

Join GitHub today

[Dismiss](#)

GitHub is home to over 28 million developers working together to host and review code, manage projects, and build software together.

[Sign up](#)

Branch: master ▾

[esp-who](#) / [docs](#) / [en](#) / [get-started](#) / [ESP-EYE_Getting_Started_Guide.md](#)[Find file](#)[Copy path](#) [NatashaLi-ESPRESSIF](#) Doc/update esp eye guides

5970bcc 14 days ago

[1 contributor](#)

191 lines (106 sloc) 8.29 KB

ESP-EYE Getting Started Guide

[\[中文\]](#)

What You Need

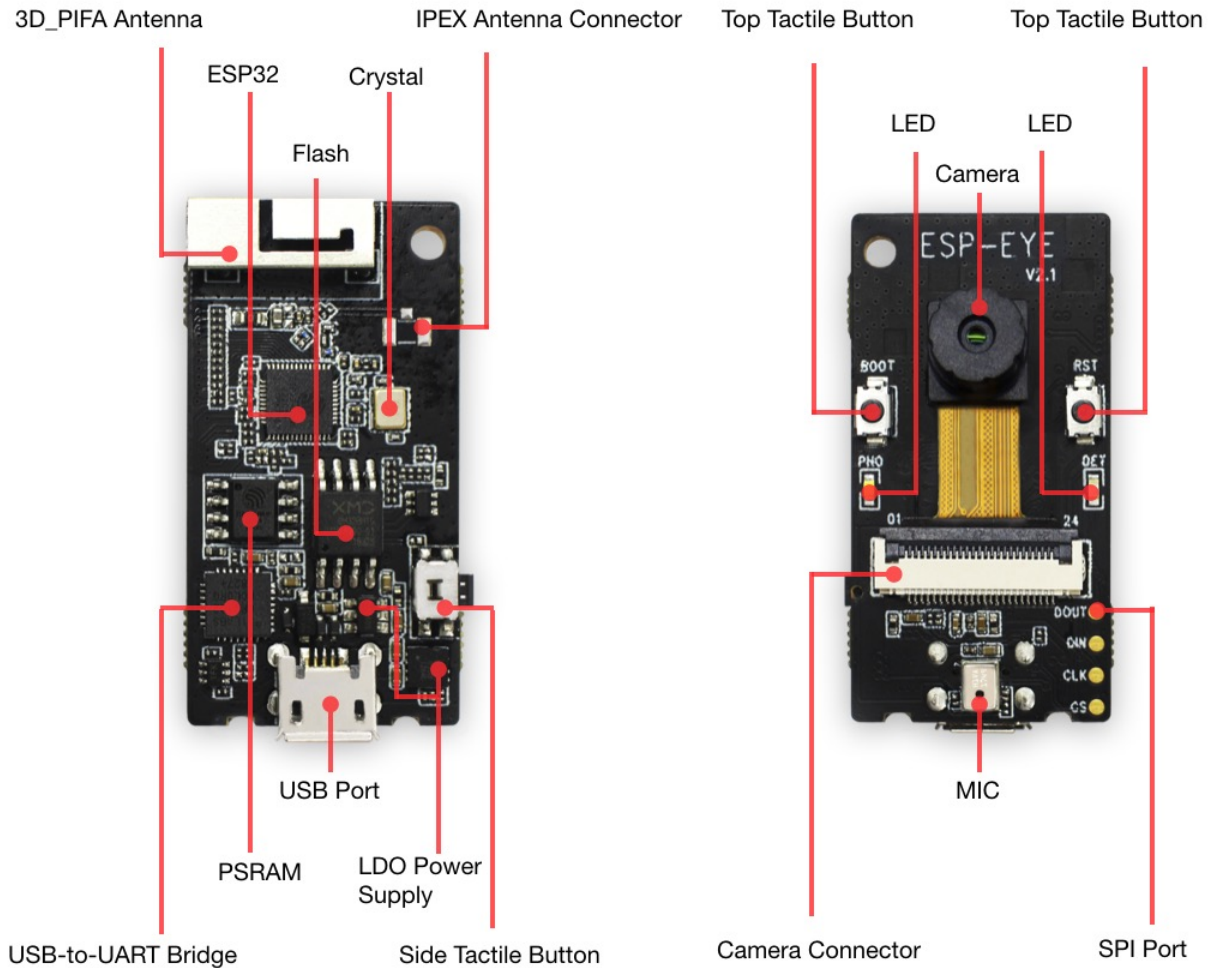
- 1 × ESP-EYE V2.1 board
- 1 × Micro USB cable
- 1 × PC with Windows, Linux or Mac OS

Overview

ESP-EYE is an ESP32-based development board that integrates a digital microphone, an 8 MB PSRAM and a 4 MB flash, while also providing an external 2-Megapixel camera. These features make the board ideal for applications relating to face detection, face recognition and speech recognition. Besides, the board can also support image transmission over Wi-Fi and debugging through a Micro USB port, which enables the development of advanced AI solutions.

Hardware Preparation

The list and figure below describe the key components, interfaces and functions of the ESP-EYE development board:



- **3D_PIFA Antenna**

A 3D PIFA antenna. With the R14 resistor users can select the external IPEX antenna, whereas with the R15 resistor they can select the 3D antenna.

- **IPEX Connector**

Used for connecting the external IPEX antenna. With the R14 resistor users can select the external IPEX antenna, whereas with the R15 resistor they can select the 3D antenna.

- **ESP32 Chip**

A 2.4 GHz Wi-Fi and Bluetooth combo chip.

- **Crystal Oscillator**

Provides an external clock to ESP32.

- **Flash & PSRAM**

Provides memory for storing applications.

- **CP2102 USB-to-UART Chip**

Converts the USB signals to UART signals.

- **USB Port**

Provides power supply to the whole system.

- **LDO Power Supply**

Provides the required power supply to the ESP32 chip, camera and LED indicators.

- **Side Tactile Button**

A function key.

- **Top Tactile Button**

Reset/Boot button. We recommend that you do not configure this button for other uses.

- **LED Indicators**

The board has a red and a white indicator. Different flashing combinations of the red and white indicators reflect the different statuses of the board, e.g. waking up, networking, face detection, face recognition, face enrollment and face recognition.

- **Camera**

An external 2-Megapixel camera module for face detection, face recognition and Face ID enrollment.

- **Camera Connector**

Used for connecting the external camera.

- **MIC**

A digital microphone for voice control functions.

- **SPI Port**

A reserved port for data transmission.

Software Development

ESP-EYE supports firmware downloading with a Linux, MacOS or Windows PC. At present, users must set up a toolchain in the development environment before starting software development.

Preparation

- Go through [Get Started](#) to set up the toolchain in your PC.
- Connect the ESP-EYE to your PC with a Micro USB cable.
- Launch your development-environment tool, such as Terminal (Linux/MacOS) or MinGW (Windows).

Getting ESP-WHO

To obtain a local copy, open your Terminal (such as Terminal in a Linux environment), go to the directory in which you want to store ESP-WHO, and clone the repository by using the `git clone` command:

```
git clone --recursive https://github.com/espressif/esp-who.git
```

ESP-WHO will be downloaded into an automatically-generated folder named `esp-who`.

Do not miss the `--recursive` option. If you have already cloned ESP-WHO without this option, please run another command to get all the submodules: `git submodule update --init --recursive`

Setting up a Path to ESP-WHO

Please follow the instruction described in [Setup Path to ESP-IDF](#) to configure the `IDF_PATH` variable to `esp-who/esp-idf`.

Downloading Firmware

This section describes the steps for downloading firmware to ESP-EYE (taking the Linux operating system as an example):

- Power up ESP-EYE by connecting it to your PC with a USB cable.
- Run `ls /dev/ttyUSB*` in your Terminal to check if the board has connected itself to your PC successfully. If the connection is successful, you will see a newly generated item named e.g. `/dev/ttyUSB0` on your list, after running the command.
- Go to an example directory, such as `cd esp-who/examples/single_chip/recognition_solution`.
- Run `make defconfig` to complete the default configuration.
- Run `make menuconfig` and go to `Serial flasher config -> Default serial port` to configure the device name, according to the name of the item you saw in the second step, e.g. `/dev/ttyUSB0`. Then, save it and exit.
- Run `make flash` to download the firmware.

Checking Logs

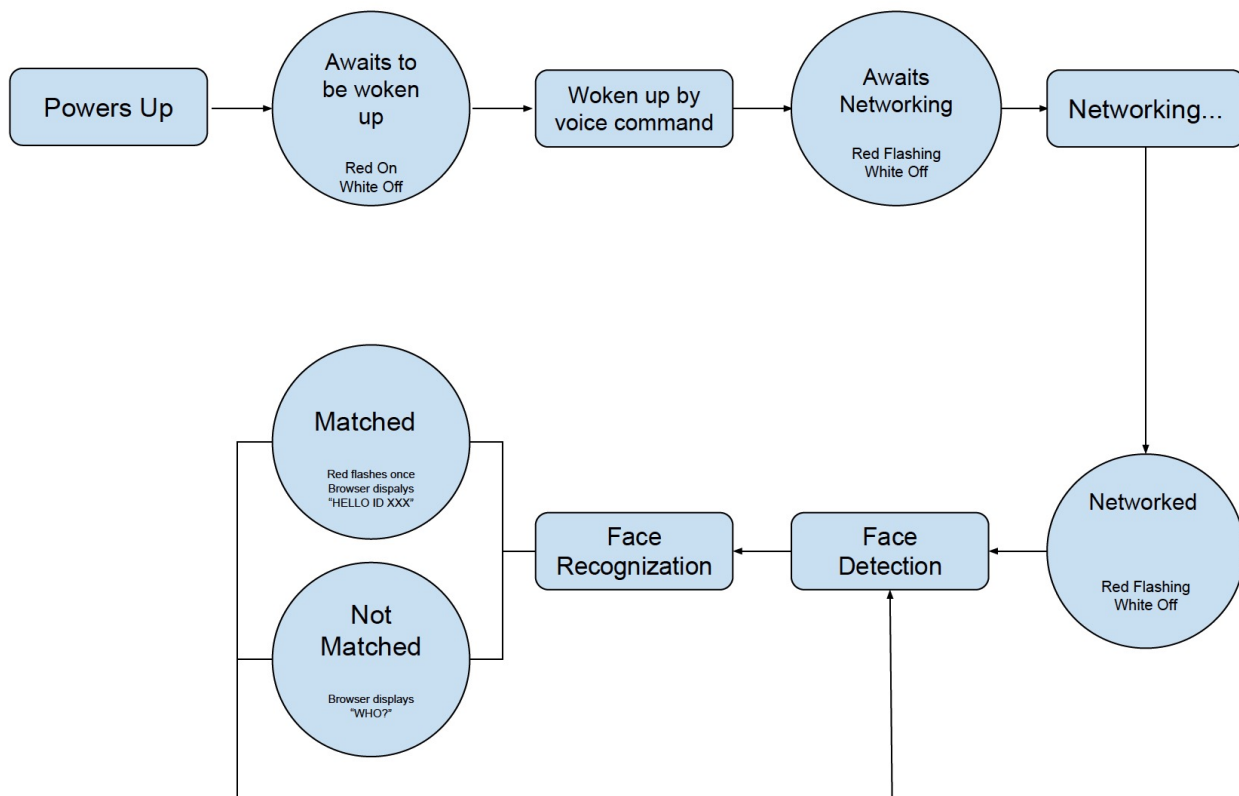
This section describes the steps for checking the logs on your Terminal (taking the Linux operating system as an example):

- Launch your Terminal.
- Run `make monitor`.

Note: The ESP-EYE development board will reboot after this operation.

Key Process

The figure below describes the workflow of ESP-EYE:



1. Voice Wake-up

ESP-EYE awaits to be woken up after powering up (Red LED on and white LED off). The board wakes up after recognizing the wake-up command "Hi Lexin", and then awaits for networking (Red LED flashing and white LED off). Subsequently, users can initiate the networking.

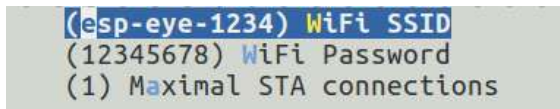
2. Networking

Users can connect their PCs or mobile phones to ESP-EYE's Wi-Fi (by default), with the following information:

- Username: esp-eye-xxxx (xxxx should be the board's MAC address)
- Password: not needed

Alternatively, users can also follow the steps below to configure the username and password of the board's Wi-Fi connection:

- Launch your Terminal.
- Run `make menuconfig` and complete the configuration, as instructed in the figure below:



```
(esp-eye-1234) WiFi SSID  
(12345678) WiFi Password  
(1) Maximal STA connections
```

Note: After reconfiguring the Wi-Fi username and password, you will have to restart from the point of downloading firmware.

3. Face Detection

ESP-EYE starts the face detection after networking. Users can see the real-time image captured by the board, through their browser (address: `192.168.4.1/face_stream`). During this step, the red LED is off and the white LED is on.

4. Face Recognition

After detecting a face, ESP-EYE will start the face recognition if there are any enrolled Face IDs stored in the board:

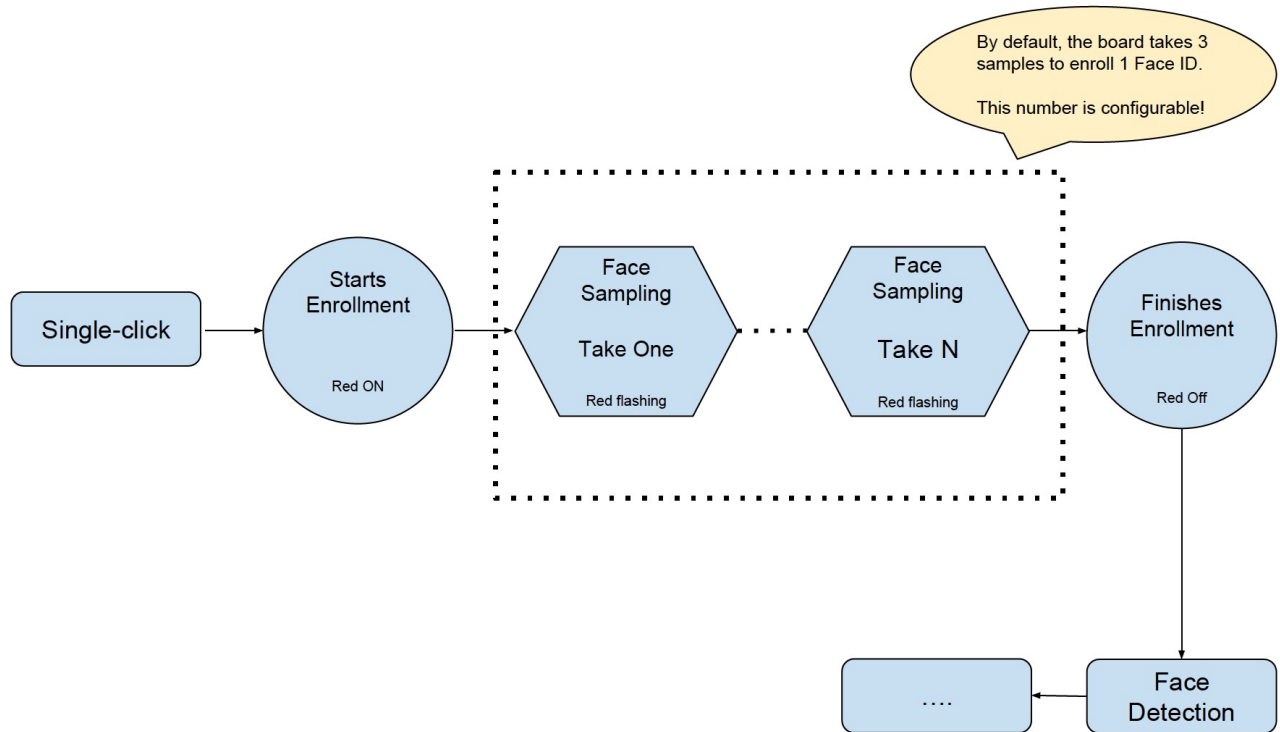
- When there is a match, the red LED on the board flashes once and the browser displays **HELLO ID XXX**.
- When there is no match, the board shows no signs and the browser displays **WHO?**.

If there is no enrolled Face ID, the board continues the face-detecting process. You should enroll at least one Face ID if you want to start face recognition.

5. Add/delete a Face ID

The users can add/delete a Face ID after the network is successfully established.

5.1 Add a Face ID



- Single-click the Side Tactile Button to enroll a new Face ID. At this point, the red LED is on and the browser displays **START ENROLLING**;
- Once you put a face in front of the camera, the face-sampling starts automatically. The red LED flashes whenever the board gets a face sample and the browser displays the ordinal number of the current face sample, i.e. **THE 1st SAMPLE** etc. By default, the board has to take three samples to add one Face ID. Users can configure the number of samples needed for one Face ID. (Please adjust your position/distance from the camera and try again if you cannot see the red LED flashing for some time).
- After the Face ID enrollment, the red LED on the board is off and the browser displays **ENROLLED FACE ID XXX**;
- The board enters Face Detection after the Face ID enrollment.

Currently, ESP-EYE can enroll up to 10 Face IDs. Please note that the maximum number of enrolled Face IDs can be configured according to how users allocate the flash memory. However, we recommend a number that is no greater than 30.

5.2 Delete a Face ID

- Double-click the Side Tactile Button to delete an existing Face ID.
- After that, the board deletes the earliest record of all the existing enrolled Face IDs. The white LED on the board flashes, and the browser displays: **XXX ID(S) LEFT**.

Troubleshooting

The board returns to the "awaiting to be woken up" status when there are network anomalies, such as "network disconnection" and "network timeout".