

# Laird Linux Software Integration Guide

## SD45/SD50 Modules

*Application Note*

v2.1

### NOTES FOR FIRST INTEGRATION

This document explains the steps required to fully integrate the SD45/SD50 and MSD45/MSD50 Linux software packages.

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.

### INTEGRATION

#### Ready Your Board Support Package

Before integrating the Linux software package, disable all device software from attempting to manage the wireless connection. This includes, but is not limited to, NetworkManager, Connman, wpa\_supplicant, and others. We also expect that the customer's platform supplies the DHCP solution for use with the wireless interface. We can provide guidance if needed on this subject. The WB45 and WB50 software releases can be used as a guide for implementing a DHCP solution if your board support package doesn't already include one. Contact Laird support for more details.

#### Test Your System Using the Stock Driver and Firmware

Before trying to integrate the driver, we recommend that you validate your hardware and Linux kernel configuration by building and trying to use the stock ath6kl driver and QCA's stock firmware. Enable the ath6kl driver module in your kernel, build it and attempt to use it. If you are able to at least connect to an open AP with the stock driver and firmware, then you know for sure that the underlying SDIO hardware and the MMC subsystem is configured correctly and we can eliminate these as issues if you have problems with the drivers and firmware.

When bringing up the Laird drivers and firmware, if you have trouble communicating to the chip, this will be one of the first support suggestions to verify your configuration.

#### CHIP\_PWD\_L GPIO pin

The CHIP\_PWD\_L pin must be connected to a GPIO on the host processor. On the WB45NBT/WB50NBT, we do this and our driver automatically controls this pin. However, for the MSD, we cannot know what GPIO you chose

to connect the pin to, and the GPIO subsystem in Linux is not backported. Therefore the backport driver for the MSD does not control this pin. You must make sure that the pin is set high (please refer to the HIG for specifications) before loading the drivers.

Holding this pin low will hold the MSD45/MSD50 in reset and is also the lowest power state for the device. If you wish to use suspend to bring your device to a low-power state, you should first unload the driver modules and then assert the GPIO connected to the CHIP\_PWD\_L pin. Only then should you suspend. To wake the system, first wake it, de-assert the pin, and then load the ath6kl drivers.

## Supporting Libraries and Programs

Before integration of our enhanced wireless software package, several prerequisite libraries and programs should be added to your device's root file system. We recommend that they be added through your device's board support package if possible.

- **libnl** – libnl is the set of libraries used by userspace programs to communicate to the kernel's netlink interface. We recommend using version 3.2.25 or greater if possible. For more information visit <http://www.infradead.org/~tgr/libnl/>
- **OpenSSL version 1.0.1h** - OpenSSL is the encryption library that our supplicant uses. See the Building OpenSSL section below for details on integrating OpenSSL correctly. For more information visit <https://www.openssl.org/>

## Laird Backports Driver

Backports is a software package that allows Linux users to port a driver from a newer Linux kernel back to an older Linux kernel. We have created a version of Linux Backports that contains all customizations to the ath6kl driver provided with the Linux kernel. The current Backports releases are based on the 4.4 kernel and generate ath6kl drivers for kernel versions 3.0 to 4.4.

If you already have contacted Support and have access to our download site, please visit:  
<http://www.lairdtech.com/products/msd45n> or <http://www.lairdtech.com/products/msd50n>.

Download the Backports release that corresponds to the software release that you are integrating. Once you have downloaded the correct release, extract the archive.

On your host build system, complete the following steps:

```
set -a
CROSS_COMPILE=${CROSS_COMPILE}
ARCH=${TARGET_CPU}
KLIB_BUILD=${DEV_PATH}/${LINUX_DIR}
KLIB=${TARGET_ROOTFS_ON_HOST}
set +a
make defconfig-laird
make
make install
```

Note the following:

- `${CROSS_COMPILE}` – The location of your cross compiler
- `${TARGET_CPU}` – Your target CPU architecture

- `${DEV_PATH}/${LINUX_DIR}` – The location of your already-configured kernel build directory
- `${TARGET_ROOTFS_ON_HOST}` – The location of your rootfs to install the modules into and fix kernel module dependency files

The following files are generated by this process:

- `ath6kl_core.ko` ----- at `./drivers/net/wireless/ath/ath6kl`
- `ath6kl_sdio.ko` ----- at `./drivers/net/wireless/ath/ath6kl`
- `cfg80211.ko` ----- at `./net/wireless/cfg80211.ko`
- `compat.ko` ----- at `./compat/compat.ko`

If the `make install` command is not working then these files must be added to your device's root filesystem in `/lib/modules/<kernel name>/kernel/<each module path above>/<module>.ko` .

If you encounter dependency issues when inserting the ath6kl driver modules, run `depmod -a` to regenerate the kernel's module dependency file on a running device with the modules installed.

The following is the correct order of module loading:

- `compat.ko`
- `cfg80211.ko`
- `compat_firmware_class.ko` (optional, only if needed by your kernel)
- `ath6kl_core.ko`
- `ath6kl_sdio.ko`

We recommend that `ath6kl_core` is loaded with the following module parameters:

```
recovery_enable=1 heart_beat_poll=1000
```

These module parameters enable automatic firmware recovery on a radio firmware crash and heartbeat detection for recovering unresponsive radio firmware.

## Laird-Provided Binary Files

If this is a new integration and you have not already been provided the following files, please contact [ews.support@lairdtech.com](mailto:ews.support@lairdtech.com).

If you already have contacted Support and have access to our download site, please visit: <http://www.lairdtech.com/products/msd45n> or <http://www.lairdtech.com/products/msd50n>. Download the appropriate version of the software release for Linux and extract the archive file. You also must extract the `rootfs.tar` archive file contained within the release archive file.

## Laird Firmware Files (SD45)

These can be found under `/lib/firmware/ath6k/AR6003/hw2.1.1/` after extracting the `rootfs.tar`. These files should be placed by your build system into `/lib/firmware/ath6k/AR6003/hw2.1.1/` on your root file system.

- `bdata.bin` – This should be a symbolic link to the real `calData_XXXXXXXXXX_XXXX_CCA.bin`.
- `calData_XXXXXXXXXX_XXXX_CCA.bin` – This is the board data file which includes unique settings and calibration for the module.
- `fw-4.bin` – This should be a symbolic link to the real `fw_v3.x.x.x.bin`.
- `fw_v3.x.x.x.bin` – This is the enhanced radio firmware for normal runtime.
- `athtcmd_ram.bin` – This should be a symbolic link to the real `athtcmd_ram_v3.x.x.x.bin` (optional).

- **aththcmd\_ram\_v3.x.x.x.bin** – This is the firmware to use when wanting to run **aththtestcmd** for regulatory testing (optional).

### Laird Firmware Files (SD50)

These can be found under **/lib/firmware/ath6k/AR6004/hw3.0/** after extracting the rootfs.tar. These files should be placed by your build system into **/lib/firmware/ath6k/AR6004/hw3.0/** on your root file system.

- **bdata.bin** – This should be a symbolic link to the real 50NBTBDF0xxxx.bin
- **50NBTBDF0xxxx.bin** – This is the board data file which includes unique settings and calibration for the module.
- **fw-5.bin** – This should be a symbolic link to the real fw\_v3.x.x.x.bin.
- **fw\_v3.x.x.x.bin** – This is the enhanced radio firmware for normal runtime.
- **utf.bin** – This should be a symbolic link to the real utf\_v3.x.x.x.bin (optional).
- **utf\_v3.x.x.x.bin** – This is the firmware to use when wanting to run **aththtestcmd** for regulatory testing (optional).

### Laird Executable Files

These can be found under **/usr/bin/** after extracting the rootfs.tar. These files should be placed by your build system into **/usr/bin/** on your root file system.

- **sdcsupp** – This is the supplicant. It provides enhanced authentication and controls the wireless connection from userspace. It should be started after the ath6kl driver has been loaded during system initialization. We recommend the following parameters when starting sdcsupp:

```
sdcsupp -iwlan0 -Dnl80211 -s
```

The **-i** specifies the wireless interface, **-D** enables using nl80211 to communicate with the kernel, and **-s** allows logging to syslog.

- **sdcli** – This is the CLI to control wireless operation. For more information, see the SDC\_CLI user manual.
- **event\_mon (optional)** – This is a program to log Laird WiFi connectivity events for debugging and to optimize firmware recovery. Source for this program can be found at [https://github.com/LairdCP/wb-event\\_mon](https://github.com/LairdCP/wb-event_mon). See our WB Logging and Troubleshoot Guide and SDK documentation for further information. If used, we recommend the following flags:

```
event_mon -ologging -b0x0000003FA3008000 -m
```

The **-o** specifies that all events should be logged to syslog, **-b** specifies a bitmask of the most commonly used events, and **-m** specifies radio state manager mode that optimizes firmware recovery.

- **dhcp\_injector (optional)** – This is for injecting DHCP events into SDK Events. See our SDK documentation for further information.
- **aththtestcmd (optional)** – This is the Atheros test command program for regulatory and manufacturer testing. This should not be provided in normal production release and should only be used for testing.

### Laird Libraries Files

These can be found under **/usr/lib/** after extracting the rootfs.tar. These files should be placed by your build system into **/usr/lib/** on your root file system.

- **libsd\_sdk.so.1** – This is a symbolic link to the current version of our SDK.
- **libsd\_sdk.so.1.0** – This is the Wi-Fi SDK and is required for the CLI and supplicant to work. This SDK can also be used to allow your applications to interface with the wireless package in C or C++. See our SDK documentation for more information.

## REVISION HISTORY

Version	Date	Notes	Approver
1.0	15 Oct 2014	Initial Release	Brian Wagner
1.1	24 Feb 2015	Changed backports information to our version, changed references to Android specific file integration to Linux specific integration, added information on integrating regulatory database.	Dan Kephart
1.2	30 Apr 2015	Added sections on , , it and added information on compat_firmware kernel module.	Steve deRosier
2.0	28 Mar 2016	Added SD50 information	Brian Wagner
2.1	19 Jan 2017	Updated for Kernel v4.4	Dan Kephart