**Document For Creating Access point**

**Build the u-boot image**

Export arm to ARCH

=> export ARCH=arm

Export freescale path to PATH

=> export PATH=$PATH:/opt/freescale/usr/local/gcc-4.6.2-glibc-2.13-linaro- multilib-2011.12/fsl-linaro-toolchain/bin

Export Cross compiler path

=> export CROSS\_COMPILE=arm-none-linux-gnueabi-

=> make

**Build the kernel image**

Export arm to ARCH

=> export ARCH=arm

Export freescale path to PATH

=> export PATH=$PATH:/opt/freescale/usr/local/gcc-4.6.2-glibc-2.13-linaro- multilib-2011.12/fsl-linaro-toolchain/bin

Export Cross compiler path

=> export CROSS\_COMPILE=arm-none-linux-gnueabi-

Build Defconfig using below command

=> make imx6s\_defconfig

=> make -j4

=> make uImage

**SD-CARD Flash steps**

=> sudo umount /dev/sdb

=> sudo fdisk /dev/sdb

u [switch the unit to sectors instead of cylinders]

d [repeat this until no partition is reported by the 'p' command ]

n [create a new partition]

p [create a primary partition]

1 [the first partition]

16384 [starting at offset sector #16384, i.e. 8MB, which leaves enough space for the kernel, the boot loader and its configuration data]

<enter>[using the default value will create a partition that spans to the last sector of the medium]

w [ this writes the partition table to the medium and fdisk exits]

Re-insert the SD-card

=> sudo mkfs.ext4 /dev/sdb1

Copying the boot loader image

=> sudo dd if=u-boot.bin of=/dev/sdb bs=512 seek=2 skip=2 conv=fsync

Copying the kernel image

=> sudo dd if=uImage of=/dev/sdb bs=512 seek=2048 conv=fsync

Copying the root file system

=> mkdir /home/manoj/mountpoint

=> sudo mount /dev/sdb1 /home/manoj/mountpoint

Extract rootfs package to certain directory

=> gunzip rootfs.ext2.gz

=> sudo mount -o loop -t ext2 rootfs.ext2 /home/manoj/rootfs

=> cd /home/manoj/rootfs

=> sudo cp -a \* /home/manoj/mountpoint

=> sudo umount /home/manoj/mountpoint

**WIFI Driver Build Procedure**

Open the wlan source directory inside Wifi Source code

=> cd /<path to wlan\_src>/

Export Linux Kernel Source directory to the KERNELDIR

=> export KERNELDIR=/<example:home/IMX6/Linux/Kernel/linux-3.0.35>/

Export arm to ARCH

=> export ARCH=arm

Export the freescale path to PATH

=> export PATH="/opt/freescale/usr/local/gcc-4.6.2-glibc-2.13-linaro-multilib- 2011.12/fsl-linaro-toolchain/bin:$PATH"

Export Cross compiler

=> export CROSS\_COMPILE="arm-none-linux-gnueabi-"

Copy the Patch\_for\_Wifi\_driver patch file to linux Kernel directory and use the below command to add the patch

=> patch -p1 < ../Patch\_for\_Wifi\_driver.patch

Use make command to build driver

=> make *(mlan.ko and sd8xxx.ko driver file will be exist in wlan\_src directory)*

**Copying some needed files to rootfs**

=> cp sd8787\_uapsta.bin /media/(sdcard name)/lib/firmware/mrvl/.

*(if mrvl directory is not there then create a directory and copy to that directory.)*

Wi-Fi driver copied to sdcard

=> cp sd8xxx.ko mlan.ko /media/(sdcardname)/.

=> cp -rf bin\_sd8787/\* /media/(sdcard name)/Install-bz/bin/.

**After booting**

Login : root

change the directory to file system

=> cd /

Initialize the wifi driver

=> insmod mlan0.ko

=> insmod sd8xxx.ko

change directory to uaputl.exe

=> cd Install-zn/bin/

To run DHCP server (In udhcpd.conf file change *interface is uap0*)

=> ./uaputl.exe bss\_start

=> ifconfig uap0 192.168.0.101

=> udhcpd -S /etc/udhcpd.conf

**Creating Access point**

Create Open authentication

=> ./uaputl.exe bss\_stop

=> ./uaputl.exe sys\_cfg\_auth 0 // set Auth Open

=> ./uaputl.exe sys\_cfg\_protocol 1 // None

=> ./uaputl.exe sys\_cfg\_cipher 0 0 // cipher none

=> ./uaputl.exe bss\_start

Create Open authentication with WEP cipher

=> ./uaputl.exe bss\_stop

=> ./uaputl.exe sys\_cfg\_auth 0 // set Auth Open

=> ./uaputl.exe sys\_cfg\_11n 0 // Disable 802.11n

=> ./uaputl.exe sys\_cfg\_protocol 2 // WEP

=> ./uaputl.exe sys\_cfg\_cipher 0 0 // cipher none

=> ./uaputl.exe sys\_cfg\_wep\_key 0 1 55555 // char length should be 5,10,13,26

=> ./uaputl.exe bss\_start

Create Shared authentication with WEP cipher

=> ./uaputl.exe bss\_stop

=> ./uaputl.exe sys\_cfg\_auth 1 // set Auth shared

=> ./uaputl.exe sys\_cfg\_11n 0 // Disable 802.11n

=> ./uaputl.exe sys\_cfg\_protocol 2 // WEP

=> ./uaputl.exe sys\_cfg\_cipher 0 0 // cipher none

=> ./uaputl.exe sys\_cfg\_wep\_key 0 1 55555 // char length should be 5,10,13,26

=> ./uaputl.exe bss\_start

Create WPA-PSK with TKIP encryption

=> ./uaputl.exe bss\_stop

=> ./uaputl.exe sys\_cfg\_protocol 8 // set WPA auth

=> ./uaputl.exe sys\_cfg\_11n 0 // Disable 802.11n

=> ./uaputl.exe sys\_cfg\_cipher 4 4 // TKIP [PAIRWISE\_CIPHER GROUP\_CIPHER]

=> ./uaputl.exe sys\_cfg\_wpa\_passphrase 1234567890

=> ./uaputl.exe bss\_start

Create WPA2-PSK with TKIP encryption

=> ./uaputl.exe bss\_stop

=> ./uaputl.exe sys\_cfg\_protocol 32 // set WPA2 auth

=> ./uaputl.exe sys\_cfg\_11n 0 // Disable 802.11n

=> ./uaputl.exe sys\_cfg\_cipher 4 4 // TKIP [PAIRWISE\_CIPHER GROUP\_CIPHER]

=> ./uaputl.exe sys\_cfg\_wpa\_passphrase 1234567890

=> ./uaputl.exe bss\_start

Create WPA,WPA2-PSK(mixed mode) with TKIP encryption

=> ./uaputl.exe bss\_stop

=> ./uaputl.exe sys\_cfg\_protocol 40 // set WPA/WPA2 auth

=> ./uaputl.exe sys\_cfg\_11n 0 // Disable 802.11n

=> ./uaputl.exe sys\_cfg\_cipher 4 4 // TKIP [PAIRWISE\_CIPHER GROUP\_CIPHER]

=> ./uaputl.exe sys\_cfg\_wpa\_passphrase 1234567890

=> ./uaputl.exe bss\_start

Create WPA-PSK with AES encryption

=> ./uaputl.exe bss\_stop

=> ./uaputl.exe sys\_cfg\_protocol 8 // set WPA auth

=> ./uaputl.exe sys\_cfg\_cipher 8 8 // AES CCMP [PAIRWISE\_CIPHER GROUP\_CIPHER]

=> ./uaputl.exe sys\_cfg\_wpa\_passphrase 1234567890

=> ./uaputl.exe bss\_start

Create WPA2-PSK with AES encryption

=> ./uaputl.exe bss\_stop

=> ./uaputl.exe sys\_cfg\_protocol 32 // set WPA2 auth

=> ./uaputl.exe sys\_cfg\_cipher 8 8 // AES CCMP [PAIRWISE\_CIPHER GROUP\_CIPHER]

=> ./uaputl.exe sys\_cfg\_wpa\_passphrase 1234567890

=> ./uaputl.exe bss\_start

Create WPA,WPA2-PSK(mixed mode) with AES encryption

=> ./uaputl.exe bss\_stop

=> ./uaputl.exe sys\_cfg\_protocol 40 // set WPA/WPA2 auth

=> ./uaputl.exe sys\_cfg\_cipher 8 8 // AES CCMP [PAIRWISE\_CIPHER GROUP\_CIPHER]

=> ./uaputl.exe sys\_cfg\_wpa\_passphrase 1234567890

=> ./uaputl.exe bss\_start

/ / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / / /

**Note:-**

* insmod sd8xxx.ko . It shows error means again rebuild the Wi-Fi driver. When ever you made change in kernel you should rebuild Wi-Fi driver also.
* Bootloader, Kernel ,Rootfs files present in freescale website.
* wlan\_src and sd8787\_uapsta.bin present in marvell website.
* Every Wi-Fi CONFIG commands present in README file inside wlan\_src directory.
* If you set channel 12, if it shows invalid channel we need to change the country code to IN

for that command is ./uaputl.exe sys\_cfg\_80211d country IN, it works channel 12, 13.