

June 2023

About this guide

This guide describes the steps to set-up an Eclipse based development environment for the **esp32-s3-wroom-1** kit in an Oracle Virtual Box running Linux Debian 10 hosted in Windows 11.

The software used to prepare this document is listed below. See the Resources section for the download sites.

- Host OS Windows 11 Pro
- Oracle VirtualBox Version 7.0.8 r156879 (Qt5.15.2)
- Guest OS Linux Debian
- ESP IDF
- Java 20.0.1 2023-04-18
- Eclipse IDE for C/C++ Developers 2022-09
- Python 3.11.1
- Git

Summary

Next steps need to be completed in the Windows machine

1. **Install Window drivers for the USB On-The-Go (OTG) (USB Type-C) port present in the esp32-s3**
2. Install Oracle Virtual Box with Linux Debian
3. Add the esp32-s3 USB devices to the VM filters

Following steps are done in Linux running in the VM

1. Add account which is going to be used to run Eclipse to the sudoers file
2. Install Java 20
3. Install Python 3.11.1
4. Install Eclipse IDE for C/C++ Developers 2022-09
5. Install ESP IDF plugin for Eclipse
6. **Update access rules for openocd and JTAG interface**
7. **Update security groups for account running eclipse**
8. Install Git
9. Create a New Launch Configuration to start the debugger

The steps in bold might be required even in case you are not using the Eclipse IDE.

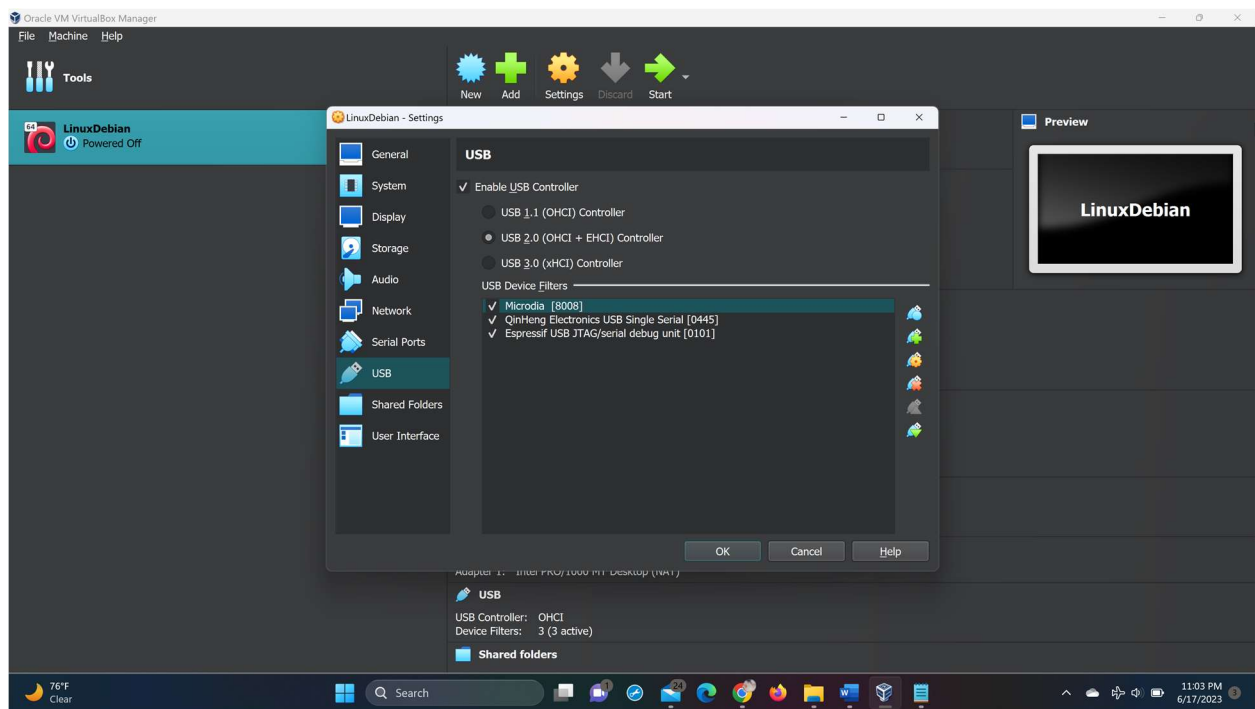
Detailed steps

1. The drivers for the USB OTG port in the esp32-s3 are not present in Windows 11 by default. Open a PowerShell command prompt and run next command

```
Invoke-WebRequest 'https://dl.espressif.com/dl/idf-env/idf-env.exe' -OutFile .\idf-env.exe; .\idf-env.exe driver install --espressif
```

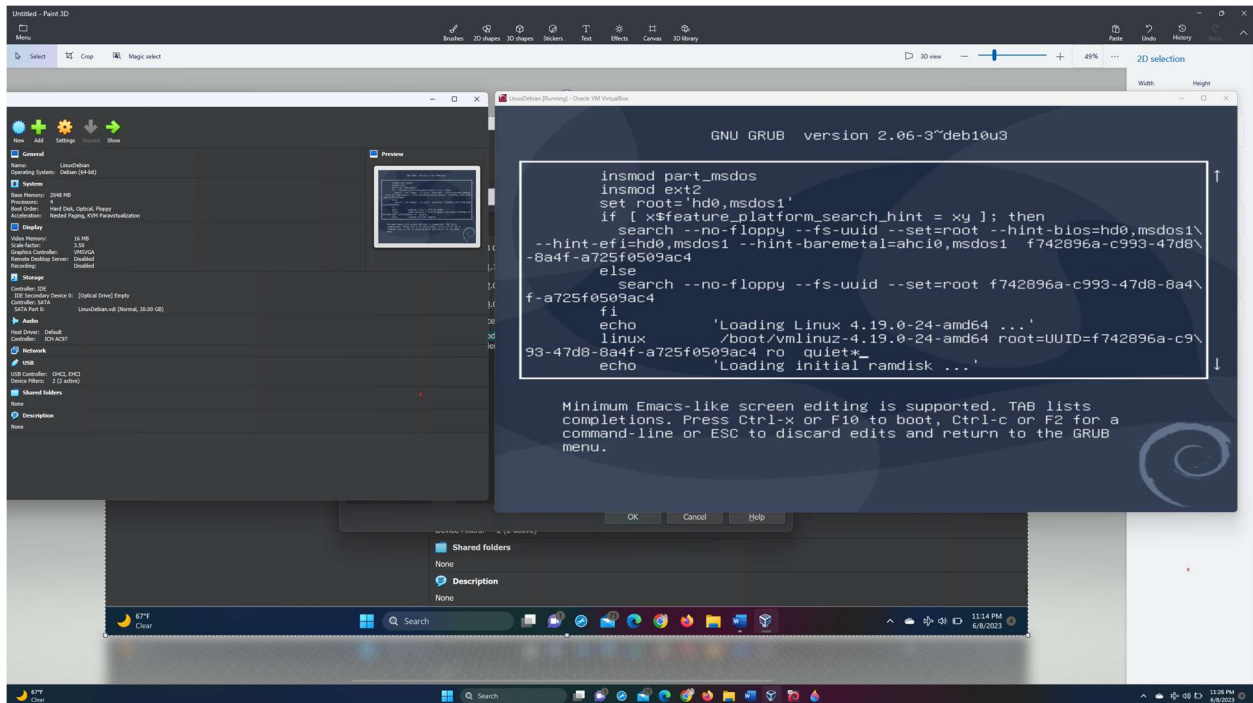
After executing above command, when esp32-s3 board is plugged to your system, you will see the entry *Espressif USB JTAG/serial debug unit* in the Windows Device Manager.

2. Install Oracle Virtual Box with Linux Debian 10. For this exercise the unattended installation was used.
3. Add the ESP32 USB devices to the VM filters (USB filters allow you to automatically mount USB devices in VirtualBox VM).
 - a. Under USB Settings add the filters for your ESP32. You can plug and unplug your ESP32 to identify the device(s) associated to it.



4. Start your VM
5. Your regular linux account will need sudo access.
 - a. Start Linux in recovery mode, when GRUB starts, press e to edit the loader configuration. Look for the linux kernel line and edit it to start in single mode by changing the word quiet to single.

- b. Open the command line by pressing `ctrl + c`
- c. Change the root password using
password root
- d. Exit from the command line and let the system start by pressing `Ctrl + x`. Logging as root using the password you set in the step d.
- e. Edit the sudoers file to grant sudo access to your regular account
- f. After changes to sudoers file are complete, resume Linux booting process



6. Install Java

```
sudo apt install ./jdk-20_linux-x64_bin.deb
```

7. Install Python

```
sudo apt update
```

```
sudo apt upgrade
```

```
sudo apt install build-essential zlibg-dev libncurses5-dev libgdbm-dev libnss3-dev libssl-dev libreadline-dev libffi-dev  
libsqlite3-dev wget libbz2-dev zlibg-dev liblzma-dev
```

```
wget https://www.python.org/ftp/python/3.11.1/Python-3.11.1.tgz
```

```
tar -xvf Python-3.11.1.tgz
```

```
cd Python-3.11.1
```

```
sudo ./configure --enable-optimizations
```

```
sudo make -j 2
```

```
sudo make install
```

to check installation

```
python3.11 -V
```

8. Install Eclipse

Copy the eclipse bundle to the directory where you decide to install it and then execute next command

```
tar xzvf eclipse-cpp-2022-09-R-linux-gtk-x86_64.tar.gz
```

then change your working directory

```
cd eclipse
```

Start eclipse using sudo

```
sudo ./eclipse &
```

9. Install ESP IDF plugin for Eclipse

- a. Installing IDF Plugin using update site URL
- b. Installing ESP-IDF
- c. Installing ESP-IDF Tools

For this task follow the steps described in <https://github.com/espressif/idf-eclipse-plugin/blob/master/README.md>

After installing the ESP-IDF toolset you will have a directory containing

- cmake
Cross compiler
- xtensa-esp-elf-gdb/bin/xtensa-esp32s3-elf-gdb
Debugger
- openocd-esp32/v0.11.0-esp32-20221026/openocd-esp32/bin/openocd
Open-On-Chip Debugger

10. Make sure the the rules file for openocd (/etc/udev/rules.d/60-openocd.rules) and the rules file for the esp32 JTAG interface (/etc/udev/rules.d/70-esp32s3-jtag.rules) contain next lines

```
ATTRS{idVendor}=="303a", ATTRS{idProduct}=="1001", MODE="660", GROUP="plugdev", TAG+="uaccess"  
ATTRS{idVendor}=="303a", ATTRS{idProduct}=="1002", MODE="660", GROUP="plugdev", TAG+="uaccess"
```

11. Add the account running eclipse to the dialout and plugdev groups

```
sudo usermod -a -G dialout your_username
```

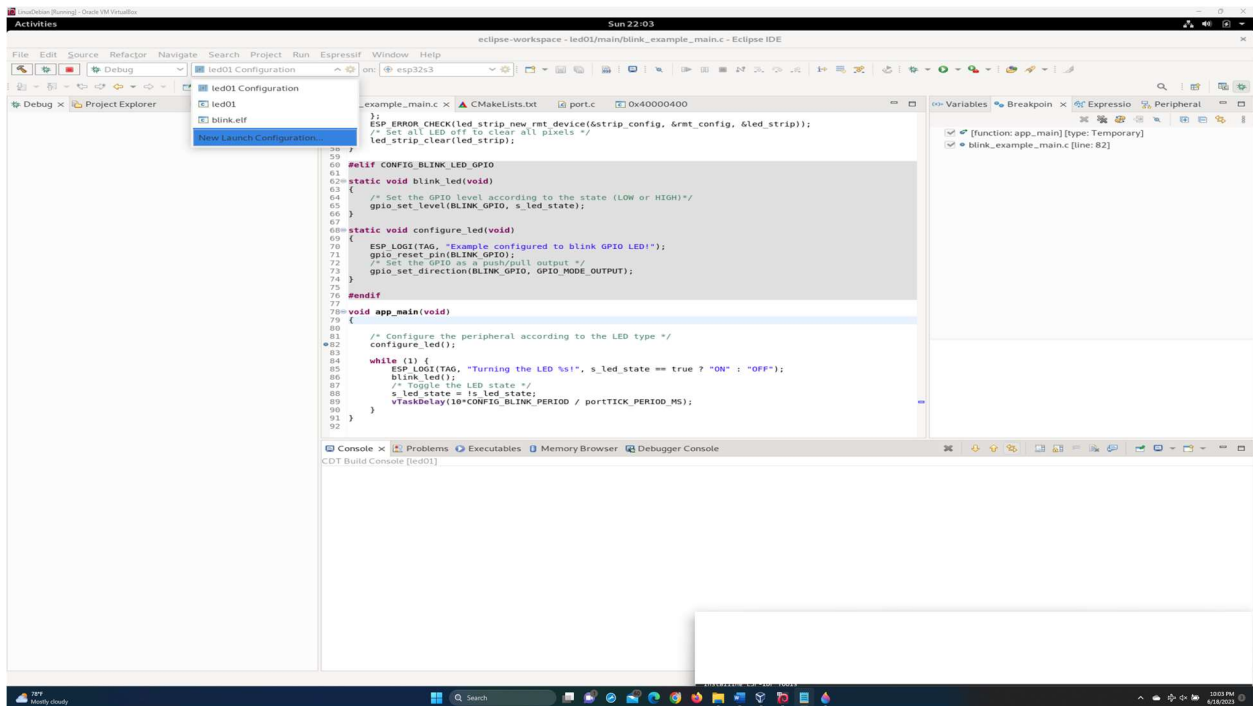
```
sudo usermod -a -G plugdev your_username
```

12. Install Git

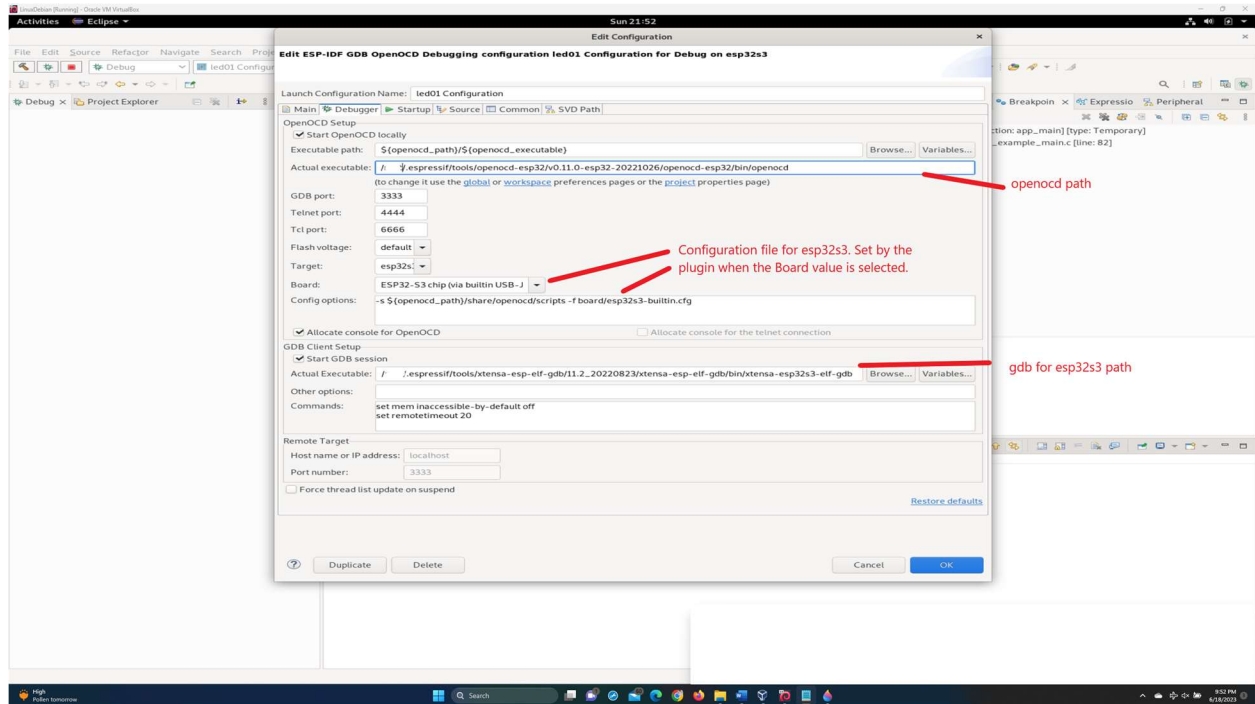
```
sudo apt-get install git
```

At this point you should be able to start eclipse, create a new project for esp32s3 and initialize the debugger. You need to complete next steps

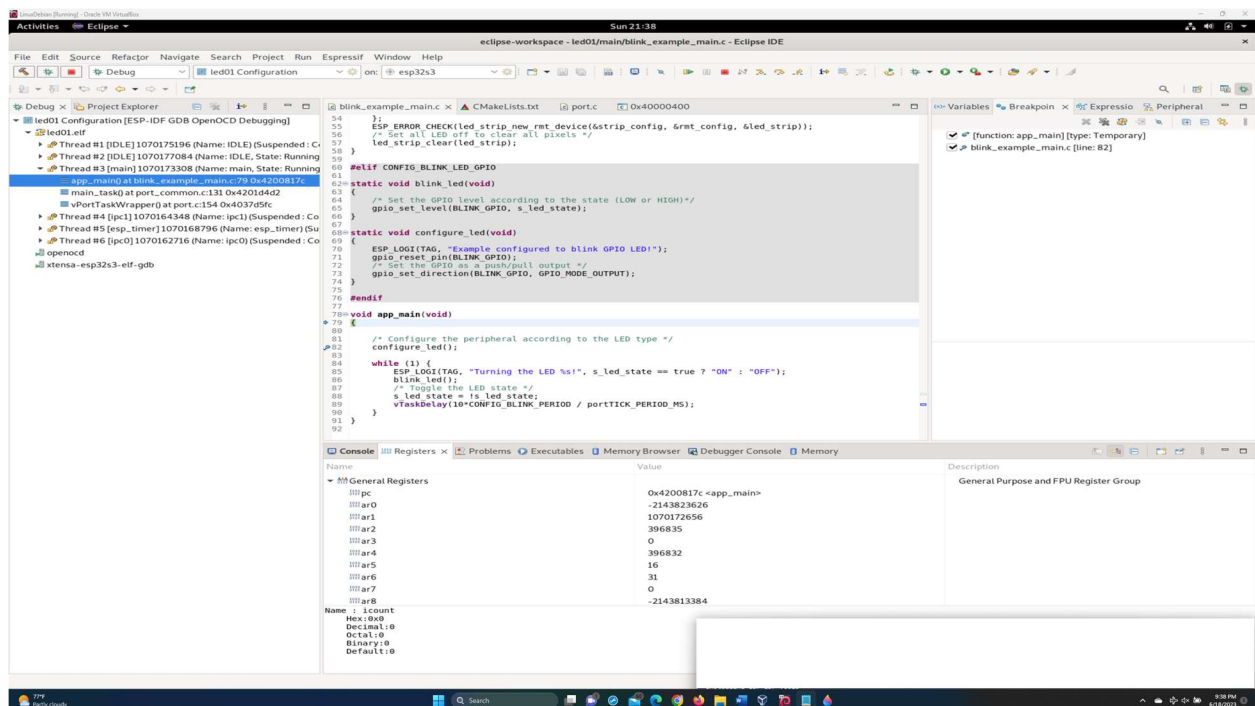
1. Create a New Launch Configuration



2. Configure the Target and Board for the new Launch Configuration. Ensure the path to openocd and xtensa-esp32s3-elf-gdb are correct. See below example.



3. Start debugging. Make sure your esp32-s3 board is plugged to your computer using the USB OTG port.



Resources

- <https://github.com/espressif/idf-eclipse-plugin/blob/master/README.md>
- <https://www.oracle.com/technetwork/java/javase/downloads/index.html>
- <https://www.eclipse.org/downloads/packages/release/2022-09/r/eclipse-ide-cc-developers>
- <https://www.python.org/downloads/>
- <https://git-scm.com/downloads>