



INVENTEK SYSTEMS
ISM4334x-L77 EVB
i.MX6 Quick Start Guide



Table of Contents

1	Overview	3
2	Required Files	3
3	i.MX6 Installation Procedure	3
3.1	Linux Install	3
3.2	Download Wireless Drivers	4
4	WL Commands	4
4.1	Introduction	4
4.2	Bringing Up Wireless Adapter	5
4.3	Scanning for a Wireless Network	5
4.4	Connect to a WI-Fi network	5
4.5	Join Script	7
5	REVISION CONTROL	8
6	CONTACT INFORMATION	8

1 Overview

The ISM43341-L77 evaluation board offers 802.11 Wi-Fi connectivity over an SDIO interface, as well as Bluetooth over a UART interface. The purpose of this document is to assist you in bringing up the Wi-Fi interface on an i.MX6 processor running the Linux operating system. There are many processor boards in the i.MX6 family. The instructions in this document will target the i.MX6 UltraLite evaluation kit (MCIMX6UL-EVK), running Linux Kernel version 3.14.38.

2 Required Files

The following files are referenced in the instructions and are made available on inventeksys.com

- `dhd.ko` Broadcom 43340/1 Dongle Host Driver
- `ISM4334X_Wifi_FW_6.10.190.49_P.bin` Wi-Fi Firmware
- `imx6ul-14x14-evk.dtb` Device Tree Blob for SDIO Interface
- `join.sh` Shell Script to Join a Network (Sample)
- `nvr.am.txt` Broadcom 43340/1 Configuration File
- `wlarm` WL Utility Compiled for ARM (Executable)

3 i.MX6 Installation Procedure

3.1 Linux Install

The Linux operating system can be installed on the i.MX6 UltraLite evaluation kit by following the instructions outlined in the quick start guide (IMX6ULTRALITEQSG) found on nxp.com. The default Linux configuration provided, however, will not support the ISM43341-L77 evaluation board without some modification as the i.MX6 processor requires control of the pin `WL_REG_ON` which is found on the 43341 module. Control of this pin is accomplished by modifying the device tree that is passed to the kernel at boot time.

The device tree contains a description of all the hardware in the system. The file that is passed to the kernel is known as the device tree blob (`.dtb`). This file is generated by compiling a number of device tree source (`.dts`) and device tree source include (`.dtsi`) files. A modified dtb file is provided with this document for the i.MX6 UltraLite evaluation kit. In the event that you are using a different i.MX6 platform you will need to recompile your device tree blob with the provided dtsi file included at the bottom of your device tree source.

The modified dtb file is loaded by rebooting the i.MX6 and stopping it at the u-boot prompt. Once at this prompt you will execute the following commands:

```
setenv fdt_file imx6ul-14x14-evk.dtb
saveenv
reset
```

The reset command will cause the kernel to boot and allow you to load the wireless drivers.

3.2 Download Wireless Drivers

After booting the i.MX6 UltraLite evaluation kit from the micro SD card you will need to transfer the file deliverables.zip to the device. File transfer can be accomplished by mounting a USB thumb drive or by hosting the file on a TFTP server. A free TFTP server for Windows can be downloaded from the URL

<http://sourceforge.net/projects/tftp-server>

Once the file deliverables.zip has been transferred to the device you will extract it using the command

```
unzip deliverables.zip
```

The extracted files contain a kernel object file (dhd.ko). This file is the wireless driver and it can be loaded via the command:

```
insmod dhd.ko firmware_path= ISM4334X_Wifi_FW_6.10.190.49_P.bin nvram_path=nvram.txt
```

4 WL Commands

4.1 Introduction

In order to communicate with wireless devices on your network there are two primary steps that you will need to perform: (1) bring up the wireless adapter and (2) connect to a network. Once you have performed these two steps you will be able to send and receive data over your Wi-Fi network using standard Linux network sockets. The configuration process is accomplished using the Linux program wl which allows you to control your wireless adapter. On some ARM platforms this program is named wlarm. The two commands are to be considered equivalent.

4.2 Bringing Up Wireless Adapter

The first step is to activate your WIFI adapter. This task is accomplished using the commands

```
./wlrnm PM 0  
./wlrnm up
```

The argument PM to the wl command sets the driver power management mode.

PM
Set driver power management mode.
0: CAM (constantly awake)
1: PS (power-save)
2: FAST PS mode

The argument up to the wlrnm command reinitializes the adapter and marks it as up.

4.3 Scanning for a Wireless Network

Once you have activated your wireless module you may want to scan to see what wireless networks are available. Displaying a list of the nearby wireless networks is accomplished using the command sequence

```
./wlrnm scan  
./wlrnm scanresults
```

The scan argument to the wl command initiates the scan and the scanresults argument displays the gathered information.

4.4 Connect to a WI-Fi network

The last step in the process of connecting to a WIFI network is to join the network. In order to accomplish this task you will need to know the WIFI network's security type and password. The join syntax is:

```
join <ssid>[key <0-3>:xxxxx]  
      [imode bss|ibss]  
      [amode open|shared|openshared|wpa|wpapsk|wpa2|wpa2psk|wpaone]
```

The following examples show how to connect to networks of various different security types.

Connect to an open network

```
./wlan join SSID imode bss
udhcpc -i eth1
```

Connect to WEP network with 64-bit encryption

```
./wlan join SSID key 63CACE42D9 imode bss
udhcpc -i eth1
```

Connect to WEP network with 128-bit encryption

```
./wlan join SSID key 014842D480B571495A4A036379 imode bss
udhcpc -i eth1
```

Connect to WPA2-PSK network

```
./wlan wsec 4
./wlan sup_wpa 1
./wlan wpa_auth 4
./wlan set_pmk PASSWORD
./wlan join SSID imode bss amode wpa2psk
udhcpc -i eth1
```

When connecting to a WPA network you will need to utilize the `wsec` and `wpa_auth` arguments to set the security bit vector and authorization mode. The options for both commands are listed below.

wsec

Gets or sets the wireless security bit vector.

```
./wlan wsec [<vector>]
```

Parameters

vector

<i>Value</i>	<i>Description</i>
Bit 0	WEP enabled
Bit 1	TKIP enabled
Bit 2	AES enabled
Bit 3	WSEC in software

Returns

A bit vector in the same form as the above vector parameter

wpa_auth

Sets the WPA authorization mode.

```
./wfirm wpa_auth [<value>]
```

Parameters

<i>Value</i>	<i>Description</i>
1	WPA-NONE
2	WPA-802.1X/WPA-Professional
3	WPA-PSK/WPA-Personal

Returns

The WPA authorization mode

4.5 Join Script

The functionality from sections 4.1-4.5 has been implemented by Inventek, and can be seen in the sample join.sh script.

5 REVISION CONTROL

Document : ISM43341	Wi-Fi SIP module
External Release	DOC-AN-20093

Date	Author	Revision	Comment
10/6/2015	KMT	1.0	Preliminary

6 CONTACT INFORMATION

Inventek Systems
2 Republic Road
Billerica Ma, 01862
Tel: 978-667-1962
Sales@inventeksys.com

www.inventeksys.com

Inventek Systems reserves the right to make changes without further notice to any products or data herein to improve reliability, function, or design. The information contained within is believed to be accurate and reliable. However Inventek Systems does not assume any liability arising out of the application or use of this information, nor the application or use of any product or circuit described herein, neither does it convey any license under its patent rights nor the rights of others.