Ultralight Computer Guide

How to make an Ultralight Computer

In this guide you will be able to find the steps to make your own ultralight computer. The steps involved will mean exploring areas of your computer that you might not have been to before and it will mean using the terminal. If you are not comfortable doing either of these then find someone who has some experience with this and do it with them.

Whilst this guide should work for you, there are many points along the way where things might break. In this case it's useful to know how to troubleshoot. The most basic form of troubleshooting takes the form of copying error codes that you see in your terminal and pasting them into google and finding others who had a similar problem and seeing if someone has found a solution for that problem. 50% of working with computers works through this process.

These steps have been designed for Mac OS. They have been successfully replicated on an Early 2015 Macbook Pro running macOS Monterey.

These steps have been taken from the H&D *Wifi Zine Throwie* workshop guide on their github page that can be found here.

1. Download the Arduino desktop app. Download an older version that is pre-2.0. I downloaded 1.8
2. Install the app. If you get a security pop-up, left-click the app and select "open" from the menu and then click the "open" button
3. Change the settings within the Arduino app. Click on the "Arduino" menu item in your top-menu bar and click "preferences".
    1. Check 'compile' and 'upload' in 'Show verbose output during:'
    2. Copy and paste the following into 'Additional Boards Manager URLs' and click 'Ok': `https://dl.espressif.com/dl/package_esp32_index.json`
4. Go to 'tools' > 'board' > 'Boards Manager...'
5. Type ESP32 into the search bar
6. In the 'Select Version' dropdown menu select 1.0.5 and click install
7. Download the Github repository for this workshop here: https://github.com/benjaminearl/ultralight-workshop
8. Find the compressed folders of the 'ESPAsyncWebServer' and 'AsyncTCP' in the 'libraries and plugins folder'
    1. decompress them and delete the '-master' part from the name at the end
9. Copy these 2 renamed folders to /Documents/Arduino/libraries on your computer
10. Restart the Arduino App
11. Open the 'ultralight-computer.ino' file that can be found inside 'code > ultralight-computer' folder in the repo
12. Once open, select the ESP32 Dev Module board under 'tools > Board'

13. Adjust your settings under the 'tools' menu as shown below (the only setting you should have to change is the Flash Mode to DIO): ``
    Board: ESP32 Dev module
    Upload Speed : 921600
    CPU Frequency : 240MHz (WiFi BT)
    Flash Frequency : 80MHz
    Flash Mode : **DIO** (is QIO by default)
    Flash Size : 4MB (32Mb)
    Partition Scheme : Default
    Core Debug Level : None
    PSRAM : Disabled
    ``
14. Click the compile button in the top left hand corner that has a tick in it.
    1. **If the compilation process is successful, it will say "DONE COMPILING" at the bottom**
    2. I received and error at this stage, so I had to change the python settings of my code by changing all instances of python (which by default is python 2) to python3. Check which version of python you have got installed on your computer and if you need to install it via this guide: https://docs.python-guide.org/starting/install3/osx/
    3. Then to change all instances of python copy and paste this command into your terminal:
       ```
       sed -i -e 's/=python /=python3 /g'
       ~/Library/Arduino15/packages/esp32/hardware/esp32/*/platform.txt
       ```
15. Download the Silabs USB communication chip driver download
16. Doubleclick "Install CP210x VCP Driver.app" to install it.
17. When it gives a security message, follow the instructions to allow the install to continue
18. Restart your computer
19. Make sure Gatekeeper is not interfering by going to System Preferences > Security and Privacy > General and seeing if there is an error message at the bottom of the page.
    1. If there is no error message, you can continue
    2. If there is an error message, GateKeeper is interrupting the driver's operation. If this is the case, click 'Allow' and confirm with administrator password, **then restart your computer**.
    3. If the error persists, find a guide on how to disable Gatekeeper on your MacOS
20. Plug in the MicroUSB into the ESP32 module and the USB into your computer
21. Restart the Arduino App
22. Check to see if the driver is installed by checking if 'SLAB_USBtoUART' shows as an option under 'tools > port'
23. Click the upload button (the one with an arrow pointing to the right) and while it says 'connecting...' in the terminal output press and hold the boot button of your ESP for 1 second.
24. If the upload is successful, you will see it say 'Done Uploading' above the terminal output
25. Download and install the ESP32FS plug-in and create a folder called 'tools' in your 'Documents/Arduino' folder
26. Your document structure should then be 'Documents > Arduino > tools > ESP32FS > tool > esp32fs.jar'

27. Restart the Arduino App. When open again, you should now be able to see an 'ESP32 Sketch Data Upload' option in your 'Tools' dropdown menu
28. Add your 1MB website to the folder called 'data' within the already existing 'data' folder of the repository. All the files you add to this directory will be uploaded to the ESP module
29. If you want to rename your Wifi network then rename the file ending in '.ssid'. Try to use only letters and numbers
30. Click the 'ESP32 Sketch Data Upload' button and wait for it to say 'SPIFFS Image Uploaded'
31. You can now unplug the ESP module from your computer and add it to the battery back module with the battery in.
32. You can now search for the wifi name that you choose, connect to it and your website should appear :)