

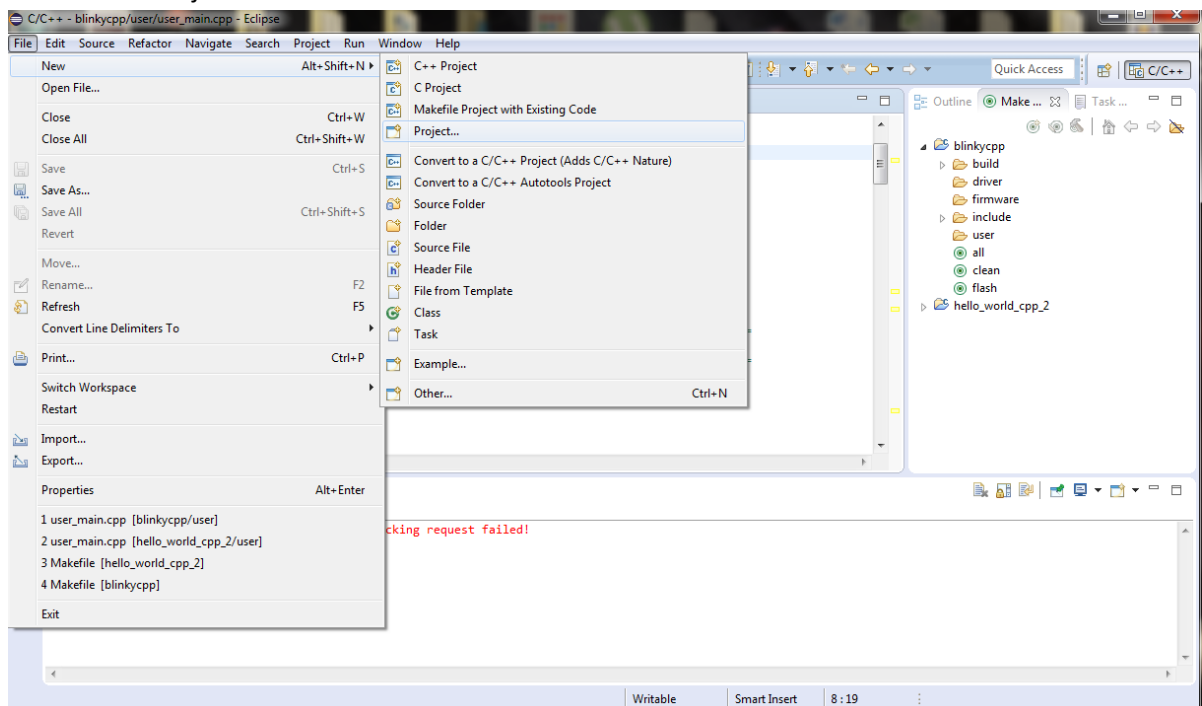
Environment Setup

1. Install [Eclipse with C/C++](#)
2. Install MingGW
 - a. I used the following link as reference [How to Install MinGW](#)
3. Install the [Unofficial Development Kit for Espressif ESP8266](#) Developed by Mikail Grigorey
 - a. This contains the official Espressif SDK
 - b. Unofficial GCC compiler for SoC Xtensa LX 106
 - c. Documentation for SoC ESP8266
 - d. Examples in the firmware source code in C
 - e. Additional tools for working with boards based on ESP8266
 - f. This also contains 2 sample hello_world projects in C++
 - g. The installation will create the directory C:\Espressif
4. In regards to Eclipse and the Eclipse workspace
 - a. In order to create new projects for the ESP8266 in C++, I had to have the workspace set up in C:\
 - b. Specifically I chose to have the workspace in C:\Espressif

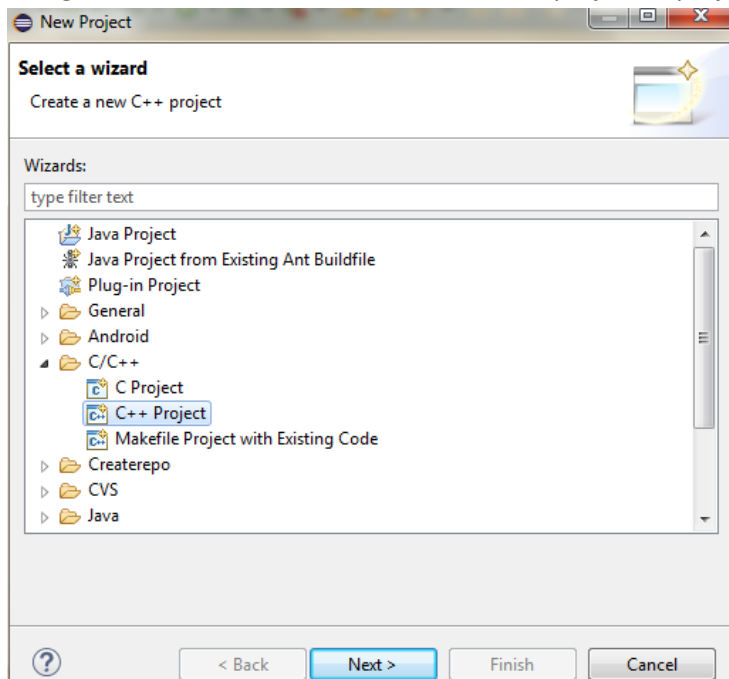
Steps in order to create a new project

I loosely followed the linked YouTube video with some modifications

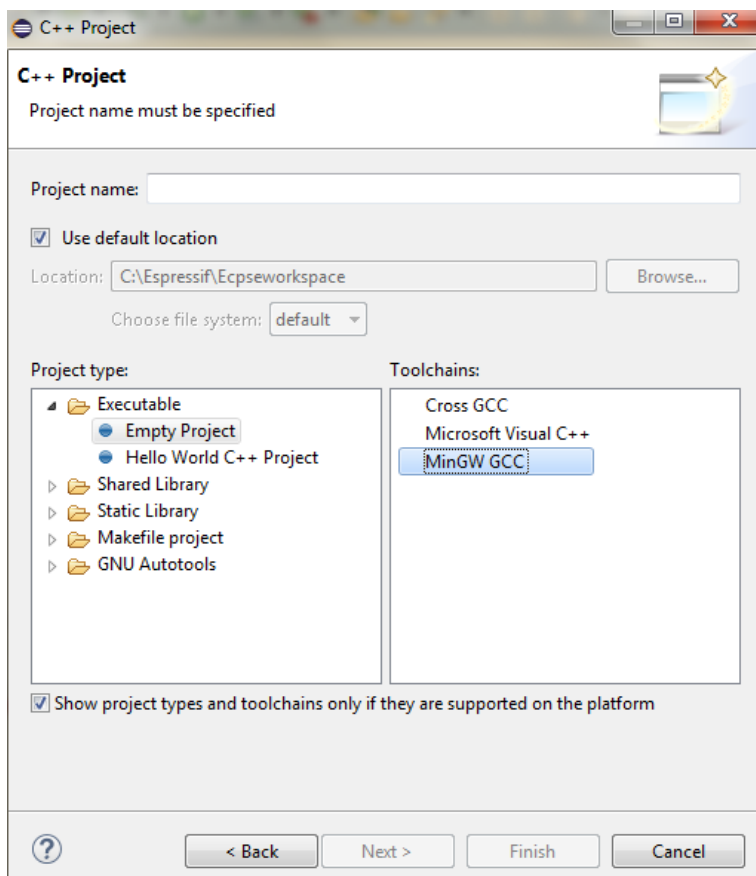
- a. [Creating New Project in Eclipse for ESP8266](#)
1. Open Eclipse
2. File->New->Project



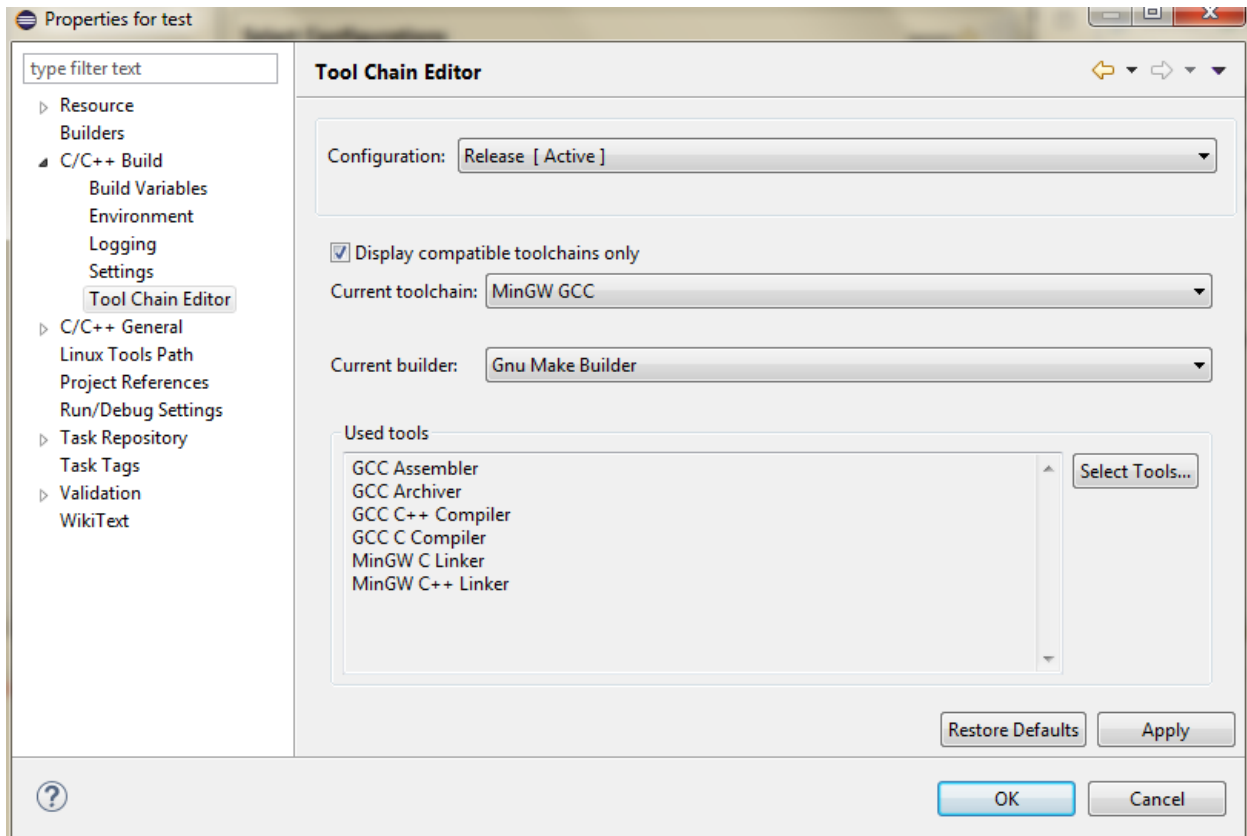
3. Navigate to the C/C++ folder and select a C++ project (C project was selected in the video)



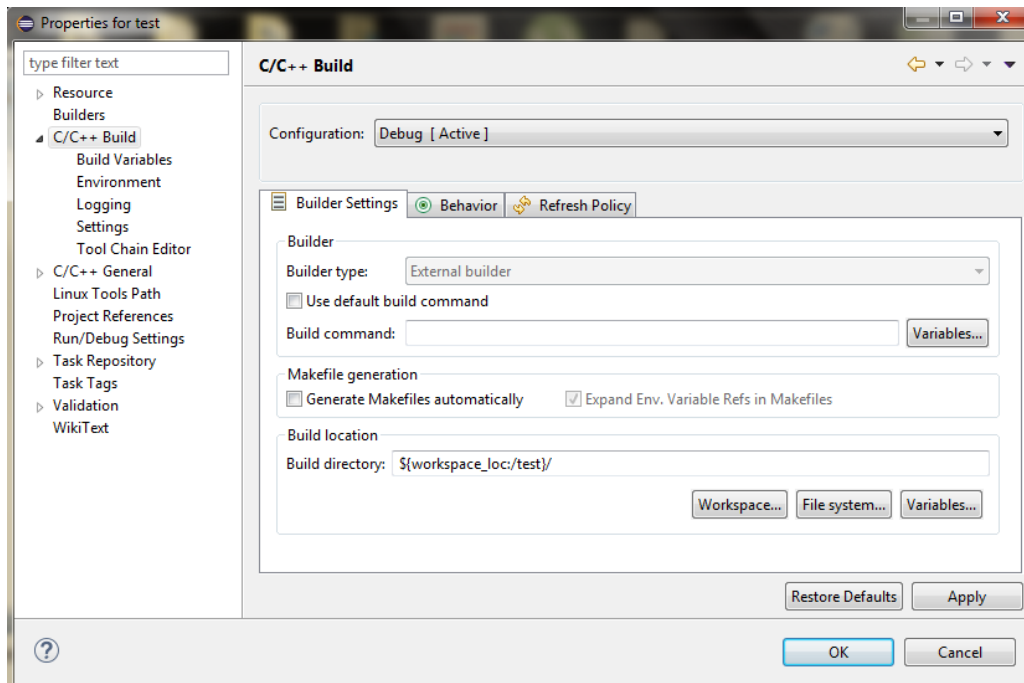
4. Press Next
5. For Project type select Empty Project
6. For Toolchains select MinGW GCC



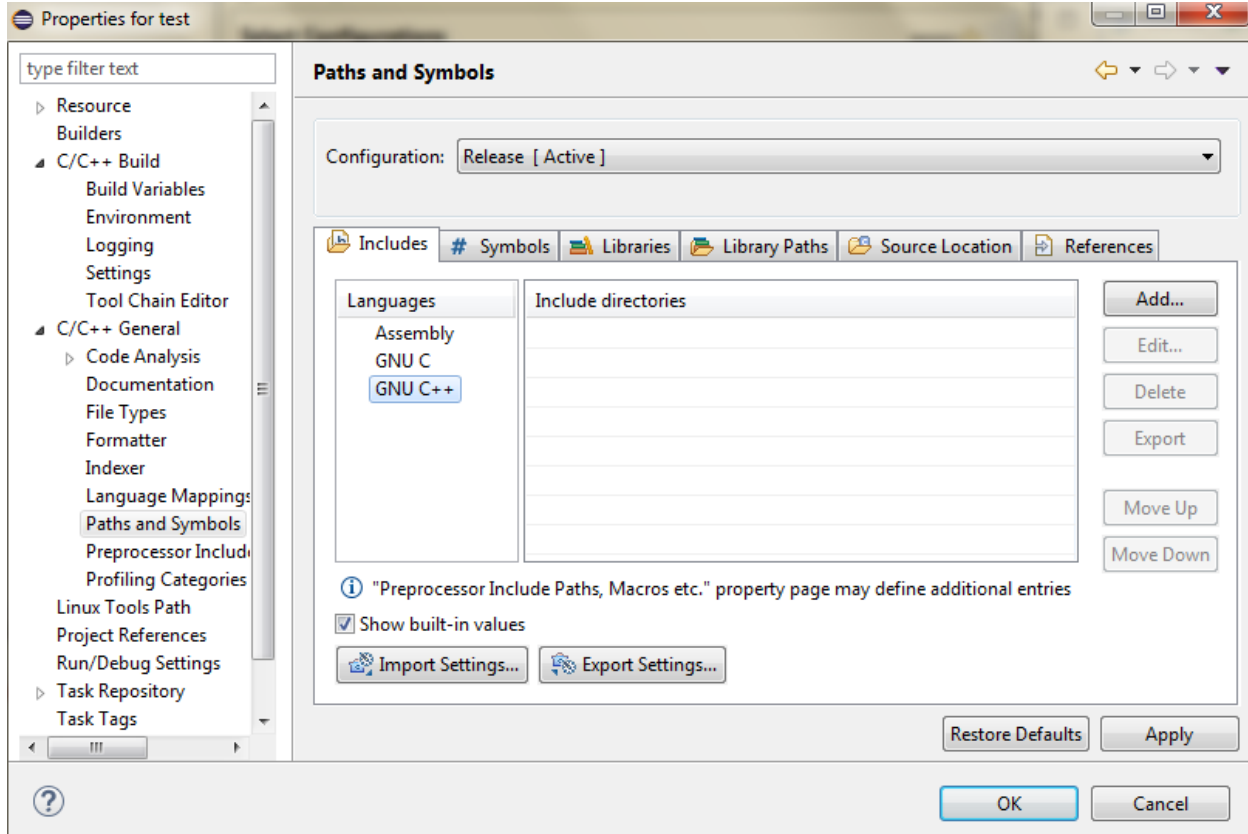
7. Name your project and press Next
8. Click Advanced Settings
9. Click the drop Down for C/C++ Build
 - a. Select Tool Chain Editor
 - b. Switch Current Builder to BE GNU Make Builder



10. Select the C/C++ Build dropdown
 - a. Unselect Use default build command
 - b. In build command write: **mingw32-make.exe -f \${ProjDirPath}/Makefile**
 - c. Unselect Generate Makefiles Automatically



11. Click the drop down for C/C++ General
 - a. Select Path and Symbols
 - b. Under Includes for Languages select GNU C++



- c. Select Add for include directories
 - i. Select Filesystems
 - ii. Select the include folder in the Espressif folder
 - 1. C:\Espressif\ESP8266\include
 - 2. Press OK
 - d. Click Add to all configurations
 - i. Press OK
 - e. Press OK again
- 12. Click Finish
- 13. On the right hand side under Make Target select the folder for your new project
- 14. Click The MakeNewTarget
 - a. Make a target named all
 - b. Make a target named clean
 - c. Make a target named flash
- 15. On the left side in project explore select the folder for your project
 - a. Right Click and select New->Folder
 - i. Make a folder named user
 - b. Select the folder user
 - i. Right click and select New->Source File
 - 1. Name this file user_main.cpp
 - ii. Right click the user folder again and select New->Header File
 - 1. Name this file user_config.h
 - c. Right Click the project folder again and select New->File
 - i. Name the file Makefile
 - d. Copy the contents of a working MakeFile (I used the one from hello_world_cpp_2) into this file
- 16. On the right side in MakeTarget, under your newly created project, double click the all target
 - a. This will compile the code in your project
 - b. This will create the folders build and firmware
- 17. Double click the flash target in order to flash code to the ESP8266