

Laird Linux LWB Software Integration Guide

Sterling-LWB Module

v1.5

NOTES FOR FIRST INTEGRATION

This document explains the required steps for integrating the Sterling LWB Linux software package.

We recommend that you thoroughly analyze each step of the process. Each individual step should be integrated separately and manually tested. Attempting all of the steps at once without testing, will likely cause bugs that are difficult to troubleshoot and confusing to untangle.

Integrating a Wi-Fi driver into specific Linux platform code can be challenging and may require platform-specific changes. If the following information is confusing or does not provide the proper results, please check <https://laird-ews-support.desk.com/> for further assistance.

We recommend you consult Laird's *CS-GUIDE-IntegrateFreescale* for detailed information on how to integrate the LWB into a host microprocessor platform.

INTEGRATION

WLREG_ON and BT_REG_ON Pins

The WLREG_ON and BT_REG_ON pins must be connected to GPIOs on the host processor. You must ensure that the GPIO pins you select are set high before loading the drivers or initializing Bluetooth.

These pins control the lowest power state of the Sterling LWB. They must be appropriately configured to function correctly. We would recommend using device trees to control the configuration and state of the GPIO pins.

Building the Drivers

To build the drivers, follow these steps:

1. Download the LWB Linux backports driver source code and binary firmware. Using your web browser, visit the following: <http://www.lairdtech.com/product-categories/embedded-wireless/wi-fi-and-wi-fi-bt-modules>
2. Select the Sterling-LWB product, go to the Kits & Software tab, and download the Linux backports package *Sterling-LWB Software Version Backports (930-0075)* to **/home/user/930-0075.zip**.
The binary firmware packages are available for three regulatory regions.
3. Select the firmware package which matches the region in which you intend to operate your end device.
 - For the USA or Canada, use the *Sterling-LWB Firmware Package (480-0079)* to **/home/user/480-0079.zip**.

- For the EU and Australia use the *Sterling-LWB Firmware Package (480-0080)* to **/home/user/480-0080.zip**.
- For Japan, use the *Sterling-LWB Firmware Package (480-0116)* to **/home/user/480-0116.zip**.

Table 1: Regional firmware packages

Regulatory Domain	Firmware Package Name
USA, IC	480-0079.zip
ETSI, RCM	480-0080.zip
Giteki	480-0116.zip

4. Extract the LWB driver source code and firmware.

Your driver directory should appear as follows (irrelevant columns deleted).

```
ls -l
drwxrwxr-x [...] fsl-release-bsp
drwxrwxr-x [...] laird-backport-<version>
drwxrwxr-x [...] lib
```

5. Generate the configuration that guides the build process for the regulatory domain in which you intend to operate the end device.

```
For the USA or Canada: make defconfig-lwb-fcc
ETSI and RCM: make defconfig-lwb-etsi
Giteki: make defconfig-lwb-jp
```

6. Set the environment variables that tell the driver build about your cross-development environment.

```
export CROSS_COMPILE="${HOME}/projects/fsl-release-bsp/build-imx6ul-fb/tmp/sysroots/x86_64-linux/usr/bin/arm-poky-linux-gnueabi/arm-poky-linux-gnueabi-"
export ARCH="arm"
export KLIB_BUILD="${HOME}/projects/fsl-release-bsp/build-imx6ul-fb/tmp/work/imx6ulevk-poky-linux-gnueabi/linux-imx/4.1.15-r0/build"
```

7. Invoke *make* to build the drivers.

```
make
```

Observe four kernel modules were built. We'll copy these onto the board's Root File System in a moment.

```
find . -name '*.ko'
./drivers/net/wireless/brcm80211/brcmfmac/brcmfmac.ko
./drivers/net/wireless/brcm80211/brcmutil/brcmutil.ko
./net/wireless/cfg80211.ko
```

```
./compat/compat.ko
```

The WL_REG_ON and BT_REG_ON signals need to be asserted to enable radio operation. We recommend that these signals be connected to GPIO lines on your host processor. You may eventually wish to use device tree modifications and regulator sub-system to control these signals, however, for early driver integration the Linux GPIO system is suitable.

For further information on device trees and manual control of the GPIO lines, consult the *CS-GUIDE-IntegrateFreescale* available from the [documentation download section of the Sterling product page](#).

BLUETOOTH

BT_REG_ON

Confirm that the BT_REG_ON pin is pulled high before using Bluetooth. If this line is not correctly asserted, Bluetooth does not function.

Patching Firmware to the Bluetooth Radio

The Bluetooth radio requires that firmware is patched to the radio for proper use. If firmware is not patched to the radio, it does not have a valid MAC address and may behave strangely. The firmware is contained in the .hcd file within the firmware folder alongside the package downloaded from the Laird website.

We recommend using the *brcm_patchram_plus* utility provided to correctly patch the firmware file to the radio.

```
wget
https://github.com/LairdCP/brcm_patchram/archive/brcm_patchram_plus_1.1.tar.gz
md5sum brcm_patchram_plus_1.1.tar.gz
3c03e03ce4ce11ea131702779906c6b3 brcm_patchram_plus_1.1.tar.gz
```

To patch the firmware to the Bluetooth radio, follow these steps:

1. Extract and build the patchram plus application. For example...

```
tar xf brcm_patchram_plus_1.1.tar.gz -C projects/
cd projects/brcm_patchram-brcm_patchram_plus_1.1/
make
[warnings scroll past]
arm-poky-linux-gnueabi-gcc -march=armv7-a -mfloat-abi=hard -mfpu=neon -
mtune=cortex-a7 --sysroot=/home/user/projects/fsl-imx-fb/4.1.15-
1.2.0/sysroots/cortexa7hf-vfp-neon-poky-linux-gnueabi -Wl,-O1 -Wl,--hash-
style=gnu -Wl,--as-needed brcm_patchram_plus_usb.o -lbluetooth -o
brcm_patchram_plus_usb
```

Note: For more details, please consult the *CS-GUIDE-IntegrateFreescale* documentation.

2. Use the following commands with *brcm_patchram_plus* to patch the radio:

```
brcm_patchram_plus --patchram {path to .hcd firmware file} --enable_hci --  
no2bytes --tosleep 1000 {path to /dev/ UART interface} &
```

Downloading and Using the BlueZ Bluetooth Stack

Laird supports the BlueZ stack for use with the Sterling LWB. The latest user space application code can be downloaded from the following link: <http://www.bluez.org/download/>

Note: Certain BlueZ functions are dependent on the Linux kernel driver that you are using.

Once the radio has been patched, the Bluetooth interface can be brought up using the following:

```
hciconfig hci0 up
```

BASIC BRING UP QUICK START

Assuming that the hardware is connected correctly, the Device Trees are configured correctly to assert WL_REG_ON and BT_REG_ON, and that everything on your system works, the LWB should then be tested against an open AP. This validates basic functionality with a minimum of effort and extra software and settings.

The following are the basic steps:

1. Physically connect the module, configure the kernel, integrate files, and so on...
2. Power on your board and login via a shell.
3. Load the drivers.
4. Configure the GPIO and set it HIGH.
5. Use **ifconfig** to enable the wlan0 interface.
6. Use **iw** to connect to the open AP.
7. Use a DHCP client to get an IP address.
8. Ping some other device across the wireless network.

If everything is set up correctly, the above steps result in the LWB being connected to an AP, getting an IP address, and passing data. If you can do this, then the LWB is integrated correctly.

The below script is from our WB50N-RDVK and shows the test steps. After running these steps, pinging or running iperf against a server across the wireless network should work.

```
modprobe brcmfmac  
ifconfig wlan0 up  
iw wlan0 connect ccopen1  
sleep 1  
udhcpc -i wlan0
```

[lwb-start.sh]

DEBUGGING

Driver Output

Upon successful initialization of the driver, the following is placed in the kernel log:

```
mmc0: new high speed SDIO card at address 0001
brcmfmac: F1 signature read @0x18000000=0x1531a9a6
brcmfmac: brcmf_sdio_drivestrengthinit: No SDIO Drive strength init done for chip 43430 rev 1 pmurev
24
brcmfmac: brcmf_c_preinit_dcnds: Firmware version = wl0: Dec 29 2015 15:56:15 version 7.45.41.24
(r608913) FWID 01-4e412465
```

If you do not see the above, most likely you have not released the WL_REG_ON pin to a high value.

Wpa_supplicant Usage

The native wpa_supplicant can be used for Wi-Fi setup and configuration. To enable the wpa_supplicant, use the following command:

```
wpa_supplicant -Dnl80211 -c /etc/wpa_supplicant.conf -i wlan0 -B
```

The wpa_supplicant can be configured with the wpa_cli. The following steps allows configuration of an open wireless network:

```
wpa_cli disconnect
wpa_cli add_network
wpa_cli set_network 1 ssid '"[name]"'
wpa_cli select_network 1
wpa_cli enable_network 1
```

REVISION HISTORY

Version	Date	Notes	Approver
1.1	06 June 2016	Initial Release	Brian Wagner
1.2	22 July 2016	Added Bluetooth Support and integration instructions. Added note on BT_REG_ON Clarified section on device tree modifications. Removed unnecessary references to bcmhd driver. Updated the location of the Sterling LWB binary files	Andy Dobbing
1.3	06 Sept 2016	Removal of legacy information	Steve deRosier
1.4	01 Dec 2016	Removed incorrect and duplicated information	Andrew Dobbing
1.5	10 Jul 2018	Updated firmware package descriptions and MD5 sums; Add Giteki regulatory	Eric Bentley

© Copyright 2018 Laird. All Rights Reserved. Patent pending. Any information furnished by Laird and its agents is believed to be accurate and reliable. All specifications are subject to change without notice. Responsibility for the use and application of Laird materials or products rests with the end user since Laird and its agents cannot be aware of all potential uses. Laird makes no warranties as to non-infringement nor as to the fitness, merchantability, or sustainability of any Laird materials or products for any specific or general uses. Laird, Laird Technologies, Inc., or any of its affiliates or agents shall not be liable for incidental or consequential damages of any kind. All Laird products are sold pursuant to the Laird Terms and Conditions of Sale in effect from time to time, a copy of which will be furnished upon request. When used as a tradename herein, *Laird* means Laird PLC or one or more subsidiaries of Laird PLC. Laird™, Laird Technologies™, corresponding logos, and other marks are trademarks or registered trademarks of Laird. Other marks may be the property of third parties. Nothing herein provides a license under any Laird or any third party intellectual property right.