



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China



60 Series Radio Software Integration Guide

21 Mar 2018

330 - 0261 - R1.4

Revision History

Version	Date	Notes	Approver
1.0	28 July 2017	Preliminary Draft	Andrew Dobbing
1.1	31 Aug 2017	Light edits	Dave Drogowski
1.2	14 Sep 2017	Added to BT bring-up commands	Andrew Dobbing
1.3	18 Sep 2017	Updated GitHub info, added DVK_2230C SDR104 info and made minor syntax updates	Andrew Dobbing
1.4	21 Mar 2018	Updated language to reference support for kernel 3.0 to 4.9	Dave Drogowski

Contents

1 Introduction	4
1.1 Terminology	4
1.2 Scope	4
2 Required Materials	4
2.1 Cautions and Warnings	5
2.2 Build Host Linux Distribution and Version	5
3 Enabling the Radio Hardware for Operation	5
4 Modifying the 60 Series Radio DVK for SDR104 Operation	5
4.1 Obtaining Source from GitHub	6
5 Building BackPorts	6
5.1 Notes on the wpa-supPLICANT	6
5.2 Prepare Your Board Support Package	6
5.3 Introduction to Backports	6
5.4 Commands to build Backports	6
5.5 Running the 60 Series Driver	7
6 Bluetooth	8
6.1 Loading the Bluetooth driver	8

1 Introduction

This document details how to fully integrate the Laird Sterling-60 Wi-Fi and Bluetooth Linux software package for the ST60_SIPT and ST_60_2230C modules using the Laird Backports package. We recommend 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. Integrating a Wi-Fi driver into specific Linux platform code can be challenging and may require platform-specific changes. For help, downloads, and further documentation, please visit <https://laird-ews-support.desk.com/> for further assistance.

1.1 Terminology

The following terminology is applicable to this document:

- **ST60-2230C** - Sterling-60 Radio on M.2 form factor
- **ST60-SIPT** - Sterling-60 Radio in SIP form factor
- **DVK_SU60-2230C** - Sterling-60 Radio M.2 form factor Development Kit
- **DVK_SU60-SIPT** - Sterling-60 Radio SIP form factor Development Kit
- **RDVK-50** - Laird's 50 Series Radio Development Kit. Used for the evaluation of all Laird radios

1.2 Scope

The scope of this document is limited to providing instruction on where to obtain the Laird backports package, how to run the backports package and how to build the Sterling 60 series drivers for your Linux version of choice and their inclusion into your filesystem.

It is assumed the reader has an understanding of Linux and building Linux software, the basic operation of the build system employed within the host platform to which the 60 Series radio is utilized. Laird recommends the use of Buildroot as a build system for embedded Linux systems. However, the specifics related to your build system are beyond the scope of this document. If you are using Buildroot, please consult the Laird RDVK documentation which utilizes a Buildroot distribution. If you are using Open Embedded or Yocto, please consult the documentation regarding those distributions. The remainder of this document assumes that you are using an RDVK-50 as your development platform to evaluate the 60 series radio. You should make the appropriate modifications to suit your development environment.

2 Required Materials

The following are required for this integration:

- Laird Sterling-60 development kit with SDIO or PCIE card form factor
- Host development platform running Linux version 3.0 or greater (if you are using kernel versions prior to 3.18, please contact Laird)
- RDVK-50 or other suitable host processor platform with an available SDIO or PCIE for the WLAN interface and an available UART port for the Bluetooth interface

2.1 Cautions and Warnings

Adhere to all cautions and warnings included with all materials. The 60 Series radio and DVK can be damaged by electrostatic discharge (ESD); handle accordingly.

2.2 Build Host Linux Distribution and Version

This guide was produced using a build host running Ubuntu 16.04.1. While it is possible to use a different version of Ubuntu or an entirely different Linux distribution, doing so may require a different procedure. Build hosts running distributions other than Ubuntu 16.04.1 are beyond the scope of this document.

3 Enabling the Radio Hardware for Operation

The radio requires you to de-assert the PCIE_W_DISABLE_N signal to allow the radio to operate. Consult the DVK schematic and user guide for the location of the PCIE_W_DISABLE_N signal. Depending on which DVK you are using this pin is situated on different connectors. If you are using the Radio Development Kit (RDVK_50) from Laird, connect a jumper cable between the RDVK-50 development board pin I2C.4 and the following connectors on the DVK.

- **DVK_SU60-2230C** - Pin J23.4
- **DVK_SU60-SIPT** - Pin J8.6

If you are using a different development kit such as an MSC-Q7 or iMX.UL, please consult the schematics and documentation for those devices to select a suitable GPIO to control the PCIE_W_DISABLE_N signal. You also need to make the appropriate changes to the device trees for your development board.

4 Modifying the 60 Series Radio DVK for SDR104 Operation

By default the 60 series radio DVK boards (DVK-60-SIPT and DVK-60-2230C) have an SDIO bus configured to operate at 3.3v for HS operation. To modify the DVK board for SDR104 operation, the board

must be configured to operate at 1.8V. The method required to enable this operation depends on the DVK that being used. On a DVK_SIPT board cut the solder bridge SB4. On a DVK_2230C it is just necessary to ensure that J4 and J7 have been removed.

4.1 Obtaining Source from GitHub

The source for Laird's distribution is available at our publicly facing GitHub repo <https://github.com/LairdCP/Release-Packages/tree/master/ST60/>. The repo contains the firmware binary and driver backports package.

5 Building BackPorts

5.1 Notes on the wpa-supPLICANT

Laird recommends that you use wpa_supplicant 2.6. This is the latest wpa_supplicant available and fixes a great many bugs found in older versions of the supplicant. The 60 series radio has been tested using the wpa_supplicant 2.6. Failures on a customer platform which are related to Wi-Fi performance with a prior version of the supplicant are most likely to be related to the supplicant version being used.

5.2 Prepare 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, Conman, wpa_supplicant, and others. We also expect that the customer's platform supplies the DHCP solution for use with the wireless interface.

5.3 Introduction to Backports

Laird recommends that you use the LTS 4.9 kernel. However, that isn't always possible for your processor of choice or development platform. Laird uses the Backports package to allow you to take the driver package Laird releases, which is tested at a known kernel version (currently Linux kernel version 4.9), and apply the driver in a standard way to the kernel version you are using. The current backports package can be used to build the driver for kernel versions between 3.0 and 4.9.

5.4 Commands to build Backports

After extraction, run following commands to cross-compile backport driver modules:

```
1 set -a
2 CROSS_COMPILE=${CROSS_COMPILE}
3 ARCH=${TARGET_CPU}
4 KLIB_BUILD=${DEV_PATH}/${LINUX_DIR}
5 set +a
6 make defconfig-lrdmwl
7 make
```

The following kernel driver modules will be generated by this process and need to be loaded on the target:

```
1 lrdmwl.ko ----- at ./drivers/net/wireless/laird/lrdmwl/
2 lrdmwl_sdio.ko --- at ./drivers/net/wireless/laird/lrdmwl/
3 mac80211.ko ----- at ./net/mac80211/
4 cfg80211.ko ----- at ./net/wireless/
5 compat.ko ----- at ./compat/
```

5.5 Running the 60 Series Driver

Plug in the USB cable from USB2 on the RDVK-50 development kit to the USB port on your laptop, and confirm your serial port assignment is correct. Using `dmesg` to log the output from the RDVK-50, you should see indications that the USB serial device is recognized and the enumeration of the USB device corresponding to the RDVK-50. It should indicate something resembling the following:

```
1 FTDI USB Serial Device converter now attached to ttyUSB0
```

The `lrdmwl` driver expects firmware to be in `/lib/firmware/lrdmwl/` on the target. The firmware can be found in the `laird-sterling-60` tarball. The software should connect to all security types supported by the `wpa_supplicant 2.6` and have normal throughput.

Insert the SDIO adapter into the SD slot. Confirm that `wlan0` is present by looking at the output of the command **`ip addr`**.

To load the Wi-Fi drivers, use the following commands:

```
1 insmod compat/compat.ko
2 insmod net/wireless/cfg80211.ko
3 insmod net/mac80211/mac80211.ko
4 insmod drivers/net/wireless/laird/lrdmwl/lrdmwl.ko
5 insmod drivers/net/wireless/laird/lrdmwl/lrdmwl_sdio.ko
```


6 Bluetooth

The Laird driver also provides support for Bluetooth on the 60 Series radio, including downloading the Bluetooth firmware over the primary interface and establishing a connection over the Bluetooth interface.

6.1 Loading the Bluetooth driver

To load the Bluetooth drivers and bring up the BT interface, type the following at the prompt on the RDVK-50:

```
1 insmod net/bluetooth/bluetooth.ko
2 insmod drivers/bluetooth/hci_uart.ko
3 hciattach /dev/ttyUSB0 any 3000000 1
```

You can confirm that the BT interface is present and has been assigned a MAC address by typing the following:

```
1 hciconfig
```

Next, bring up the HCI interface as follows:

```
1 hciconfig hci0 up
2 hcitool -i hci0 cmd 3f 0087 0x05
```

You can now use the `hcitool` to operate the BT interface. Typing the command `hcitool` will provide a list of available commands.