



## Restore or Upgrade Firmware on ESP8266 (ESP-01) Module Using Arduino MKR



by ksmith3036

Arduino MKR series, Arduino Zero or compatibles, based on SAMD21 or SAMD51 MCU's with at least 8 Kbytes of SRAM and 16 Kbyte of flash, working at 3.3 volt.

These MCU's have native USB for PC connection and hardware UART for connection to ESP.

The method might also work for other 3.3V Arduino compatibles, but then the sketch must be set to `FlashingMode` instead of `AutoMode`.

(ESP pinout diagram by [blinkmaker](#))

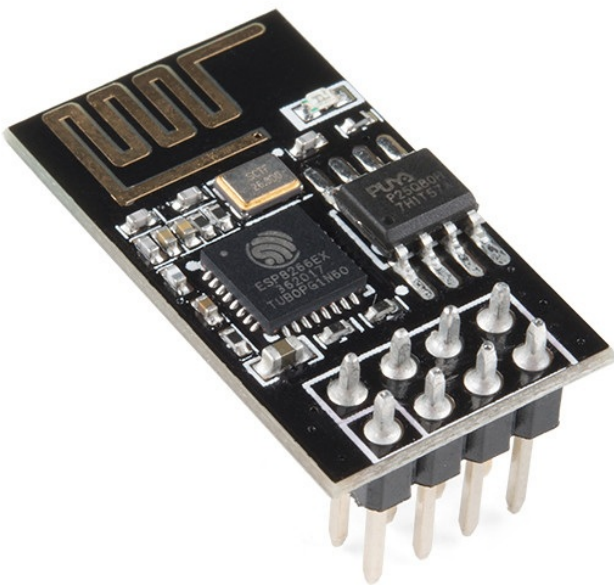
Supplies: Arduino MKR, Arduino Zero or other SAMD21 and SAMD51-based Arduino compatible.

USB cable for Arduino

ESP8266 (ESP-01)

7 breadboard leads

Arduino sketch for [EspSerialPassthrough](#).

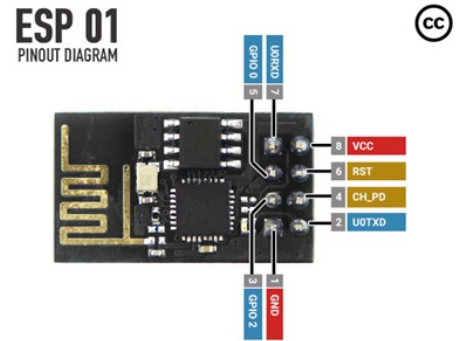


---

### Step 1: Hardware Setup

Connect the ESP8266 (ESP-01) to the Arduino like this:

Arduino pin	ESP8266 (ESP-01) pin	Purpose
8 (D8)	CH_PD / EN (the marking differ)	Chip select / Chip enable signal. Used to enable and reset the ESP.
9 (D9)	GPIO0	Select flashing mode in ESP.
10 (D10)	RESET	ESP reset.
RX (13 on MKR)	TX	Serial
TX (14 on MKR)	RX	Serial
GND	GND	Ground
VCC (3.3V/3V3)	3V3	3.3V power



## Step 2: Firmware From Espressif

Firmware may be found at <https://www.espressif.com/en/support/download/at>.

Content of firmware download from espressif:

Filename	Size
ESP8266_NonOS_AT_Bin_V1.7.4\DS_Store	8196
ESP8266_NonOS_AT_Bin_V1.7.4\ESP8266 NonOS AT Release Note.pdf	117598
ESP8266_NonOS_AT_Bin_V1.7.4\bin\DS_Store	10244
ESP8266_NonOS_AT_Bin_V1.7.4\bin\blank.bin	4096
ESP8266_NonOS_AT_Bin_V1.7.4\bin\boot_v1.7.bin	4080
ESP8266_NonOS_AT_Bin_V1.7.4\bin\esp_init_data_default_v08.bin	128
ESP8266_NonOS_AT_Bin_V1.7.4\bin\at\DS_Store	8196
ESP8266_NonOS_AT_Bin_V1.7.4\bin\at\README.md	2182
ESP8266_NonOS_AT_Bin_V1.7.4\bin\at\1024+1024\user1.2048.new.5.bin	413444
ESP8266_NonOS_AT_Bin_V1.7.4\bin\at\1024+1024\user2.2048.new.5.bin	413444
ESP8266_NonOS_AT_Bin_V1.7.4\bin\at\512+512\user1.1024.new.2.bin	413444
ESP8266_NonOS_AT_Bin_V1.7.4\bin\at\512+512\user2.1024.new.2.bin	413444
ESP8266_NonOS_AT_Bin_V1.7.4\bin\at\sdio\README.md	1764
ESP8266_NonOS_AT_Bin_V1.7.4\bin\at\sdio\1024+1024\user1.2048.new.5.bin	461732
ESP8266_NonOS_AT_Bin_V1.7.4\bin\at\sdio\1024+1024\user2.2048.new.5.bin	461732

First part of ESP8266\_NonOS\_AT\_Bin\_V1.7.4\bin\at\README.md:

# Notice:

AT firmware becomes larger since it supports more functions. So, we provide two firmwares here:

\* 1024+1024: Normal AT, which is compiled from ESP8266\_NONOS\_SDK/examples/at, and uses mbedTLS lib.

\* 512+512: Nano AT, which is compiled from ESP8266\_NONOS\_SDK/examples/at\_nano, and uses SSL lib that supports less cipher suites.

It is suggested to use normal AT(1024+1024), if your flash size is 2MB or larger.

# BOOT MODE

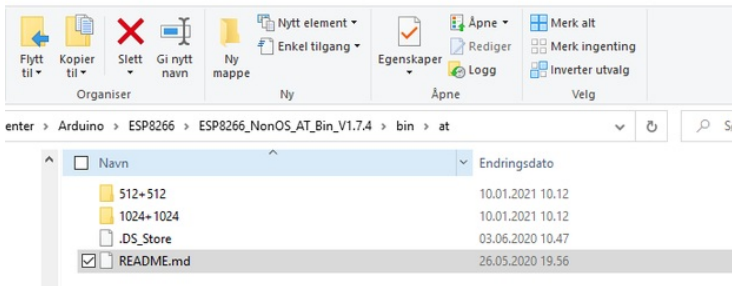
## download

### Flash size 8Mbit: 512KB+512KB

boot\_v1.2+.bin 0x00000  
user1.1024.new.2.bin 0x01000  
esp\_init\_data\_default.bin 0xfc000  
blank.bin 0x7e000 & 0xfe000

### Flash size 16Mbit-C1: 1024KB+1024KB

boot\_v1.2+.bin 0x00000  
user1.2048.new.5.bin 0x01000  
esp\_init\_data\_default.bin 0x1fc000  
blank.bin 0xfe000 & 0x1fe000



### Step 3: 8 Mbit Firmware Information

For an ESP8266 with 8 Mbit / 1 Mbyte flash, configure the ESP8266 Download Tool according to the information from the README.md file like below. Some deduction must be done, to find the correct files.

Text from README.md	Remark	Firmware file	Start address
boot_v1.2+.bin 0x00000	This means to choose a file with bootloader with version 1.2 or newer	bin\boot_v1.7.bin	0x0
user1.1024.new.2.bin 0x01000	Finds this file in the 512+512 folder	bin\at\512+512\user1.1024.new.2.bin	0x1000
esp_init_data_default.bin 0xfc000	Matches with esp_init_data_default_v08.bin	bin\esp_init_data_default_v08.bin	0xfc000
blank.bin 0x7e000 & 0xfe000	With two start addresses, this file must be flashed into two addresses	bin\blank.bin	0x7e000 0xfe000

### Step 4: 16 Mbit Firmware Information

For an ESP8266 with 16 Mbit / 2 Mbyte flash, configure the ESP8266 Download Tool according to the information from the README.md file like below. Some deduction must be done, to find the correct files.

If having a flash of 16 Mbit or more or wanting to use the AT-SDIO version for 16 Mbit flash, refer to the ESP8266\_NonOS\_AT\_Bin\_V1.7.4\bin\at\_sdio\README.md file.

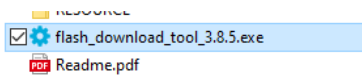
Text from README.md	Remark	Firmware file	Start address
boot_v1.2+.bin      0x00000	This means to choose a file with bootloader with version 1.2 or newer	bin\boot_v1.7.bin	0x0
user1.2048.new.5.bin    0x01000	Finds this file in the 512+512 folder	bin\at\1024+1024\user1.2048.new.5.bin	0x1000
esp_init_data_default.bin 0x1fc000	Matches with esp_init_data_default_v08.bin	bin\esp_init_data_default_v08.bin	0x1fc000
blank.bin      0xfe000 & 0x1fe000	With two start addresses, this file must be flashed into two addresses	bin\blank.bin	0xfe000
		bin\blank.bin	0x1fe000

## Step 5: ESP Firmware Download Tool

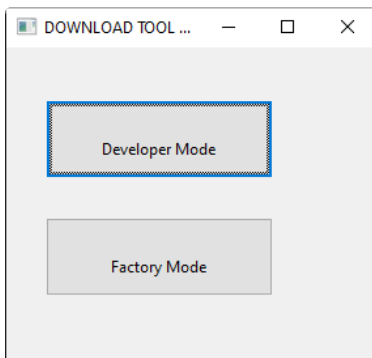
The ESP firmware flashing tool may be downloaded from <https://www.espressif.com/en/support/download/other-tools>.

Unpack and run the flash download tool.

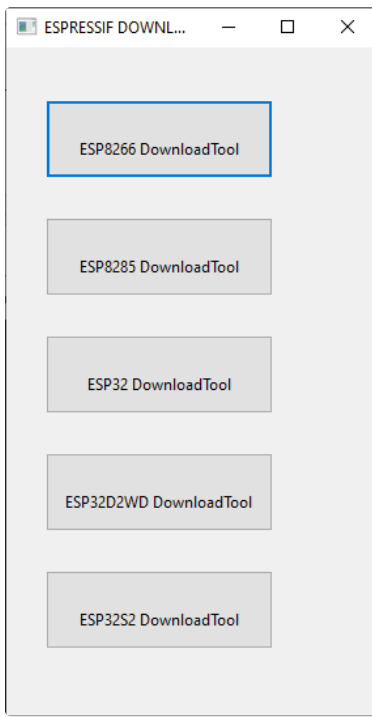
First a black window appears, then after up to 20-30 seconds, a selection box appears.



## Step 6: Select Developer Mode



## Step 7: Select ESP8266 DownloadTool

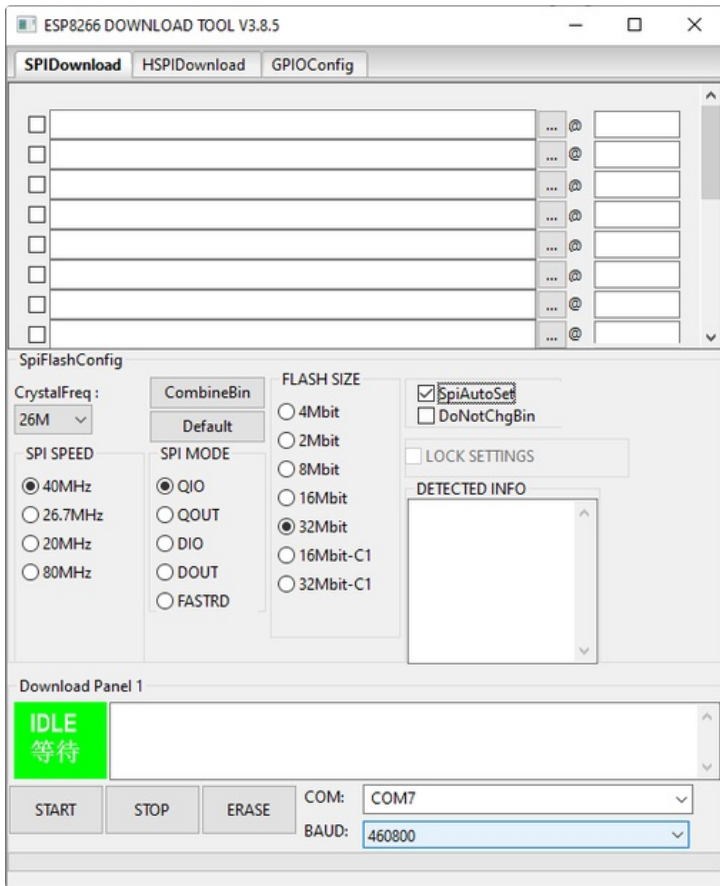


---

## Step 8: Read Information From ESP

Use SpiAutoSet to get parameters from ESP, by checking SpiAutoSet and pressing START with no files selected for flashing. *File names may be listed, but they must be unchecked.*

When using an Arduino MKR or Zero, baud rate could be set at up to 1500000 baud. If using an Arduino without native USB, 115200 should be used, and PASSTHROUGH\_MODE should be set to FlashingMode in the sketch.



## Step 9: Statuswindow

The command window shows the working. Initially the flashing program connect using 115200 baud, sending a stub loader to the ESP. If another baud rate than 115200 is selected in the user interface, the baud rate is switched. Switching baud rates will only work for Arduinos with native USB connection to the PC, since they in fact ignore the baud rate between the PC and

Arduino, always running at full USB speed, which for an SAMD21 is 12 Mbit/sec. The EspSerialPassthrough sketch detects the baud rate change command, and changes the baud rate between the Arduino and the ESP accordingly.

```

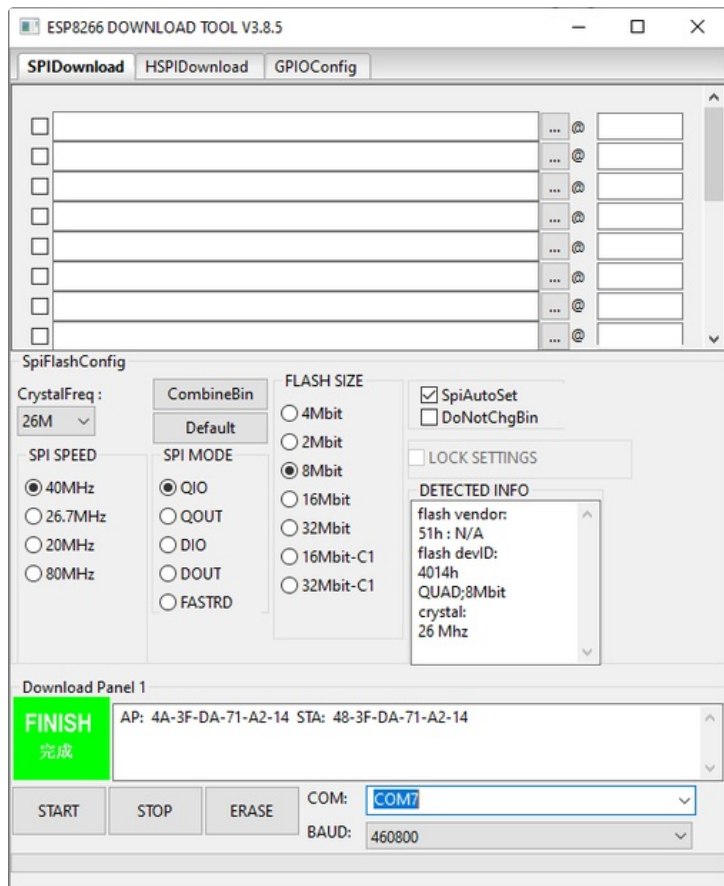
C:\Users\kaare\Documents\Arduino\ESP8266\flash_download_tool.
=====
CONNECT BAUD: 115200
=====
...._Uploading stub...
Running stub...
Stub running...
Changing baud rate to 460800
Changed.
-----
crc_efuse_4bit: 0
crc_calc_4bit: 4
48bit mac

is stub and send flash finish

```

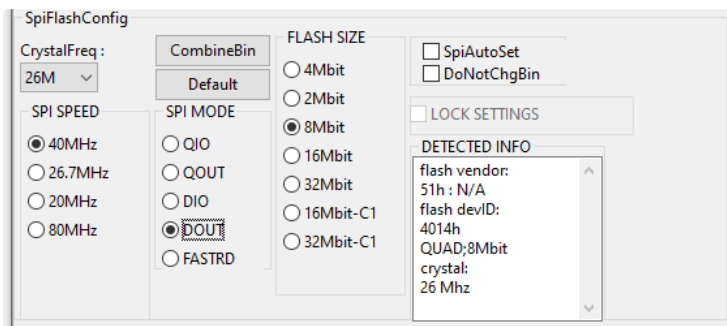
## Step 10: The Result for an ESP-01 / ESP8266 With 8 Mbit Flash

The program finds the crystal frequency (CrystFreq) and flash size. In my case it keeps the SPI MODE set to QIO. Several tries flashing with SPI mode set to QIO failed, but an Internet search found that DOUT might be necessary to select manually.



## Step 11: Select SPI MODE

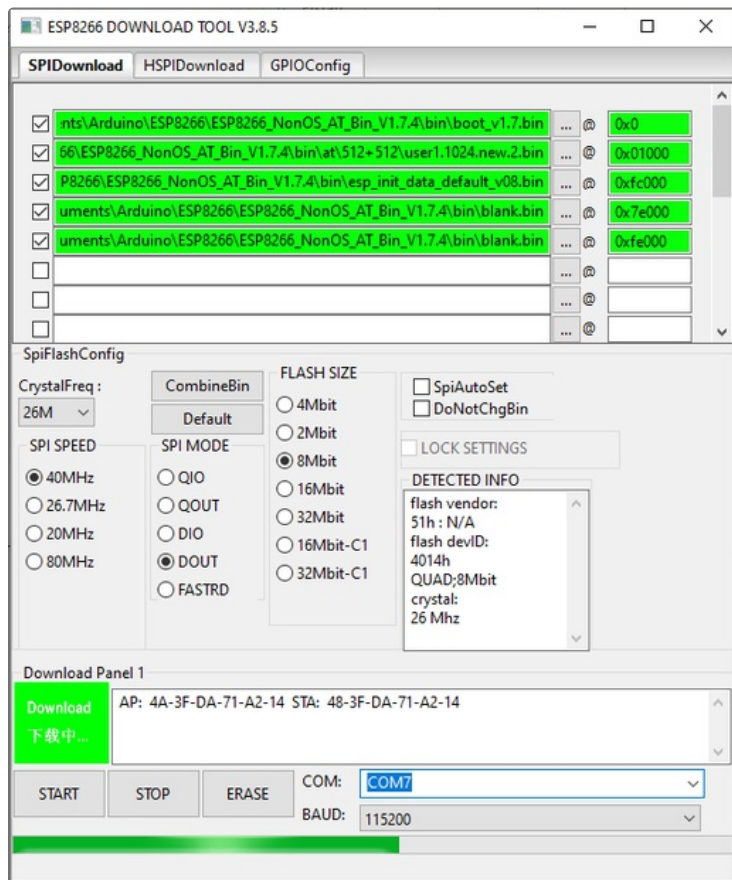
**Uncheck SpiAutoSet** and select the appropriate SPI MODE, which for my ESP8266 was **DOUT**. SPI Mode seems to control the way the ESP flashes itself.



## Step 12: Flash Firmware

Select the correct files to flash, according to the README.md file of the ESP firmware (information extracted into tables above), and click START.

For baud rates, using an Arduino MKR with SAMD21 microcontroller, connected to the PC using native USB, I successfully have flashed with baud rates up to **150000**.



## Step 13: Verify Successful Flashing

Reset the Arduino, which makes the Arduino program reset the ESP8266.

Open a serial monitor or terminal emulator like Putty. Make sure to send both NewLine and Carriage Return as line feeds. May manually be entered as Ctrl-M + Ctrl-J instead of using the ENTER key.



```
Serial COM7 - Arduino MKR NB 1500
Port open
ets Jan 8 2013, rst cause:1, boot mode:(3,4)
load 0x40100000, len 2592, room 16
tail 0
chksum 0xf3
load 0x3ffe0000, len 764, room 8
tail 4
chksum 0x92
load 0x3ffe21fc, len 676, room 4
tail 0
chksum 0x22
csum 0x22

2nd boot version : 1.
ets Jan 8 2013, rst cause:1, boot mode:(3,7)
load 0x40100000, len 2592, room 16
tail 0
chksum 0xf3
load 0x3ffe0000, len 764, room 8
tail 4
chksum 0x92
load 0x3ffe21fc, len 676, room 4
tail 0
chksum 0x22
csum 0x22

2nd boot version : 1.7(5d5f877)
SPI Speed : 40MHz
SPI Mode : DOUT
SPI Flash Size & Map: @0b1(512K@+512K@)
jump to run user1 @ 1000

correct flash map
V2
No
,**rf cal sector: 251
freq trace enable 0
rf[112] : 00
rf[113] : 00
rf[114] : 02

SDK ver: 3.0.4(9532ceb) compiled @ May 22 2020 16:26:04
phy ver: 1156_0, pp ver: 10.2

***HW2FI CONNECTED
WIFI GOT IP
```

---

## Step 14: Run the AT+GMR Command

The output should match the newly flashed firmware

```
AT+GMR
AT version:1.7.4.0(May 11 2020 19:13:04)
SDK version:3.0.4(9532ceb)
compile time:May 27 2020 10:12:17
Bin version(Wroom 02):1.7.4
OK
```

---

## Step 15: Arduino Sketch

The Arduino sketch may be found at Github: <https://github.com/ksmith3036/EspSerialPassthrough>

