Strong Arm Of The Law

A WEBRADIO FOR THE MINI2440

Search

The Wi-Fi WebRadio Project

Mar
10
2011

Getting the Wi-Fi to work.

tutorials, webradio Add comments

After setting up a working distro on the board, the next step is making it connect to a network via a wi-fi usb dongle. In the end i choose a TL-WN321G from TP-LINK which is cheap (12 euros here in Italy) and widely available. I tried also the TL-WN721N but i couldn't make it work as its drivers are new and not included in the version of the kernel I'm using (2.6.32 from the buserror git). Not everything went smooth with the TL-WN321G either as the ones you can buy now (named TL-WN321Gv4) are shipped with the newer RT3070 chipset for which you have to build your own module from the Ralink drivers source. So i decided to put together all the information i got and make a simple tutorial hoping that it will be useful to someone.

We'll assume you have a working cross-compiler and have compiled your kernel and have its source in your system.

Download the latest driver for the RT3070 chipset from the Ralink site.

Unpack it and start editing the Makefile in the main directory. We are going to modify it to tell him where to search the kernel headers.

Around the line 25 you'll find this:

```
#PLATFORM: Target platform
PLATFORM = PC
```

We will change it to :

```
#PLATFORM: Target platform
PLATFORM = ARM
```

Now we'll go to line 170 and we'll find something like:

```
ifeq ($(PLATFORM),PC)
```

Here we'll tell the makefile where we he can find the linux headers, so we'll change it to :

```
ifeq ($(PLATFORM),ARM)
LINUX_SRC= /path/to/kernel
LINUX_SRC_MODULE= /path/to/kernel/drivers/net/wireless/
endif
```

Obviously change the right kernel path.

Now we have to add the right USB ID to common/rtusb_dev_id.c (yes, the driver is so shitty). Around line 36 you'll find this:

```
/* module table */
USB_DEVICE_ID rtusb_dev_id[] = {
#ifdef RT3070
{USB_DEVICE(0x148F,0x3070)}, /* Ralink 3070 */
```

Now add to that structure this line:

Blog's Manifesto

This blog is born as a build log of a Wi-FI webradio based upon the mini2440 arm9 board from FriendlyArm.
I'm not a linux guru or the developer guy, I'm much more into hardware than software. So i started this project to learn more about embeded linux systems and how to develop application for them.

Archives

March 2011 (2)

Blogroll

- Bill's Blog
- buserror's blog
- diycinema blog

Meta

- Log in
- Entries RSS
- Comments RSS

```
MODULE_LICENSE("GPL");
```

Change this:

```
#include "rt_config.h"
// Following information will be show when you run 'modinfo'
// *** If you have a solution for the bug in current version of driver,
please mail to me.
// Otherwise post to forum in ralinktech's web site(www.ralinktech.com) and
let all users help you. ***
MODULE_AUTHOR("Paul Lin ");
MODULE_DESCRIPTION("RT2870 Wireless Lan Linux Driver");
#ifdef CONFIG_STA_SUPPORT
#ifdef MODULE_VERSION
MODULE_VERSION(STA_DRIVER_VERSION);
#endif
#endif // CONFIG_STA_SUPPORT //
```

To this:

```
#include "rt_config.h"
// Following information will be show when you run 'modinfo'
// *** If you have a solution for the bug in current version of driver,
please mail to me.
// Otherwise post to forum in ralinktech's web site(www.ralinktech.com) and
let all users help you. ***
MODULE_AUTHOR("Paul Lin ");
MODULE_DESCRIPTION("RT2870 Wireless Lan Linux Driver");
MODULE_LICENSE("GPL");
#ifdef CONFIG_STA_SUPPORT
#ifdef MODULE_VERSION
MODULE_VERSION(STA_DRIVER_VERSION);
#endif
#endif // CONFIG_STA_SUPPORT //
```

Now one last step before the compiling we have to change the os/linux/config.mk:

- Change HAS_WPA_SUPPLICANT=n to HAS_WPA_SUPPLICANT=y
- Change HAS_NATIVE_WPA_SUPPLICANT_SUPPORT=n to HAS_NATIVE_WPA_SUPPLICANT_SUPPORT=y

And around line 322 you'll find this:

```
#kernel build options for 2.4
```

We'll add this to the if table of the script:

```
ifeq ($(PLATFORM),ARM)
#EXTRA_CFLAGS := -v $(WFLAGS) -I$(RT28xx_DIR)/include
EXTRA_CFLAGS := -D__KERNEL__ -v $(WFLAGS) -I$(RT28xx_DIR)/include
endif
```

Oh and i almost forget this, the makefile script will search for this file RT3070STA.dat but we have only RT2870STA.dat in the main directory of the sources, so we'll just do this:

```
cp RT2870STA.dat RT3070STA.dat
```

Okay now we are ready to cross-compile the driver just issue your usual make command with the right environment variable:

```
make
```

And we'll get our rt3070sta.ko under the os/linux directory and the right firmware file rt2870.bin in the common directory.

Now we just need to copy the these files to our rootfs, as I'm using a rootfs on a sd card we'll just mount the right file system and then copy the files:

```
mount /dev/ /mnt
sudo mkdir -p /etc/Wireless/RT2870STA
sudo cp /path/to/drivers/RT2870STA.dat /etc/Wireless/RT2870STA/
sudo dos2unix /etc/Wireless/RT3070STA/RT3070STA.dat
sudo chmod +x /etc/Wireless/RT3070STA/RT3070STA.dat
sudo cp /path/to/drivers/rt2870.bin /lib/firmware/
sudo   cp   /path/to/drivers/rt3070sta.ko   /lib/modules/2.6.32-rc8/kernel/
drivers/staging/rt3070/
```

Beware the module must be in the stating directory otherwise it won't work.

Now just run depmod and load the new module:

```
depmod -a
modprobe rt3070sta.ko
```

Say hello to your new ra0 device !

This isn't all my work but just an assembly of the puzzle you can found in the Internet, the sources for this work are:

- http://electoday.com/bbs/archiver/?tid-10317.html didn't understood a word but it helped me to get a hold of what i had to do to make things work.
- http://ubuntuforums.org/showthread.php?t=1285828 useful tutorial about compiling the Ralinks drivers.
- http://www.linuxforums.org/forum/wireless-internet/161550-rt3070sta-module-license-unspecified-taints-kernel-solved.html this one saved my day helping me solving the "taints the kernel" bug.

Hope this will help someone 😉 .

Posted by XAmarokX at 3:37 pm

Tagged with: drivers, linux, mini2440, ralink, wifi

Like

DISQUS

Add New Comment

Login

Showing 5 comments

Sort by popular now ▾

Bruno_F

Hi there.
This can be helpful to me, and for that, i would be extremely grateful, but I have a question before following your steps and enriching your so-far great and simple tutorial with the experience I'll gain from it.
The mini2440 don't change much from the mini6410, as long as the kernel goes, if I'm right. Do you believe, or know if these steps will work on mini6410?
I'm in an engineering development group (almost-last-year engineering student), in which we are creating and embedded system which needs to transfer data via wireless, and since we already spent our not-so-great budget in the TL-WN321G (TE7WN321Gv4), we need to get this working under it. We are using Ubuntu 10.04 as system (not sure about the version, but I'm sure that it's Ubuntu).
Any help or thoughts are appreciated.

1 year ago 1 Like

Like Reply

mtanh282

Hi.

Bruno_F

Hello there again.
On kernel 2.6.36 the driver for the WN321G is already available. While tweaking the kernel to have the staging folder on kernel/drivers, I found this option:
Device Drivers --->
[*] Staging Drivers --->
[*] Ralink 2870/3070 wireless support
After selecting those two, the WN321G worked.

11 months ago Like Reply

Bruno_F

Hello again. After resolving other problems related to the Tiny6410 (said wrong that time), started to create the drivers for the linux 2.6.36 kernel. I have a cross compiler ambient set (arm-linux-gcc 4.3.2 working well, with compiled codes working right on board), and tried to follow the steps (changing and checking directories).
Ends out that a thing were a bit different:
In the Makefile (where you change the linux kernel target, ifeq ($(PLATFORM),ARM)) there was another line:
CROSS_COMPILER =
I added arm-none-linux-gnueabi- to it (CROSS_COMPILER = arm-none-linux-gnueabi-)
After all changes, I used the make (only typed make on the terminal, with no other arguments (I didn't understand the "with the right environment variable").Turns out the compilation went okay (and as seemed, it used the arm-none-linux-gnueabi-gcc to cross compile, as the various lines reports about the compiling told).
Although, the file rt3070sta.ko didn't show up, only the rt5370sta.ko was there.Is there something I'm doing wrong? I'll try to fiddle a little more while I wait a response. If something works, I'll update the info here, thanks.

11 months ago Like Reply

XAmarokX

I don't know if this will work for the mini6410 kernel, but as this is just a modification to make the drivers cross-compile for an arm target i believe it might work.
I've done this under a 10.10 ubuntu with the cross-compile tools generated by openembedded.
Goold luck for your project and let me know if this works :-).

1 year ago Like Reply

Subscribe by email RSS

Trackback URL http://embedded.abstract-b

blog comments powered by DISQUS

© 2012 Strong Arm Of The Law Suffusion theme by Sayontan Sinha