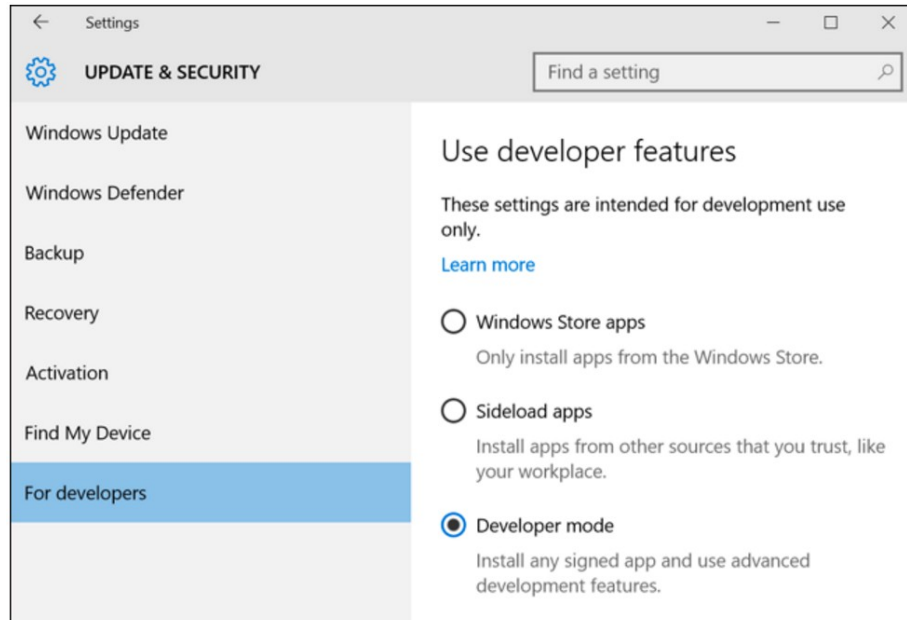


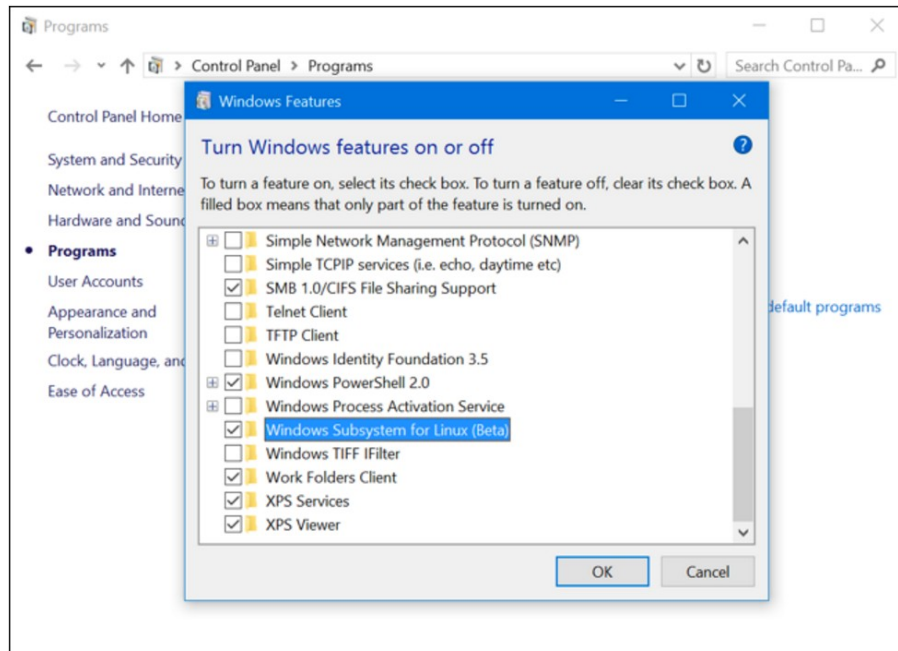
Directions for Setting up ECE 486 programming environment on Windows 10:

1. Enable Linux subsystem for Windows

- a. Open settings app, go to Update & Security > For Developers. Activate the “Developer Mode” switch here to enable Developer Mode.



- b. Open the Control Panel, click “Programs,” and click “Turn Windows Features On or Off” under Programs and Features. Enable the “Windows Subsystem for Linux (Beta)” option in the list here and click “OK.”



c. Reboot Computer

2. Install and update Ubuntu:

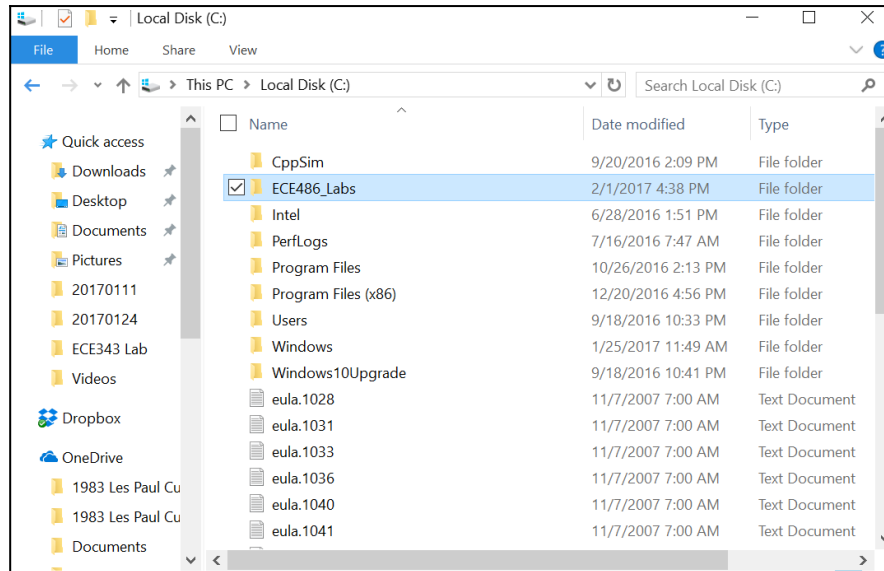
- a. Install the (free) Ubuntu app from the Windows store.
- b. Run the app, and set up your linux user name and password. This can be the same or different from your windows login and password – just remember your credentials.
- c. Update Ubuntu using the following commands

```
sudo apt-get update
```

```
sudo apt-get dist-upgrade
```

3. Set up a shared directory between your Windows and linux systems.

- a. Go to the C drive on the windows file explorer and create a folder called "ECE486\_Labs"



b. To create a link between the folders, In the Ubuntu shell, enter the following commands:

```
cd ~
ln -s /mnt/c/ECE486_Labs
ls
```

c. You should see an ECE 486 Folder appear in your home directory. Anything you drop into this folder from the Windows side will be available in the bash shell.

4. Download cross compiler and other required software for programming environment

```
sudo apt-get install gcc g++ git autoconf automake libusb-1.0-0-dev cmake
```

```
sudo apt-get install gcc-arm-none-eabi libnewlib-arm-none-eabi
```

```
sudo apt-get install unzip
```

***If you're doing the install from off campus, you'll need to be running the Umaine VPN to perform the following steps (accessing the supplemental web site, or accessing the git server).***

5. Install C libraries required for ECE486.

```
wget -N http://web.eece.maine.edu/~hummels/classes/ece486/src/stmdevel.zip
unzip -d stmdevel stmdevel.zip
cd stmdevel/build
make
sudo make install
cd ..
```

6. Install git by following setup instructions here:

[http://web.eece.maine.edu/~hummels/classes/ece486/docs/git\\_instructions.pdf](http://web.eece.maine.edu/~hummels/classes/ece486/docs/git_instructions.pdf)

- Clone your repository into ECE\_486 Labs folder, changing XX to your group number

```
cd ECE486_Labs
git clone git@gitlab.eece.maine.edu:donald.hummels/ece486_2017_group_XX.git
```

- You can now use git in the bash shell just as in any other Linux system

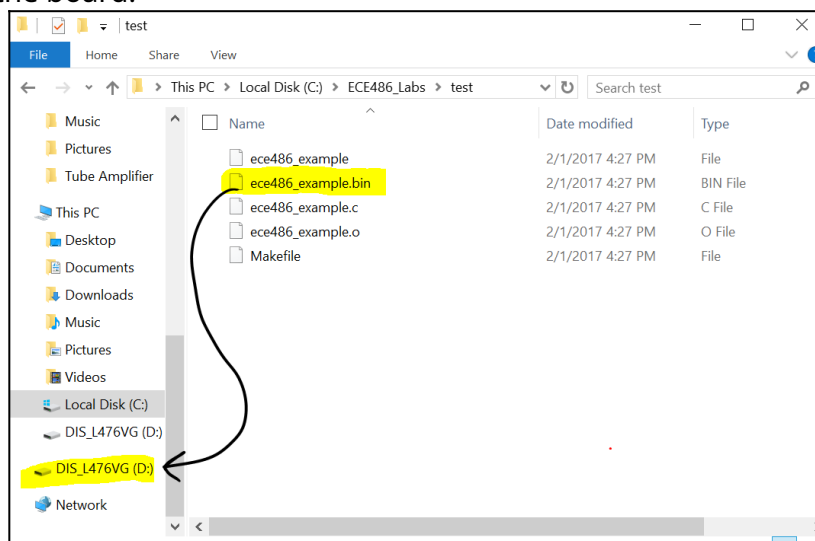
7. Try programming the board:

a. I created a new directory in the ECE486\_Labs folder called test. Save the test code from <http://web.eece.maine.edu/~hummels/classes/ece486/stm32l476.html> into the test directory, and the appropriate makefile below the testcode on the website.

b. Inside the bash shell, navigate to the test directory. You need to remove the .txt from Makefile.txt in order to build the code. The .txt is added to the file name when windows Downloads the file from the class website. Then run make to compile and build the code.

```
cd ~/ECE486_Labs/test
mv Makefile.txt Makefile
```

c. If all was successful, make created a binary file in the test directory. To flash the board, plug it in, and navigate to the test folder in Windows explorer. Drag and drop the binary file onto the board.



8. Install a decent text editor for Windows unless you want to program through the command line with nano. I recommend atom from git.

<https://atom.io>

## 9. Notes:

- The Linux subsystem is limited, not all features work. These installation instructions do not install st-link or any debugger. The Linux subsystem cannot run graphical applications
- The binary file must be dragged and dropped onto the board as the STM32L476 Board virtual drive does not mount to the Linux subsystem, and there does not appear to be an easy way to make this happen.
- You must have the Windows 10 anniversary update for this to work. Most will have this by default