wi2wi®

Wireless 2 Wireless

WLAN Module W2SW0025

User Manual

Revision 0.2

January 29, 2016

Disclaimers

Wi2Wi, Inc. PRODUCTS ARE NOT AUTHORISED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE MANAGING DIRECTOR OF Wi2Wi, Inc.

The definitions used herein are:
a) Life support devices or systems are devices which (1) are intended for surgical implant into the body, or (2) support or sustain life and whose failure to perform when properly used in accordance with the instructions for use provided in the labeling can reasonably be expected to result in a significant injury to the user. b) A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

Wi2Wi does not assume responsibility for use of any of the circuitry described, no circuit patent licenses are implied and Wi2Wi reserves the right at any time to change without notice said circuitry and specifications.
The content of this document is to be treated as strictly confidential and is not to be disclosed, Reproduced or used, except as authorized in writing by Wi2Wi, Inc.

IC RSS Statement:

English:
This device complies with Industry Canada license-exempt RSS standard(s). Operation is subject to the following two conditions:

1) This device may not cause harmful interference;
2) This device must accept any interference received, including interference that may cause undesired operation of the device.

French:
Cet appareil est conforme à Industrie Canada une licence standard RSS exonérés (s). Son fonctionnement est soumis aux deux conditions suivantes:

1) Cet appareil ne doit pas provoquer d'interférences
2) Cet appareil doit accepter toute interférence reçue, y compris les interférences pouvant provoquer un fonctionnement indésirable de l'appareil.

FCC Part 15 Statement:

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions:

1) This device may not cause harmful interference, and
2) This device must accept any interference received, including interference that may cause undesired operation.

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses, and radiates radio frequency energy; and if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

a) Reorient or relocate the receiving antenna.
b) Increase the separation between the equipment and receiver.
c) Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
d) Consult the dealer or an experienced radio/TV technician for help.

The FCC requires the user to be notified that any changes or modifications made to this device that are not expressly approved by Wi2Wi may void the user"s authority to operate the equipment.

Integrator Guidance:

- Only the antenna(s) described in the filings under this FCC ID or equivalent antenna(s) with equal or lesser gain may be used with this transmitter. Any new antenna type, or higher gain antenna would require a Class II permissive change.
- If the operation of the equipment is for portable use (within 20 cm of user), or where co-location configuration use is required; the end product, including the transmitter will require re-evaluation in accordance to the FCC rules.

Labeling:

The final end product must be labeled in a visible area with the following:

- "Contains FCC ID: XXXXXXXX, IC: YYYYYYYYY"

where XXXXXXXX, YYYYYYYYY are the approved FCC/IC ID for the device being installed.

The grantee's FCC/IC ID can be used only when all FCC/IC compliance requirements are met.

Table of Contents

List of Figures

List of Tables:

<u>Revision History:</u>

Revision	Revision Date	Originator	Changes
0.1			
0.2			

1. Introduction

This document describes how to set up the W2SW0025 EVK to evaluate Wi-Fi performance of the W2SW0025 host based module. It also describes how to compile the Linux reference drivers and provides some basic usage example.

1.1 W2SW0025 Module

W2SW0025 is a complete wireless subsystem featuring 802.11 b/g/n WLAN capabilities in a small form factor module. The W2SW0025 module is designed to simplify the process of adding wireless capability without lengthy design cycles or complex RF circuit. It is completely tested for functionality and performance along with coexistence with other wireless standards.
W2SW0025 has been fully optimized for throughput and receive sensitivity using careful design practices Based on world-class silicon from Marvell.

1.2 W2SW0025 EVK Board

The W2SW0025 EVK board is a development platform based on the W2SW0025 with integrated USB and SDIO. This EVB shows the abilities of the W2SW0025 and it is perfect for learning and developing USB-based applications using the W2SW0025. The W2SW0025 evaluation kit includes an evaluation board which can be used as a reference design for the W2SW0025 modules.

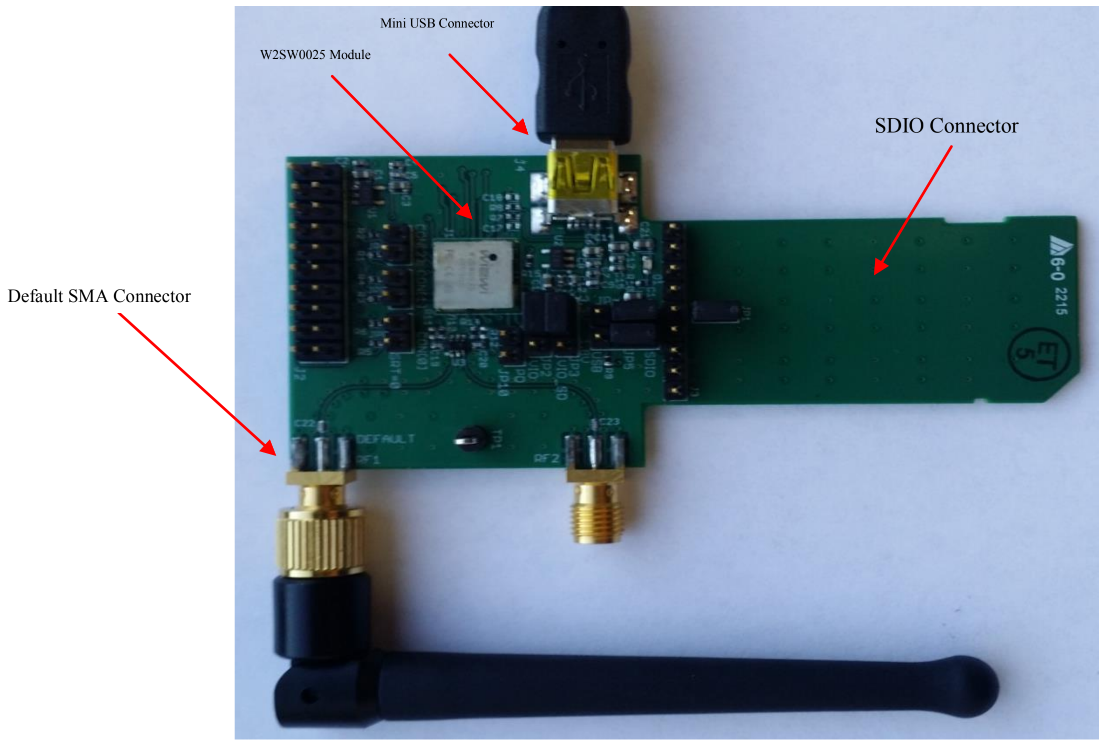


Figure 1: W2SW0025 Evaluation Kit Overview

2. Scope

2.1 Pre-requisites

This section details the pre-requisites for SDK hardware usage. A machine with the following specification:

1. PC with Linux Operating System version x.x.x above
2. At least 2GB of hard disk space available
3. Administrator rights on the machine

2.2 Setup

This section details the system setup to evaluate and use the driver

***** Insert the picture

2.3 Release Package Contents

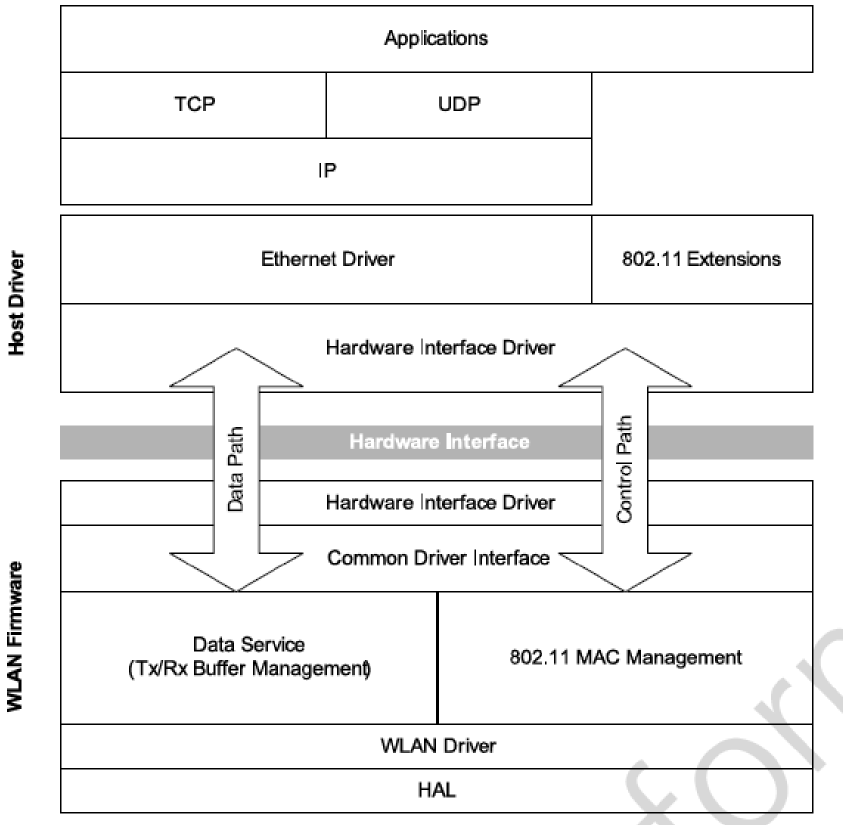
The W2SW0025 Release package is delivered as a tar ball named in the format W2SW0025.LINUX.x.y.z.tgz, where

x.y.z – identifies the software package.

The package contains the following files/folders:

1) Readme.txt
2) Releasenotes.txt
3) Documents
4) WiFi Driver Software and README_WLAN, README_UAP, README_WDT.

2.4 Driver and Firmware Architecture



*** redraw the diagram

2.5 Driver Building

2.5.1 Kernel Configuration

The versions of the W2SW0025 package and the Linux OS that are used for this document are:

- W2SW0025 reference driver version x.x.x
- Linux 3.10.17

The drivers should be able to support Linux kernel versions from 2.6.31 to 3.10 or newer (not verified thoroughly). Older or more recent kernels might require some patches due to changed kernel APIs.

W2SW0025 depend on the MMC/SDIO stack of the Linux kernel, so it must be enabled on the target Linux system, the following must be selected in the kernel configuration "kernel.config" file while compiling the Linux kernel.

```
CONFIG_WIRELESS_EXT=y

CONFIG_WEXT_PRIV=y

CONFIG_CFG80211=y
```

For older kernels which are <3.2, it is advised to use compat-wireless (now named backports) to provide recent versions of the kernel's 802.11 APIs to support all the driver features. In this case,cfg80211 has to be compiled as a module (CONFIG_CFG80211=m).

2.5.2 Compile Time Driver Configuration

The W2SW0025 driver package contains the firmware image, the Wi-Fi driver sources and also a release notes that describes the tested hardware platform, supported features, bug fixes and known limitations of the release. The package comes as several archives that are packed into each other. Follow the steps mentioned below to extract the driver package.

```
# tar xf USB-8801-FC18-X86-14.68.36.p38-C3X14047_B0-app-src.tar

# tar xf USB-8801-FC18-X86-14.68.36.p38-C3X14047_B0-GPL-src.tar

# tar xf USB-8801-FC18-X86-14.68.36.p38-C3X14047_B0-mlan-src.tar
```

The Wi-Fi driver has several compile-time configuration options that can be set in the driver's Makefile. Change to the wlan_src subdirectory and ensure that the following are enabled in "Makefile".

```
# Enable STA mode support

        CONFIG_STA_SUPPORT=y
# Enable uAP mode support
        CONFIG_UAP_SUPPORT=y
# Manufacturing firmware support
        CONFIG_MFG_CMD_SUPPORT=y
```

2.5.3 Building

Follow the below mentioned steps to compile the driver.

1. Goto source code directory wlan_src/. (*** copy all contents from three different directories to GPL directory).
2. make clean
3. make build
4. The driver and utility binaries can be found in ../bin_xxxx directory. The driver code supports Linux kernel up to 3.4.

2.6 WiFi Functional mode Driver Installation

1. Copy firmware image "FwImage/usb8766_uapsta.bin" to "/lib/firmware/mrvl/" directory, create the directory if it doesn't exist.
2. Plug-In SDIO/USB card in SDIO/USB slot in case of using EVB.
3. Connect USB power supply to SDIO/USB EVB.
4. When the module is detected on the SDIO interface, the driver will automatically download the firmware to it, initialize the hardware, and register the network interfaces. The following is the kernel log after inserting the SDIO card.

```
mmc1: new high speed SDIO card at address 0001
vendor=0x02DF device=0x9119 class="0" function=1
rx_work=0 cpu_num=1
Wlan: FW download over, firmwarelen=449964 downloaded 449964
WLAN FW is active
fw_cap_info=0xf03, dev_cap_mask=0xffffffff
```

5. Install WLAN driver, by providing the drv_mode, max_sta_bss, max_uap_bss as input module parameters. The bit settings of drv_mode are,

   Bit 0 : STA
   Bit 1 : uAP
   Bit 2 : WIFIDIRECT

- max_sta_bss: Maximum number of STA BSS (default 1, max 1), sta_name: Name of the STA interface (default: "mlan").
- max_uap_bss: Maximum number of uAP BSS (default 1, max 2), uap_name: Name of the uAP interface (default: "uap")
- max_wfd_bss: Maximum number of WIFIDIRECT BSS (default 1, max 1), wfd_name: Name of the WIFIDIRECT interface (default: "wfd")
- Install cfg80211 driver, for example to install cfg80211 driver

  ```
  modprobe cfg80211
  ```
- For example, to install USB87xx driver,

  ```
  insmod mlan.ko
  insmod usb8766.ko [drv_mode=3] [fw_name=mrvl/usb8766_uapsta.bin]
  ```
- To load driver in STA only mode,

  ```
  insmod mlan.ko
  insmod usb8766.ko drv_mode=1 [fw_name=mrvl/usb8766_uapsta.bin]
  ```
- To load driver in uAP only mode,

  ```
  insmod mlan.ko
  insmod usb8766.ko drv_mode=2 [fw_name=mrvl/usb8766_uapsta.bin]
  ```
- To switch mode between STA only, uAP only and uAPSTA in run time,

  ```
  echo drv_mode=1 > /proc/mwlan/config          // STA mode
  echo drv_mode=2 > /proc/mwlan/config          // uAP mode
  echo drv_mode=3 > /proc/mwlan/config          // uAPSTA mode
  ```

- when the driver is installed properly the following new network interfaces will be created (using *ifconfig* command)

  ```
  mlan0   --      Wi-Fi station mode
  uap0    --      Wi-Fi micro access point mode
  wfd0    --      Wi-Fi Direct
  ```
- To switch mode between STA only, uAP only and uAPSTA in run time,

  ```
  echo drv_mode=1 > /proc/mwlan/config          // STA mode
  echo drv_mode=2 > /proc/mwlan/config          // uAP mode
  echo drv_mode=3 > /proc/mwlan/config          // uAPSTA mode
  ```

6. Uninstall WLAN driver,

   ```
   ifconfig mlanX down
   ifconfig uapX down
   ifconfig wfdX down (*** Check this)
   rmmod usb8xxx
   rmmod mlan
   ```

2.7 MFG mode Driver Installation

1. To load driver with MFG firmware file, use mfg_mode=1 when insmod WLAN driver and specify MFG firmware name if needed.
2. There are some other parameters for debugging purpose etc. Use modinfo to check details.
   - drvdbg=<bit mask of driver debug message control>
   - mac_addr=xx:xx:xx:xx:xx:xx <override the MAC address (in hex)>
   - auto_ds=0|1|2 <use MLAN default | enable auto deepsleep | disable auto deepsleep>
   - ps_mode=0|1|2 <use MLAN default | enable IEEE PS mode | disable IEEE PS mode>
   - max_tx_buf=2048|4096|8192 <maximum AMSDU Tx buffer size>
   - cfg_11d=0|1|2 <use MLAN default | enable 11d | disable 11d>
   - req_fw_nowait=0|1 <use request_firmware API (default) | use request_firmware_nowait API>
   - init_cfg=<init config (MAC addresses, registers etc.) file name>

     e.g. copy init_cfg.conf to firmware directory, init_cfg=mrvl/init_cfg.conf

- cal_data_cfg=<CAL data config file name>
  e.g. copy cal_data.conf to firmware directory, cal_data_cfgrvl/cal_data.conf
- cfg80211_wext=<bit mask of CFG80211 and WEXT control>
  Bit 0: STA WEXT
  Bit 1: uAP WEXT
  Bit 2: STA CFG80211
  Bit 3: uAP CFG80211
- skip_fwdnld=0|1 <enable FW download support (default) | disable FW download support>
- wq_sched_prio: Priority for work queue
- wq_sched_policy: Scheduling policy for work queue
  (0: SCHED_NORMAL, 1: SCHED_FIFO, 2: SCHED_RR, 3: SCHED_BATCH, 5: SCHED_IDLE)
- Please note that, both wq_sched_prio and wq_sched_policy should be provided as module parameters. If wq_sched_policy is (0, 3 or 5), then wq_sched_prio must be 0. wq_sched_prio should be 1 to 99 otherwise.
- usb_aggr=0|1|2 <use MLAN default (disabled) | enable USB aggr | disable USB aggr>

3. WiFi Usage Examples

3.1 WiFi Station Mode

Use the following commands to check the system in WiFi Station Mode.

```
mlanutl mlan0 countrycode IN                # India
mlanutl mlan0 bandcfg 11                    # Enable infra mode with bands bgn
mlanutl mlan0 passphrase "1;ssid=MyAP;passphrase=12345678"
# configuring the Passsord to connect to specific AP
iwpriv mlan0 reassoctrl 1                   #
iwconfig mlan0 essid MyAP                   # Connect to specific AP
udhcpc -i mlan0                             # Request for dynamic IP Address
```

3.2 WiFi Access Point Mode

Use the following commands to check the system in WiFi Access Point mode.

```
./uaputl.exe sys_cfg_ssid wi2wi_25                 # set AP SSID as "wi2wi_25"
./uaputl.exe sys_cfg_channel 1                     # set AP primary channel to 1
./uaputl.exe sys_cfg_2040_coex 0          #0 disable  #1 enable 20/40MHz coexistence
# enable 802.11n mode with short guard interval, 40MHz channel bandwidth:
./uaputl.exe sys_cfg_11n 1 0x116e 3 0 0xff
./uaputl.exe sys_cfg_rates 0xc 0x12 0x18 0x24 0x30 0x48 0x60 0x6c
./uaputl.exe sys_cfg_80211d state 1 country US # enable 802.11d, set country
```

```
# configure encryption:
./uaputl.exe sys_cfg_auth 0
./uaputl.exe sys_cfg_protocol 1 # Open
./uaputl.exe sys_cfg_wpa_passphrase hyderabad # passphrase "hyderabad"
./uaputl.exe sys_cfg_cipher 8 8 # CCMP
./uaputl.exe bss_start # start the AP
ifconfig uap0 192.168.1.1 # Assign IP Address
#Also enable DHCP server.
```

3.3 WiFi Direct Mode

3.4 Data Transfer

Data transfer can be done using iperf commands as shown below:

```
TCP server:     iperf   -s
TCP Client:     iperf   -c      'Dest IP' 'time' 'interval'
```

Note: Underlined arguments are optional

Ex: `iperf -c 192.168.1.1 -t 10 -i 1`

The -t argument specifies the test duration time in seconds, default is 10 secs.

The -i argument indicates the interval in seconds between periodic bandwidth reports.

3.5 Driver Proc and Debug commands

Driver proc and debug commands are explained in " README_MLAN.txt" file which is given with release package.

3.6 Driver Configuration commands

The following commands are used to send additional commands to the Marvell MLAN card via the Linux device driver. The mlanX parameter specifies the network device that is to be used to perform this command on. It could be mlan0, mlan1 etc.

mlanutl is used to configure the additional parameters available for WiF driver. For more details refer " USER MANUAL FOR MLANUTL" in " README_MLAN.txt" file which is given with release package.

4. Configuration of Kernels

4.1 SDIO Stack Options

If SDIO is the interface to the Host processor, it has to be ensured that the SDIO stack related modules are compiled in the kernel. If the SDIO stack modules are not present, follow the steps below to enable SDIO support in the kernel.

1) Navigate to the Linux kernel source folder. This is usually in /usr/src/kernels/linux-<kernel-version>
2) Execute the „make menuconfig„ command to open the Kernel Configuration menu.
3) Scroll down to the “Device Drivers --->” option and hit Enter.
4) In the new menu, scroll down to the “MMC/SD/SDIO card support --->” option and press „M‟ to modularize the “MMC/SD/SDIO card support” feature and hit Enter.
5) In the new menu, press „M‟ to modularize the following options:
    a. MMC block device driver
    b. Secure Digital Host Controller Interface support
    c. SDHCI support on PCI bus
6) Hit the Tab key to select Exit and hit Enter. Repeat this till you are asked whether you want to save the configuration. Select “Yes” and hit Enter. If the above options are already selected, the menuconfig screen will exit immediately.

4.2 Kernel Compilation

Execute the steps listed below to compile the kernel with the above options enabled.

1) Navigate to the kernel source folder.
2) Execute the “make” command.
3) Execute the “make modules_install” command.
4) Execute the “make install” command. This ensures that the customized kernel is installed and the boot loader is updated appropriately.
5) Reboot the system to boot with the customized kernel.

5. Driver building for Embedded Platforms

The platforms supported for the current release are listed below:

1) Freescale i.MX6

The sections below explain the usage of the binaries on these platforms

5.1 Driver Building for i.MX6 SoloLite EVK

5.1.1 Requirements

- W2SW0025 Evaluation Kit contents are as follows:
  - W2SW0025 Module Evaluation Board
  - USB-to-microUSB Cable
  - SDIO Adaptor Cable
  - SPI Adaptor Cable
  - USB Pen Drive
- i.MX 6SoloLite Evaluation Kit. The kit contents are as follows:
  - Board: MCIMX6SLEVK
  - Cables: Micro USB-B-2-USB-Type A male, V2.0
  - Power supply: 100/240 V input, 5 V, 2.4 A output W/AC adaptor
  - Two SD cards: Programmed Android™
- Linux PC with Serial-to-USB drivers installed – this will be used to communicate with the i.MX6 platform.
- Software Requirements
  - Toolchain, BSP and Ubuntu Linux OS package for i.MX6 - Kernel version 3.0.35.
  - W2SW0025 Release Package

5.1.2 Hardware Setup

1) Connect the i.MX6 board to the Linux PC using the USB-to-microUSB cable – the cable has to be connected to port J26 (microUSB) of the board.
2) Connect the SDIO Evaluation Board (EVB) to the i.MX6 board using the SDIO adaptor or USB-to-microUSB cable, depending on which Host Interface is needed.
   a. i.MX6 + SDIO EVB: Connect SDIO Adapter to SD3 port of i.MX6
3) Preparing the MMC Card: An SD/MMC memory card is required to transfer the bootloader and kernel images for initializing the partition table and copy the root file system. This is included in the i.MX6 Evaluation Kit, but is programmed for Android OS. Refer to the i.MX_6SoloLite_EVK_Linux_User's_Guide.pdf document provided by Freescale as part of the L3.0.35_4.1.0_LINUX_MMDOCS documentation package, to prepare the SD/MMC card for Linux OS with kernel version 3.0.35.

5.1.3 Cross Compiling the Driver for i.MX6 SoloLite EVK

5.2 Driver Building for i.MX6 WandBoard

5.2.1 Requirements

- W2SW0025 Evaluation Kit contents are as follows:
    - W2SW0025Module Evaluation Board
    - USB-to-microUSB Cable
    - SDIO Adaptor Cable
    - SPI Adaptor Cable
    - USB Pen Drive
    - USB 5V power supply
- Wandboard Eval Kit. The kit contents are as follows:
    - Cables: One micro A/B-type USB cable, One serial RS232 cable, One RJ45 crossed cable
    - Power supply: Universal input AC/DC power supply
    - Board: Imx6 Wandboard solo board.
    - Display with a VGA interface, such as PC monitor etc...
    - Cables: HDMI to VGA cable, USB to micro USB cable, SDIO to micro SD cable ,
    - 16GB Micro SD card
    - SD card reader
    - W2SW0025 SDIO EVB .
    - Power supply: Supply to VGA Monitor , Universal input AC/DC power supply rated 3V,5A

- Linux PC – this will be used to flash the sd card with wandboard bootloader,bootimage and rootfs.
- Software Requirements
    - Toolchain, BSP and Ubuntu Linux OS package for Wandboard - Kernel version 2.6.34.
    - W2SW0025 Release Package
- Linux PC – this will be used to flash the sd card with wandboard bootloader,bootimage and rootfs.
- Connect the wand board to the Linux PC using the micro A/B-type USB cable and power up the board.

5.2.2 Host PC for Wandboard

5.2.2.1 source package

1. The setup of host environment is provided as example on Fedora Linux

#uname -aLinux cpu307 3.6.10-4.fc18.i686.PAE #1 SMP Tue Dec 11 18:15:08 UTC 2012 i686 i686 i386 GNU/Linux

2. Create a directory

```
#mkdir /home/username/WandBoardBuild
#cd /home/username/WandBoardBuild
```

3. Download the Wand board-sdk-20140519 source package to host PC (Commands provided are for) from the download-link given below

   http://www.wandboard.org/images/downloads/wandboard-sdk-20140519.tar.xz

4. Extract the downloaded wand board package. It will contain u-boot-2013.10-wand boot loader, Linux-3.0.101-4.1.0-wand, docs, firmware and precompiled binaries.

```
#tar xvf wandboard-sdk-20140519.tar.xz
```

5. Download required tool chain for the IMX6 wand board solo.(arm-2009q1-203)

```
#wget https://sourcery.mentor.com/GNUToolchain/package4571/public/arm-none-linux-gnueabi/arm-2009q1-203-arm-none-linux-gnueabi-i686-pc-linux-gnu.tar.bz2

#tar xvjf arm-2009q1-203-arm-none-linux-gnueabi-i686-pc-linux-gnu.tar.bz2
```

   This will create a directory arm-2009q1

5.2.2.2 U-boot compilation procedure

1. Paste the following two lines in the file wandboard-sdk-20140519/u-boot-2013.10-wand/arch/arm/cpu/armv7/mx6/u-boot-spl.lds

```
.ARM.exidx : { *(.ARM.exidx*) } > .sram
  .got : { *(.got*) } > .sram
```

   Copy the above two lines Below these two lines

```
. = ALIGN(4);
.rodata : { *(SORT_BY_ALIGNMENT(.rodata*)) } >.sram
```

2. Go to U-boot-2013.10-wand folder in wand board-sdk-20140519 source package. To compile use the command
   make ARCH=arm CROSS_COMPILE=< give the path to the tool chain in your machine > –j4 wandboard

   Ex: make ARCH=arm CROSS_COMPILE=/work/WandBoardBuild/arm-2009q1/bin/arm-none-linux-gnueabi- -j4 wandboard.

3. After successful compilation, **SPL** and **u-boot.img** binaries will be generated in the current folder.

5.2.2.3 Kernel Compilation Procedure

1. Changes needed for kernel compilation
   a) Make the following changes to board-wand.c file located at

   ```
   #cd </path/to/extracted/wandboard/package/>/linux-3.0.35-wand/arch/arm/mach-mx6/
   ```
   b) Search for the function definition of wand_init_sd and delete(-)/add(+)

   lines as shown below.

   ```
   static void wand_init_sd(void) {
   int i;
   ..................
   ..................

   IMX6_SETUP_PAD( EIM_DA9__GPIO_3_9 );

   /*Add mmc devices in reverse order ,so mmc0 always is boot sd (SD3)*/

       -for(i=2; i>=0; i--) {

                   -imx6q_add_sdhci_usdhc_imx(i, &wand_sd_data[i]);

       -}

                   +imx6q_add_sdhci_usdhc_imx(2, &wand_sd_data[2]);

                   +imx6q_add_sdhci_usdhc_imx(0, &wand_sd_data[0]);

       +}
   ```

   Note: Add the lines in the same sequence as mentioned above.

5.2.2.4 Steps to generate uImage

1. Enter the following commands

   ```
   #cd < /path/to/extracted/wand/board/package/>/linux-3.0.101-4.1.0-wand/

   cp wandboard_defconfig .config

   make ARCH=arm CROSS_COMPILE= < give the path to the tool chain in your machine > menuconfig
   ```

2. Enable wireless extensions sysfs files option present at

   Networking support ------>
   Wireless --------->
   [*] Wireless extensions sysfs files

3. Enable IEEE 802.11 for Host AP

   Device Drivers ------>
   Network device support ----->
   Wireless LAN ----->

   <*>IEEE 802.11 for Host AP (Prism2/2.5/3 and WEP/TKIP/CCMP)

4. make ARCH=arm CROSS_COMPILE= < give the path to the tool chain in your machine > –j4 uImage modules

   Ex:
   make ARCH=arm CROSS_COMPILE=/work/WandBoardBuild/arm-2009q1/bin/arm-none-linux-gnueabi- uImage modules

5. Upon successful compilation, **uImage** binary will be generated in the following path.
   cd "</path/to/extracted/wandboard/package/>/linux-3.0.101-4.1.0-wand/arch/arm/boot".

5.2.2.5 Card Flashing Procedure

1. Micro SD card with 8GB capacity is required for flashing process.
2. Format the MicroSD card and Plug the MicroSD card to hostPC.
3. The IMX6 ROM will look for a bootloader at byte offset 1024 in the SD card. If using the recommended SPL based boot, SPL binary should start here.
4. SPL binary can be copied to SD card by

   cd </path/to/extracted/wandboard/package/>/u-boot-2013.04-wand/
   "sudo dd if=SPL of=/dev/sdX bs=1k seek=1 oflag=dsync"
   (Where **sdX** is the SD card drive detected at the time of plugging SD card on to PC, can be checked by „**fdisk –l**" command)

5. The SPL looks for a u-boot 69KB into SD card, u-boot.img can be copied to SD card by

   cd < /path/to/extracted/wandboard/package/>/u-boot-2013.04-wand/
   "sudo dd if=u-boot.img of=/dev/sdX bs=1k seek=69 oflag=dsync"

6. The default u-boot expects a kernel one megabyte into the card. Hence, the kernel can be installed to SD card by
   Using the following command

   cd <"/path/to/extracted/wandboard/package/>/linux-3.0.101-4.1.0-wand/arch/arm/boot/".

"sudo dd if=uImage of=/dev/sdX bs=1M seek=1"

WARNING: Make sure you use the right device for your SD card, Using the wrong device name can overwrite the content on your hard drive.

7. Make sure that the memory card is not used by any other device. For that give the below command
   "**Umount /dev/sdX1**"

8. Now create a partition on micro SD card by entering following **„fdisk –l"** command.

9. Now enter **fdisk /dev/sdX** command to partition micro SD card.

10. Format the created partition by entering following command
   "mkfs.ext3 /dev/sdX1"

11. Copying root file system to partition on micro SD card
    1. Mount the micro SD card to some mount point.
       "mount /dev/sdX1 /mnt"
    2. Extracting the root file system contents to micro SD card partition
       #tar -xvf ubuntu-14.04.1-minimal-armhf-2015-01-20.tar.xz
       After extracting you will get folder named **"ubuntu-14.04.1-minimal-armhf-2015-01-20"**

       cd </to/that/folder>
       Ex: cd ubuntu-14.04.1-minimal-armhf-2015-01-20.
       It will contain
       1. armhf-rootfs-ubuntu-trusty.tar
       2. image-builder.project
       3. User_password.list.

       Now
       # tar -xvf armhf-rootfs-ubuntu-trusty.tar -C /mnt

    3. After copying unmount the micro SD card partition
       "umount /mnt "

6. References

6.1 Specifications

- IEEE 802.11 b/g/n Wireless LAN Specification
- SDIO HS 4-bit Specification

6.2 Trademarks, Patents and Licenses

- Trademarks: Wi-Fi
- Licenses: 88W88801 Software from Marvell

6.3 Other

- W2SW0025-DEV: Development Kit