WLAN  
Host Driver Design

Copyright © Imagination Technologies Limited. All Rights Reserved.

This document is strictly confidential. Neither the whole nor any part of the information contained in, nor the product described in, this document may be adapted or reproduced in any material form except with the written permission of Imagination Technologies Limited. Imagination Technologies, the Imagination logo, PowerVR, MIPS, Meta, Ensigma and Codescape are trademarks or registered trademarks of Imagination Technologies Limited. All other logos, products, trademarks and registered trademarks are the property of their respective owners. This document can only be distributed subject to the terms of a Non Disclosure Agreement or Licence with Imagination Technologies Limited.

Filename : WLAN.Host Driver Design  
Version : 1.0.31 Not Issued - Live Document  
Issue Date : 22 Jan 2014  
Author : Imagination Technologies Limited

Contents

[1. Introduction 3](#_Toc378273341)

[2. Overview 3](#_Toc378273342)

[3. Driver design 3](#_Toc378273343)

[4. Driver code 4](#_Toc378273344)

List of Figures

**No table of figures entries found.**

IF{{Internal}}

Document History

| Issue | Date | Changes/Comments |
| --- | --- | --- |
| 1.0.26 | 01 Oct 2013 | External Issue. |
| 1.0.27 | 16 Dec 2013 | External Issue |
| 1.0.30 | 22 Jan 2014 | External Issue |
| 1.0.31 | 22 Jan 2014 | External Issue |

END{{Internal}}

# Introduction

This document describes about IMG WLAN host driver design.

# Overview

IMG WLAN host driver is a mac80211 dependant linux kernel driver. It’s a loadable kernel module.

Figure 1 below shows the IMG wlan driver architecture. It has three core modules mac80211\_if, lmac\_if and hal\_if.

mac80211\_if

lmac\_if

hal\_if

Figure 1: IMG Wlan driver architecture

* mac80211\_if module interfaces with mac80211 module in linux kernel and implements all mac80211 ops.
* lmac\_if module implements framework to encap/decap messages to/from lmac firmware
* hal\_if module has ISR, TX and RX tasklets.

Host to firmware messages are communicated with CMD\_TX, CMD\_CHANNEL etc

Firmware to host messages are communicated with EVENT\_RX, EVENT\_TX\_DONE etc

There is a core module which handles beacon programming, EDCA parms, reset command proc etc.

# Driver design

When Driver loads into the kernel, hal\_init() is called. Hal\_init() requests IO memory regions, initializes TX, RX tasklets and registers hal\_irq\_handler and finally registers mac80211\_ops.

Some of the driver config parms are read from proc/umac/parms and stored into wifi\_params struct.

When user creates virtual interface, add\_interface() op is called and programs firmware with vif\_index, mac\_addr and mode (STA/AP/ADHOC) with IF\_ADD command.

On removal of virtual interface, remove\_interface() op is called and sends IF\_REMOVE command to firmware.

When the interface is activated, start() op gets called and driver issues CMD\_RESET with value LMAC\_ENABLE to firmware and waits for EVENT\_RESET\_COMPLETE. Programs RX\_buffer cmds to hw, inits TxQs.

When interface is deactivated stop() op gets called and driver issues CMD\_RESET with value LMAC\_DISABLE to firmware and waits for EVENT\_RESET\_COMPLETE and releases all pending outstanding cmds.

Config() op takes care of channel info programming.

Tx() op supports transmission of frames. Mac80211 gives dot11 pkt skb (either mgmt/data) to driver through this op. Driver allocs Tx\_buf and submits to firmware through CMD\_TX. Completion of this pkt is handled through EVENT\_TX\_DONE.

Firmware indicates receive pkts through EVENT\_RX. Driver extracts skb from Rx\_buf and indicates to mac80211 through ieee80211\_rx().

# Driver code

Driver global data stored in struct mac80211\_dev {}.

Config parms are stored in struct wifi\_params {}

Hal\_if code is in hal\_hostport.c

Lmac\_if code is lmac\_if.c

Mac80211\_if code is in 80211\_if.c

Driver source tree:

Projects/WLAN/uccp420/uccp/drivers/softmac/linux/3\_10

inc/ contain include header files

patches/ contain any required kernel patches.

src/ contain all c files.