Prerequisites & Board Setup

This assumes you are running Ubuntu 18.04 on your local machine, as how this project was performed.

1 - Perform the necessary `git` operations:

```
git config --global user.name "Firstname Lastname"
git config --global user.email "firstname.lastname@arm.com"
git config --global color.diff auto
```

2 - Workspace Initialization

Create a directory to clone the following repository into:

```
git clone https://git.linaro.org/landing-teams/working/arm/arm-reference-platforms.git
cd arm-reference-platforms
python3 sync_workspace.py
```

Select **Development Boards (1) Juno (1) 64 bit Stack (1)**

You will now be presented with **two options** of **Building from Source** or **Use Prebuilt Configurations**.

Below are the steps needed for both options

2a - Building from Source **(I used this option)**

Issues and Workarounds

After running the aforementioned `python3` script, you will most likely be prompted to install some packages through `sudo apt-get install`, install these packages.
These should be no problem. Re-run the script after installing.

However, a problem was surfaced at the next step, when prompted to install the `pip2` packages, particularly only when installing the Jupyter package.

Below is a workaround:

1. install ALL pip2 packages requested except Jupyter:
   `pip2 install IPython ......`

2. upgrade pip:
   `pip install --user --upgrade pip`

3. install the problematic jupyter package:
   `pip install --user --upgrade jupyter`

Next Selections & Starting to Build from Source

Select **Linux Kernel & Userspace Filesystem (1) Linaro/ArmLT LSK (2) Busybox (1)**

It is recommended that after the workspace stops building, you start building the kernels from source through the provided shell scripts.

Building kernels requires you to navigate to the `/build-scripts/` directory. There is documentation contained in the directory, **it is best to read it beforehand**. However, just execute the building of everything at the same time:

```
build-scripts/build-all.sh all
```

After this, visit the other page on connecting to the board & flashing the kernel.

NB:

Note that the `all` command only applies to the `build-all.sh` shell file. All the others need to do `clean`, `build`, and `package` in that order. You can choose to build individual targets, e.g. for Linux:

```
build-scripts/build-linux.sh build
```

2b - Prebuilt Configurations

Using prebuilt configurations mean that there is an option to choose one of the five settings to build.

Select **All (6)** and simply wait for the script to retrieve the pre-configured image files.

End of Chapter- See Juno Board Connection & Flashing Workflow