

# **Raspberry Pi based Wireless Access Point**

Module & Makefile editing

# RTL8188



The wireless network adapter pictured at left uses a RealTek RTL8188 microprocessor. In order to operate this with the Raspberry Pi, the proper linux specific drivers had to be installed.

Next steps:

- While the wireless adapter did come with a driver on a CD, it was not sufficient to be used on the RPi. Therefore, a driver was downloaded from the manufacturer's website.
- The driver was unzipped and directories changed. The "tar ball" needed to be decompressed, per the driver's documentation.
- With the "tar ball" decompressed, the Makefile required editing, with the specifics of the RPi's processor having to be added.

```
ifeq ($(CONFIG_PLATFORM_ARM_1176JZFS),y)
EXTRA_CFLAG += -DCONFIG_LITTLE_ENDIAN
ARCH := arm
CROSS_COMPILE := arm-none-linux-gnueabi-
KSRC := /lib/modules/3.6.11+/modules.builtin
MODDESTDIR := /lib/modules/3.6.11+/kernel/drivers/net/wireless/
MODULE_NAME := wlan
endif
```

- A new directory was created at /lib/modules/3.6.11+/kernel/drivers/net/wireless/rtl8192cu to house the driver files, including the Makefile.

- After attempting to make the driver, it appeared that a new “clean” kernel source tree might be required. The file `linux-rpi-3.6.y.tar.gz` was obtained from <http://raspberrypi.stackexchange.com>
- The file was decompressed and the “clean” kernel source tree was linked into the existing kernel.
- The make of the driver was reattempted. With the “clean” kernel source tree, fewer errors were encountered; however, the “make” was still unsuccessful due to the makes inability to parse whether the chip uses Big Endian or Little Endian (despite the variable being set in the Makefile).
- The solution to the previous error required altering one of the driver files: `/lib/modules/3.6.11+/kernel/drivers/net/wireless/rtl8192cu/include/autoconf.h`

where a define was added to the beginning of the file stating

```
#define CONFIG_LITTLE_ENDIAN 1
```

- A make was reattempted, where the little endian error was resolved.
- The next error to emerge during make also involves the available driver files. The compilation is failing due to the following error: “selected processor does not support ARM mode ‘cpsid i”

Of course, it should support that Assembly command, based upon the version of ARM (v6) that the Pi uses. It seems, now, to be a matter of convincing the compiler of which version of ARM it may use (since, from research, it does appear that this particular command is only relevant for version 6 and higher)