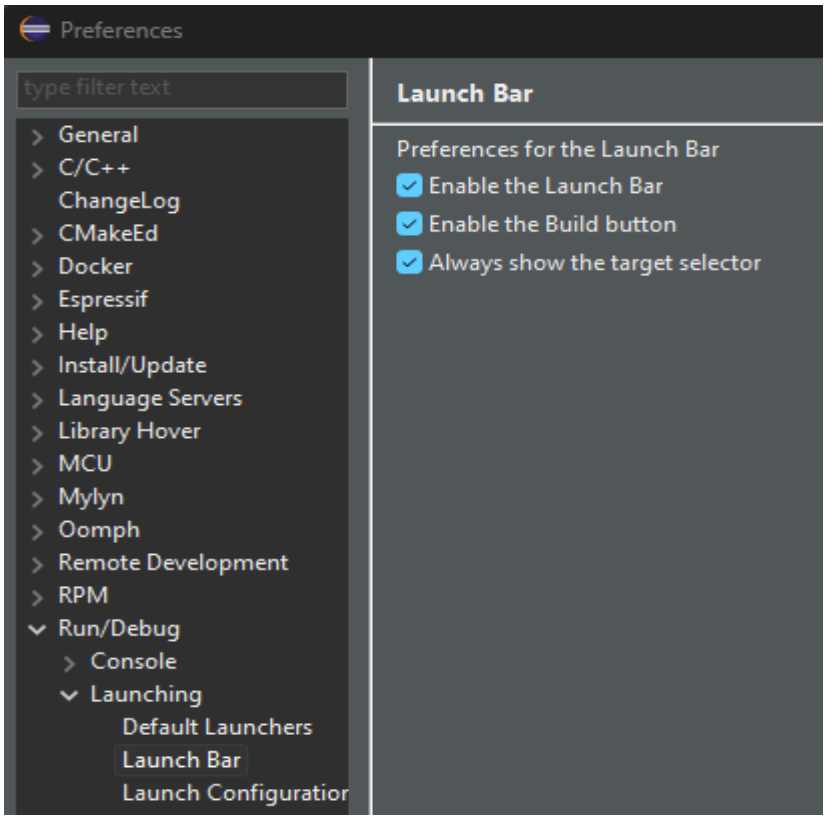
Everything About ESP32, ESP-IDF on Eclipse, ESP-PROG and JTAG debugging

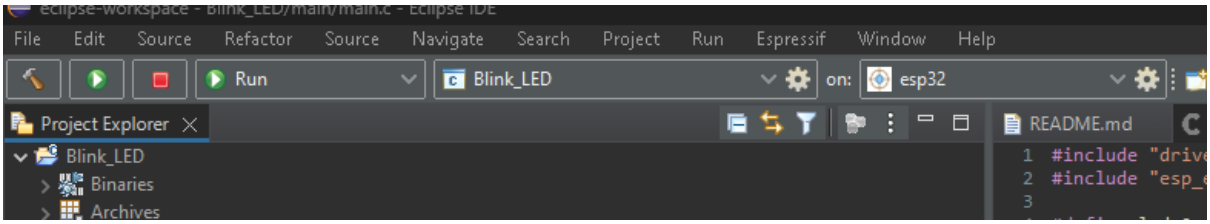
1. Install Eclipse IDE from: https://www.eclipse.org/downloads/packages/installer

2. Install ESP-IDF from e.g. https://github.com/espressif/esp-idf/releases/download/v5.2.1/esp-idf-v5.2.1.zip
    a. I installed to C:\ESP-IDF\esp-idf-v5.2.1
    b. Set-up ESP-IDF as per here: https://docs.espressif.com/projects/esp-idf/en/stable/esp32/get-started/windows-setup.html
    c. Command and PowerShell shortcuts will be added to your desktop, use these (Prefer PowerShell) to run python commands without using Eclipse e.g.
        i. Idf.py set-target esp32
        ii. Idf.py menuconfig (open blue config window)
        iii. Idf.py build
        iv. Idf.py -p COM30 flash
        v. Idf.py -p COM30 flash monitor (Serial output from ESP32 is displayed in the PowerShell window.

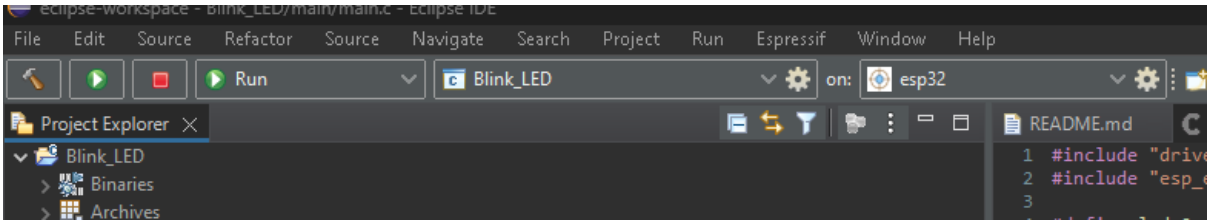
3. Project Launch Bar missing in Eclipse plug-in for IDF
    a. This is vital and not easy to discover, why is it not the default??
    b. In my case I had to turn it on in Window->Preferences->Launching->Launch Bar->Enable the Launch Bar. Took me a while to find it.
    c. in eclipse 2022-09 the "Launch Bar" setting is located under Window->Preferences->Run/Debug->Launching



4. Once the Project Launch Bar is added the vital green LAUNCH IN RUN MODE button appears, which uploads the code to the target.



5. The LAUNCH TARGET button can be used to specify the target hardware e.g. ESP32 and the associated COM port.



6. Format source code = highlight and Ctrl + Shift + F, or right click file e.g. main.c and > source > format. Or better install CppStyle by going to > Help > Eclipse Marketplace Select the Eclipse logo at the bottom and in the search box at the top put cppstyle and when found install it. Next go to Window -> Preferences -> CppStyle and set these two paths:
    a. Clang-format path: C:\Espressif\tools\esp-clang\16.0.1-fe4f10a809\esp-clang\bin\clang.exe
    b. Cpplint path: C:\Espressif\python_env\idf5.2_py3.11_env\Scripts\cpplint.exe
    c. And tick the boxes: Enable cpplint and Run clang-format on file save

7. 5: Install latest ESP-IDF:
    a. https://github.com/espressif/idf-eclipse-plugin/releases
    b. https://dl.espressif.com/dl/idf-eclipse-plugin/updates/beta/

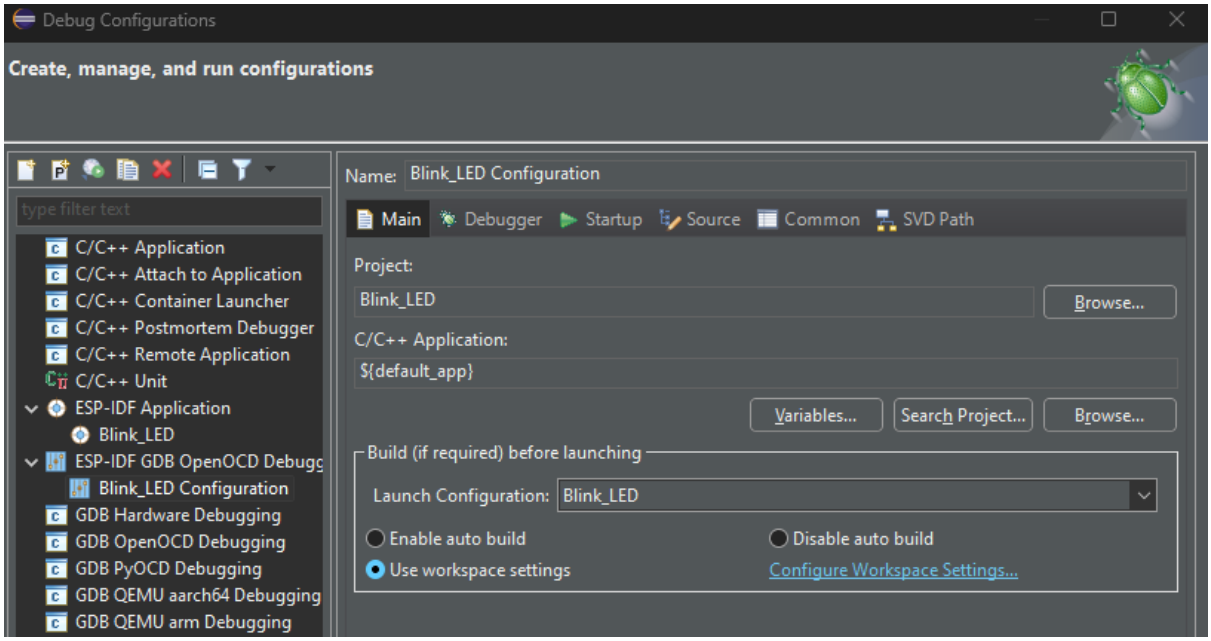
8. Eclipse HELP = https://help.eclipse.org/latest/index.jsp

9. Install Python, from Microsoft Store, by running the following command from a PowerShell window: C:\Users\Steven\AppData\Local\Microsoft\WindowsApps\python3.EXE
    a. Then when running >Espressif > Product Information from Eclipse it produces output without warning messages.

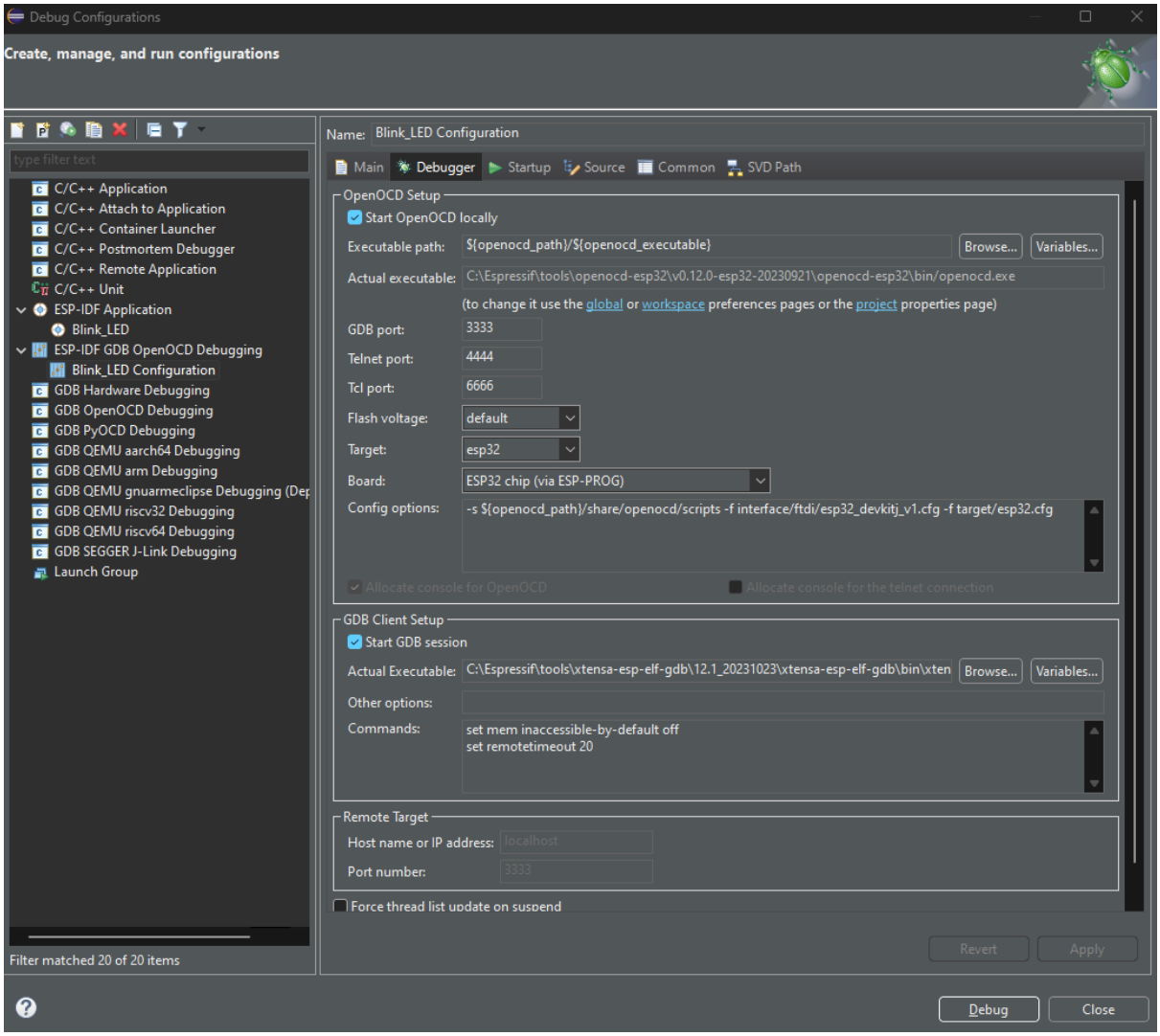
Everything About ESP32, ESP-IDF on Eclipse, ESP-PROG and JTAG debugging

10. Flashing message in info box at bottom right corner of Eclipse: “update rpm packages proposal list” and CPU usage very high with cooling fan at high speed!!
    a. It's a bug in the "Linux Tools" plugin. It seems that the plugin is unable to create the file specified at Window -> Preferences -> Specfile Editor -> RPM Completions -> Path to packages list file and therefore keeps trying to create it.
    b. Quick solution is: Go to >Windows > Preferences > RPM > Specfile Editor > RPM Completions and untick – Automatically build the RPM package proposals list
        i. Before you can do this go to >Windows > Preferences > RPM > Rpmlint and set he path to: C:\Users\Steven\.p2\pool\plugins\org.eclipse.linuxtools.rpm.rpmlint_1.1.0.202403061658.jar the path can be found by using the Everything app to search for rpmlint.
    c. Better solution: Create a text file e.g. new.txt in C:\Users\Steven\ and rename it to .pkglist i.e. C:\Users\Steven\.pkglist and leave the box ticked.
    d. For more info see: https://bugs.eclipse.org/bugs/show_bug.cgi?id=428302

11. Start debugging:
    a. Check OpenOCD is installed by opening the ESP-IDF PowerShell console and typing:
        i. PS C:\ESP-IDF\esp-idf-v5.2.1> openocd –version, which should produce:
            1. Open On-Chip Debugger v0.12.0-esp32-20230921 (2023-09-21-13:40)
    b. Connect ESP-PROG to computer and configure ports using Zadig
        i. See: https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-guides/jtag-debugging/configure-ft2232h-jtag.html
    c. Connect ESP-PROG to target hardware.
    d. Go to > Run > Debug Configurations > and right click ESP-IDF GDB OpenOCD Debugging and choose New Configuration.
    e. On the MAIN tab browse for the Project e.g. Blink_LED



f. On the DEBUGGER tab configure as below:



g. The other tabs can be left as default.

h. Click the Debug button to start debugging.