

# SJ2 Development Environment

There are two major components of the development environment:

- Compile a program for the ARM processor (such as the SJ2 board)
- Compile a program for your host machine (your laptop)

Get started with the development environment by either:

1. Download and install Git and then clone the [SJ2-C repository](#)
2. Go to the [SJ2-C repository](#) and download the zip file

---

## Compile the SJ2 project

Most of the documentation related to the ARM compiler is captured in a few README files that you can [read here](#). We will not repeat the details here so please read the linked article. You can watch the following video to get started:

- [Youtube: Compile project](#)
- [Youtube: Scons build system](#)

---

## Hands-on

After setting up the SJ2-C development environment, try the following:

- Compile the FreeRTOS sample project (`lpc40xx_freertos`)
- Load it onto the processor
- Modify the program (maybe printf statement), and load/run it again
- Use a serial terminal program
  - Recommend: <https://libhal.github.io/web-serial/>
- Type "help" at the terminal window. Also try "help <command name>"
- Use all of the possible commands, and briefly skim through the code at `handlers_general.c` to get an idea of how the code works for each command.

---

## Troubleshooting

Computer cannot recognize the SJ2 development board.

- This error normally happens because of missing Silicon Lab driver.
- Solution: check the install folder inside the development packet (...sjtwo-c-master\installs\drivers). Please Install the driver, then start the computer and try to connect the device again.

"No such file or directory " after running Scons command.

- Please check, if the directory to the development folder has a name, which contain white space.
- Solution: Don't use directory with spaces.

Cannot recognize the command Scons.

- This error normally happens when Scons is not installed, there are corruption during the installation Scons packet. Sometimes, you need to upgrade the pip to latest version in order to install Scons
- Solution: Please check the pip version and upgrade the pip to latest version, then reinstall the Scons if necessary. After installation, restart the computer and try the Scons command again.

"Sh 1 : python : not found" after running Scons command

- It might appear when you have a multiple python versions, or you already had a python with different management packet (For example, python is installed in your machine through Anaconda, etc ). As the result, the python path might not setup correctly.
- Solution: Please check out these two article for your best option:
  - Window users: <https://datatofish.com/add-python-to-windows-path/>
  - Linux users: [https://www.tutorialspoint.com/python3/python\\_environment.htm](https://www.tutorialspoint.com/python3/python_environment.htm)
- To make it simple, You can also uninstall the python environment, and download the latest python version [here](#) then reinstall it again ( check in Add Python x.y to PATH at the beginning of the installing option )

Python3 is present, and "python" is not available (such as new Mac OS)

Add the following lines to you `~/.zshrc`

```
alias python=python3
alias pip=pip3export PATH=$PATH:"$(python3 -m site --user-base)/bin"
```

### VMs are not recommended

- While it is possible to pass the serial (COM) port to the VM, it can be really tricky.
- Unless you have prior experience with serial port passthrough, using VMs for this class is *not recommended*.

- If you are on Windows and want to use Linux, use **WSL1** instead of **WSL2 or VMs**.
- 

## Compile x86 project

x86 stands for instruction set for your laptop, which means that the project can be compiled and run on your machine without having to compile, load, and run it on your hex project. Being able to compile a project for your x86 host machine also provides the platform for being able to run unit-tests.

[Youtube: x86 FreeRTOS Simulator](#)

---

## SJ2 Board Startup

- The real boot location is actually at `entry_point.c`
  - Initial values of RAM are copied from Flash memory's `*data section`
    - See `startup__initialize_ram()` at `startup.c`
  - ARM core's floating point unit, and interrupts are initialized
  - Clock and a timer API is initialized
  - Peripherals and sensors are initialized
  - Finally, call to `main()` is made
- 

## Unit-Test Framework

TODO

---

## Extras!

The development environment contains built-in code formatting tool. Each time you compile, it will first reformat the source code according to preset Google coding format.

---

Revision #12

Created 5 years ago by [Preet Kang](#)

Updated 1 year ago by [isa\\_team](#)