Application Note

Quick Start Guide for ATWILC3000 on SAMA5D4-EK Android platform

AN-XXXX

Prerequisites

- **Hardware Prerequisites**
  - SAMA5D4EK board
  - Atmel WILC3000 evaluation board
  - Micro USB Cable (TypeA / MicroB)
  - USB to Serial Adaptor (for DBGU port)
- **Build Prerequisites**
  - Linux Host PC
  - Android Software Package

Introduction

The WILC3000 is a single chip IEEE 802.11 b/g/n RF, baseband, MAC, Bluetooth 4.0 and FM receiver optimized for low-power mobile applications. The WILC3000 utilizes highly optimized 802.11 – Bluetooth coexistence protocols. It provides multiple peripheral interfaces including UART, SPI, I2C and SDIO.

This quick start guide describes how to integrate the Atmel WILC3000 evaluation board via SDIO for Wi-Fi and USART for Bluetooth in SAMA5D4EK Android platform. The following links also are available to get more information on Atmel wireless drivers, Android/Linux kernel and prebuilt images.

- Atmel SAMA5D4-EK: http://www.atmel.com/tools/sama5d4-ek.aspx.
- Atmel Linux For SAM Site: http://www.at91.com/linux4sam
- Android4sam: http://www.at91.com/android4sam/bin/view/Android4SAM/
- Atmel ATWILC3000 Site: https://github.com/atwilc3000

Table of Contents

1. ATWILC3000 Linux Software Package

The latest WILC3000 is available on the AtmelSmartConnect. Download the ATWILC3000 driver and firmware.

```
git clone https://github.com/atwilc3000/driver.git
git cone https://github.com/atwilc3000/firmware.git
```

For more information, visit the GitHub wiki pages: https://github.com/atwilc3000/driver/wiki

2. SAMA5D4EK Software Package

This chapter describes how to get the SAMA5D4EK Android SDK and build the images. The Android platform for AT91 is maintained at the Android4Sam. See the following instructions to get the SDK and build images. This instruction is based on NAND flash boot mode.

2.1 Download Android source code

```
$ mkdir android4sam_v4.4_rc2
$ cd android4sam_v4.4_rc2
$ repo init -u git://github.com/Android4SAM/platform_sammanifest.git -b
android4sam_v4.4_rc2
$ repo sync
```

2.2 Configure and build Android source code

```
$ . build/envsetup.sh
$ lunch sama5d4-eng
$ make
```

2.3 Generate Android image

```
$ mkubi_image -b sama5d4
```

Find **system_ubifs-SAMA5D4-ANDROID-4.4.2_r2.img** and **userdata_ubifs-SAMA5D4-ANDROID-4.4.2_r2.img** in the Android root directory if successful.

2.4 Download Kernel source code

```
$ git clone git://github.com/Android4SAM/linux-at91.git
$ cd linux-at91
$ git checkout -b linux-at91 Android4sam_v4.4_rc2
```

2.5 Configure and build

First copy the root directory in the Android source code directory.

```
$ cp –r <android-working-dir>/out/target/product/sama5d4/root ../<linix-working-dir>
```

Then, issue the following commands to build the kernel, modules and dtbs.

```
$ make mrproper
$ make ARCH=arm sama5_android_defconfig
$ make ARCH=arm CROSS_COMPILE=(path_to_cross-compiler/cross-compiler-prefix-) zImage
$ make ARCH=arm CROSS_COMPILE=(path_to_cross-compiler/cross-compiler-prefix-)
$ make ARCH=arm CROSS_COMPILE=(path_to_cross-compiler/cross-compiler-prefix-) dtbs
```

3. Flash images

Refer to here to flash prebuilt images and new images built with the Android and kernel source codes in the corresponding board.

3.1 Prebuilt images

The prebuilt images are provided in the Android4Sam. Download the following demo package for NAND boot.

- SAMA5D4EK: Android-4.4.2_r2-sama5d4-nandboot-rc2.tgz

For SAMA5D4EK board, push PB4 (RESET) while pushing PB3. Then, the boards go to boot mode to flash the prebuilt image. Run the batch file, ***sama5d4_nandflash.bat*** in Windows. For Linux, run the script file like ***sama5d4_nandflash.sh***.

3.2 Engineering images built with source codes

This section describes how to flash engineering images to the SAMA5D4EK board. If Android and kernel are successfully built, the following outputs are generated.

- <android-working-dir>/system_ubifs-SAMA5D4-ANDROID-4.4.2_r2.img
- <android-working-dir>/userdata_ubifs-SAMA5D4-ANDROID-4.4.2_r2.img
- <kernel-working-dir>/linux-at91/arch/arm/boot/zImage
- <kernel-working-dir>/linux-at91/arch/arm/boot/dts/sama5d4ek.dtb

All of outputs should be downloaded to the target board. Run the batch or script file as done with prebuilt image.

4. Bring up ATWILC3000

This section describes how to integrate the WILC3000 driver into the SAMA5D4EK Android platform. The patch file for the SAMA5D4EK is available in https://github.com/atwilc3000/patch. The following sections will briefly explain what to port the WILC3000 into the SAMA5D4EK Android platform.

4.1 Android: /device/atmel/sama5d4/BoardConfig.mk

The ATWILC3000 WLAN and Bluetooth are enabled in android with the following configuration in the BoardConfig.mk file.

```
BOARD_WIFI_VENDOR := atmel
ifeq ($(BOARD_WIFI_VENDOR), atmel)
        CONFIG_DRIVER_WEXT :=y
        BOARD_WPA_SUPPLICANT_DRIVER := NL80211
        WPA_SUPPLICANT_VERSION      := VER_0_8_X
        BOARD_WPA_SUPPLICANT_PRIVATE_LIB := lib_driver_cmd_nmc
        BOARD_HOSTAPD_DRIVER        := NL80211
        BOARD_HOSTAPD_PRIVATE_LIB   := lib_driver_cmd_nmc

        WIFI_FIRMWARE_LOADER              := ""

        BOARD_WLAN_DEVICE := wilc3000

        WIFI_DRIVER_FW_PATH_STA := "AUTO"
        WIFI_DRIVER_FW_PATH_AP  := "AUTO"
        WIFI_DRIVER_FW_PATH_P2P := "AUTO"

        BOARD_HAVE_BLUETOOTH :=true
        BOARD_HAVE_BLUETOOTH_ATMEL := true
        SW_BOARD_HAVE_BLUETOOTH_NAME := wilc3000
endif
```

4.2 Android: /device/atmel/common/config/Android_Copy.mk

```
PREBUILD_FIRMWARE := wilc3000

ifeq ($(PREBUILD_FIRMWARE), wilc3000)
PRODUCT_COPY_FILES += \
        $(LOCAL_PATH)/wilc3000.ko:system/lib/modules/wilc3000.ko
endif
```

4.3 Android: /device/atmel/sama5d4/device.mk

The Bluetooth feature is enabled with the following configuration in the device.mk file.

```
PRODUCT_COPY_FILES += \
        frameworks/native/data/etc/android.hardware.bluetooth.xml:system/etc/permissions/android.hardware.bluetooth.xml \
        frameworks/native/data/etc/android.hardware.bluetooth_le.xml:system/etc/permissions/android.hardware.bluetooth_le.xml \
```

4.4 Android: /device/atmel/sama5d4/init.sama5-pda.rc

This file sets owner, group and also permission for the WLAN and Bluetooth like the following configuration.

```
insmod /system/lib/modules/wilc3000.ko
chown bluetooth net_bt_stack /dev/at_bt_pwr
#UART device
chmod 0660 /dev/ttyS2
chown bluetooth net_bt_stack /dev/ttyS2
```

The WPA supplicant is also configured with the following configuration for WLAN Station, AP and P2P mode.

```
service wpa_supplicant /system/bin/logwrapper /system/bin/wpa_supplicant -dd \
        -iwlan0 -Dnl80211 -c/data/misc/wifi/wpa_supplicant.conf \
        -O/data/misc/wifi/sockets \
        -e/data/misc/wifi/entropy.bin -g@android:wpa_wlan0
        class main
        socket wpa_wlan0 dgram 660 wifi wifi
        disabled
        oneshot

service p2p_supplicant /system/bin/logwrapper /system/bin/wpa_supplicant -dd \
        -ip2p0 -Dnl80211 -c/data/misc/wifi/p2p_supplicant.conf \
        -e/data/misc/wifi/entropy.bin -N \
        -iwlan0 -Dnl80211 -c/data/misc/wifi/wpa_supplicant.conf \
        -O/data/misc/wifi/sockets \
        -g@android:wpa_wlan0
        class main
        socket wpa_wlan0 dgram 660 wifi wifi
        disabled
        oneshot
```

4.5 Android: /hardware/atmel/sama5dx/Android.mk

```
modules := libagl gralloc hwcomposer audio liblights hardwareloader camera libbt wlan
```

4.6 Kernel: /arch/arm/boot/dts/sama5d4.dtsi

The SAMA5D4EK supports USART DMA with adding the followings in the USART2 section.

```
usart2: serial@fc008000 {
        compatible = "atmel,at91sam9260-usart";
        reg = <0xfc008000 0x100>;
        interrupts = <29 IRQ_TYPE_LEVEL_HIGH 5>;
        atmel,use-dma-rx;
        atmel,use-dma-tx;
        dmas = <&dma1
                (AT91_XDMAC_DT_MEM_IF(0) | AT91_XDMAC_DT_PER_IF(1))
                (AT91_XDMAC_DT_PERID(16) | AT91_XDMAC_DT_DWIDTH(0x0))>,
                <&dma1
                (AT91_XDMAC_DT_MEM_IF(0) | AT91_XDMAC_DT_PER_IF(1))
                (AT91_XDMAC_DT_PERID(17) | AT91_XDMAC_DT_DWIDTH(0x0))>;
        dma-names = "tx", "rx";
        pinctrl-names = "default";
        pinctrl-0 = <&pinctrl_usart2 &pinctrl_usart2_rts &pinctrl_usart2_cts>;
        status = "disabled";
};
```

The USART2 conflicts with ISI in ISI_VSYNC and HSYNC so it is disabled. However, the WILC3000 uses USART2 for RTS and CTS. So, the ISI should be disabled in the sama5d4ek.dts file.

4.7 Kernel:/arc/ar,/boot/dts/sama5d4.dts

The USART2 should be enabled in this device tree for the WILC3000 Bluetooth while the ISI should be disabled.

```
isi: isi@f0008000 {
        status = "disabled"; /* For RTS and CTS, wilc3000 uses USART2 disabling ISI. */
};
```

The following statement should be included to enable USART2.

```
usart2: serial@fc008000 {
        status = "okay";
};
```

4.8 Kernel: /arch/arm/configs/sama5_android_defconfig

The followings should be added in the sama5_android_defconfig to support WILC3000 via SDIO and Bluetooth via USART interface.

```
CONFIG_ATMEL_SMARTCONNECT=y
CONFIG_WILC3000=m
CONFIG_WILC3000_SDIO=y
# Bluetooth device drivers
CONFIG_BT_HCIBTUSB=y
CONFIG_BT_HCIBTSDIO=y
CONFIG_BT_HCIUART=y
CONFIG_BT_HCIUART_H4=y
CONFIG_BT_HCIUART_BCSP=y
CONFIG_BT_HCIUART_LL=y
CONFIG_BT_HCIUART_3WIRE=y
```

4.9 Kernel: /drivers/net/wireless/Makefile, Kconfig

The followings should be added to build WILC3000 driver in kernel tree.

```
source "drivers/net/wireless/atmel/Kconfig"
```

```
obj-$(CONFIG_ATMEL_SMARTCONNECT) += atmel/
```

Create **atmel** directory in the wireless directory if not available. Then, copy the wilc3000 kernel driver into the **atmel** directory.

4.10 Kernel: wilc3000.ko

Building the kernel generates **wilc3000.ko** in the ***{kernel-working-dir}/drivers/net/wireless/atmel/wilc3000***. It should be placed in the ***{android-working-dir}/device/atmel/common/config/.*** Then, build Android again to make new images. Make sure copy two android images also in the kernel root directory again and make new kernel zImage also. Flash four outputs as described in section 4.

5. Hardware Consideration

This section shows how to connect the ATWILC3000 EVB to SAMA5D4EK for WiFi and Bluetooth. The ATWILC3000 should be connected to SAMA5D4EK via SDIO for WLAN and USART for Bluetooth.

5.1 Bluetooth

The HCI UART transport layer uses the following configurations:

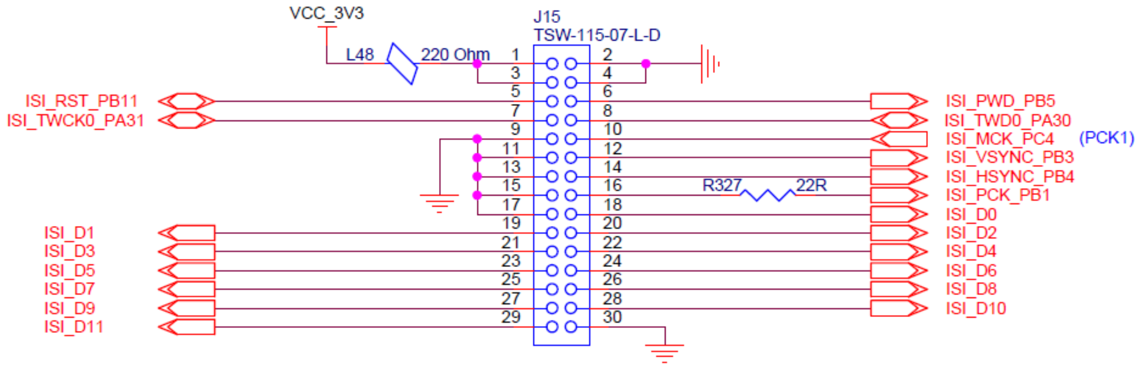
- data length: 8 bits
- parity: no parity
- stop bit: 1stop bit
- flow control: RTS/CTS
- baud rate: vendor specific

- flow-off response time: vendor specific

The local RXD should be connected to the remote TXD and the local RTS should be connected to the remote CTS and vice versa.

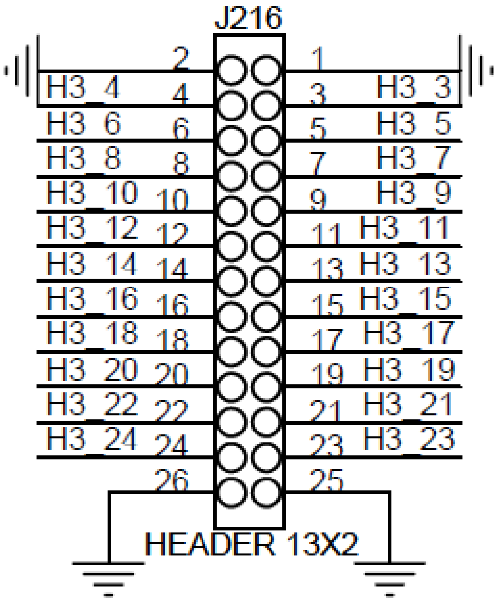
For Bluetooth, the USART2 of SAMA5D4EK is connected to the ATWILC3000 EVB. However, USART2 conflicts with the ISI on the SAMA5D4EK board. So, it's required to disable the ISI and enable the USART2 in the device tree file for the ATWILC3000 demo. Refer to the section 5.6 and 5.7. The following figure is I/O expansion, J15. The expansion, J15 has the TXD2 on pin 6, RXD2 on pin 14, RTS2 on pin 5 and CTS2 on pin 12.

Figure 1 SAMA5D4EK I/O Expansion



The TXD2 on the SAMA5D4EK should be connected to the RXD on the ATWILC3000 EVB, J216 pin18. In contrast, the RXD2 on the SAMA5D4EK should be connected to the TXD on the ATWILC3000 EVB, J216 pin19. The RTS2 on the SAMA5D4EK should be connected to the CTS on the ATWILC3000 EVB, J216 pin16 and the CTS2 should be connected to the RTS on the ATWILC3000 EVB, J216 pin24. Make sure open J227, J805 and J901 on ATWILC3000 EVB before running the Bluetooth.

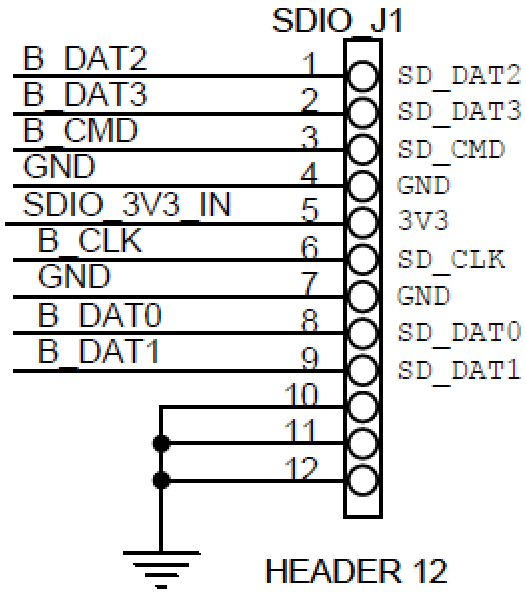
Figure 2 ATWILC3000 EVB J216



5.2 WiFi

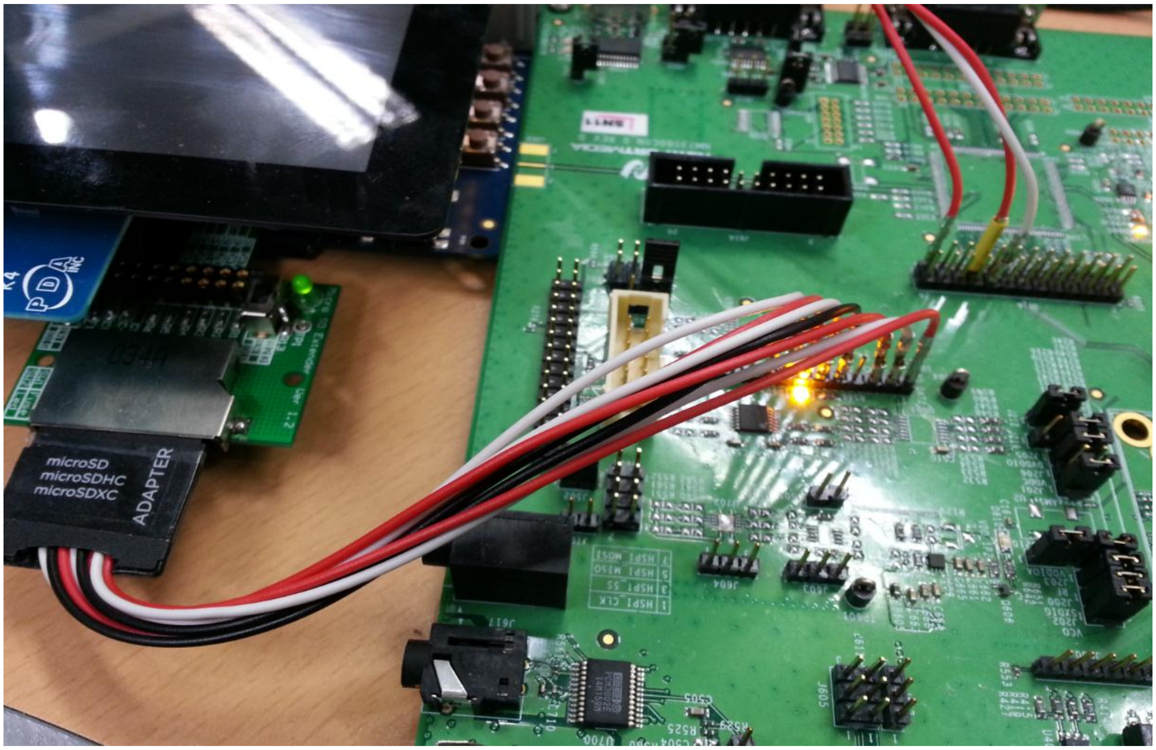
For the WLAN, the MCI1 MicroSD socket J14 on the SAMA5D4EK should be connected to the SDIO_J1 on the ATWILC3000 EVB. The following figure is the SDIO_J1 pin on the ATWILC3000.

Figure 3 ATWILC3000 SDIO_J1 Pin



The SD_DAT0 to SD_DATA3 should be connected to support SDIO 4-bit mode. In addition, SD_CMD, SD_CLK and GND should be connected to the SAMA5D4EK board. The following is real picture showing how to connect ATWILC3000 EVB to SAMA5D4EK for SDIO interface. The Micro SD socket is recommended like the following picture.

Figure 4 ATWILC3000 to SAMA5D4EK via SDIO



6. Conclusion

This Quick Start Guide described how to integrate the Atmel WILC3000 Combo driver in the Android platform with SAMA5D4-EK board.

7. Revision history

Doc. Rev.	Date	Comments
XXXXXA	11/2014	Initial document release
	11/2014	Add wilc3000 Bluetooth
	2/2015	Update the contents

Enabling Unlimited Possibilities®

Atmel Corporation
1600 Technology Drive
San Jose, CA 95110
USA
Tel:(+1)(408) 441-0311
Fax: (+1)(408) 487-2600
www.atmel.com

Atmel Asia Limited
Unit 01-5 & 16, 19F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
HONG KONG
Tel:(+852) 2245-6100
Fax: (+852) 2722-1369

Atmel Munich GmbH
Business Campus
Parkring 4
D-85748 Garching b. Munich
GERMANY
Tel:(+49) 89-31970-0
Fax: (+49) 89-3194621

Atmel Japan G.K.
16F Shin-Osaki Kangyo Bldg.
1-6-4 Osaki, Shinagawa-ku
Tokyo 141-0032
JAPAN
Tel: (+81)(3) 6417-0300
Fax: (+81)(3) 6417-0370

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. EXCEPT AS SET FORTH IN THE ATMEL TERMS AND CONDITIONS OF SALES LOCATED ON THE ATMEL WEBSITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS AND PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and products descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.