Setting Up Cross Compiling For the Onion Omega 2

What you will need:

- Linux Environment – recommended to use Ubuntu
- Time – Lots of time

> Note: If you are using the vdi file given out in the lab then step one and two are already done for you, and the ugpio library is already installed.

this guide will be following the onion setup guide on their website but will be making changes where appropriate, and if any errors occur, come to the WEEF TA office and they will fix appropriately.

1 Setting up LEDE build System

This section will go over installing the LEDE build system and getting the correct libraries. Things in *italics* will be items to run in your command line terminal.

1. Run The following Commands to install the necessary packages and git repositories:

   Note: This doc assumes this is done from the root directory.

   ```
   $ sudo apt-get install -y git wget subversion build-essential libncurses5-dev zlib1g-dev gawk flex quilt git-core unzip libssl-dev python-dev python-pip libxml-parser-perl
   $ git clone https://git.lede-project.org/source.git
   ```

   This will take a fair bit of time depending on your internet speed.

2. Now download the various libraries you might need to compile with for your omega programs

   i) Library for GPIO pin reading

a) Make a folder that is readable by the menuconfig:

```
$ cd source

$ mkdir package/libs/libugpio

$ cd package/libs/libugpio
```

b) Now get the ugpio package

```
$ wget https://dev.openwrt.org/export/35269/packages/libs/libugpio/Makefile
```

2 Installing the Base Build Chain

This section will go over the config of make and getting the libraries correctly installed.

```
$ cd source

$ make menuconfig
```

Note: if a GUI does not show up you need to make sure you are in the source directory, if that does not fix it go back to step one. The following section specifies what should be picked for each of the necessary settings

- Target Device: MediaTek Ralink MIPS
- Subtarget: MT76x8 based boards
- Target Profile Multiple Devices
- Target Devices Onion Omega2
  Onion Omega 2+

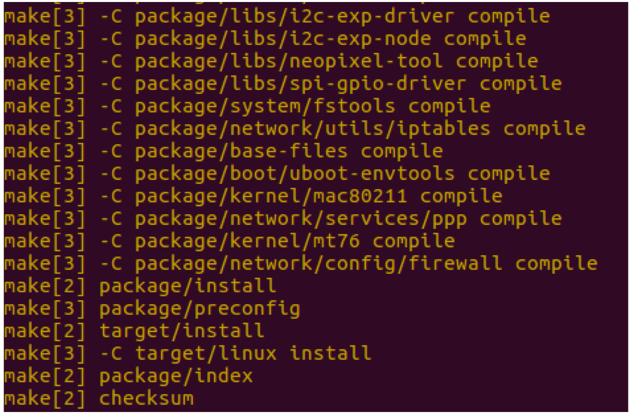
  To save your configuration go back to the main menu and use the left arrow key to get to save Do not change the name from .config, then exit the menuconfig.
- run

  ```
  $ make
  ```

  in the source folder this process will take a long time

Important: If a line similar to "make build process failed" comes up, come to the WEEF TA office, and they will help you finish this process.

- Upon completion your terminal should look something like this:



```
make[3] -C package/libs/i2c-exp-driver compile
make[3] -C package/libs/i2c-exp-node compile
make[3] -C package/libs/neopixel-tool compile
make[3] -C package/libs/spi-gpio-driver compile
make[3] -C package/system/fstools compile
make[3] -C package/network/utils/iptables compile
make[3] -C package/base-files compile
make[3] -C package/boot/uboot-envtools compile
make[3] -C package/kernel/mac80211 compile
make[3] -C package/network/services/ppp compile
make[3] -C package/kernel/mt76 compile
make[3] -C package/network/config/firewall compile
make[2] package/install
make[3] package/preconfig
make[2] target/install
make[3] -C target/linux install
make[2] package/index
make[2] checksum
```

3 Installing Extra Libraries

This section will go over how to compile the libraries on the Omega2 onto your build chain

Note: For this project do not install any node or python libraries, this is a c++ course not a python or javascript course for more information on external library support for Omega 2 click here: Compiling External Libraries

- Clone the following Github repository to get the needed files for installing the libraries while in the root directory

```
$ cd ~
$ git clone https://github.com/OnionIoT/OpenWRT-Packages.git
```

As of writing the following libraries will not compile and should **not** be used

- onion-st7735-driver
- libv8m-rb
- onion-ili9341-driver

1. Find the library you would like to install onto your buildchain, and move it to the following directory, then run the menuconfig again.

```
$ mv <path to package> ~/source/package/libs
$ cd ~/source
$ make menuconfig
```

2. Now in the menu there should be a new heading called ”Onion”, select it go to libs, and select the library you just added. Save and exit the menuconfig.

   ```
   $ make
   ```

3. That will compile your external libraries that you selected and you will now be ready to compile with them.

4 Compiling with xCompile.sh

Note: Before continuing make sure the xCompile.sh script, and the makefile from the example is inside of the folder that your .c file is.

- Change the following line in xCompile.sh: TOOLCHAIN_NAME=”toolchain-mipsel_24kc_gcc-5.4.0_musl <– extra characters here remove them” TARGET_NAME=”target-mipsel_24kc_musl”
- Edit the makefile Target to be the name of your program without the file extension

```
$ sh xCompile.sh -buildroot /root/source/ -lib <put Libraries Here>
```