The Audiophiles How-To Guides

How-To Guide for Attempted Bluetooth

Section 1

I. Installing the proper libraries

Run the following commands from the terminal on the host:

1) “> sudo apt-get install bluez”

2) “> sudo apt-get install libbluetooth-dev”

Section 2

I. Configuring the kernel

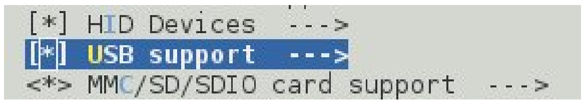
Navigate to the linux kernel directory on the host.

“> cd $HOME/cmpt433/private/linux-2.6.30.4”

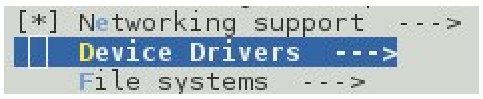
Run the command “make menuconfig”

“> make menuconfig”

Navigate to “Device Drivers” in the menu and then “USB support”:



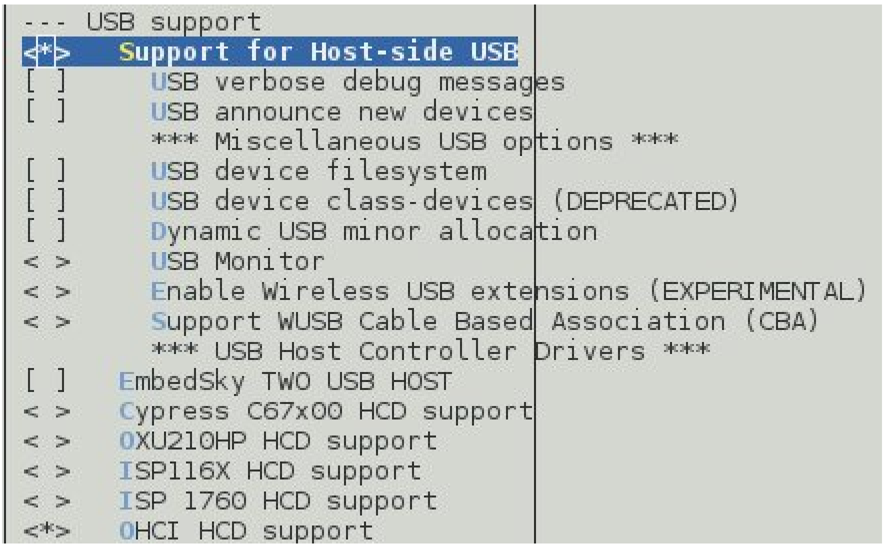
```
[*] HID Devices  --->
[*] USB support  --->
<*> MMC/SD/SDIO card support  --->
```



```
[*] Networking support  --->
    Device Drivers  --->
    File systems  --->
```

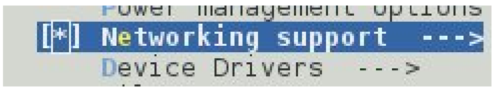
Press “Spacebar” to activate menu options:

In “USB support”



```
--- USB support
<*>   Support for Host-side USB
[ ]     USB verbose debug messages
[ ]     USB announce new devices
        *** Miscellaneous USB options ***
[ ]     USB device filesystem
[ ]     USB device class-devices (DEPRECATED)
[ ]     Dynamic USB minor allocation
< >     USB Monitor
< >     Enable Wireless USB extensions (EXPERIMENTAL)
< >     Support WUSB Cable Based Association (CBA)
        *** USB Host Controller Drivers ***
[ ]   EmbedSky TWO USB HOST
< >   Cypress C67x00 HCD support
< >   OXU210HP HCD support
< >   ISP116X HCD support
< >   ISP 1760 HCD support
<*>   OHCI HCD support
```

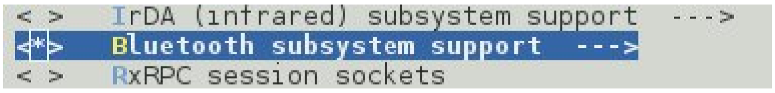
Navigate to “Networking support”in the main menu:



```
[*] Networking support  --->
    Device Drivers  --->
```

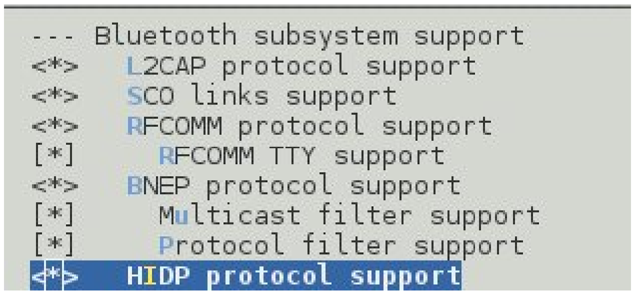
Activate using “Space bar”:

<*> Bluetooth subsystem support



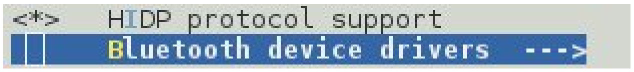
```
< >   IrDA (infrared) subsystem support  --->
<*>   Bluetooth subsystem support  --->
< >   RxRPC session sockets
```

Activate under “Bluetooth subsystem support” using “space bar”:



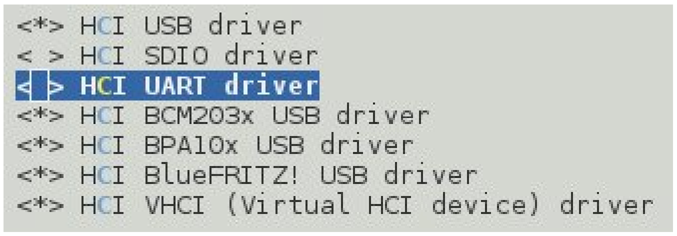
```
--- Bluetooth subsystem support
<*>   L2CAP protocol support
<*>   SCO links support
<*>   RFCOMM protocol support
[*]     RFCOMM TTY support
<*>   BNEP protocol support
[*]     Multicast filter support
[*]     Protocol filter support
<*>   HIDP protocol support
```

Navigate to “Bluetooth device drivers”:



```
<*>   HIDP protocol support
          Bluetooth device drivers  --->
```

Activate the following options using “space bar”:



```
<*> HCI USB driver
< > HCI SDIO driver
< > HCI UART driver
<*> HCI BCM203x USB driver
<*> HCI BPA10x USB driver
<*> HCI BlueFRITZ! USB driver
<*> HCI VHCI (Virtual HCI device) driver
```

Exit out of the makemenu configuration and make sure to save your changes.

Rebuild your kernel and transfer it to your device via U-Boot using the commands provided in the QuickStart Guide.

II. Installing the proper libraries

Before we attempt to install bluez we need to install some dependency libraries:

1) expat -- required by dbus

a) First download the package from:
   http://sourceforge.net/projects/expat/
b) Second untar it by running the command:
   "> tar xvf expat-<version number>.tar.gz"
c) cd into the directory
   "> cd expat-<version number>"
d) run the configure script by excuting the following command:
   ">./configure --prefix "path-to-directory for installation" --host arm-linux
e) now execute "make" to ensure everything builds properly
   "> make"
d) Once we have a proper "make" without any errors we can run "make install" to install to the directory we specified in the "configure" section.
   "> make install"

2) dbus -- required by bluez

a) Download d-bus from the following link:
   http://dbus.freedesktop.org/releases/dbus/dbus-1.5.0.tar.gz
b) Untar it now by executing the follow command:
   "> tar xvf dbus-1.5.0.tar.gz"
c) Configure it by running the following command:
   "> ./configure --prefix=<path-to-dbus-install> --host=arm-linux --with-x=no ac_cv_have_abstract_sockets=yes "CC=arm-linux-gcc -I<path-to-expat-include> -L<path-to-expat-libraries>"
d) make it now:
   "> make"
e) Once we've confirmed a proper make without any errors run:
   "> make install"

3) zlib -- required by glib

a) Download z-lib from the following link:

http://zlib.net/zlib-1.2.5.tar.gz

b) Untar it using the following command:

“>tar xvf zlib-1.2.5.tar.gz”

c) Configure the package now with the following command:

“>./configure”

d) make it now:

“> make”

e) Once we’ve confirmed a proper make without errors run:

“> make install”

6) libiconv -- required by glib

a) Download libiconv from here:

http://ftp.gnu.org/pub/gnu/libiconv/libiconv-1.14.tar.gz

b) Untar it by executing the following command:

“> tar xvf libiconv-1.14.tar.gz“

c) Configure the package now with the following command

“> ./configure --prefix=<path-to-install-to> --host=arm-linux“

d) make it now:

“> make”

e) Once we’ve confirmed a proper make without errors run:

“> make install”

4) glib -- required by bluez

a) Download the glib from here:

http://ftp.gnome.org/pub/gnome/sources/glib/2.30/glib-2.30.1.tar.bz2

b) Untar the file using the following command:

“> tar xvf glib-2.30.1.tar.bz2”

c) Configure the package by running the following command:

“> ./configure --prefix=<path-to-install-to> --host=arm-linux “CC=arm-linux-gcc -I<path-to-libiconv-includes> -L<path-to-libiconv-lib>”

d) Run make once configure is complete:

“> make”

e) Once make completes properly run:

“> make install”

5) Bluez -- required to use bluetooth libraries

a) Download bluez from the following link:

http://mirror.anl.gov/pub/linux/bluetooth/bluez-4.96.tar.gz

b) Untar the downloaded package using the following command:

“> tar xvf bluez-4.96.tar.gz”

c) Configure the package now with the following command:

“> ./configure --prefix=/home/vla22/SFU/cmpt433/private/bluez --host=arm-linux “CC=arm-linux-gcc -I<path-to-your-include> -L<path-to-your-libraries>

DBUS_LIBS=”-L<path-to-dbus-libs>” DBUG_CFLAGS=”-I<path-to-dbus-includes>” GLIB_LIBS=”-L<path-to-glib-libs>” GLIB_CFLAGS=”-I<path-to-glib-includes>”

d) Run make now:

“> make”

e) Once we’ve got a proper make without errors:

“> make install”

III. Compiling arm bluetooth programs

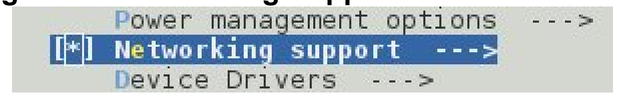
a) Compile your code by using the following command:

”>arm-linux-gcc -o <executable-name> <cpp-files-to-compile> -lbluetooth”

<u>How-To Guide for Wireless</u>

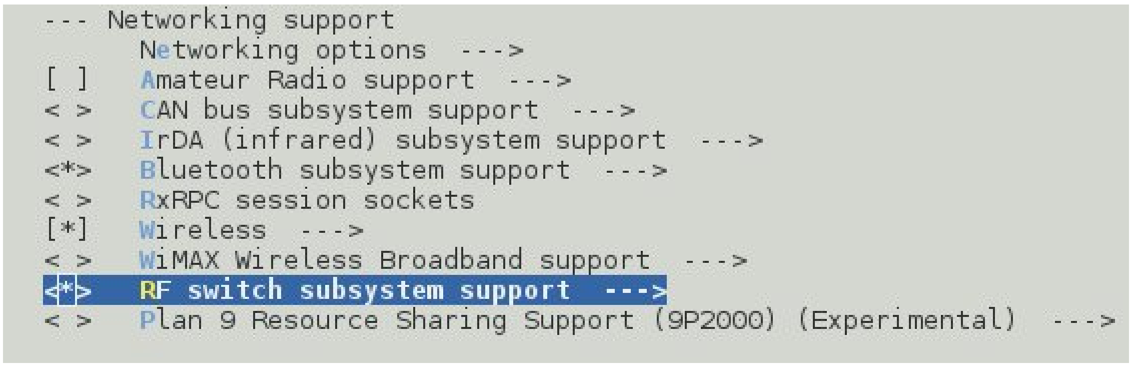
Section 1 - Enabling the necessary options in the linux kernel

Navigate to “Networking Support” in the main menu:



```
         Power management options  --->
     [*] Networking support  --->
         Device Drivers  --->
```

Enable the following options using “spacebar”:



```
 --- Networking support
         Networking options  --->
 [ ]     Amateur Radio support  --->
 < >     CAN bus subsystem support  --->
 < >     IrDA (infrared) subsystem support  --->
 <*>     Bluetooth subsystem support  --->
 < >     RxRPC session sockets
 [*]     Wireless  --->
 < >     WiMAX Wireless Broadband support  --->
 <*>     RF switch subsystem support  --->
 < >     Plan 9 Resource Sharing Support (9P2000) (Experimental)  --->
```

Exit out of the makemenu configuration and make sure to save your changes.

Rebuild your kernel and transfer it to your device via U-Boot using the commands provided in the QuickStart Guide.

Section 2 - Inserting the necessary drivers or *.ko files

Run the following commands on the host:

“> insmod bcm203x.ko

“> insmod option.ko

“> insmod scsi_wait_scan.ko

“> insmod input-polldev.ko

“> insmod mac80211.ko
“> insmod rt2x00lib.ko
“> insmod rt2x00usb.ko
“> insmod rt73usb.ko

Section 3 - Issuing the proper commands upon startup to connect

“> ifconfig wlan0 up”
“> iwlist wlan0 scan”
“> iwconfig wlan0 essid <essid you gained from the scan>”
“> udhcpc -i wlan0”

Section 4 - Resolve missing driver for Realtek

Download the driver:
http://download.wireless-driver.com/driver/Realtek/RTL8188CUS/RTL8188CUS_v2.0.1212.zip

Extract its contents into ~/cmpt433/private/ and rename the directory to rtl8192cu/, so it’s ~/cmpt433/private/rtl8192cu/

cd to ~/cmpt433/private/rtl8192cu/
Run the command “sudo sh install.sh” in this directory

There will now be a directory ~/cmpt433/private/rtl8192CU_linux_v2.0.1212.20101208
This is the driver you want to include in your kernel

Include RTL8192CU into the build process by executing the following commands:

“> cp rtl8192CU_linux_v2.0.1212.20101208 ~/cmpt433/private/linux-2.6.30.4/drivers/net/wireless/RTL8192CU”

Then Navigate to the folder by running the follow command:

“> ~/cmpt433/private/linux-2.6.30.4/drivers/net/wireless/

Edit Kconfig:

“> vim Kconfig”

Add:

```
config RTL8192CU
        tristate "Realtek 8192C USB WiFi"
        depends on USB
        ---help---
          Help message of RTL8192CU
```

Config ARLAN

Change directories to
“> cd ~/cmpt433/private/linux-2.6.30.4/
and run
“>make menuconfig”

Navigate to Device Drivers -> Network device support -> Wireless LAN
and set
<*> Realtek 8192CU USB Wifi (new)

After enabling your driver, exit out of the program and save your changes.
Rebuild your kernel and transfer it to your device via U-Boot.

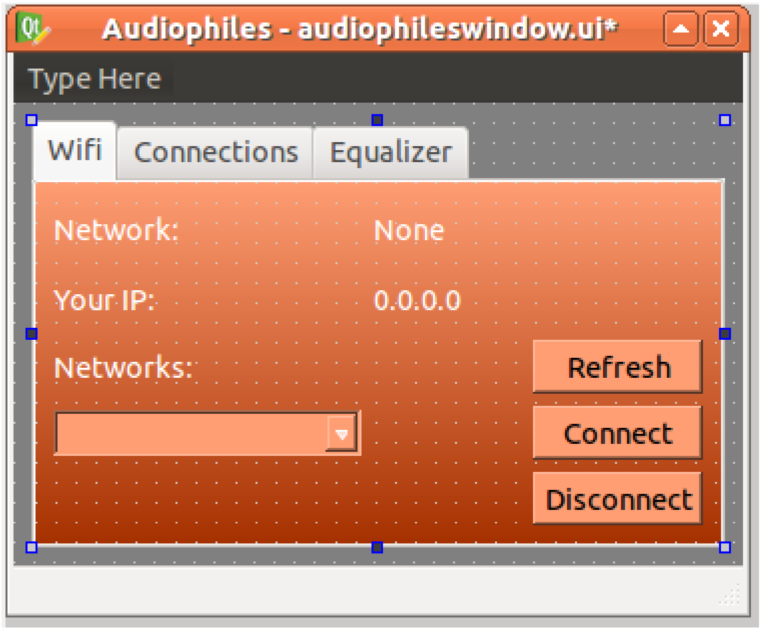
References:
http://www.thinkwiki.org/wiki/How_to_setup_Bluetooth
http://www.daimi.au.dk/~rolft/liwas/docs/CrossCompilingBluez.html
http://www.cs.sfu.ca/CourseCentral/433/bfraser/other/DriverCreationGuide.pdf

---

How-To Guide for Qt CSS

With your Qt Designer open and your project opened:
For a global CSS stylesheet which effects all parts of your project:
Right click on your *MainWindow* in the Object Inspector
Select Change styleSheet...
This displays a prompt in which you can add CSS the same as with HTML.
Similar steps can be taken by right clicking on appropriate widgets to just edit their CSS.

The following is a short example of CSS used in Qt:



```
#centralwidget{
        background: gray;
}
#tabConnections{
        background: QLinearGradient(x1: 0, y1: 0, x2: 0, y2: 1, stop: 0 #FF9D73,
        stop: 1 #A63100);
}
#tabMixer{
        background: QLinearGradient(x1: 0, y1: 0, x2: 0, y2: 1, stop: 0 #FF9D73,
        stop: 1 #A63100);
}
#tabEqualizer{
        background: QLinearGradient(x1: 0, y1: 0, x2: 0, y2: 1, stop: 0 #FF9D73,
        stop: 1 #A63100);
}
#tabWifi{
        background: QLinearGradient(x1: 0, y1: 0, x2: 0, y2: 1, stop: 0 #FF9D73,
        stop: 1 #A63100);
}
QPushButton{
        color:black;
        background-color: #FF9D73;
}
QComboBox{
        background: #FF9D73;
        color: white;
}
QLabel{
        color: white;
}
QCheckBox{
	color: black;
	background-color: #FF9D73;
}
```