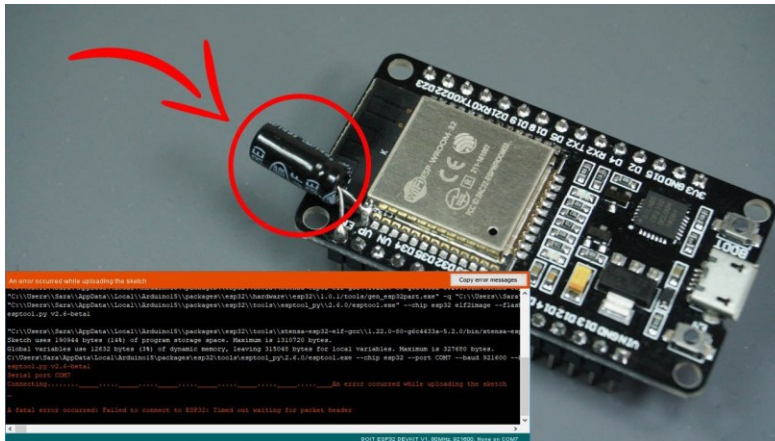


[SOLVED] Failed to connect to ESP32: Timed out waiting for packet header

Learn how to fix the Fatal Error Occurred: **“Failed to connect to ESP32: Timed out waiting for packet header”** error when trying to upload new code to your ESP32 board once for all.



Why are you getting this error?

Some [ESP32 development boards](#) (read [Best ESP32 boards](#)) don't go into flashing/uploading mode automatically when uploading a new code.

This means that when you try to upload a new sketch to your ESP32, the Arduino IDE fails to connect to your board, and you get the following error message:

```
An error occurred while uploading the sketch
Copy error messages

"C:\Users\Sara\AppData\Local\Arduino15\packages\esp32\hardware\esp32\1.0.1/tools/gen_esp32part.exe" -q "C:\Users\Sara\AppData\Local\Arduino15\packages\esp32\tools\esptool_py\2.6.0/esptool.exe" --chip esp32 elf2image --flash_esp32.py v2.6-beta1

"C:\Users\Sara\AppData\Local\Arduino15\packages\esp32\tools\xtensa-esp32-elf-gcc\1.22.0-80-g64433a-5.2.0/bin/xtensa-esp32-elf-gcc.exe" -x -c "C:\Users\Sara\AppData\Local\Arduino15\packages\esp32\hardware\esp32\1.0.1/tools/gen_esp32part.exe" -q "C:\Users\Sara\AppData\Local\Arduino15\packages\esp32\tools\esptool_py\2.6.0/esptool.exe" --chip esp32 --port COM7 --baud 921600 --esptool.py v2.6-beta1
Sketch uses 190944 bytes (14%) of program storage space. Maximum is 1310720 bytes.
Global variables use 12632 bytes (3%) of dynamic memory, leaving 315048 bytes for local variables. Maximum is 327680 bytes.
C:\Users\Sara\AppData\Local\Arduino15\packages\esp32\tools\esptool_py\2.6.0/esptool.exe --chip esp32 --port COM7 --baud 921600 --esptool.py v2.6-beta1
Serial port COM7
Connecting.....An error occurred while uploading the sketch
-----
A fatal error occurred: Failed to connect to ESP32: Timed out waiting for packet header

DOIT ESP32 DEVKIT V1, 80MHz, 521600, None on COM7
```

Holding the BOOT/FLASH button

One of the ways to solve this is holding-down the **“BOOT/FLASH”** button in your ESP32 board while uploading a new sketch at the same time. But having to worry about this every time you want to upload new code can be tedious, specially when you're testing and debugging your code. There is a way to fix this once for all – no need to hold down the **“BOOT/FLASH”** button anymore.

How to fix the Error?

To make your ESP32 board go into flashing/uploading mode automatically, you can **connect a 10 uF electrolytic capacitor between the EN pin and GND**.

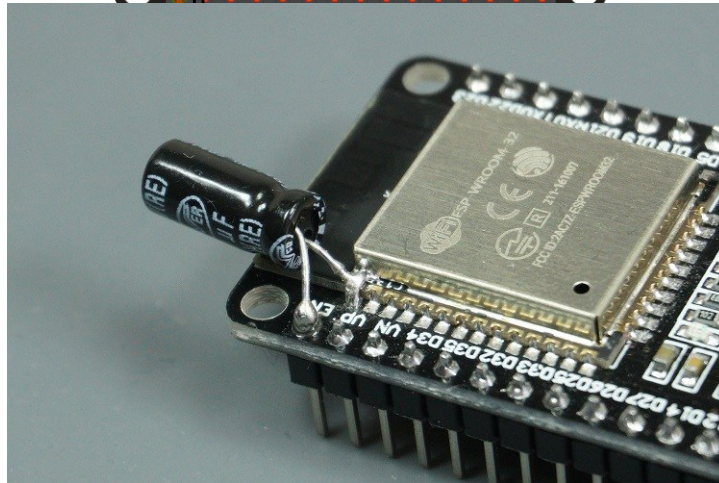
You may want to test this setup first on a breadboard to make sure it works for your ESP32 development board.

Note: electrolytic capacitors have polarity. The white/grey stripe indicates the negative lead.

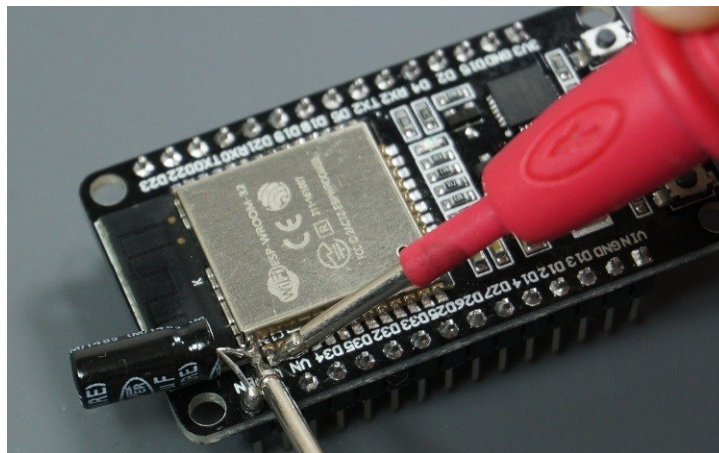
If it works, then you can solder the **10 uF electrolytic capacitor** to the board. Since the **EN** and **GND** pins are far apart from each other, you can simply connect the capacitor between the **EN** and the **GND** of the ESP32 chip as shown in the schematic diagram below:

Recommended: [ESP32 Pinout Reference](#), Which GPIO pins should you use?

The following figure shows how many ESP32 pins are soldering a capacitor. It doesn't occupy much space, and fortunately you won't get more trouble.



Before trying to upload a new code, you should check the connections with a multimeter in continuity mode – check that you haven't inadvertently soldered anything to the next pin.



If everything is soldered properly, you won't need to press the BOOT button when uploading new code. You also won't get the Fatal Error Occurred: "Failed to connect to ESP32: Timed out waiting for packet header".

```
Done uploading.
Writing at 0x00010000... (14 %)
Writing at 0x00014000... (28 %)
Writing at 0x00018000... (42 %)
Writing at 0x0001c000... (57 %)
Writing at 0x00020000... (71 %)
Writing at 0x00024000... (85 %)
Writing at 0x00028000... (100 %)
Wrote 191104 bytes (99404 compressed) at 0x00010000 in 1.5 seconds (effective 1013.8 kbit/s)...
Hash of data verified.
Compressed 3072 bytes to 144...

Writing at 0x00008000... (100 %)
Wrote 3072 bytes (144 compressed) at 0x00008000 in 0.0 seconds (effective 1755.4 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...

9 DOIT ESP32 DEVKIT V1, 80MHz, 921800, None on COM7
```

Wrapping Up

We hope you've found this trick useful and it solved your problem. Thanks to Ben Hall for the suggestion.