# HOW TO USE ESP8266 WITH ARDUINO IDE

**This document applies for the following products:**
**ESP8266-EVB; ESP8266-EVB-BAT; ESP8266-EVB-BAT-BOX**

Document revision B, February 2017



All boards produced by Olimex LTD are ROHS compliant

# DISCLAIMER

© 2017 Olimex Ltd. Olimex®, logo and combinations thereof, are registered trademarks of Olimex Ltd. Other product names may be trademarks of others and the rights belong to their respective owners.

The information in this document is provided in connection with Olimex products. No license, express or implied or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Olimex products.

The hardware designs of the development boards mentioned in this guide (ESP8266-EVB; ESP8266-EVB-BAT; ESP8266-EVB-BAT-BOX) are considered open source. Anyone can download the original schematics and board design files. The files describing the hardware are made with CadSoft's EAGLE PCB design software – the software has a free version suitable for studying any board schematic or board design.

The part of the software written by Olimex is released under GPL. However, the examples that we distribute are based on the SDK libraries and projects released by Espressif Systems. Make sure that you are acquainted with Espressif Systems' license agreement before using, publishing, or distributing any code based on our software.

It is possible that the pictures in this manual differ from the latest revision of the board.

The products described in this document are subject to continuous development and improvements. All particulars of the products and their use contained in this document are given by OLIMEX in good faith. However, all warranties implied or expressed, including but not limited to, implied warranties of merchantability or fitness for purpose, are excluded. This document is intended only to assist the reader in the use of the products. Olime Ltd. shall not be liable for any loss or damage arising from the use of any information in this document or any error or omission in such information or any incorrect use of the product.

These evaluation boards are intended for use in engineering development, demonstration, or evaluation purposes only and are not considered by Olimex to be a finished end-product fit for general consumer use. Persons handling the products must have electronics training and observe good engineering practice standards. As such, the goods being provided are not intended to be complete in terms of required design-, marketing-, and/or manufacturing-related protective considerations, including product safety and environmental measures typically found in end products that incorporate such semiconductor components or circuit boards.

Olimex currently deals with a variety of customers for products, and therefore our arrangement with the user is not exclusive. Olimex assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein.

## Table of Contents

# CHAPTER 1: PURPOSE OF THIS DOCUMENT

This document aims to assist beginners in the first-time programming of ESP8266-EVB, ESP8266-EVB-BAT, or ESP8266-EVB-BAT-BOX, via Arduino IDE. Experienced ESP8266 users would probably find this document redundant.

The document structure is as follows:

- All the required hardware and software for a basic setup is listed in "CHAPTER 2: REQUIREMENTS".
- In "CHAPTER 3: SETUP" there are specific details on how to set everything together.
- "CHAPTER 4: UPLOADING BLINKING LED" shows how to upload a simple blinking LED example to the board.
- "CHAPTER 5: FURTHER READING" provides links to more ESP8266-related resources.
- The document's revision and ordering information are available in "CHAPTER 6: ORDERING CODES AND DOCUMENT REVISION".

# CHAPTER 2: REQUIREMENTS

The minimal connection requires several prerequisites.

## 2.1 Hardware requirements

Note that you might already own all the necessary hardware listed below. For example, as long as you have a fitting 5V power supply unit or USB<->serial cable there is no need to purchase new ones.

### 2.1.1 ESP8266 board

The main component of the setup. You need one of the following three boards, which are suitable for beginners:

- ESP8266-EVB
- ESP8266-EVB-BAT
- ESP8266-EVB-BAT-BOX

*MOD-WIFI-ESP8266-DEV and MOD-WIFI-ESP8266 are not suitable for beginners with ESP8266. The main difficulty if you use one of those two boards would be changing the boot mode of the board. This would require cutting and soldering of SMT jumpers (or shortening pins via wires).*

### 2.1.2 Power supply adapter

The ESP8266-EVB board gets powered via the barrel power jack named "PWR". You need to apply exactly 5V DC.

The corresponding power supply plug (or female jack) that fits the DC barrel has 2.1mm inner diameter and 5.5mm outer diameter. The length of the male plug should be between 10mm and 14mm.

*We distribute affordable 220V<->5V adapters that fit the board's barrel jack. Search our web-shop for SY0605E and SY0605E-CHINA.*

### 2.1.3 USB <-> serial cable

You might already have such a cable. In case you don't, you would need to get such an adapter cable due to a couple of reasons:

- the voltage levels of board's RXD and TXD are lower than the voltage levels of computer's TXD and RXD (even if your computer has a native serial port). The adapter takes care of the level adjustment;

- most modern computers lack serial port.

The are two very important considerations to make, when selecting the cable that would work for you:

- the cable needs to fit the pins of UEXT connector of the board – if it has free female leads, for example;

- the cable has the proper driver support for the operating system of your personal computer.

*We distribute affordable USB to serial adapter cables with female leads. Search our web-shop for USB-SERIAL-CABLE-F and BB-CH340T.*

### 2.1.4 Personal computer

In order to install the Arduino IDE you need a computer which is able to:

- run Arduino IDE;

- work properly with your serial cable;

- access the Internet.

## 2.2 Software requirements

### 2.2.1 Driver for the USB<->serial cable

Contact the supplier of your USB to serial adapter cable for such drivers. After the drivers are installed make sure the cable is properly recognized by the operating system.

A properly installed serial cable should be listed as a COM port by the operating system.

For example, under Windows, in "Windows Device Manager" a properly installed cable gets listed like this:
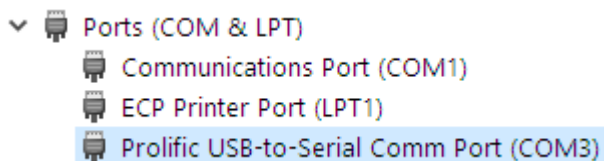


### 2.2.2 Arduino IDE

Download and install Arduino IDE from the official web-site of Arduino.

Get a version 1.6.4 or newer. It is important that the Arduino IDE version that you get has the additional boards manager.

We used Arduino 1.6.6 during the preparation of this document. If you want to have the exact same behavior as us, use Arduino 1.6.6.

# CHAPTER 3: SETUP

Now that we have all the prerequisites ready we need to set everything together.

## 3.1 Hardware setup

The layout hardware connections that we are interested can be found on the next page.

The hardest part of the connection is probably connecting the USB<->serial cable to the UEXT connector of ESP8266-EVB. The important part here is to connect the GND of the cable to the GND pin (#2) of the UEXT connector. Then TXD of the cable to RXD of the UEXT (#4). Finally, RXD of the cable to TXD of the UEXT (#3).
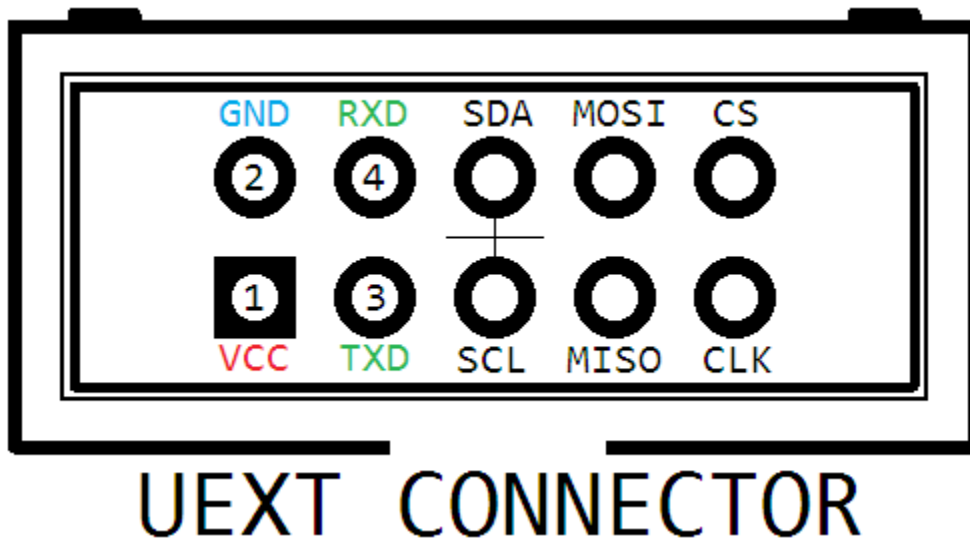


For connecting the USB<->serial cable we are interested only in pins #2, #3 and #4.
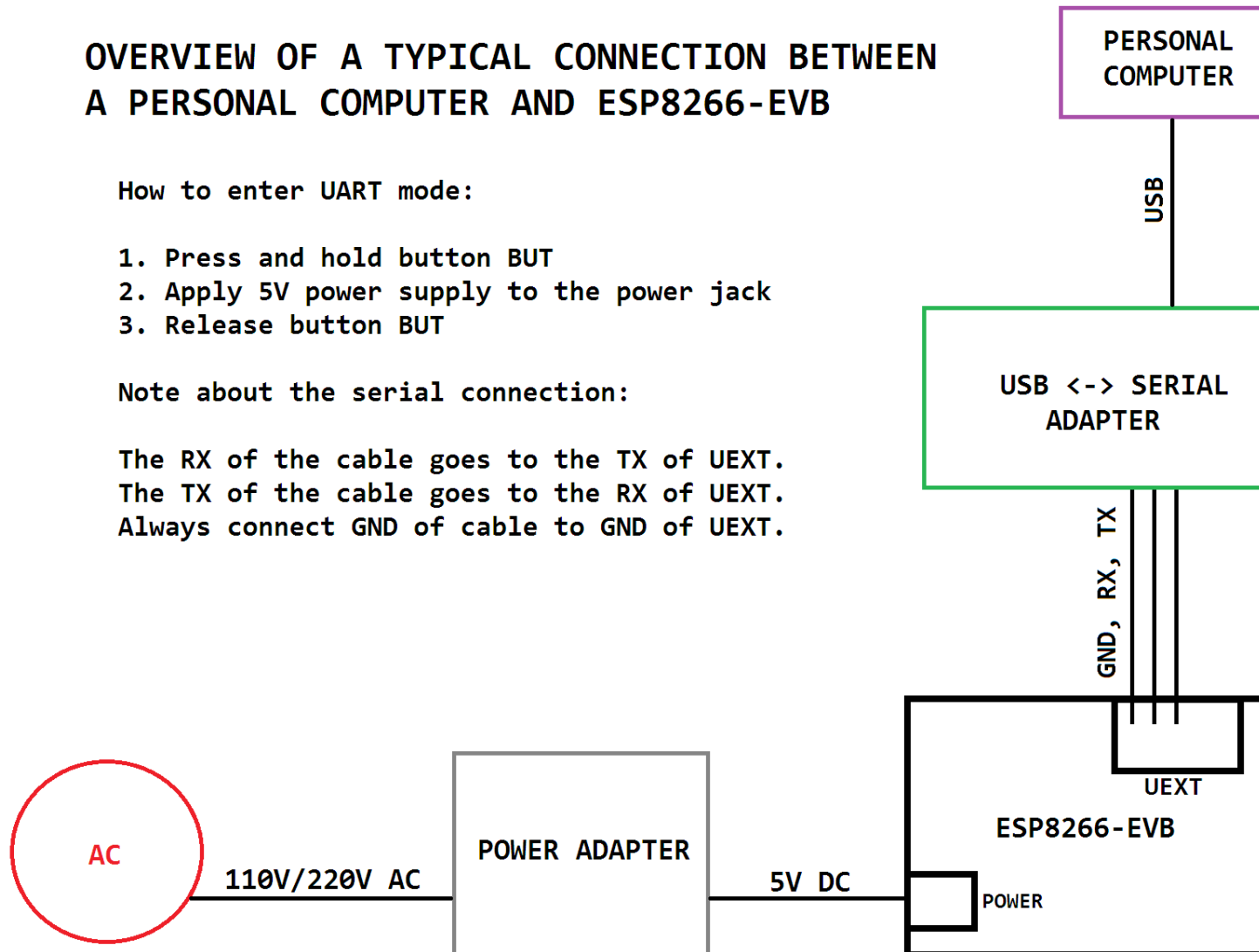
## OVERVIEW OF A TYPICAL CONNECTION BETWEEN A PERSONAL COMPUTER AND ESP8266-EVB

How to enter UART mode:

1. Press and hold button BUT
2. Apply 5V power supply to the power jack
3. Release button BUT

Note about the serial connection:

The RX of the cable goes to the TX of UEXT.
The TX of the cable goes to the RX of UEXT.
Always connect GND of cable to GND of UEXT.

PERSONAL COMPUTER

USB

USB <-> SERIAL ADAPTER

GND, RX, TX

UEXT

ESP8266-EVB

AC

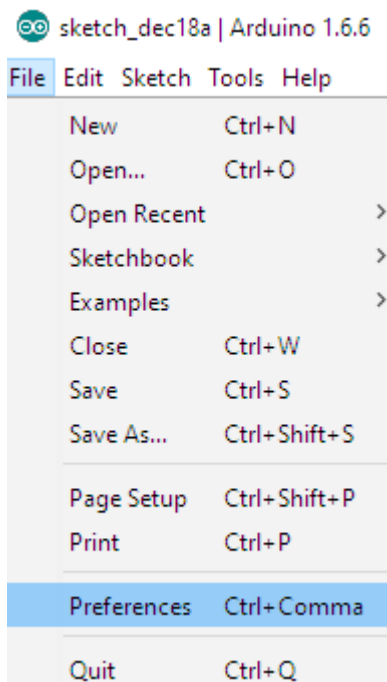110V/220V AC

POWER ADAPTER

5V DC

POWER

## 3.2 Software setup

Before we are able to compile and upload a demo example, we need to prepare Arduino IDE to work with ESP8266 boards.

This is relatively simple. Follow the steps below:

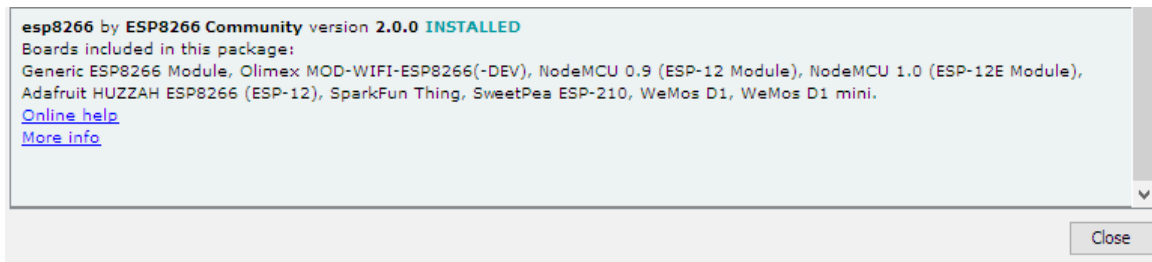- Start Arduino.

- Select File → Preferences as shown below:



- A dialog window would pop-up.

- The only thing we need to edit here is the field called "Additional Boards Manager URLs:". In the blank space we need to provide a link to the .json file which would add ESP8266 support to Arduino. By the time of writing the correct address is:

  http://arduino.esp8266.com/stable/package_esp8266com_index.json

  The field should look similar to this:



- Click "OK" to apply the changes and close the dialog window

- Navigate to Tools → Board → Boards Manager… and install the esp8266 platform. A properly installed library should show similar to this:

esp8266 by ESP8266 Community version 2.0.0 INSTALLED
Boards included in this package:
Generic ESP8266 Module, Olimex MOD-WIFI-ESP8266(-DEV), NodeMCU 0.9 (ESP-12 Module), NodeMCU 1.0 (ESP-12E Module),
Adafruit HUZZAH ESP8266 (ESP-12), SparkFun Thing, SweetPea ESP-210, WeMos D1, WeMos D1 mini.
Online help
More info

Close

- Now at the bottom of the selector Tools → Board you should see different ESP8266 boards available. This is shown below:
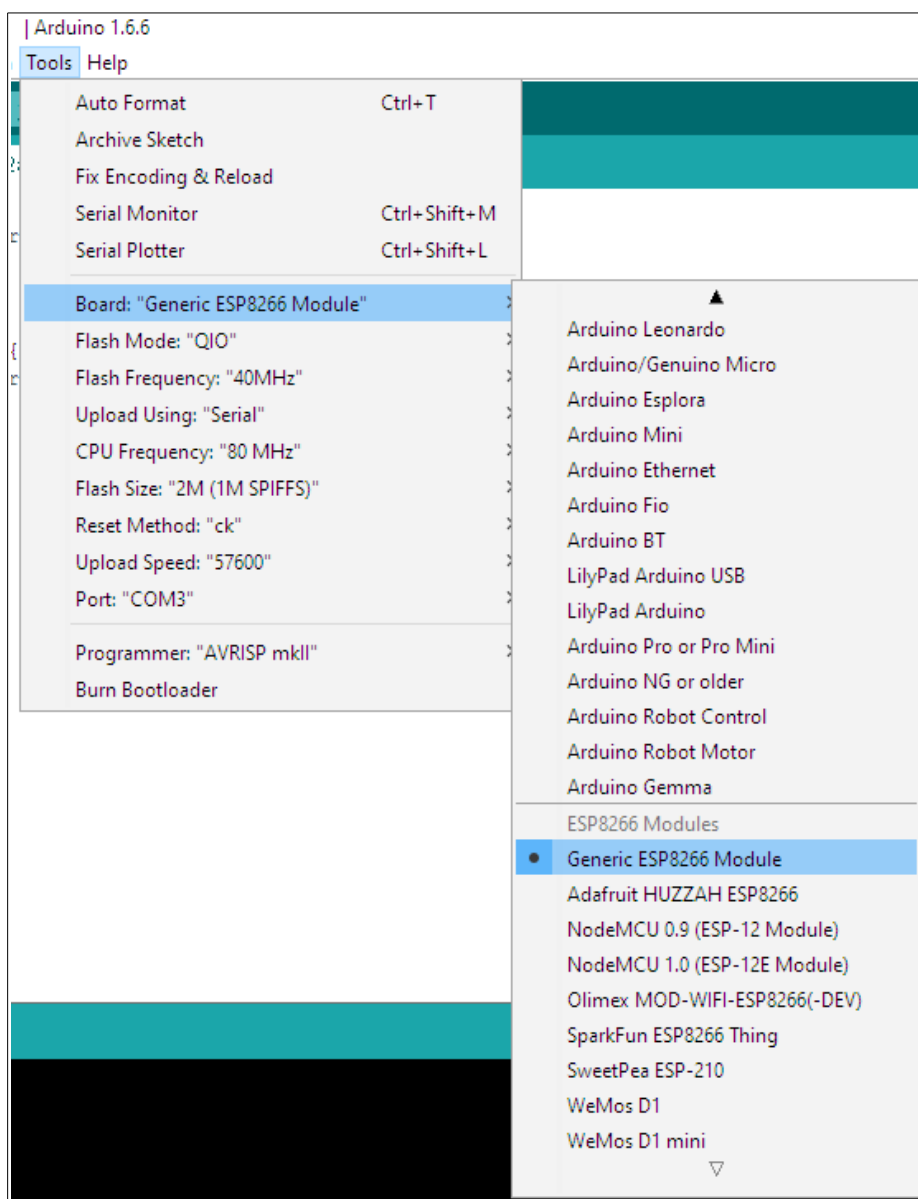
At this point we have a properly configured setup. Now we can proceed to upload a simple blinking LED example (the "Hello, World!" of electronics).

# CHAPTER 4: UPLOADING BLINKING LED

We need a properly configured hardware and software setup as described previously in chapters 2 and 3. Then we proceed as follows:

- Start Arduino IDE.

- Load the blinking LED example. Navigate to File → Examples → 01.Basics → Blink and click on it. This is shown below:
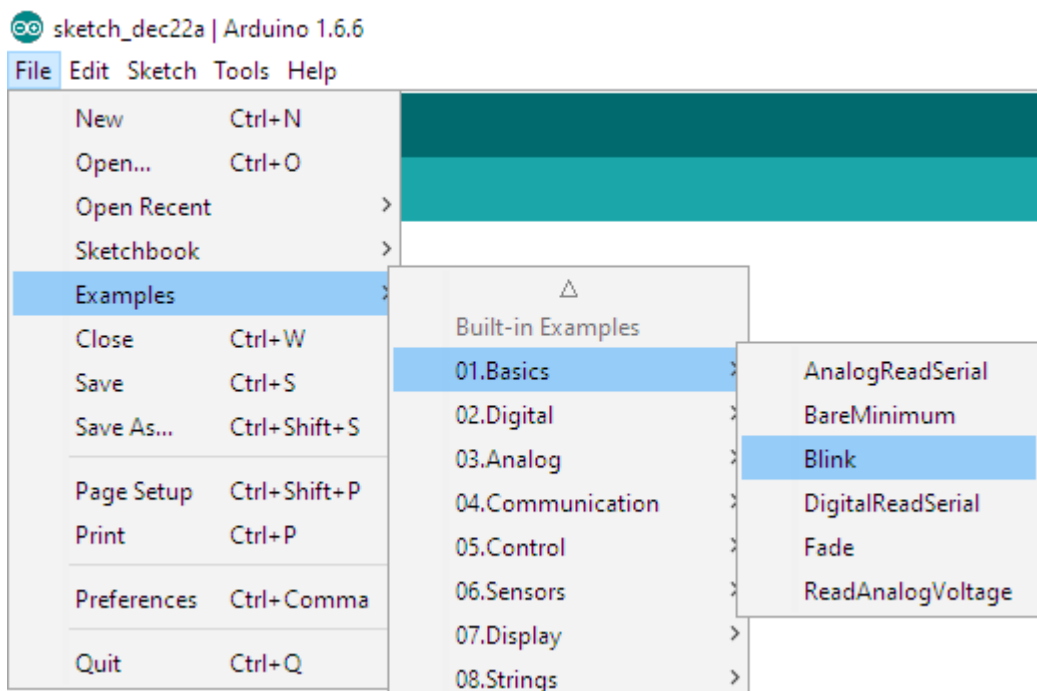


- A new window with code would open. Edit the code in order to reflect the hardware of ESP8266-EVB boards. By default, the example assumes the LED is connected to digital pin #13  (like in most Arduino boards). In our case the relay that has a LED is connected to digital pin #5. This information can be acquired from the schematic of the board. Replace all occurrences of  "13" with "5". The code should look as follows:

```
void setup() {
        // initialize digital pin 5 as an output.
        pinMode(5, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
        digitalWrite(5, HIGH);   // turn the LED on (HIGH is the voltage level)
        delay(1000);            // wait for a second
        digitalWrite(5, LOW);   // turn the LED off by making the voltage LOW
        delay(1000);            // wait for a second
}
```

- Save the edited code.

- Connect the board to the serial cable, and the cable to the USB of the computer (this is detailed in chapter 3 above).

- Identify the serial port number which the cable created in your personal computer.

- In Arduino IDE select "Generic ESP8266 Module" board. This is done from Tools → Board → Generic ESP8266 Module.

- Edit the settings for upload! Navigate to Tools and edit the settings as show below (except for the COM port number which you identified previously; use the COMx number of your setup):
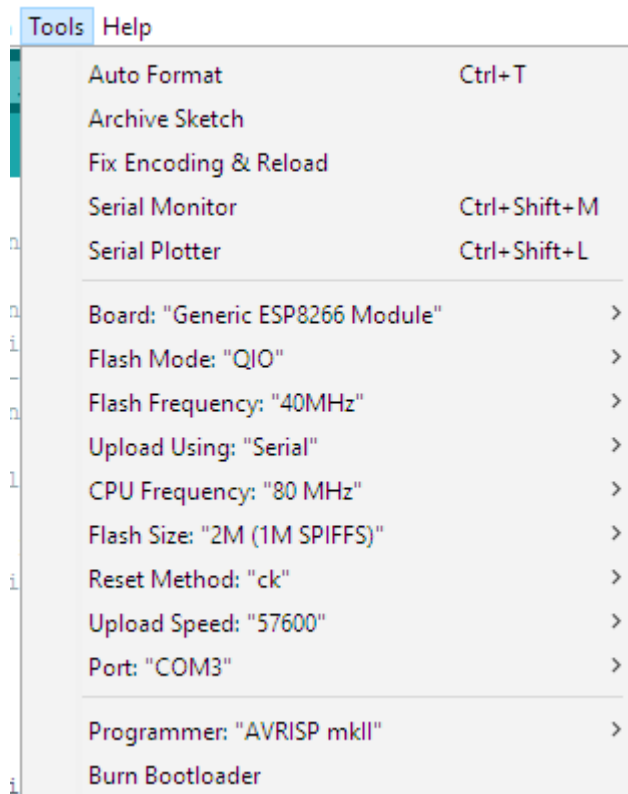


The important settings (specific for Olimex ESP8266-EVB boards) are:

◦ Board: "Generic ESP8266 Module"

◦ Flash Mode: "QIO"

◦ Flash Size: "2M (1M SPIFFS)"

◦ Upload Speed: "57600"

◦ Port: "COMx"

- Enter bootloader mode and power the board. How to do it: press and hold the button of the ESP8266-EVB board; apply 5V of power to the PWR jack; release the button.

  You need to perform the above sequence every time before uploading new code to the board.

- In Arduino IDE compile and upload the example to the board. Simply click button "Upload".

- If everything goes smooth you will also hear that our blinking LED example goes an extra mile and also switches the relay.

# CHAPTER 5: FURTHER READING

In this chapter you would find links to places with useful information.

- The best location for ESP8266-related information is the official Espressif Systems' forum here:

  http://bbs.espressif.com/

- The best place to read more about the ESP8266 support for Arduino IDE is this one:

  https://github.com/esp8266/Arduino

- The custom ESP8266 firmware and examples provided by Olimex can be found here:

  https://github.com/OLIMEX/ESP8266

- The link to the official web-site of Olimex is here:

  https://www.olimex.com

# CHAPTER 6: ORDERING CODES AND DOCUMENT REVISION

## 6.1 Ordering codes

The exact names of some products mentioned in this guide can be found below. The products can be found in our web-shop with these specific product names:

**ESP8266-EVB** – evaluation board for ESP8266 with relay, button, UEXT, all GPIOs available on 0.1" header

**ESP8266-EVB-BAT** – evaluation board for ESP8266 with relay, button, UEXT, Li-Po battery connector with charger and all GPIOs available on boxed connector

**ESP8266-EVB-BAT-BOX** – evaluation board for ESP8266 with relay, button, UEXT, Li-Po battery connector with charger; all GPIOs available on boxed connector and Hammond box

**USB-SERIAL-CABLE-F** – affordable serial to USB cable with female leads

**BB-CH340T** – affordable serial to USB cable (remember to get female-female jumper cables with it)

**SY0605E** – affordable power supply adapter 5V/1A 50Hz/220V(suitable for most European countries)

You can also find a big number of extension boards that can be connected to the UEXT of the board and also some of the components (if damage occurs and you want to replace the part yourself). Make sure to use the search box.

You can purchase directly from our online shop or from any of our distributors. Note that usually it is faster and cheaper to purchase Olimex products from our distributors. List of confirmed Olimex LTD distributors and resellers: https://www.olimex.com/Distributors.

Please visit https://www.olimex.com/ for more info.

## 6.2 Document revision

| Document revision | Changes | Modified page |
|---|---|---|
| A, 22.12.15 | Initial document release | All |
| B, 13.02.17 | Corrected "hardware pin" mentions to "digital pin". | 11 |