



FLASHING THE ONEBUS

Abstract

This guide describes the steps to be followed to flash the OneBus to the latest firmware

Stéphane Ghenne
Revision 0 | 2021-07-18

How to flash the OneBus

1. Install STM32CubeProgrammer

The program can be downloaded [here](#)

2. Connect the Nucleo F767Zi board to your PC

This can be done using a USB cable on the CN1 micro USB connector of the board.

3. In STM32CubeProgrammer click on the green 'connect' button

This button can be found in the top right of the program window. If the connection is successful, the status dot above the button will turn to green and say 'Connected', as can be seen in the screenshots below. The target information on the bottom right of the window should list the characteristics of the F767Zi board.

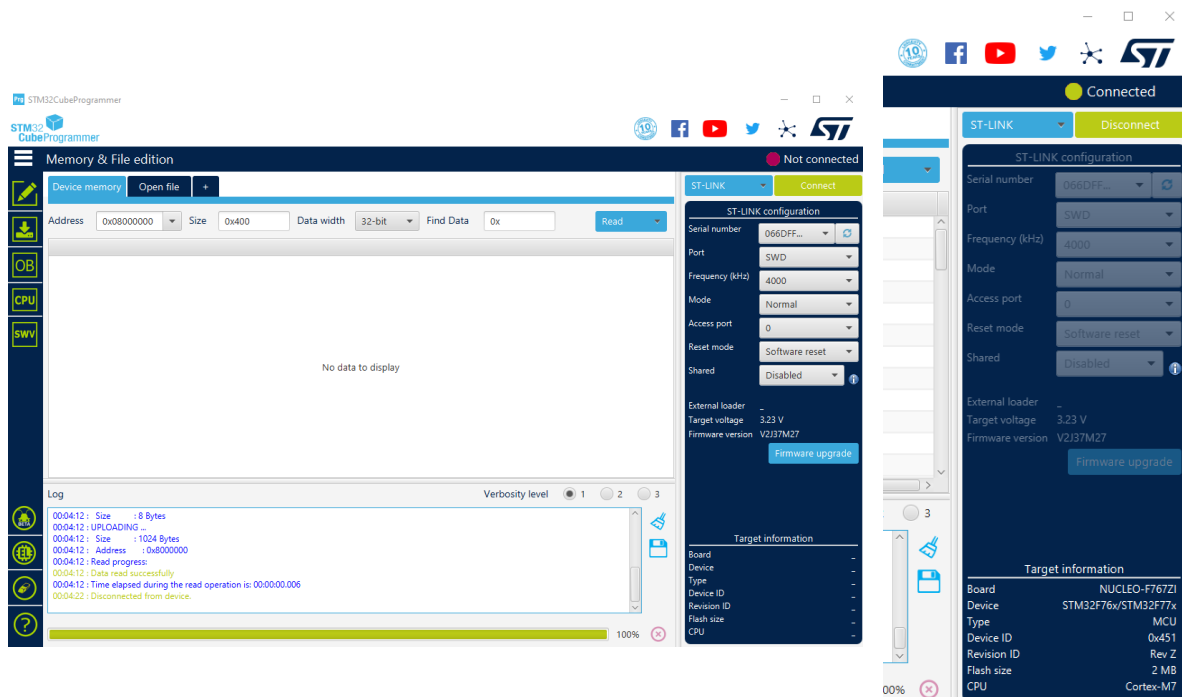


Figure 1: STM32CubeProgrammer window. On the top right you can find the 'connect' button in green. [left]. The status dot turns green for a successful connection [right].

4. Click on the 'Erasing and programming' button

This button can be found on the left-hand side menu.

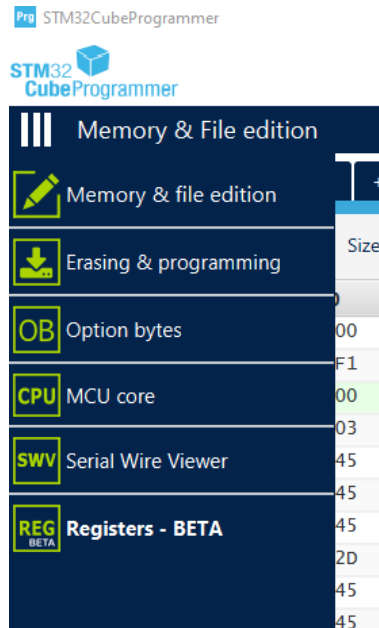


Figure 2: STM32CubeProgrammer left hand side menu. Click on 'Erasing & programming' button.

5. Select the firmware to flash with the 'Browse' button

The latest OneBus firmware can be downloaded [here](#). Make sure to check the 'Verify programming' and 'Run after programming' boxes. The other options can be left by default, as can be seen on the screenshot below.

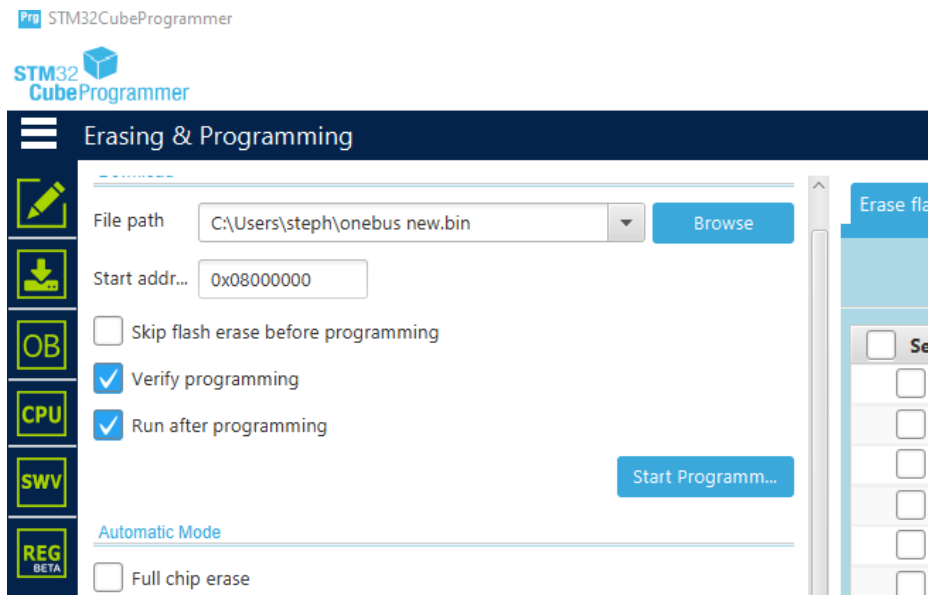


Figure 3: The 'Erasing and programming' window. Browse to the firmware file and check 'Verify programming' and 'Run after programming'.

6. Click on 'Start Programming'

If all went well, you will see 'Start operation achieved successfully' and 'Download verified successfully'

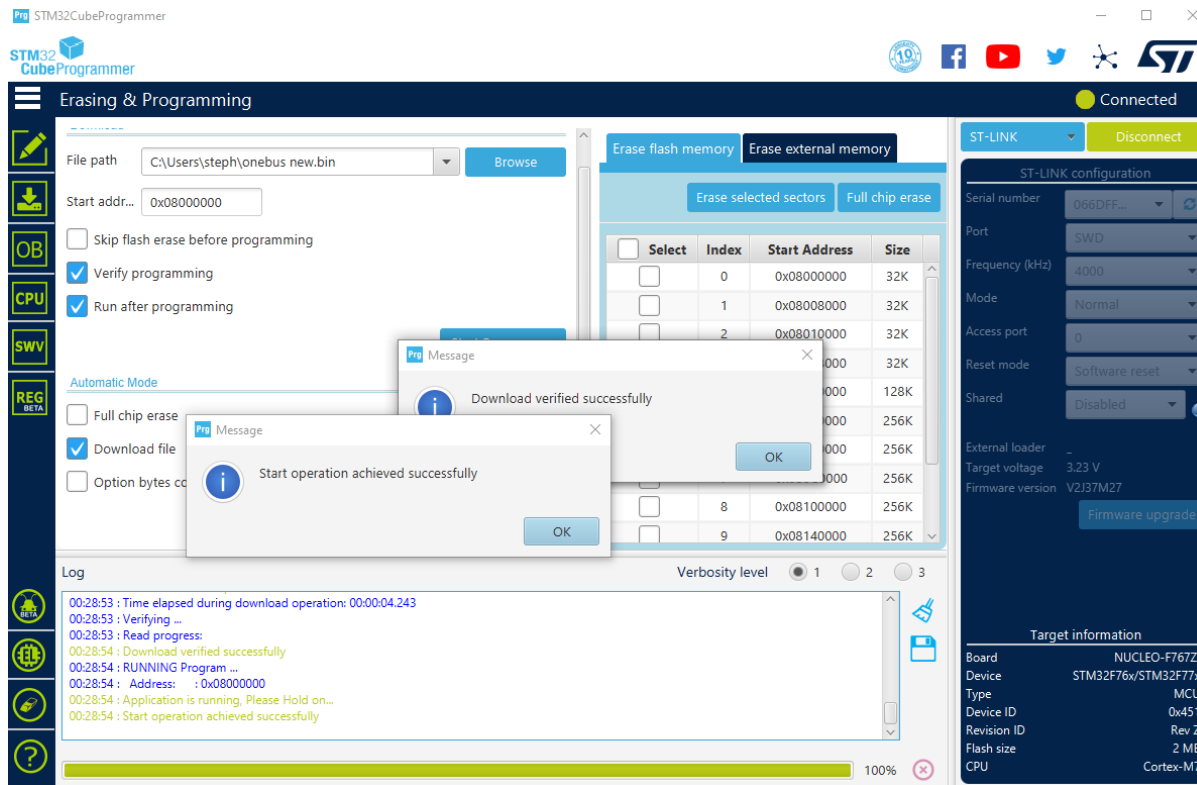


Figure 4: Pop-up windows will confirm the successful programming.

7. Click on the green 'Disconnect' button

The OneBus has been successfully flashed. After clicking the 'Disconnect' button, you can safely quit STM32CubeProgrammer.

If the Nucleo or ESP8266 card you are using have been flashed before with the OneBus software, no additional steps are required. For a new Nucleo card, you will need to follow the steps described in §8. For a new ESP8266, follow §9 as well.

8. Initialize the OneBus

To enable a user to keep his settings when flashing his device, some configuration parameters are kept outside of the firmware memory.

These steps are only required for a Nucleo card never flashed with the OneBus firmware before.

For a new installation, these settings need to be initialized by default for the OneBus to work properly.

8.1 Initialize the OneBus with default settings

To initialize the OneBus with default settings, the following manipulations of the two push buttons ('B1 User' and 'B2 Reset') on the Nucleo board are required:

- 1 Press the 'B1 User' button and while keeping it pressed, press the 'B2 Reset' button.

- 2 Release the 'B2 Reset' button, but keep the 'B1 User' button pressed. After 3 seconds, the red LED lights up, and the screen displays 'OneBus default config'.
- 3 You can now release the 'B1 User' button. As soon as the red LED turns off, the configuration of the OneBus is complete.

8.2 Check the COM Port of the Nucleo board in the Windows Device manager

Write down the COM Port number of the Nucleo board. In this example the COM port is 8.

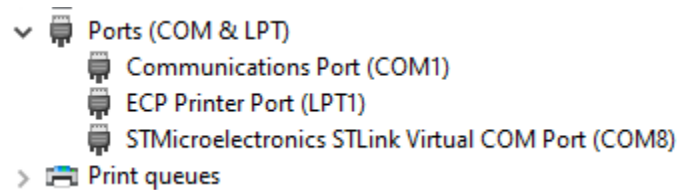


Figure 5: The COM port of the Nucleo device in the Windows Device Manager.

8.3 Install and start PuTTY

PuTTY can be downloaded [here](#). Once downloaded, start the program. The settings need to be as follows:

- Serial line: COM8 (the COM port the Nucleo card is on)
- Speed: 460800
- Connection type: Serial

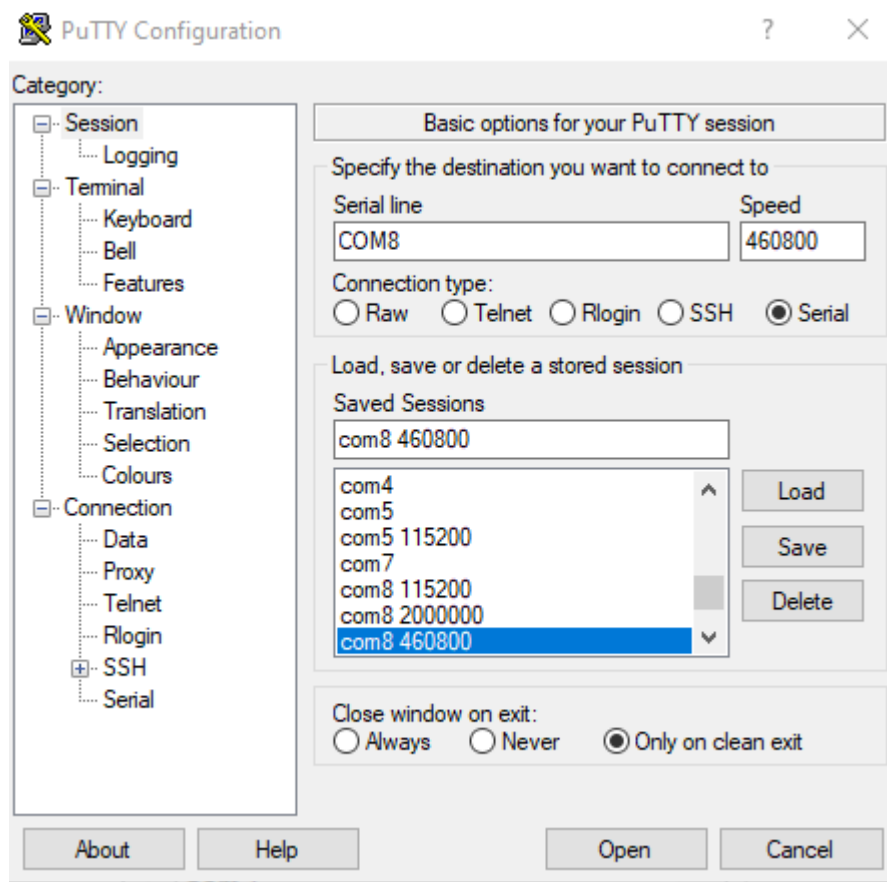
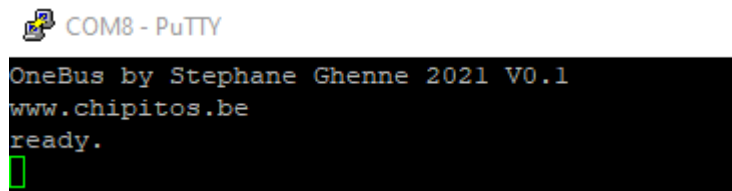


Figure 6: Starting and configuring PuTTY.

Click on 'Open'. After resetting the OneBus, if all went well, you should see the following message in PuTTY:



```
COM8 - PuTTY
OneBus by Stephane Ghenne 2021 V0.1
www.chipitos.be
ready.
█
```

Figure 7: PuTTY message for successful initialization of the OneBus.

Your OneBus is now initialized.

If the ESP8266 WiFi card is new, it must also be initialized for use with the OneBus. This can be achieved by following the steps below (§9).

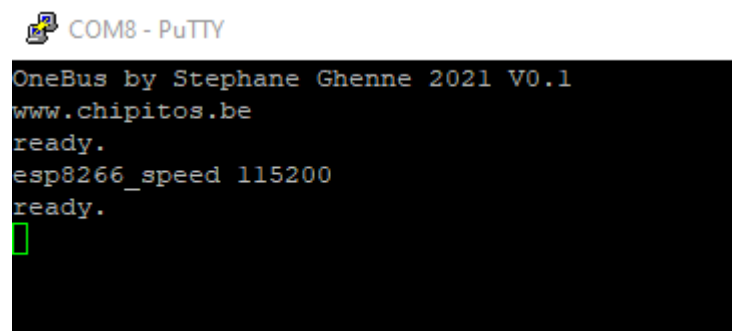
9. Initialize the ESP8266

By default, in a new ESP8266, the baud rate is set to 115200. This speed is too slow to show the data transfer in real time as is done in the OneBus. For a proper operation, the speed needs to be set to 460800 bauds.

This section assumes a working PuTTY connection with the OneBus. To do this, we refer to §8.3.

9.1 Configure the OneBus for communication with the ESP8266 at 115200 bauds

This can be done by using the command 'esp8266_speed 115200' through PuTTY.

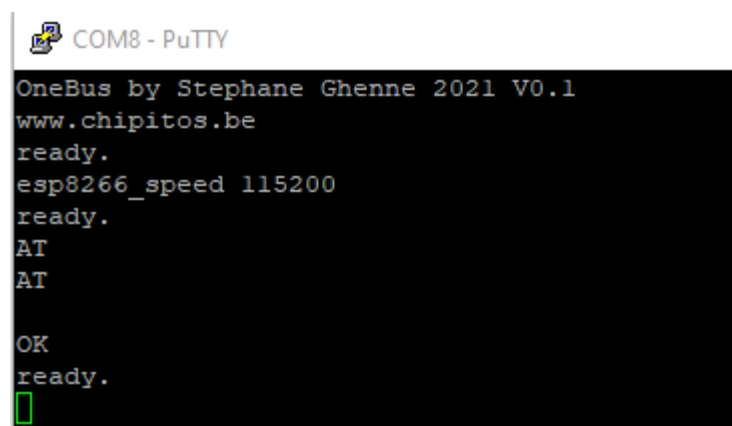


```
COM8 - PuTTY
OneBus by Stephane Ghenne 2021 V0.1
www.chipitos.be
ready.
esp8266_speed 115200
ready.
█
```

Figure 8: Using the 'esp8266_speed 115200' command in PuTTY.

9.2 Use the 'AT' command

If working correctly, the ESP8266 should respond 'OK'.

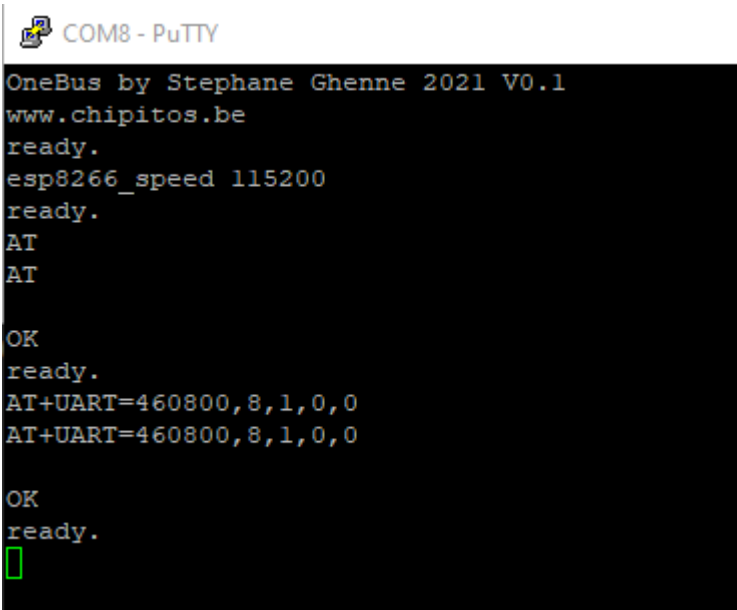


```
COM8 - PuTTY
OneBus by Stephane Ghenne 2021 V0.1
www.chipitos.be
ready.
esp8266_speed 115200
ready.
AT
AT
OK
ready.
█
```

Figure 9: Using the AT command, and receiving 'OK' for 115200 bauds.

9.3 Configure the ESP8266 for 460800 bauds

Use the command “AT+UART=460800,8,1,0,0”. If all goes well, you should receive an ‘OK’ reply.

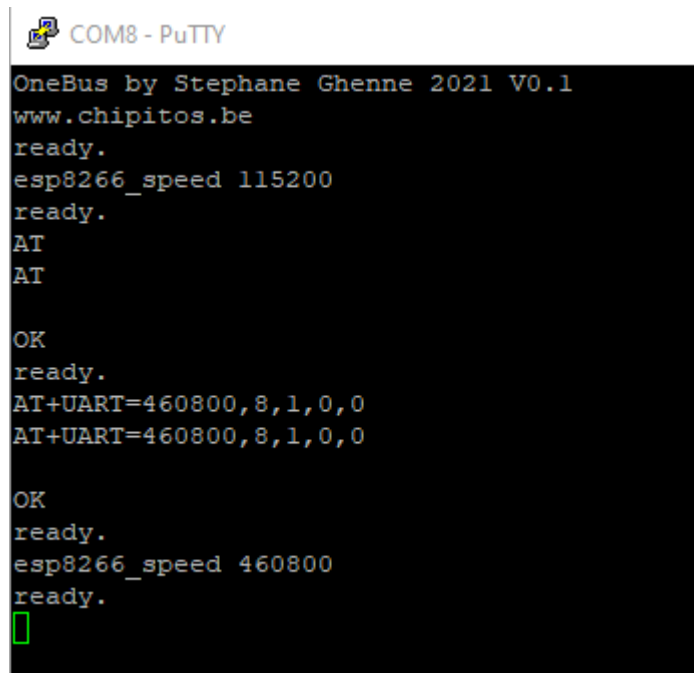


```
COM8 - PuTTY
OneBus by Stephane Ghenne 2021 V0.1
www.chipitos.be
ready.
esp8266_speed 115200
ready.
AT
AT
OK
ready.
AT+UART=460800,8,1,0,0
AT+UART=460800,8,1,0,0
OK
ready.
█
```

Figure 10: Configuring the ESP8266 to 460800 bauds.

9.4 Configure the OneBus to be able to communicate with the ESP8266 at 460800 bauds

This can be set with the command ‘esp8266_speed 460800’.

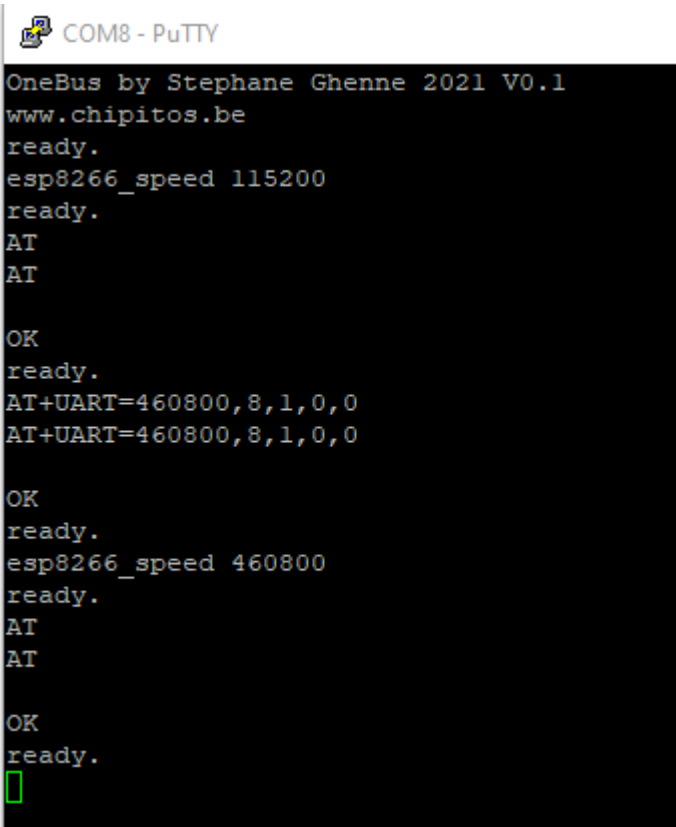


```
COM8 - PuTTY
OneBus by Stephane Ghenne 2021 V0.1
www.chipitos.be
ready.
esp8266_speed 115200
ready.
AT
AT
OK
ready.
AT+UART=460800,8,1,0,0
AT+UART=460800,8,1,0,0
OK
ready.
esp8266_speed 460800
ready.
█
```

Figure 11: Configuring the OneBus for 460800 bauds.

9.5 Use the 'AT' command

If working correctly, the ESP8266 should respond 'OK'.



```
COM8 - PuTTY
OneBus by Stephane Ghenne 2021 V0.1
www.chipitos.be
ready.
esp8266_speed 115200
ready.
AT
AT

OK
ready.
AT+UART=460800,8,1,0,0
AT+UART=460800,8,1,0,0

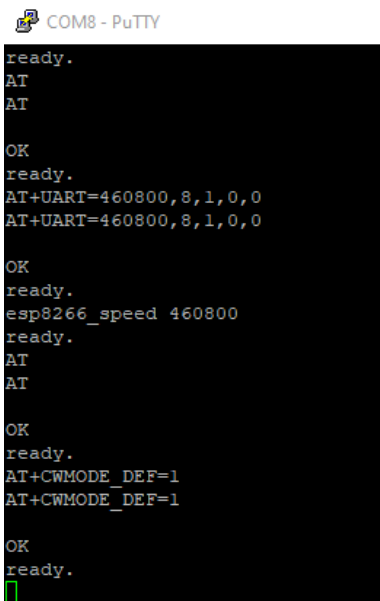
OK
ready.
esp8266_speed 460800
ready.
AT
AT

OK
ready.
█
```

Figure 12: Using the AT command, and receiving 'OK' for 460800 bauds.

9.6 Set the ESP8266 in station mode

This can be done with the command "AT+CWMODE_DEF=1" and check if you receive an 'OK' reply.



```
COM8 - PuTTY
ready.
AT
AT

OK
ready.
AT+UART=460800,8,1,0,0
AT+UART=460800,8,1,0,0

OK
ready.
esp8266_speed 460800
ready.
AT
AT

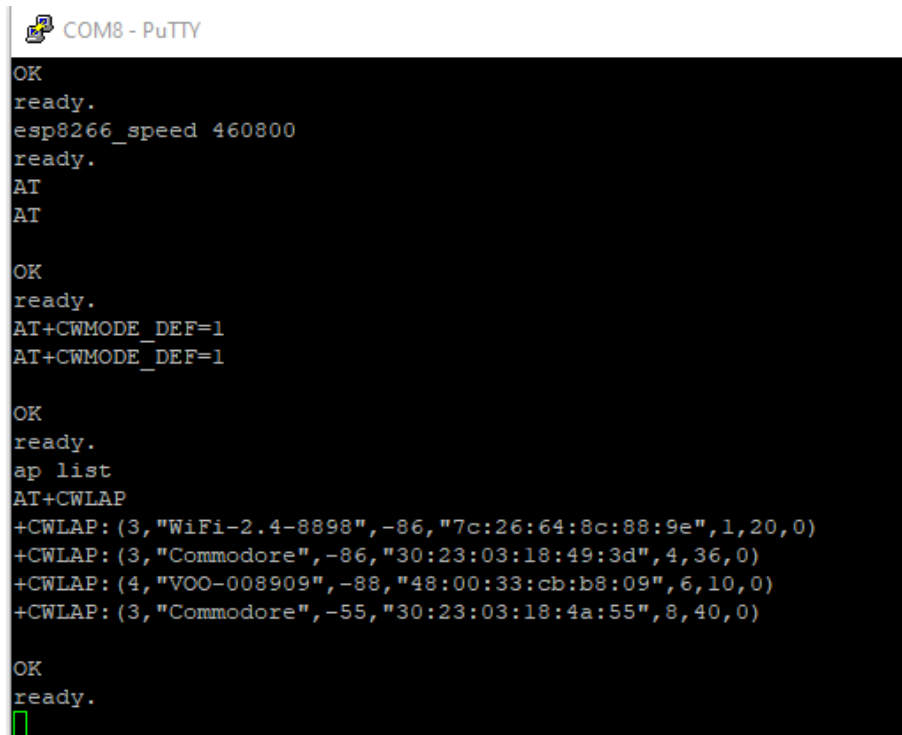
OK
ready.
AT+CWMODE_DEF=1
AT+CWMODE_DEF=1

OK
ready.
█
```

Figure 13: Setting the ESP8266 in station mode.

9.7 Check if you are picking up access points

This is done with the command 'ap list'.



```
COM8 - PuTTY
OK
ready.
esp8266_speed 460800
ready.
AT
AT

OK
ready.
AT+CWMODE_DEF=1
AT+CWMODE_DEF=1

OK
ready.
ap list
AT+CWLAP
+CWLAP: (3, "WiFi-2.4-8898", -86, "7c:26:64:8c:88:9e", 1, 20, 0)
+CWLAP: (3, "Commodore", -86, "30:23:03:18:49:3d", 4, 36, 0)
+CWLAP: (4, "VOO-008909", -88, "48:00:33:cb:b8:09", 6, 10, 0)
+CWLAP: (3, "Commodore", -55, "30:23:03:18:4a:55", 8, 40, 0)

OK
ready.
█
```

Figure 14: Picking up all the available access points.

The ESP8266 is now correctly initialized.