

Porting BCM4330/BCM43362 WIFI to Android4.2.2

2015-08-20

The document will introduce all steps for poring BCM4330/BCM43362 WIFI module to freescale android4.2.2 BSP, it includes these contents:

- Hardware & Software Environment
- Hardware Design Based on BCM43362 module
- i.MX6 BSP configuration for WIFI module
- BCM4330/BCM43362 dirver for linux 3.0.35
- Integrated to Android4.2.2

1. Hardware & Software Environment

*Software Platform:

Kernel : Linux 3.0.35

Android: jb4.2.2

*Hardware Platform

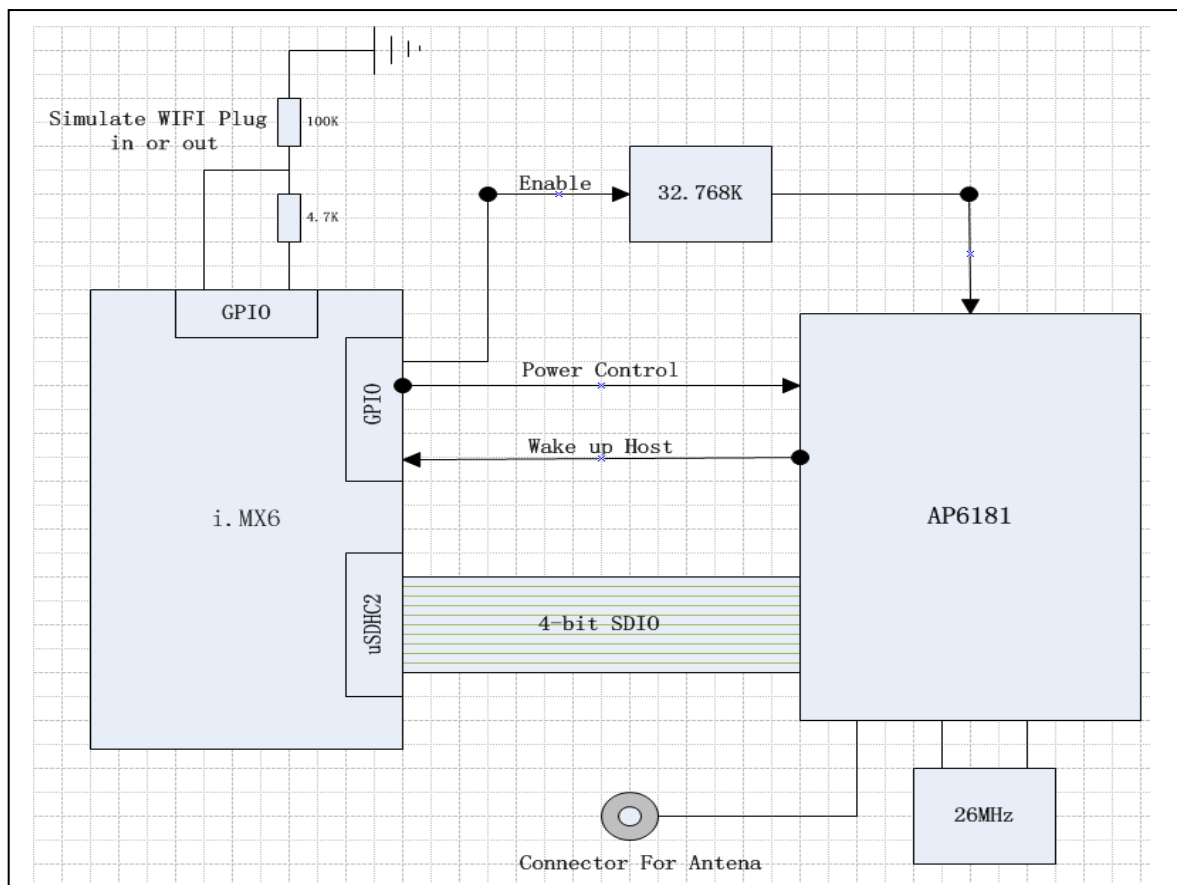
Processor: Freescale i.MX6Q/DL

Memory: DDR3 1GB

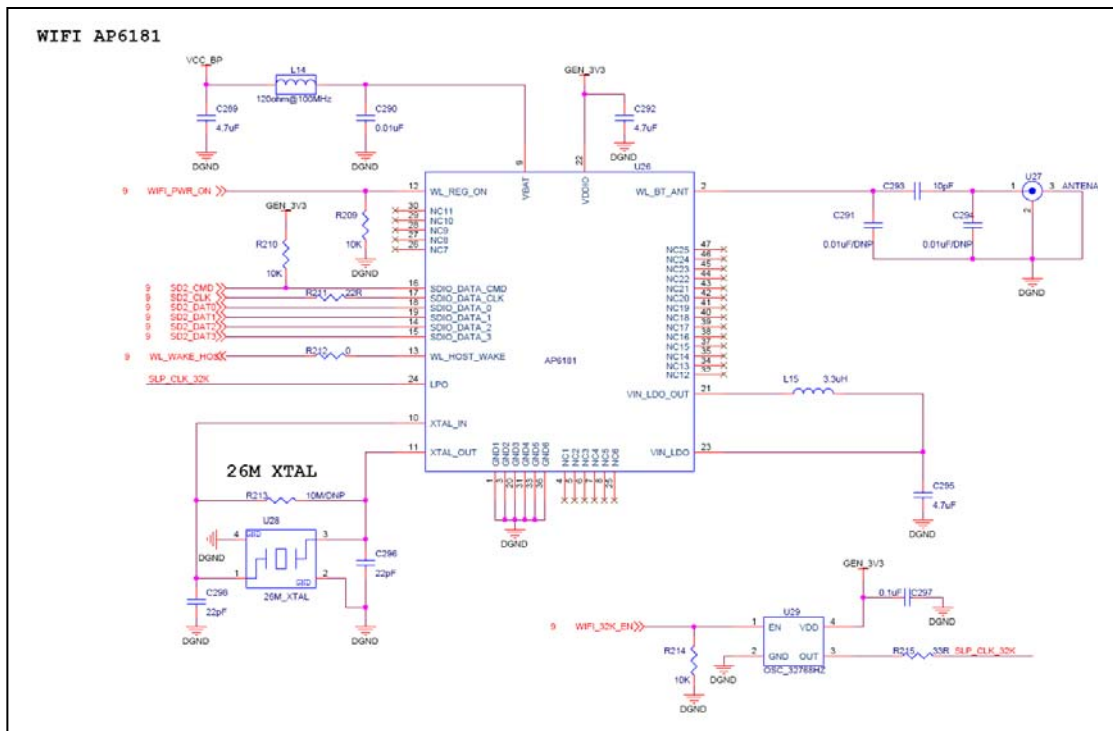
Flash: 8GB Managed Nand

2. Hardware Design Based on BCM43362 module

The following is a diagram for connections between CPU and WIFI module, AP6181 is a WIFI module based on BCM43362 chipset.



The following schematic is for AP6181:



Note:

(1)VCC_BP is 3.6~4.2V.

(2)We used 2 GPIOs to simulate WIFI card's insert or plug out.

3. i.MX6 BSP configuration for WIFI module

BSP files are in path myandroid/kernel_imx/arch/arm/mach-mx6/, we will adjust source code based on board--mx6q_sabresd.c & board-mx6q_sabresd.h.

(1) uSDHC multiplexing

open board-mx6q_sabresd.h, check if SD2 has been multiplexed, static iomux_v3_cfg_t mx6q_sabresd_pads[] = {

```

...
/* USDHC2 - For WIFI*/
MX6Q_PAD_SD2_CLK__USDHC2_CLK,
MX6Q_PAD_SD2_CMD__USDHC2_CMD,
MX6Q_PAD_SD2_DAT0__USDHC2_DAT0,
MX6Q_PAD_SD2_DAT1__USDHC2_DAT1,
MX6Q_PAD_SD2_DAT2__USDHC2_DAT2,
MX6Q_PAD_SD2_DAT3__USDHC2_DAT3,...
}

```

(2)GPIO multiplexing

We need 3 GPIOs to control AP6181: Power Control , Wake up Host control & 32.768K enable control.

In addition, in order to simulate WIFI card hot plug, we used 2 GPIOs to realize it, by this way, it is easy to make uSDHC driver reset.

Let us add multiplexing to board-mx6q_sabresd.h:

```
static iomux_v3_cfg_t mx6q_sabresd_pads[] = {
...
    /* GPIO for WIFI */
    MX6Q_PAD_EIM_A16__GPIO_2_22,    /* WIFI simulate insert or plug in */
    MX6Q_PAD_KEY_COL2__GPIO_4_10,    /* WIFI_CD */
    MX6Q_PAD_ENET_TXD1__GPIO_1_29,    /* WIFI wake up host */
    MX6Q_PAD_ENET_TXD0__GPIO_1_30,    /*WIFI Power control*/
    MX6Q_PAD_GPIO_2__GPIO_1_2,        /*WIFI 32K CLK Ctrl*/ }

```

(3) source code for board--mx6q_sabresd.c

The following source code should be added:

--GPIO definitions

In GPIO definitions, add these 2 lines:

```
#define MX6Q_SABRESD_WL_32K_EN        IMX_GPIO_NR(1, 2)
#define MX6Q_SABRESD_WL_WAKE_HOST    IMX_GPIO_NR(1, 29)
#define MX6Q_SABRESD_WL_PWR_ON       IMX_GPIO_NR(1, 30)
#define MX6Q_SABRESD_WL_PLUG_CTRL    IMX_GPIO_NR(2, 22)
#define MX6Q_SABRESD_WIFI_CD         IMX_GPIO_NR(4, 10)

```

---uSDHC2 data

```
/*SD2 is for WIFI working at 4 bit mode*/
static const struct esdhc_platform_data mx6q_solopi_sd2_data __initconst = {
    .cd_gpio = MX6Q_SABRESD_WIFI_CD,
    .keep_power_at_suspend = 1,
    .support_8bit = 0,
    .delay_line = 0,
    .cd_type = ESDHC_CD_CONTROLLER,
    .runtime_pm = 1,
};

```

---WIFI Power control

These 2 functions will be added to BCM4330/BCM4332 driver:

```
/* Add wifi power control for i.MX6Q & i.MX6DL Solo Pi */
static void bcm_wlan_power_on(int flag)
{
    if(flag==1)
    {
        printk(KERN_INFO "====wifi power on==== \n");
        gpio_set_value(MX6Q_SABRESD_WL_PWR_ON,1);
        msleep(100);
    }
}

```

```

    }
    else
    {
        printk(KERN_INFO "====wifi power on====(flag = %d \n",flag);
        gpio_set_value(MX6Q_SABRESD_WL_PWR_ON,1);
    }
}
EXPORT_SYMBOL(bcm_wlan_power_on);

static void bcm_wlan_power_off(int flag)
{
    if(flag==1)
    {
        printk(KERN_INFO "====wifi power off==== \n");
        gpio_set_value(MX6Q_SABRESD_WL_PWR_ON,0);
        msleep(100);
    }
    else
    {
        printk(KERN_INFO "====wifi power off====(flag = %d \n",flag);
        gpio_set_value(MX6Q_SABRESD_WL_PWR_ON,0);
    }
}
EXPORT_SYMBOL(bcm_wlan_power_off);

```

---Board specific initialization

In function static void __init mx6_sabresd_board_init(void), the following code should exist:

```

static void __init mx6_sabresd_board_init(void)
{
    ...
    imx6q_add_sdhci_usdhc_imx(1, &mx6q_sabresd_sd2_data);...
}

```

4. BCM4330/BCM43362 driver for linux 3.0.35

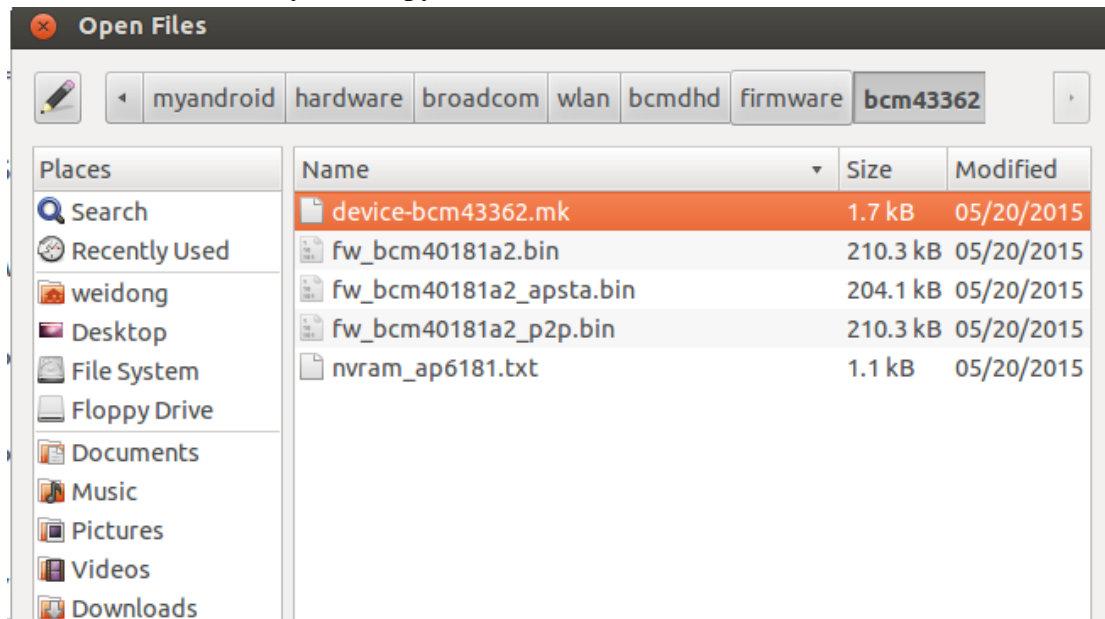
--Copy linux driver to ~/myandroid/kernel_imx/drivers/net/wireless/bcmdhd/

Before copying driver, please check if the directory is empty, if not, delete all file in it, then copy WIFI driver here.

[Note] If customer uses BCM4330 or BCM43362 module, he should ask for its linux drivers & firmware from hardware provider.

--Copy firmware to android

At path ~/myandroid/hardware/broadcom/wlan/bcmdhd/firmware/, create bcm43362 subdirectory, and copy all firmware here.



Then create a file named device-bcm43362.mk, and the following lines:

```
ifeq ($(strip $(WIFI_BAND)),802_11_BG)
BCM_FW_SRC_FILE_STA := fw_bcm40181a2.bin
BCM_FW_SRC_FILE_AP  := fw_bcm40181a2_apsta.bin
BCM_FW_SRC_FILE_P2P := fw_bcm40181a2_p2p.bin
else
BCM_FW_SRC_FILE_STA := fw_bcm40181a2.bin
BCM_FW_SRC_FILE_AP  := fw_bcm40181a2_apsta.bin
BCM_FW_SRC_FILE_P2P := fw_bcm40181a2_p2p.bin
endif

PRODUCT_COPY_FILES += \

hardware/broadcom/wlan/bcmdhd/firmware/bcm43362/$(BCM_FW_SRC_FILE_STA)
:system/etc/firmware/fw_bcmdhd.bin \

hardware/broadcom/wlan/bcmdhd/firmware/bcm43362/$(BCM_FW_SRC_FILE_AP)
:system/etc/firmware/fw_bcmdhd_apsta.bin \
```

```
hardware/broadcom/wlan/bcmdhd/firmware/bcm43362/$(BCM_FW_SRC_FILE_P2P
):system/etc/firmware/fw_bcmdhd_p2p.bin \
    hardware/broadcom/wlan/bcmdhd/firmware/bcm43362/nvram_ap6181.txt:system
/etc/firmware/nvram.txt \
    hardware/broadcom/wlan/bcmdhd/config/wpa_supplicant_overlay.conf:system/et
c/wifi/wpa_supplicant_overlay.conf \
    hardware/broadcom/wlan/bcmdhd/config/p2p_supplicant_overlay.conf:system/et
c/wifi/p2p_supplicant_overlay.conf
```

-- Kconfig in the path should be like the following:

```
config BCMDHD
```

```
    tristate "Broadcom 4329/30 wireless cards support"
```

```
    depends on MMC
```

```
    ---help---
```

```
        This module adds support for wireless adapters based on
        Broadcom 4329/30 chipset.
```

```
        This driver uses the kernel's wireless extensions subsystem.
```

```
        If you choose to build a module, it'll be called dhd. Say M if
        unsure.
```

```
config BCMDHD_FW_PATH
```

```
    depends on BCMDHD
```

```
    string "Firmware path"
```

```
    default "/system/etc/firmware/fw_bcmdhd.bin"
```

```
    ---help---
```

```
        Path to the firmware file.
```

```
config BCMDHD_NVRAM_PATH
```

```
    depends on BCMDHD
```

```
    string "NVRAM path"
```

```
    default "/system/etc/firmware/nvram.txt"
```

```
    ---help---
```

```
        Path to the calibration file.
```

```
config BCMDHD_WEXT
```

```
    bool "Enable WEXT support"
```

```
    depends on BCMDHD && CFG80211 = n
```

```
    select WIRELESS_EXT
```

```
    select WEXT_PRIV
```

```
    help
```

```
        Enables WEXT support
```

```
config DHD_USE_STATIC_BUF
    bool "Enable memory preallocation"
    depends on BCMDHD
    default n
    ---help---
        Use memory preallocated in platform
```

```
config DHD_USE_SCHED_SCAN
    bool "Use CFG80211 sched scan"
    depends on BCMDHD && CFG80211
    default n
    ---help---
        Use CFG80211 sched scan
```

-- Makefile in the path should be like the following:

```
# -DCUSTOMER_HW2 -DOOB_INTR_ONLY -DHW_OOB
-DPROP_TXSTATUS -DCUSTOM_SDIO_F2_BLKSIZE=128
-DGET_CUSTOM_MAC_ENABLE \
#-DMMC_SDIO_ABORT

DHDCFLAGS = -Wall -Wstrict-prototypes -Dlinux -DBCMDRIVER
\
    -DBCMDONGLEHOST -DUNRELEASEDCHIP -DBCMDMA32
-DBCMFILEIMAGE \
    -DDHDTHREAD -DDHD_DEBUG -DSDTEST -DBDC -DTOE
\
    -DDHD_BCMEVENTS -DSHOW_EVENTS -DBCMDBG
\
    -DBCMSDIO -DBCMLXSDMMC -DBCMPPLATFORM_BUS
-DSDIO_ISR_THREAD -DWLP2P \
    -DWIFI_ACT_FRAME -DARP_OFFLOAD_SUPPORT
\
    -DKEEP_ALIVE -DPKT_FILTER_SUPPORT \
    -DEMBEDDED_PLATFORM -DENABLE_INSMOD_NO_FW_LOAD
-DPNO_SUPPORT \
    -DDHD_USE_IDLECOUNT -DSET_RANDOM_MAC_SOFTAP
-DROAM_ENABLE -DVSDB \
    -DWL_CFG80211_VSDB_PRIORITIZE_SCAN_REQUEST
\
    -DESCAN_RESULT_PATCH -DHT40_GO -DPASS_ARP_PACKET
-DSDIO_CRC_ERROR_FIX \
    -DDHD_DONOT_FORWARD_BCMEVENT_AS_NETWORK_PKT
-DAMPDU_HOSTREORDER \
    -DSUPPORT_PM2_ONLY -DWL_SDO\
```



```
-Idrivers/net/wireless/bcmdhd -Idrivers/net/wireless/bcmdhd/include
```

```
DHDOFILES = aiutils.o bcmsdh_sdmmc_linux.o dhd_linux.o siutils.o bcmutils.o \
dhd_linux_sched.o dhd_sdio.o bcmwifi_channels.o bcmevent.o hndpmu.o \
bcmsdh.o dhd_cdc.o bcmsdh_linux.o dhd_common.o linux_osl.o \
bcmsdh_sdmmc.o dhd_custom_gpio.o sbutils.o wldev_common.o wl_android.o
```

```
obj-$(CONFIG_BCMDHD) += bcmdhd.o
bcmdhd-objs += $(DHDOFILES)
ifneq ($(CONFIG_WIRELESS_EXT),)
bcmdhd-objs += wl_iw.o
DHDCFLAGS += -DSOFTAP -DWL_WIRELESS_EXT -DUSE_IW
endif
ifneq ($(CONFIG_CFG80211),)
bcmdhd-objs += wl_cfg80211.o wl_cfgp2p.o wl_linux_mon.o dhd_cfg80211.o
DHDCFLAGS += -DWL_CFG80211 -DWL_CFG80211_STA_EVENT
-DWL_ENABLE_P2P_IF
DHDCFLAGS += -DCUSTOM_ROAM_TRIGGER_SETTING=-65
DHDCFLAGS += -DCUSTOM_ROAM_DELTA_SETTING=15
DHDCFLAGS += -DCUSTOM_KEEP_ALIVE_SETTING=28000
DHDCFLAGS += -DCUSTOM_PNO_EVENT_LOCK_xTIME=7
endif
ifneq ($(CONFIG_DHD_USE_SCHED_SCAN),)
DHDCFLAGS += -DWL_SCHED_SCAN
endif
EXTRA_CFLAGS = $(DHDCFLAGS)
ifeq ($(CONFIG_BCMDHD),m)
EXTRA_LDFLAGS += --strip-debug
endif
```

--Adding controls to WIFI driver

(1) Open `dhd_linux.c`, and GPIO definitions at the beginning.

```
#define IMX_GPIO_NR(bank, nr)                (((bank) - 1) * 32 + (nr))
#define MX6Q_SABRESD_WL_PLUG_CTRL          IMX_GPIO_NR(2, 22)
#define MX6Q_SABRESD_WIFI_CD              IMX_GPIO_NR(4, 10)
```

(2) In function `dhd_module_init(void)` of the file

Add :

```
static int __init dhd_module_init(void)
{
...
DHD_TRACE((" %s: Enter\n", __FUNCTION__));
wl_android_init();
```

```

gpio_set_value(MX6Q_SABRESD_WL_PLUG_CTRL, 0);
msleep(2);
...
}

```

(3) In function `dhd_module_cleanup(void)` of the file

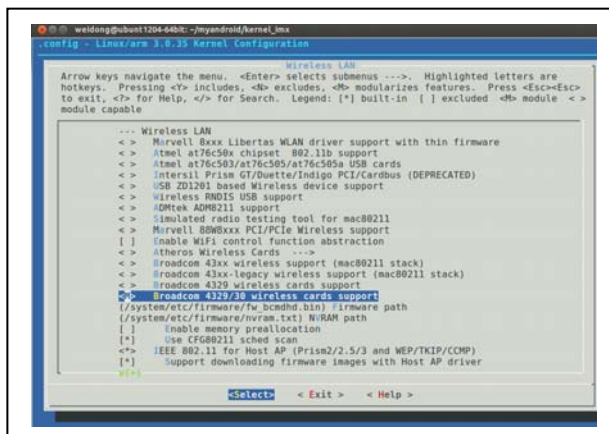
Add:

```

static void __exit dhd_module_cleanup(void)
{
...
    wl_android_exit();
    gpio_set_value(MX6Q_SABRESD_WL_PLUG_CTRL, 1);
...
};

```

--Running make menuconfig and Selecting WIFI module



(4) Compiling WIFI driver

At first, Using `make uImage` to compile linux kernel, then using `make modules` to compile all modules including wifi driver.

After compiling it, we can get wifi driver named `bcmhdh.ko` in the driver's path.

```

weidong@ubuntu1204-64bit:~/myandroid/kernel_imx$ ls ./drivers/net/wireless/bcmhdh
autils.c          bcmwdh_linux.o    bcmwifi_channels.o  dhd_proto.h      Kconfig-        uamp_api.h       wl_dbg.h
autils.o          bcmwdh.o          built-in.o          dhd_custom_gpio.c  dhd_sdio.c      linux_osl.c      wl_android.c    wldev_common.c
bcmhdh.ko         bcmwdh_sdmmc.c   dhd_bta.c          dhd_custom_gpio.c-  dhd_sdio.c-    linux_osl.o      wl_android.c-  wldev_common.h
bcmhdh.mod.c     bcmwdh_sdmmc_linux.c  dhd_bta.h         dhd_custom_gpio.o  dhd_sdio.o     Makefile         wl_android.h    wldev_common.o
bcmhdh.mod.o     bcmwdh_sdmmc_linux.c-  dhd_bus.h         dhd_dbg.h         dhd_wlfc.h     Makefile-        wl_android.o   wl_iw.c
bcmhdh.o         bcmwdh_sdmmc_linux.o  dhd_cdc.c         dhd.h             dngl_stats.h   modules.order    wl_cfg80211.c  wl_iw.h
bcmevent.c       bcmwdh_sdmmc.o    dhd_cdc.o         dhd_linux.c       dngl_wlhdr.h   sbutils.c        wl_cfg80211.h  wl_iw.o
bcmevent.o       bcmutils.c        dhd_cfg80211.c    dhd_linux.c-      hndpmu.c       sbutils.o        wl_cfg80211.o  wl_linux_mon.c
bcmwdh.c         bcmutils.o        dhd_cfg80211.h    dhd_linux.o       hndpmu.o       siutils.c        wl_cfgp2p.c    wl_linux_mon.o
bcmwdh_linux.c  bcmwifi.c         dhd_cfg80211.o    dhd_linux_sched.c  include         siutils.o        wl_cfgp2p.h    wl_linux_mon.o
bcmwdh_linux.c- bcmwifi_channels.c  dhd_common.c      dhd_linux_sched.o  Kconfig        siutils_priv.h   wl_cfgp2p.o
weidong@ubuntu1204-64bit:~/myandroid/kernel_imx$

```

(2) Debug WIFI module on linux command line

See link: <https://community.freescale.com/message/544259#544259>

5. Integrated to Android4.2.2

(1) WIFI HAL

WIFI HAL is at path `~/myandroid/hardware/libhardware_legacy/wifi/`, so copy `wifi_bcm.c` here.

Then add the following lines to Android.mk of the current path.

```
ifeq ($(BOARD_WLAN_VENDOR),INTEL)
    LOCAL_SRC_FILES += wifi/wifi_intel.c
    LOCAL_C_INCLUDES +=
$(LOCAL_PATH)/../external/wpa_supplicant_8/src/common
else
    ifeq ($(BOARD_WLAN_VENDOR),BROADCOM)
        LOCAL_SRC_FILES += wifi/wifi_bcm.c
        LOCAL_C_INCLUDES +=
$(LOCAL_PATH)/../external/wpa_supplicant_8/src/common
    else
        LOCAL_SRC_FILES += wifi/wifi.c
        LOCAL_C_INCLUDES +=
$(LOCAL_PATH)/../external/wpa_supplicant_8/src/common
    endif
endif
```

(2) BoardConfig.mk

BoardConfig.mk for evaluation board of freescale is at path ~/myandroid/device/fsl/sabresd_6dq/(take the board as an example).

In the file , the following lines should be added:

```
# Wifi
BOARD_WLAN_VENDOR           := BROADCOM
BOARD_WLAN_DEVICE_REV       := BCM43362
# for broadcom BCM43362
BOARD_WPA_SUPPLICANT_DRIVER := NL80211
WPA_SUPPLICANT_VERSION      := VER_0_8_X
BOARD_WPA_SUPPLICANT_PRIVATE_LIB := lib_driver_cmd_bcmdhd
BOARD_HOSTAPD_DRIVER        := NL80211
BOARD_HOSTAPD_PRIVATE_LIB  := lib_driver_cmd_bcmdhd
BOARD_WLAN_DEVICE           := bcmdhd
WIFI_DRIVER_FW_PATH_PARAM   := "/sys/module/bcmdhd/parameters/firmware_path"
WIFI_DRIVER_MODULE_PATH     := "/system/lib/modules/bcmdhd.ko"
WIFI_DRIVER_MODULE_NAME     := "bcmdhd"
WIFI_DRIVER_FW_PATH_STA     := "/system/etc/firmware/fw_bcmdhd.bin"
WIFI_TEST_INTERFACE         := "sta"
WIFI_DRIVER_MODULE_ARG      := "iface_name=eth1
firmware_path=/system/etc/firmware/fw_bcmdhd.bin nvram_path=/system/etc/firmware/nvram.txt"
WIFI_TEST_INTERFACE         := "sta"
```

(3) include device-bcm43362.mk in sabresd_6dq.mk

Open myandroid/device/fsl/imx6/sabresd_6dq.mk, and add the lines:

```
BOARD_WLAN_DEVICE_REV := bcm4330_b2
WIFI_BAND              := 802_11_BG
$(call inherit-product-if-exists,
```

hardware/broadcom/wlan/bcmdhd/firmware/bcm43362/device-bcm43362.mk)

(4)init.rc

Add the following lines before "# Prepare for bluetooth" :

```
# Prepare for wifi
  chmod 777 /data/misc/keystore
  mkdir /data/misc/dhcp 0770 dhcp dhcp
  chown dhcp dhcp /data/misc/dhcp
  mkdir /system/etc/wifi 0777 wifi wifi
  chmod 777 /system/etc/wifi
  chmod 777 /system/etc/wifi/wpa_supplicant.conf
  chown wifi wifi /system/etc/wifi/wpa_supplicant.conf
  mkdir /data/misc/wifi 0777 wifi wifi
  mkdir /data/misc/wifi/sockets 0770 wifi wifi
  chmod 777 /data/misc/wifi
  chmod 777 /data/misc/wifi/wpa_supplicant.conf
  chmod 777 /data/misc/wifi/p2p_supplicant.conf
  chown wifi wifi /data/misc/wifi
  chown wifi wifi /data/misc/wifi/wpa_supplicant.conf
# wpa_supplicant socket (unix socket mode)
  mkdir /data/system/wpa_supplicant 0777 wifi wifi
  chmod 0777 /data/system/wpa_supplicant
  chown wifi wifi /data/system/wpa_supplicant
```

(5)init.freescale.rc

```
  Start wpa_supplicant service & htcpd
  setprop wlan.interface "eth1"
  setprop wifi.interface "eth1"
#Script for Starting wifi BCM43362

service p2p_supplicant /system/bin/wpa_supplicant -ieth1 -Dnl80211
-c/data/misc/wifi/wpa_supplicant.conf -N \
    -ip2p0 -Dnl80211
-c/data/misc/wifi/p2p_supplicant.conf -e/data/misc/wifi/entropy.bin \
    -puse_p2p_group_interface=1

  class main
  socket wpa_eth1 dgram 660 wifi wifi
  disabled
  oneshot

service wpa_supplicant /system/bin/wpa_supplicant -ieth1 -Dnl80211 \
    -c/data/misc/wifi/wpa_supplicant.conf \
    -e/data/misc/wifi/entropy.bin

  class main
```

```
socket wpa_eth1 dgram 660 wifi wifi
disabled
oneshot
```

```
service dhcpcd_eth1 /system/bin/dhcpcd -aABDKL
class main
disabled
oneshot
```

```
service dhcpcd_p2p /system/bin/dhcpcd -aABKL
class main
disabled
oneshot
```

```
service dhcpcd_bnep0 /system/bin/dhcpcd -ABKL
class main
disabled
oneshot
```

```
service dhcpcd_eth0 /system/bin/dhcpcd -ABDKL
class main
disabled
oneshot
```

```
service iprenew_eth1 /system/bin/dhcpcd -n
class main
disabled
oneshot
```

```
service iprenew_p2p /system/bin/dhcpcd -n
class main
disabled
oneshot
```

```
service iprenew_bnep0 /system/bin/dhcpcd -n
class main
disabled
oneshot
```

(6) system/ect/wifi/wpa_supplicant.conf

in the file, the contents should be like following:

```
update_config=1
ctrl_interface=DIR=/data/system/wpa_supplicant GROUP=wifi
#ctrl_interface=eth1
```

```
eapol_version=1
ap_scan=1
fast_reauth=1
p2p_listen_channel=11
p2p_listen_reg_class=81
```

(7) The path for wifi driver

By default, system/lib/, there is no "module" sub-directory, we can create one manually, then copy bcmhdh.ko here, see following:

```
weidong@ubuntu1204-64bit:~/myandroid/out/target/product/mx6q_solopi/system/lib/modules$ ls
bcmhdh.ko
weidong@ubuntu1204-64bit:~/myandroid/out/target/product/mx6q_solopi/system/lib/modules$ █
```

Freescale TICS Team
Weidong.sun
2015-08-20