Application Note

FX30: Using Bluetooth with the Mirage IOT Card

Applies to:
- FX30
- FX30S

Introduction

The Mirage IoT Card from Talon Communications enables Wi-Fi and Bluetooth for the FX30/FX30S Programmable Gateways. The Mirage card supports Bluetooth 2.1 and BLE 4.0 with an IOT card form factor. This document explains how to set up the Bluetooth interface. The FX30 will be able to communicate with Bluetooth devices after completing the procedures described in this application note.

Note: In this document, FX30 refers to both FX30 and FX30S.

Overview

This is an overview of the process to enable Bluetooth on the FX30. You must:

1. Enable the Bluetooth compile options in the kernel
2. Add a kernel modification, then rebuild the kernel
3. Update the FX30 software
4. Copy the hci binaries from the mangOH BluetoothUtils to the FX30
5. Run the new bt_init.sh script

Note: The following procedures assume that you know how to build the FX30 kernel.

Note: For reference, see also: https://github.com/mangOH/mangOH/wiki/Bluetooth-WL18xx-driver-for-mangOH. However, do not use the WP85 stock kernel as described on the Github page.

Enable Bluetooth Options in the Kernel

Before beginning, download the FX30 3G or FX30S 3G source code package (R13.1 or later) from the Source at: https://source.sierrawireless.com/resources/airlink/software_downloads/fx30-firmware/fx30-firmware/

The source code is compressed as a .tar (tarball) file.

Note: The source code does not include Legato.

Two methods of enabling Bluetooth options are available:

Method 1: Manually enabling kernel options

This method uses bitbake to run menuconfig, allowing a user to manually select the Bluetooth kernel options. This method is similar to running a “make menuconfig” in Linux.

Note: The settings enabled using this method will not survive a “make clean”.

After extracting the source code from the tarball, enable the Bluetooth options:

1. Type `cd <workspace>` to enter your workspace.
2. Type `make dev`

   This configures your build environment for bitbake. This step automatically changes directories to `<workspace>/build_bin`
3. Type `bitbake -c menuconfig linux-yocto`
4. Enable the compile options:
   a. Enter Networking support.
   b. Under Networking support, enter:

      <M> Bluetooth subsystem support

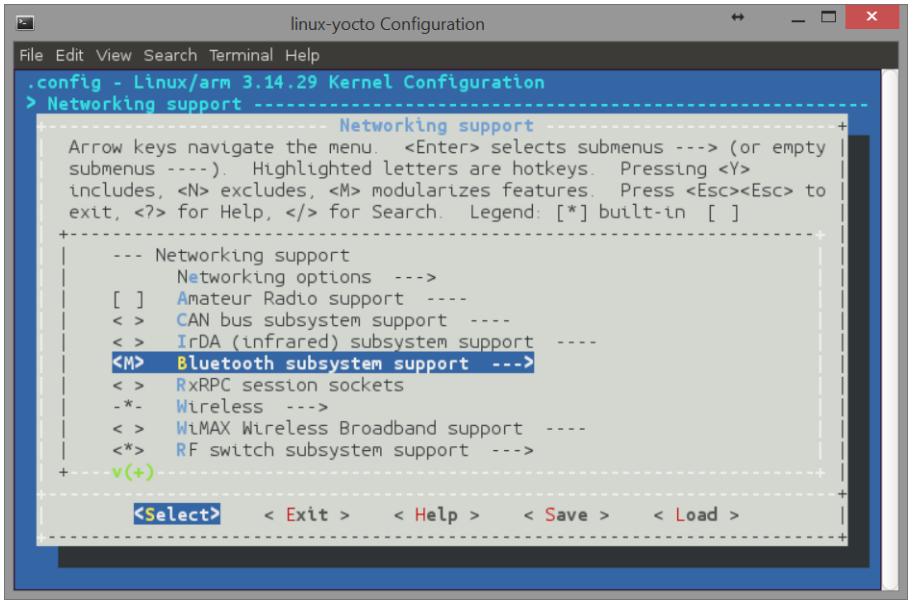

Figure 1: Networking support

   c. Under Bluetooth subsystem support, enter:

      [*] RFCOMM protocol support

      [*] RFCOMM TTY support

      <M> BNEP protocol support

[*] Multicast filter support

[*] Protocol filter support

<M> HIDP protocol support

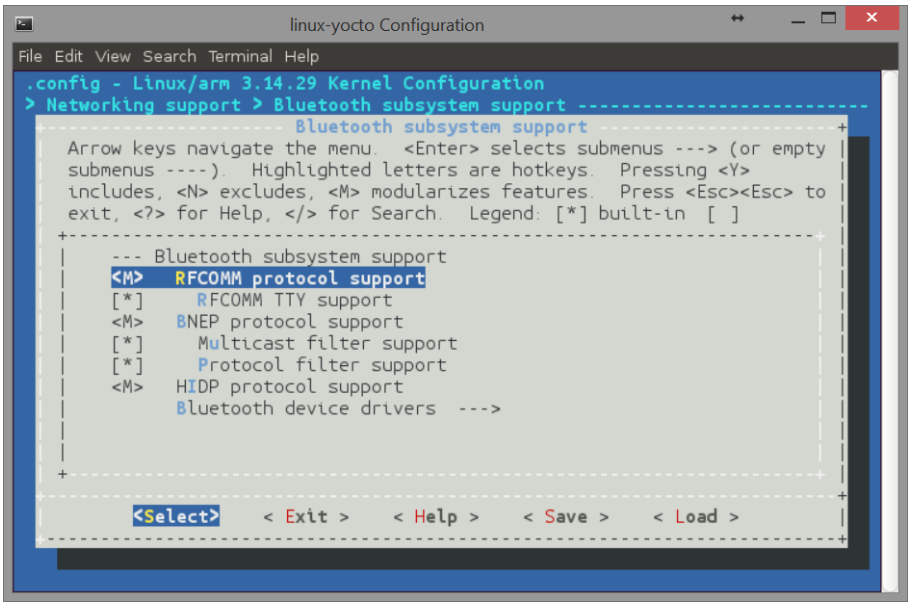

Figure 2: Bluetooth subsystem support

d. Enter Bluetooth device drivers.

e. Under Bluetooth device drivers, enter:

<M> HCI UART driver

[*] UART (H4) protocol support

[*] BCSP protocol support

[*] Atheros AR300x serial support

[*] HCILL protocol support

[*] Three-wire UART (H5) protocol support

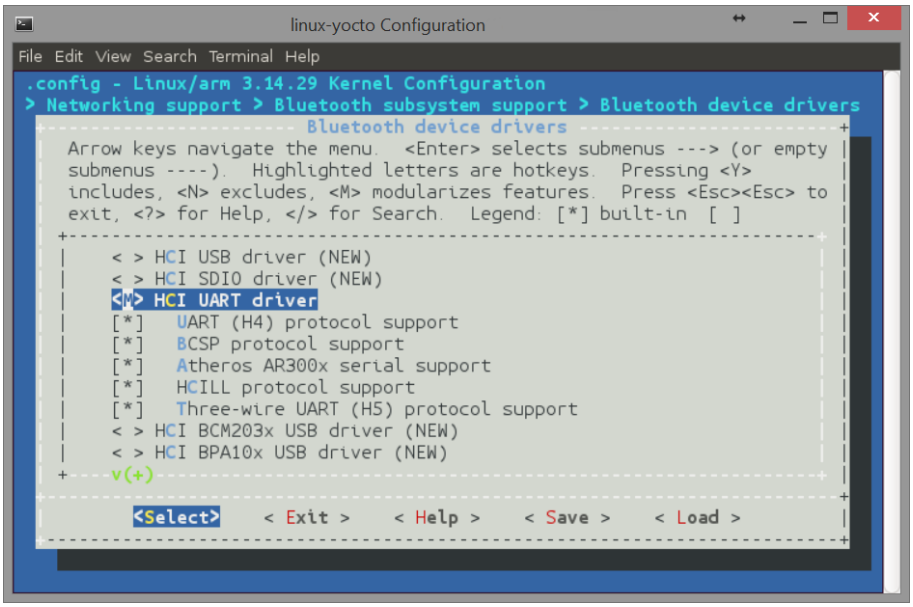

Figure 3: Bluetooth device drivers

5. Save and exit the configuration.

Method 2: Patching the kernel

This method patches the kernel via a Yocto recipe and patch files. If a “make clean” is executed, the next time the kernel is built, it will be patched and compiled.

1. Copy the file enable-bt.cfg to your workspace at the location:

```
<workspace>/meta-columbia-x/meta-columbia-x-bsp/recipes-kernel/
linux/files
```

2. Edit the linux-yocto bbappend file:

```
vi <workspace>/meta-columbia-x/meta-columbia-x-bsp/recipes-
kernel/linux/linux-yocto_3.14.bbappend
```

   a. Add the file to the SRC_URI list by appending the following to the list (anywhere within the double quotes):

```
file://enable-bt.cfg \
```

The contents of the enable-bt.cfg file are shown below:

```
CONFIG_6LOWPAN_IPHC=m
CONFIG_BT=m
CONFIG_BT_RFCOMM=m
CONFIG_BT_RFCOMM_TTY=y
CONFIG_BT_BNEP=m
CONFIG_BT_BNEP_MC_FILTER=y
CONFIG_BT_BNEP_PROTO_FILTER=y
CONFIG_BT_HIDP=m
CONFIG_BT_HCIUART=m
CONFIG_BT_HCIUART_H4=y
CONFIG_BT_HCIUART_BCSP=y
CONFIG_BT_HCIUART_ATH3K=y
CONFIG_BT_HCIUART_LL=y
CONFIG_BT_HCIUART_3WIRE=y
```

Note: Do not remove any other patch files or .cfg files.

The bbappend file will look something like this:

```
SRC_URI += " \
            file://0001.patch \
            file://0002.patch \
            file://0003.patch \
            ...
            file://enable-bt.cfg \
            "
```

Add a Kernel Modification

There is an interoperability issue between the TI chipset on the Mirage IOT card and the FX30 (WP85) UART2 interface. The TI chipset requires a 4-wire hardware flow control UART, while the IOT card slot only supports 2-wire (in software) but 4-wire in hardware. The workaround is a kernel modification to always enable flow control.

To enable flow control:

1. Enter the following in your workspace:

```
cd meta-columbia-x/meta-columbia-x-bsp/recipes-kernel/linux/files
vi 0010-ALPC-167-UART2_CTS-UART2_RTS-Default-Settings.patch
```

2. Find the section shown below and change GPIOMUX_OUT_HIGH to GPIOMUX_OUT_LOW.

```
+static struct gpiomux_setting uart2_cts_iot_default = {
+       .func = GPIOMUX_FUNC_GPIO,
+       .drv = GPIOMUX_DRV_8MA,
+       .pull = GPIOMUX_PULL_NONE,
+       .dir = GPIOMUX_OUT_LOW,
+};
```

3. Save the patch file and exit from vi:
   a. Type `:wq`
   b. Press Enter.
4. Type `exit` to leave the build environment.

Warning: *Please be aware that data into the FX30 is always flowed on, so the incoming UART buffer could overflow.*

Building the FX30 image (including kernel)

To build the FX30 image, use the `make` command.

```
make
```

Update FX30 Software

Once the build is complete, update the FX30 software as usual.

Update the FX30 image on the device from the location:

```
build_bin/tmp/deploy/images/swi-mdm9x15/boot-yocto_wp85.cwe
```

Copy Bluetooth Binaries

The Bluetooth binaries are located on github in the mangOH package: https://github.com/mangOH/Demos/tree/master/BluetoothUtil

Copy the files **gatttool**, **hciattach**, **hciconfig**, and **hcitool** from the bin folder to the FX30 `/home/root`.

Add `/home/root` to your path: `export PATH=$PATH:/home/root`

Copy the TIInit_11.8.32.bts file to this location on the FX30:

```
/legato/systems/current/apps/bluetoothUtil/read-only/bin
```

*Note: You may need to create the **bluetoothUtil/read-only/bin** directories.*

Run the bt_init.sh

To run the bt_init.sh script:

1. Copy the contents of the bt-init.sh script to the FX30 in your home directory. The contents of the script appear on page 9.

*Note: The script found in the **mangOH bluetooth util** repository will not work on the FX30.*

2. Change the permissions to the script to make it executable:
   `chmod a+x bt-init.sh`
3. Run this script using ./bt-init.sh

After running this script you should see this text:

```
PWD is: /home/root
Found a Texas Instruments' chip!
Firmware file : /legato/systems/current/apps/bluetoothUtil/read-only/bin/
TIInit_11.8.32.bts
Loaded BTS script version 1
texas: changing baud rate to 3000000, flow control to 1
Device setup complete
hci0:   Type: BR/EDR  Bus: UART
        BD Address: AA:BB:CC:DD:EE:FF   ACL MTU: 1021:6  SCO MTU: 180:4
        UP RUNNING
        RX bytes:587 acl:0 sco:0 events:28 errors:0
        TX bytes:393 acl:0 sco:0 commands:28 errors:0
root@fx30:~#
```

At this point, the Mirage IOT card is functional, and you can use HCI tools.

Using hcitools

Here are some guidelines for using hcitools.

- To scan for LE Bluetooth devices, run `./hcitool lescan`
  This will output a list of Bluetooth addresses:
  ```
  LE Scan ...
  AA:BB:CC:DD:EE:FF CC2650 SensorTag
  ```
- Run gatttool: `./gatttool -b AA:BB:CC:DD:EE:FF -I`
- When gatttool is running, you can connect to the device:
  ```
  [AA:BB:CC:DD:EE:FF][LE]> connect
  ```
  You should see something like:
  ```
  Attempting to connect to AA:BB:CC:DD:EE:FF
  Connection successful
  ```

- After connecting to the device, you can view useful information about the device:

```
[AA:BB:CC:DD:EE:FF][LE]> primary
attr handle: 0x0001, end grp handle: 0x0007 uuid: 00001800-0000-1000-8000-00805f9b34fb
attr handle: 0x0008, end grp handle: 0x0008 uuid: 00001801-0000-1000-8000-00805f9b34fb
attr handle: 0x0009, end grp handle: 0x001b uuid: 0000180a-0000-1000-8000-00805f9b34fb
attr handle: 0x001c, end grp handle: 0x0021 uuid: 0000180f-0000-1000-8000-00805f9b34fb
```

```
[AA:BB:CC:DD:EE:FF][LE]> char-desc
handle: 0x0001, uuid: 2800
handle: 0x0002, uuid: 2803
handle: 0x0003, uuid: 2a00
handle: 0x0004, uuid: 2803
            . . .
handle: 0x0071, uuid: f000ffc4-0451-4000-b000-000000000000
handle: 0x0072, uuid: 2902
handle: 0x0073, uuid: 2901
Discover descriptors finished: No attribute found within the given range
```

- Now you can read or write the files that you are interested in. You will need to understand the file contents of the device you are using.

```
[AA:BB:CC:DD:EE:FF][LE]> char-read-hnd 0x0073
Characteristic value/descriptor: 49 6d 67 20 53 74 61 74 75 73
```

Appendix: bt-init.sh contents

```
#!/bin/sh

PATH=$PATH:$EXTRA_PATH

echo "PWD is: $PWD"

stty -F /dev/ttyHSL1 115200
sleep 1

modprobe hci_uart
sleep 1
modprobe bluetooth
sleep 1
modprobe rfcomm
sleep 1
modprobe hidp
sleep 1
modprobe bnep
sleep 1

if [ ! -d "/sys/class/gpio/gpio13" ]; then
    echo 13 > /sys/class/gpio/export
fi
echo up > /sys/class/gpio/gpio13/pull
sleep 1

#mux 1  # UART1 to IoT slot 0
#mux 15 # Take IoT slot 0 out of reset
#sleep 1

hciattach /dev/ttyHSL1 texas 115200 noflow
sleep 1

hciconfig
sleep 1

hciconfig hci0 up
```

Document History

Revision number	Release date	Changes
1	December 2017	New document
2	July 2018	Added alternate Method 2: Patching the Kernel

Legal Notice

Limitation of Liability

The information in this document is subject to change without notice and does not represent a commitment on the part of Sierra Wireless. SIERRA WIRELESS AND ITS AFFILIATES SPECIFICALLY DISCLAIM LIABILITY FOR ANY AND ALL DIRECT, INDIRECT, SPECIAL, GENERAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES INCLUDING, BUT NOT LIMITED TO, LOSS OF PROFITS OR REVENUE OR ANTICIPATED PROFITS OR REVENUE ARISING OUT OF THE USE OR INABILITY TO USE ANY SIERRA WIRELESS PRODUCT, EVEN IF SIERRA WIRELESS AND/OR ITS AFFILIATES HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES OR THEY ARE FORESEEABLE OR FOR CLAIMS BY ANY THIRD PARTY.

Notwithstanding the foregoing, in no event shall Sierra Wireless and/or its affiliates aggregate liability arising under or in connection with the Sierra Wireless product, regardless of the number of events, occurrences, or claims giving rise to liability, be in excess of the price paid by the purchaser for the Sierra Wireless product.

Patents

This product may contain technology developed by or for Sierra Wireless Inc. This product includes technology licensed from QUALCOMM®. This product is manufactured or sold by Sierra Wireless Inc. or its affiliates under one or more patents licensed from MMP Portfolio Licensing.

Copyright

Trademarks

Sierra Wireless®, AirLink®, AirVantage® and the Sierra Wireless logo are registered trademarks of Sierra Wireless.

Windows® and Windows Vista® are registered trademarks of Microsoft Corporation.

Macintosh® and Mac OS X® are registered trademarks of Apple Inc., registered in the U.S. and other countries.

QUALCOMM® is a registered trademark of QUALCOMM Incorporated. Used under license.

Other trademarks are the property of their respective owners.

Contact Information

Sales information and technical support, including warranty and returns	Web: sierrawireless.com/company/contact-us/ Global toll-free number: 1-877-687-7795 6:00 am to 5:00 pm PST
Corporate and product information	Web: sierrawireless.com