Eclipse with the ESP8266

California Plug Load Research Center
Winter 2018

Justin Le & Jerry Lee

March 20, 2018

Contents

1 Introduction

In this guide, you will be instructed on how to use Eclipse IDE with the ESP8266, rather than using the Arduino IDE. This guide is based on having a Windows host so for Mac OS and Linux users, the steps may be different. Please note that the following setup has been done for Windows 10 x64 and may not work otherwise. Installation and setup instructions are adapted from original instructions made by Sean Santarsiero!

2 Setting Up Windows Environment

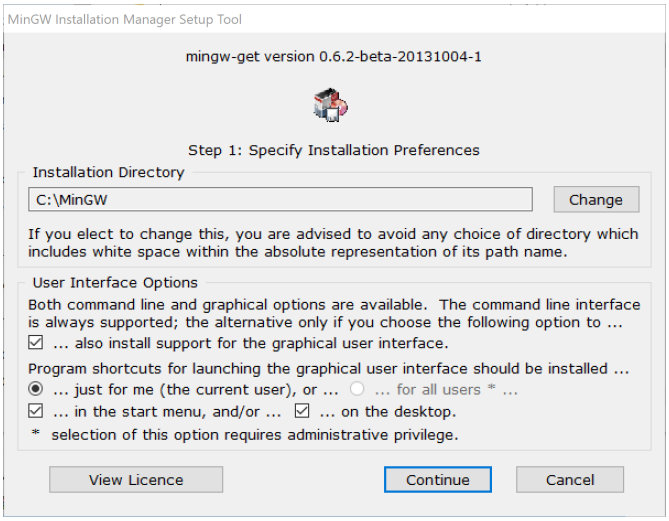
2.1 Updating Java SE Runtime Environment

Before you can setup the ESP 8266 for windows, you must have the most updated Java SE Runtime Environment. You can find this by going to the Oracle website and navigating to Java then the Java SE Download page. Or click here to the download page or follow this link `http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html`. The current version used in this guide is 8u161.

Note: You may need to uninstall any previous Java SE Runtime Environments for this to work.

2.2 Download MINGW

MinGW is an open source software development environment specifically used for creating Windows Applications. Click here or use this link to directly download the an executable file to install MinGW:
`https://sourceforge.net/projects/mingw/files/latest/download?source=files`
After the initial download, run the executable and install into your (C:) drive as shown below:



3 Install Eclipse

If you already have the following version of Eclipse then ignore this step. We will be using the Eclipse Oxygen 2 IDE for C/C++ Developers. To download this you can navigate to this link: `http://www.eclipse.org/downloads/packages/eclipse-ide-cc-developers/oxygen2` and select the Windows 64bit option. Unzip the downloaded zip file into your (C:) drive into the eclipse folder such that your path will be *C:\eclipse*.

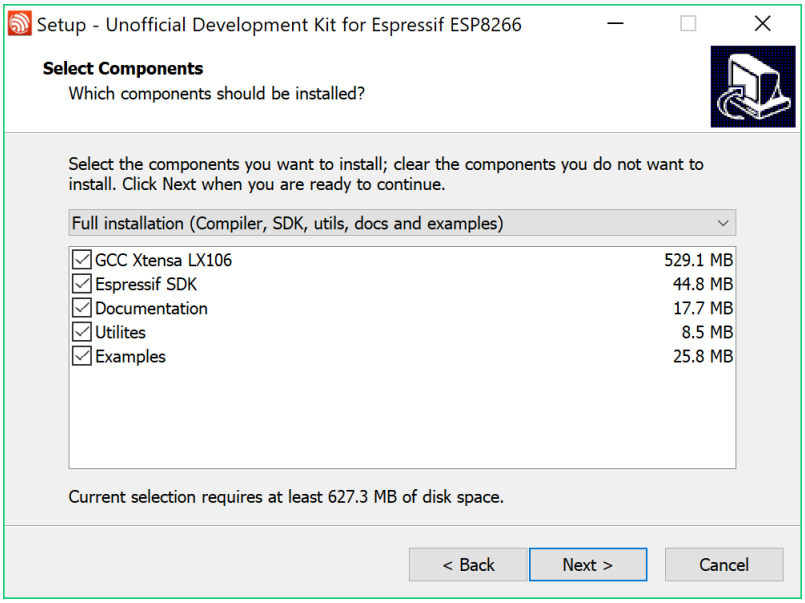
Eclipse must be unzipped into the (C:) drive because an eclipse MakeFile cannot handle spaces such that if unzipped into, for example, your Users folder, it will cut off. Example: *C:\Users\Justin Le* will become *C:\Users\Justin*.

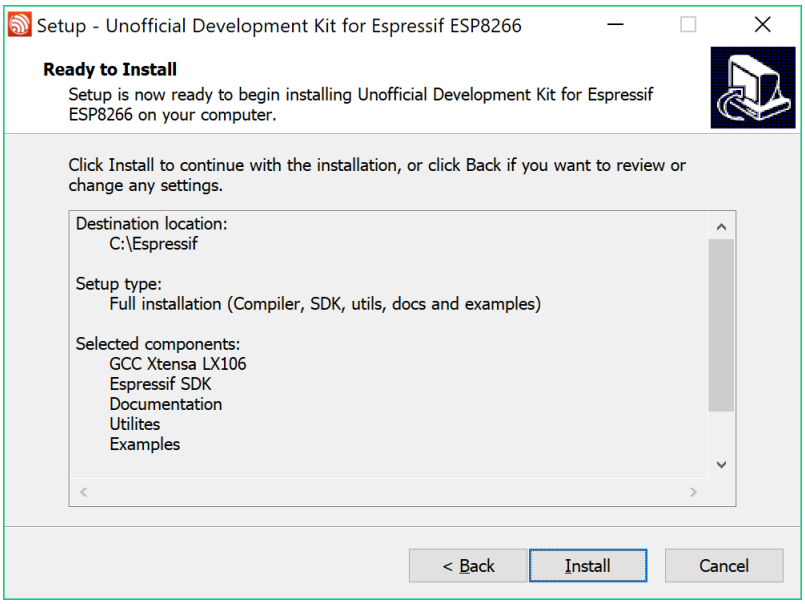
4 Installing and Setting Up Espressif SDK

4.1 Installing

Next download Developer Mikhail Grigoriev's Unofficial SDK, we are using version 2.0.9 for this guide. Using a different/newer version may lead to issues.

To download this, click here or use the following link:
http://www.nefastor.com/wp-content/uploads/2016/12/Espressif-ESP8266-DevKit-v2.0.9-x86.exe
This will download and executable file, run the executable and this will install the Espressif folder into you (C:) drive. See below for reference:

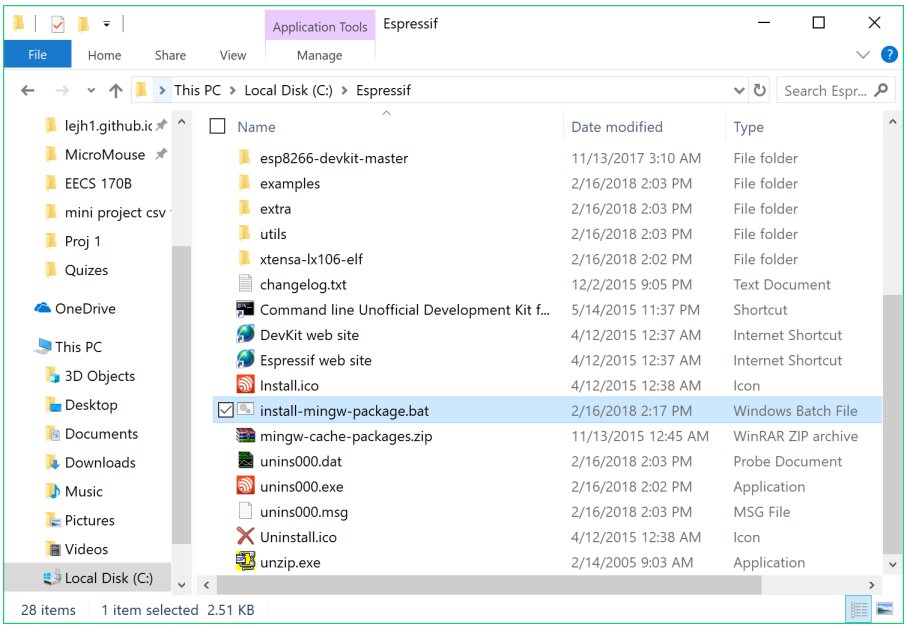




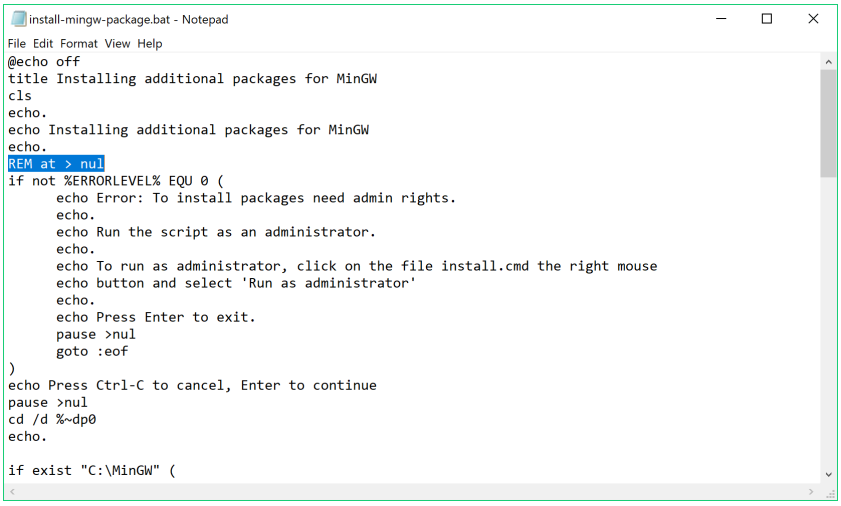
Your path to the Espressif folder should now be *C: \Espressif*.

4.2 Setting Up

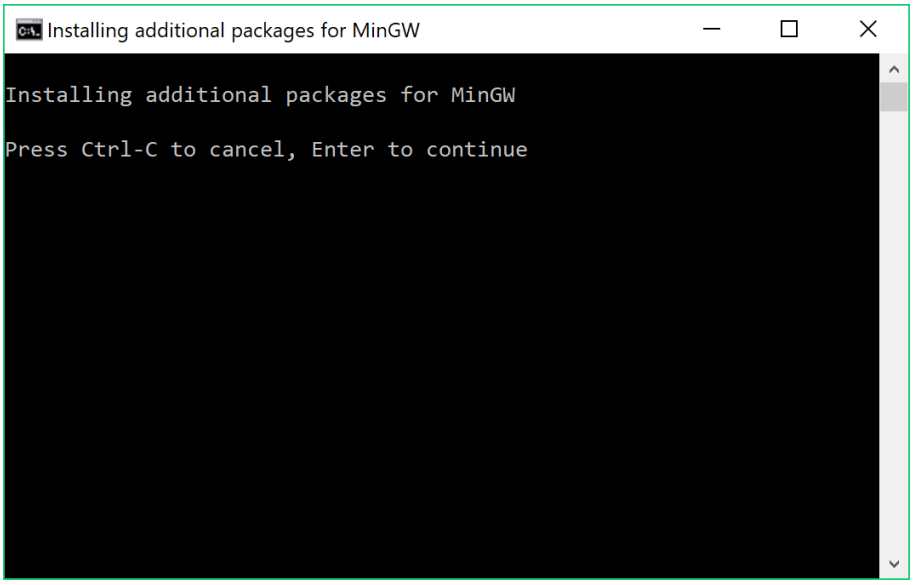
Next we have to download the development add-on tool package. To download, click here or use the following link: http://www.nefastor.com/wp -content/uploads/2016/12/Espressif-ESP8266-DevKit-Addon.zip This will download a zip file. Unzip the add-on tool package into the your Espressif folder, *C: \Espressif*. Next go into your Espressif folder and find the install-mingw-package.bat file as seen below:



Right-Click on this file and select edit. You will need to find the line "at > nul" and change it to "REM at > nul" as seen below:



Save the edited file then run the install-mingw-package.bat file. You will get the following command prompt:



Simply press *Enter* to continue with the installation. If you get issues installing, try running it as an administrator.

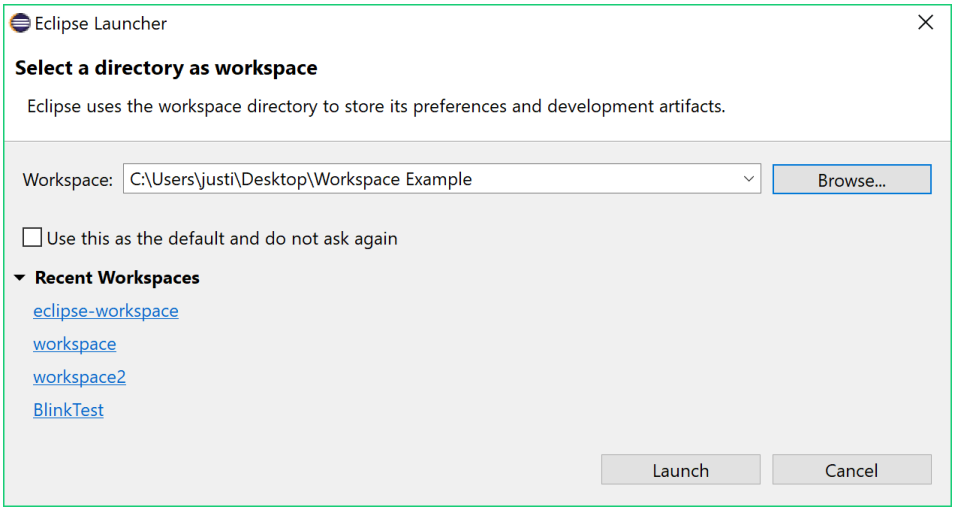
5 Example Source Code for Various Applications

This section is solely for importing existing ESP 8266 Examples for our use and reference. The examples included in the folder covers a wide variety of protocols and applications that will be useful to refer to when writing software for the ESP8266.

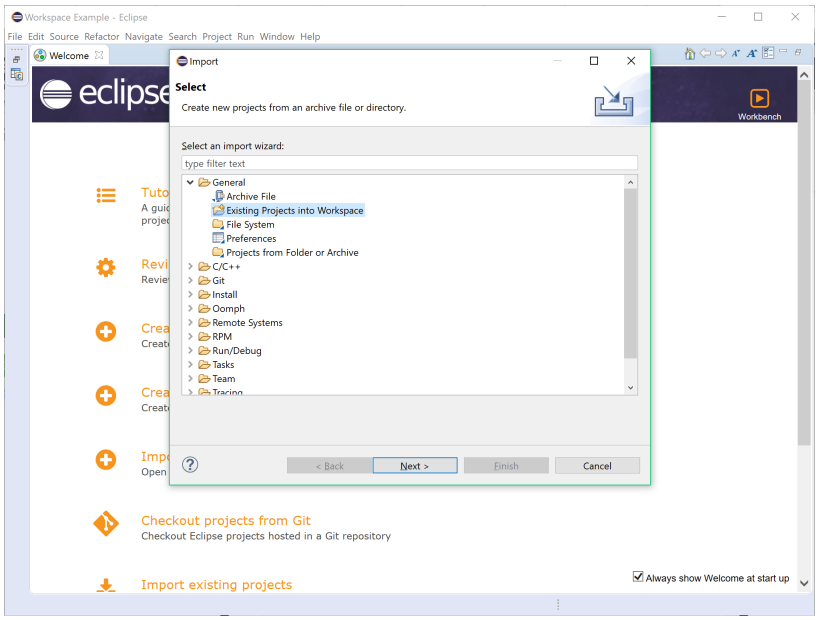
5.1 Import to Workspace

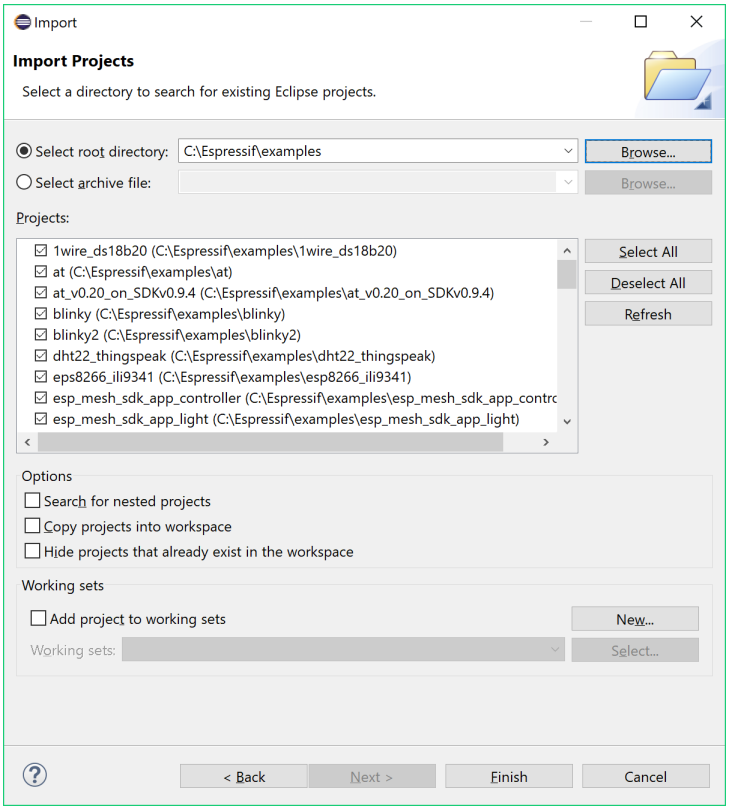
Previously in the setup, when running the Development Kit for Espressif ESP8266 executable, it downloaded the examples into Espressif so now we just need to import them.

Open Eclipse by double-clicking the *eclipse.exe* executable in your eclipse folder. You can create your own workspace by making a new folder anywhere on your computer and adding the path when prompted or use the *Browse* option to locate it as seen below. You can also use the default eclipse-workspace workspace.



From the Eclipse window, select the File Menu on the top left corner. Choose Imports, General, Existing Projects into Workspace as seen below.





Here you will specify the location of the examples by either selecting *Browse* and locating the examples folder (same path as follows) or typing it in as stated below:

C:\Espressif\examples

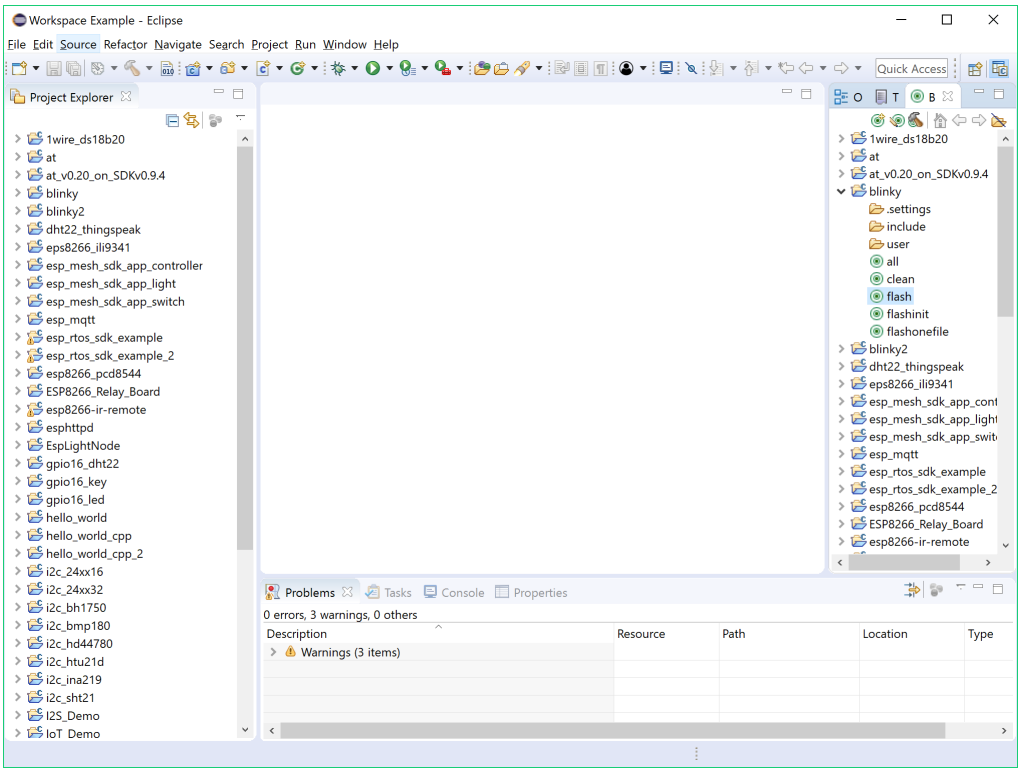
This will import the sample projects into your workspace. Now you can open the source code, modify it, and build it however you please.

6 Compile and Program

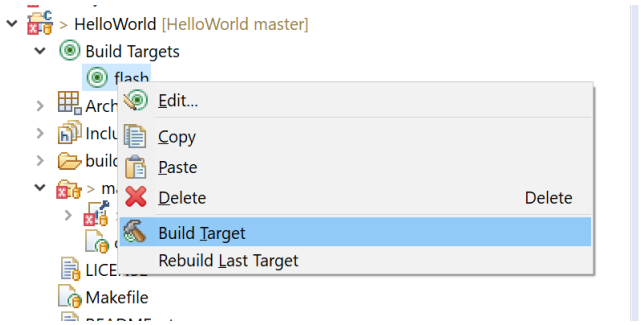
6.1 Building and Flashing

On the right side of the Eclipse application, there is a window with tabs named "Outline", "Tasks Lists", and "Build Targets". Go into "Build Targets" and find the project you want to compile. Click on flash and it should work.

Alternatively, you can go into your Project Explorer on the left side. You can open your project and view the contents by double clicking on the project folder if you do not already have the folder collapsed. From there you can double-click on "Build Targets". You will see flash and can double click there to compile and flash the code. Both methods are the same.



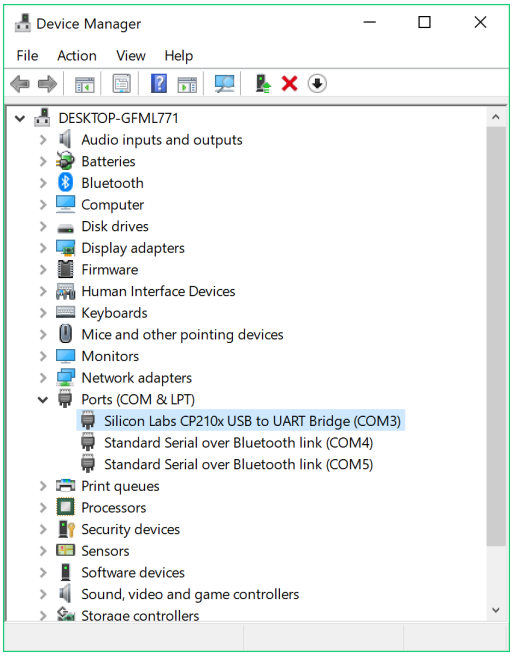
Method 1



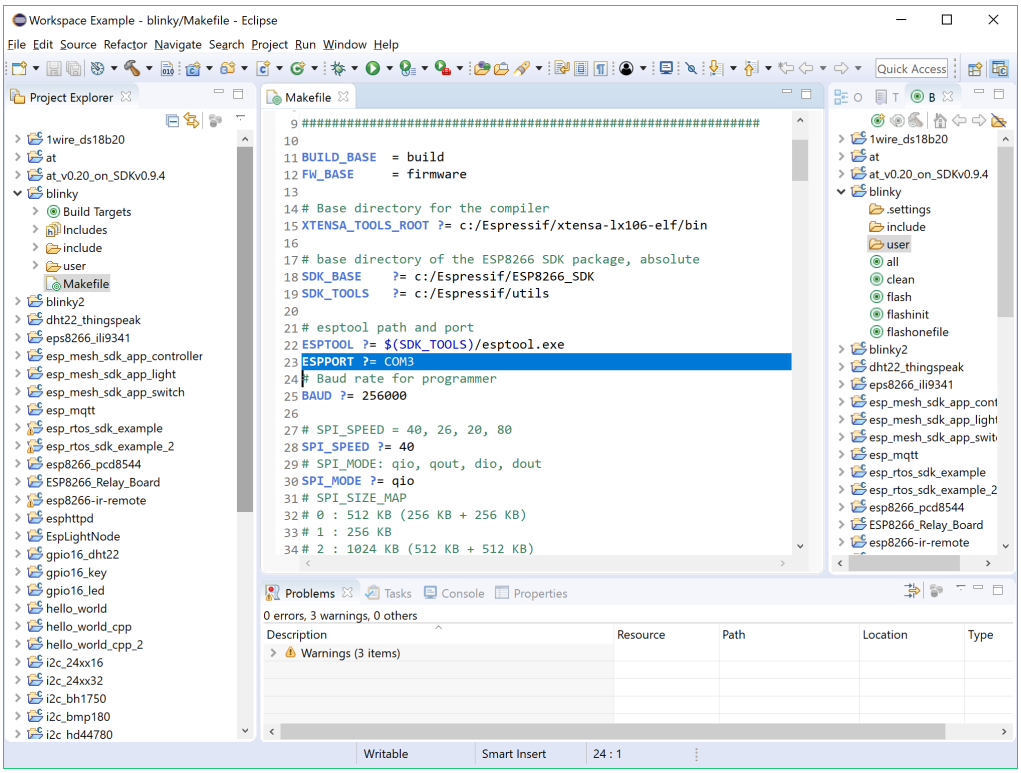
Method 2

6.2 Configuring Makefile

You will only need to configure your Makefile in the case that you receive a PORT or COM error when flashing. By default, most examples and ESP code templates will come with Makefiles that port to a port that may not be the same as your computers. To resolve this error, you must open your computers "Device Manager". You can find this by typing it into the Windows search bar. From there, make sure the ESP8266 is plugged into the computer. You should be able to see in the list "Ports (COM & LPT)". Click the arrow to open the drop down menu and you should see something connected, followed by the COM with a number in parentheses as seen below.



That is your port you need to change to in the project's Makefile. Note: Your name of the device may not be the same as the one seen above in the picture. In Eclipse, open your project's Makefile so you can edit the code.



Under "# esptool path and port" there should be an variable "ESPPORT". You will notice the port define there is not the same as your port so you much change it to your own specific computer's port. For example, on my computer, I have to use COM3. In the Makefile, I will change the port so it should look like the following:

ESPPORT ?= COM3

Now save your Makefile and try flashing again. This time you should be able to flash without any issues.