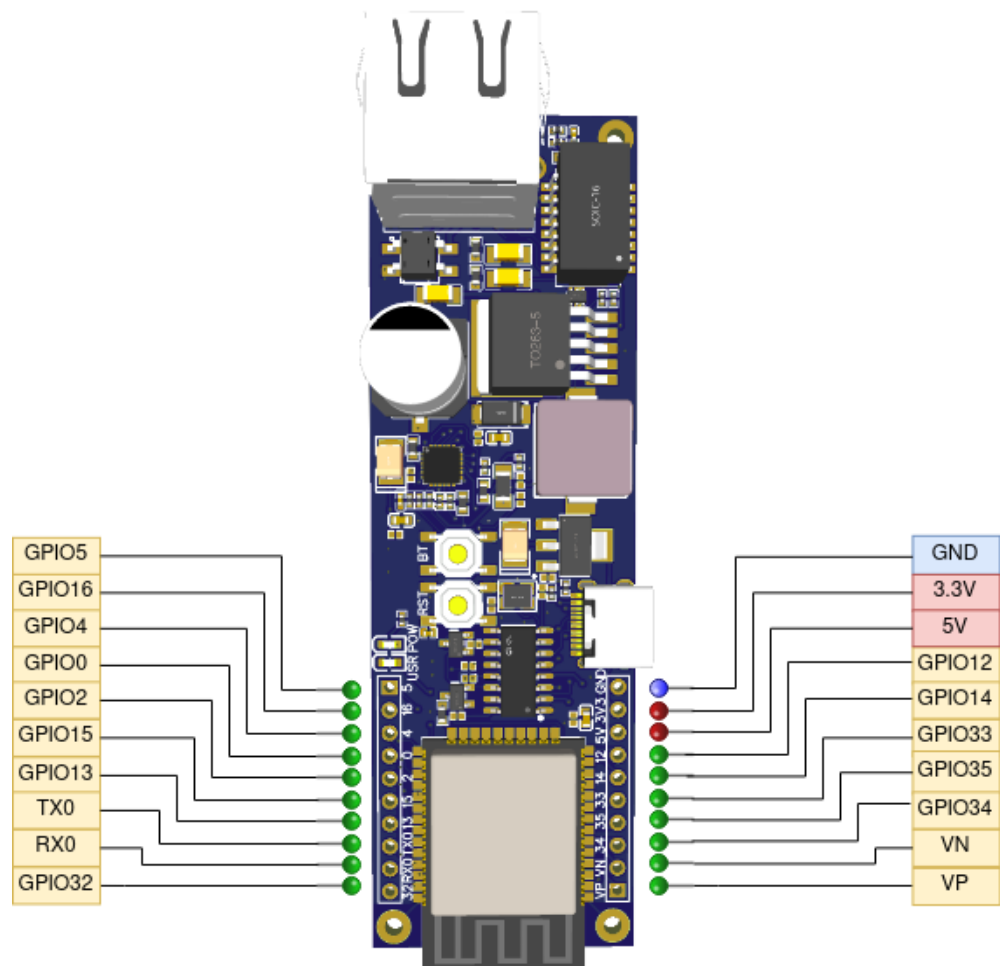


**Picture 2: Esp32-Stick-PoE-A Dimensions**

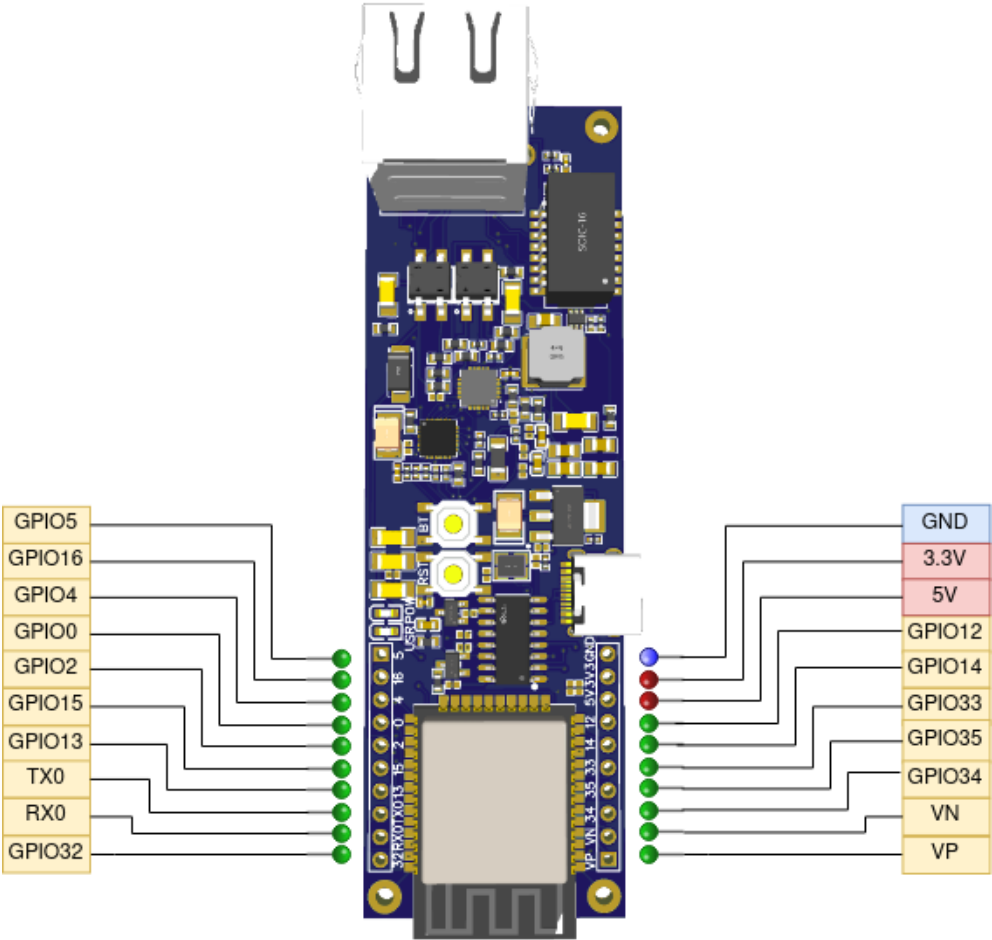


**Picture 3: Esp32-Stick-Eth Dimensions**

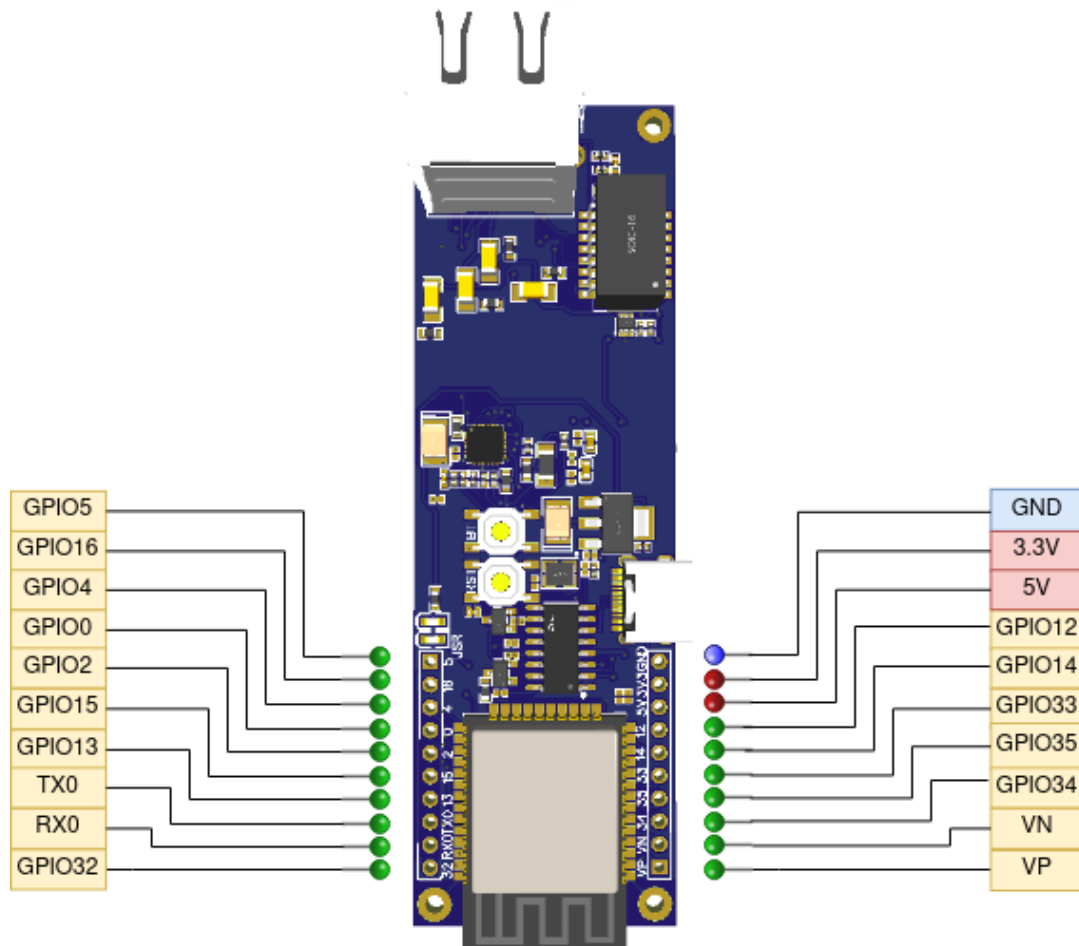
Pinouts



Picture 4: Esp32-Stick-PoE-P pinout



Picture 5: Esp32-Stick-PoE-A pinout

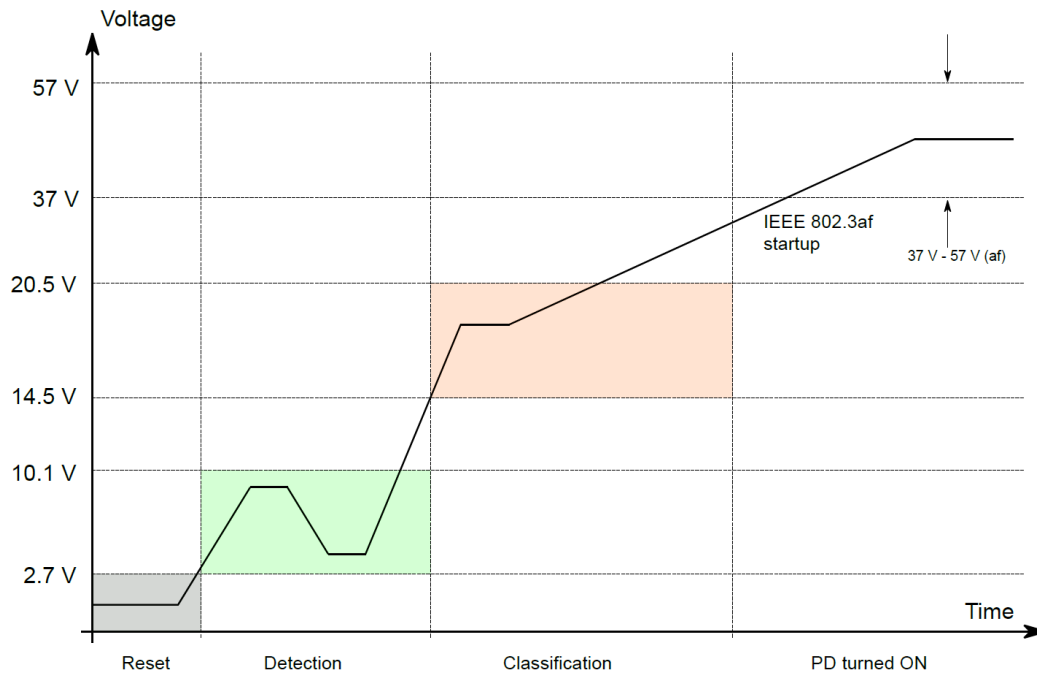


Picture 5: Esp32-Stick-Eth pinout

### Active Power over Ethernet(Esp32-Stick-PoE-A)

Power over Ethernet technology facilitates powering a Esp32-POE-A device over the same Ethernet cable as the data traffic. Compared to non-PoE devices, PoE devices feature with flexibility that allow you to easily place endpoints, sensors and drivers anywhere in the LAN or MAN, even places where it might be difficult to run a power outlet. The device is compatible with IEEE 802.3af standard (PoE can provide up to 15 W of DC power) and later IEEE 802.3at standard (PoE+ can provide up to 30 W of DC power). The POE-Jumper should be on to supply 3.3 V to ESP32 via POE. Maximum continuous operating current consumption is 600 mA.

When the Si3404 is connected via Ethernet cable to a PSE-enabled Ethernet switch, it has to provide a characteristic resistance ( $\sim 25\text{ k}\Omega$ ) to the PSE in a given voltage range (2.7–10.1 V). This is called detection. After the PSE detects the PD, the PSE increases the voltage above the classification threshold 14.5 V. Then, the PD provides the classification current to inform the PSE about its required power class (Class 1, 2, 3, or 4). Type 1 PSEs cannot provide enough power for a Class 4 PD. Type 2 PSEs have additional voltage steps before switching on the PD. After an initial classification voltage pulse, the Type 2 PSE reduces the voltage below the mark threshold level (10 V) then raises it up again to the Class event range. Last, before switching ON the dc-dc, it reduces the voltage again.



### IMPORTANT NOTICE!

There is no galvanic isolation, so the board should not be connected to POE while programming. Disconnect POE jumper or use non-POE cable and supply power via USB.

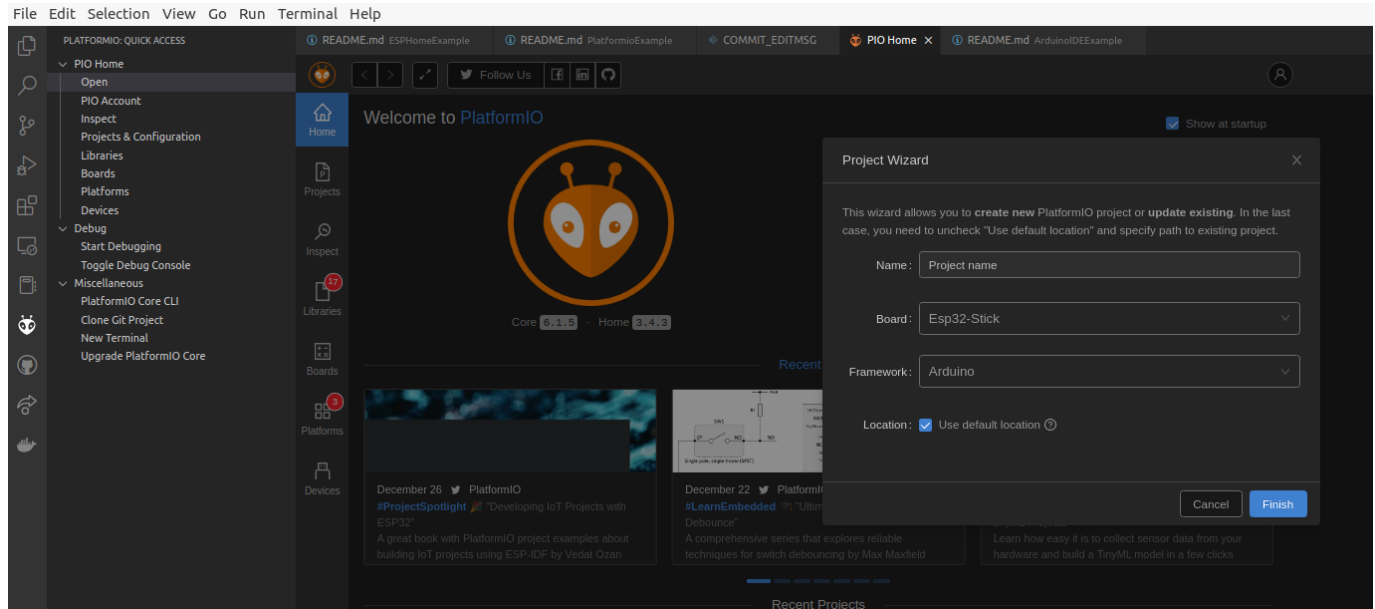
## Passive Power over Ethernet(Esp32-Stick-PoE-P)

Passive Power over Ethernet standard does not require initialization, voltage is just supplied via Ethernet pins. Voltage range is not defined, but there are some standard values like 24V and 48V. Esp32-Stick-PoE-P can work with a wide range of input voltages(5-57V).

## Programming options

Using Vscode Platformio and Arduino/Espressif IoT Development Framework:

- 1) Open Vscode and click on the Platformio extension 'ant' icon in the left menu of the left bar (extensions sometimes take longer to load) and click on 'Open' in the 'PIO Home' menu, which will bring up the 'PIO Home' tab where you can create a new project via 'Quick Access'. This will launch the 'Project Wizard' and in it you set Name, Board: ESP32-Stick, Framework: 'Espressif IoT Development Framework' or 'Arduino'. If you don't see the 'ESP32-Stick' board, go to the Platform installation in this document.



- 2) Write your code.
- 3) Connect the computer and the Esp32-Stick board with a USB-C cable.

### IMPORTANT NOTICE!

There is no galvanic isolation, so the board should not be connected to PoE while programming. While programming use non-POE cable and supply power via USB.

- 4) [Establish Serial Connection with ESP32](#).
- 5) If the PC did not recognize the connected USB device, install the driver [CH340 USB-UART drivers and software](#) or from the [project repository](#).
- 6) After connection press ctrl-alt-u (upload) to compile and upload the code.

## Uploading example project

For example project in Vscode IDE and Platformio extension go to:

<https://github.com/allexoK/Esp32-Stick-Platformio-Examples>

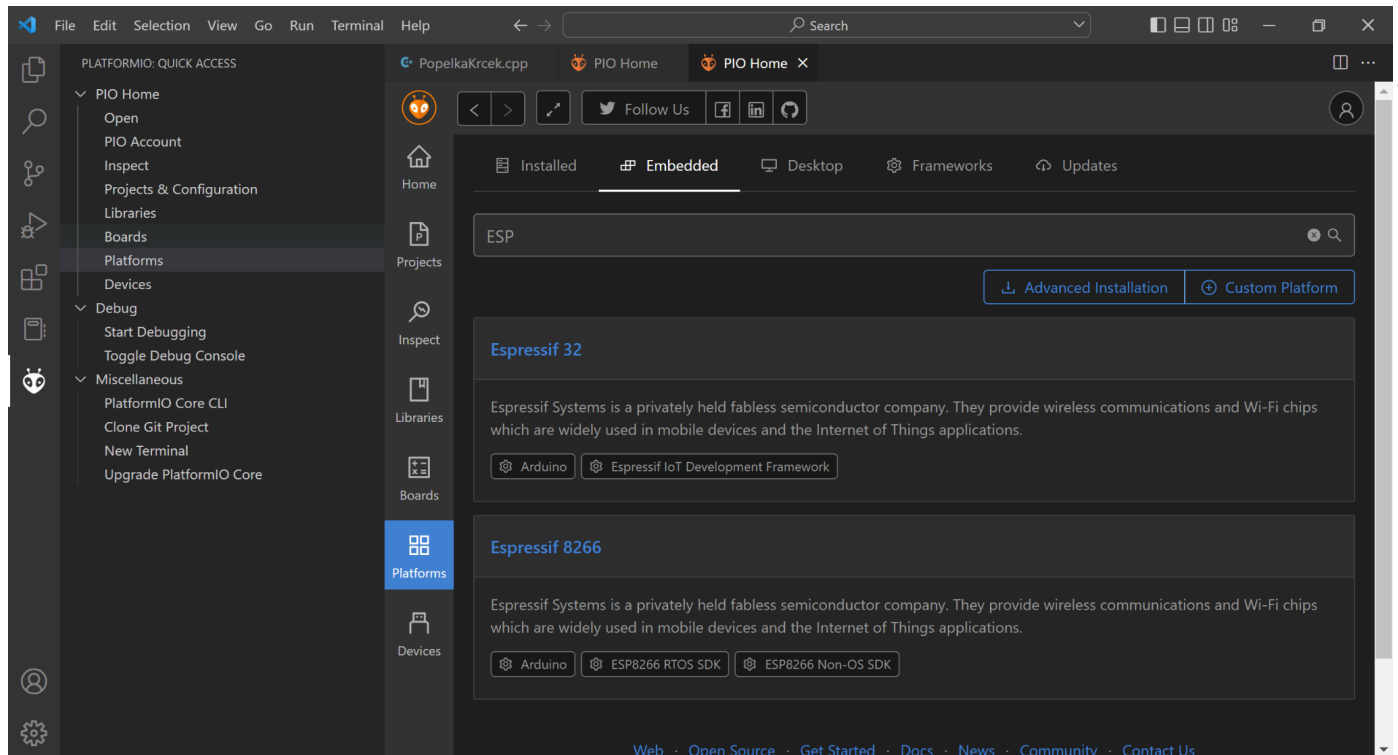
For example project in Arduino IDE go to:

<https://github.com/allexoK/Esp32-Stick-Arduino-Examples>

## Platform installation

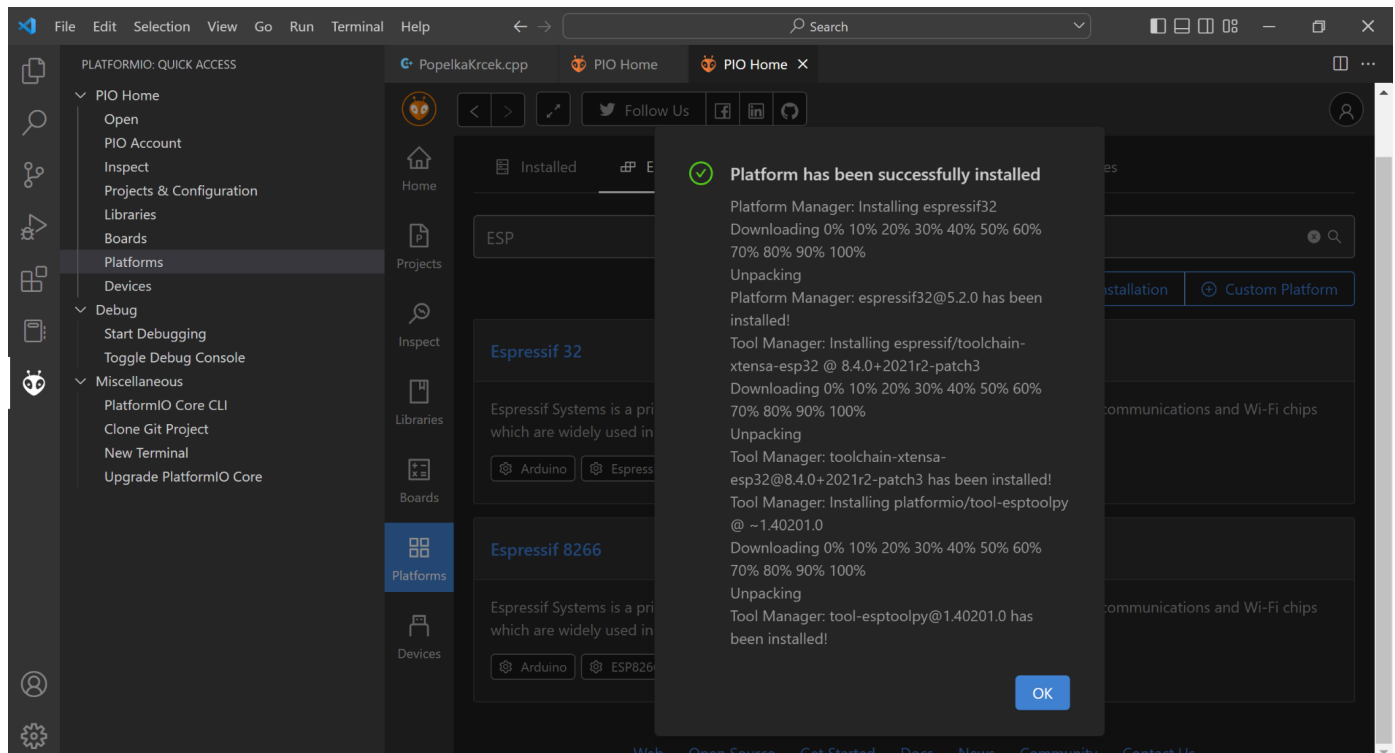
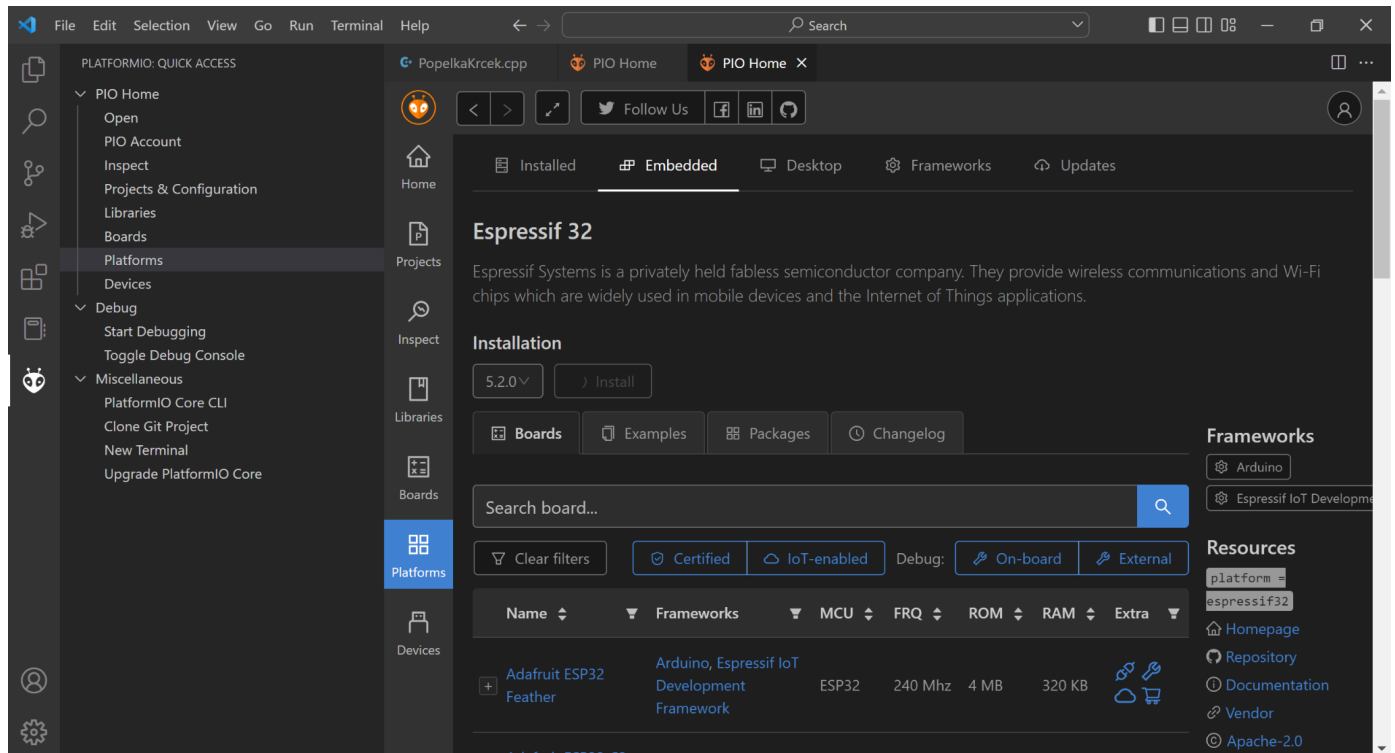
In the case, you don't see the 'Esp32-Stick' board in the Platformio 'Project Wizard' or if do you have new Visual Studio Code installation, install board/platform step by step:

- 1) Open Vscode and click on the Platformio extension 'ant' icon in the left menu of the left bar (extensions sometimes take longer to load) and click on 'Platforms'. Type 'ESP' and click Espressif 32 blue text.



- 2) Install Espressif 32 and wait until installation finishes



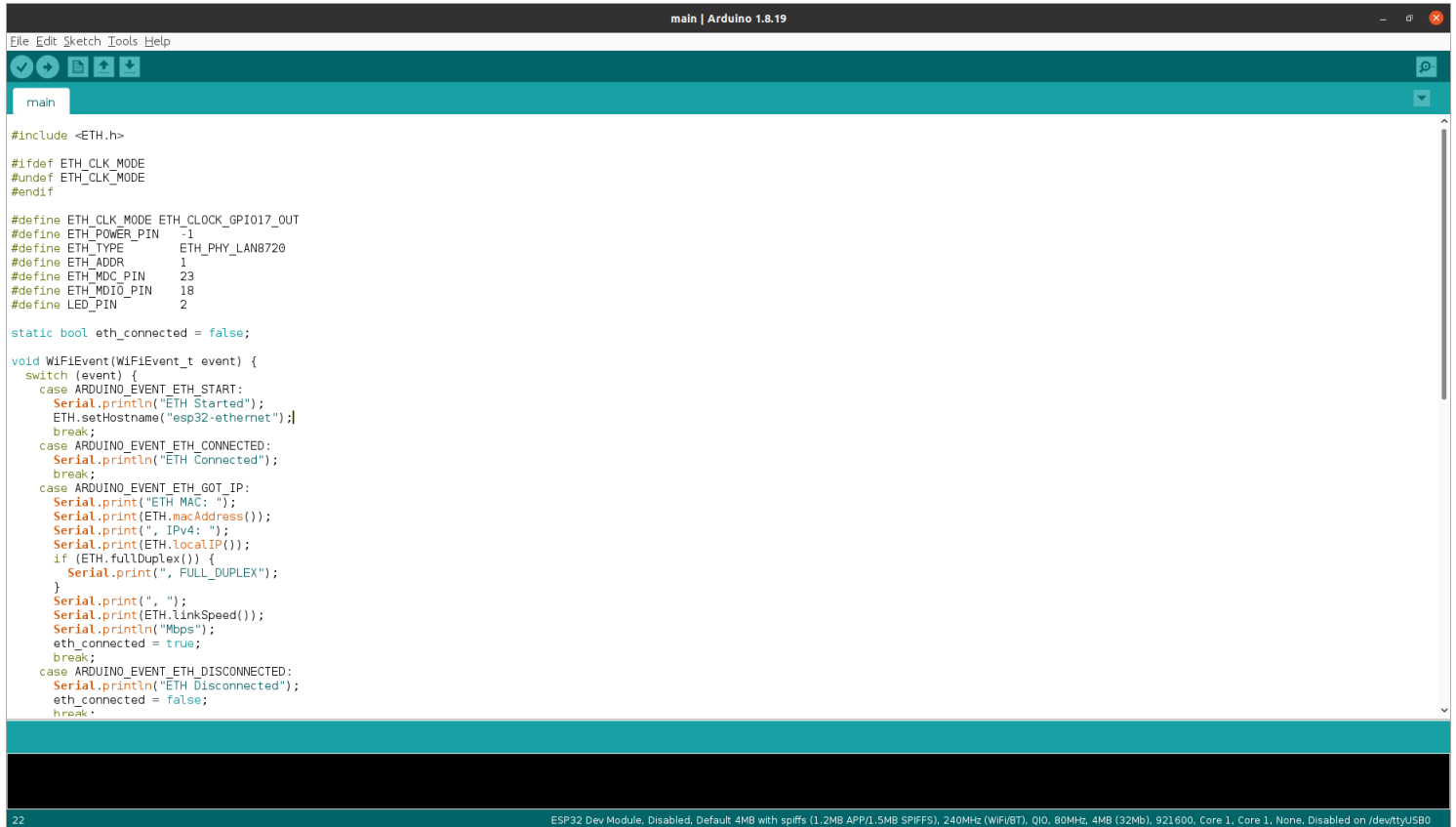


- 3) Copy 'esp32-stick.json' file to the '.platformio' directory in your home folder:
  - a) Linux: .platformio/platforms/espressif32/boards
  - b) Windows: C:\Users\user\.platformio\platforms\espressif32\boards

Recent	Name	Size	Modified
Starred	alksesp32.json	757 bytes	12 lis 2021
Home	az-delivery-devkit-v4.json	789 bytes	12 lis 2021
Desktop	bpi-bit.json	679 bytes	12 lis 2021
Documents	briki_abc_esp32.json	860 bytes	12 lis 2021
Downloads	briki_mbcwb_esp32.json	857 bytes	12 lis 2021
Music	d-duino-32.json	775 bytes	12 lis 2021
Pictures	esp32-c3-devkitm-1.json	731 bytes	12 lis 2021
Videos	esp32cam.json	842 bytes	12 lis 2021
Trash	esp32dev.json	760 bytes	12 lis 2021
Other Locations	esp32-devkitlipo.json	820 bytes	12 lis 2021
	esp32dolt-devkit-v1.json	745 bytes	12 lis 2021
	esp32dolt-espduino.json	737 bytes	12 lis 2021
	esp32-evb.json	782 bytes	12 lis 2021
	esp32-gateway.json	798 bytes	12 lis 2021
	esp32-poe.json	727 bytes	12 lis 2021
	esp32-poe-a.json	703 bytes	25 lis
	esp32-poe-iso.json	743 bytes	12 lis 2021
	esp32-pro.json	727 bytes	12 lis 2021
	esp32-s2-kaluga-1.json	821 bytes	12 lis 2021
	esp32-s2-saola-1.json	732 bytes	12 lis 2021
	esp32-stick.json	659 bytes	15:43
	"esp32-stick.json" selected (659 bytes)		

If you use Arduino IDE:

1)open your project(or example project from <https://github.com/allexoK/Esp32-Stick-Arduino-Examples>)



```
#include <ETH.h>

#ifdef ETH_CLK_MODE
#undef ETH_CLK_MODE
#endif

#define ETH_CLK_MODE ETH_CLOCK_GPIO17_OUT
#define ETH_POWER_PIN -1
#define ETH_TYPE ETH_PHY_LAN8720
#define ETH_ADDR 1
#define ETH_MDC_PIN 23
#define ETH_MDIO_PIN 18
#define LED_PIN 2

static bool eth_connected = false;

void WiFiEvent(WiFiEvent_t event) {
  switch (event) {
    case ARDUINO_EVENT_ETH_START:
      Serial.println("ETH Started");
      ETH.setHostname("esp32-ethernet");
      break;
    case ARDUINO_EVENT_ETH_CONNECTED:
      Serial.println("ETH Connected");
      break;
    case ARDUINO_EVENT_ETH_GOT_IP:
      Serial.print("ETH MAC: ");
      Serial.print(ETH.macAddress());
      Serial.print(", IPv4: ");
      Serial.print(ETH.localIP());
      if (ETH.fullDuplex()) {
        Serial.print(", FULL_DUPLEX");
      }
      Serial.print(", ");
      Serial.print(ETH.LinkSpeed());
      Serial.println("Mbps");
      eth_connected = true;
      break;
    case ARDUINO_EVENT_ETH_DISCONNECTED:
      Serial.println("ETH Disconnected");
      eth_connected = false;
      break;
  }
}
```

22 ESP32 Dev Module, Disabled, Default 4MB with spiiffs (1.2MB APP/1.5MB SPIFFS), 240MHz (WiFi/BT), QIO, 80MHz, 4MB (32Mb), 921600, Core 1, Core 1, None, Disabled on /dev/ttyUSB0

2)Go to Tools->Board->Esp32 Arduino->Esp32 Dev Module. This will set the board to 'Esp32 Dev Module'.

3)Upload your code using upload button.