ATWILC3000 Getting Started

For Linux and Android Users

AN-XXXX

Prerequisites

- **Hardware Prerequisites**
    - SAMA5D4EK Android board
    - SAMA5D3 Xplained Linux board
    - Atmel WILC3000 evaluation board (NM73000C0N_0 REV 0)
    - Micro USB Cable (TypeA / MicroB)
    - USB to Serial Adaptor (for DBGU port)

- **Build Prerequisites**
    - Linux Host PC
    - Android Software Package
    - Linux Buildroot Software Package
    - ATWILC3000 prebuilt images

Introduction

The ATWILC3000 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 ATWILC3000 utilizes highly optimized 802.11 – Bluetooth coexistence protocols. It provides multiple peripheral interfaces including UART, SPI, I2C and SDIO.

This getting started guide describes how to integrate the Atmel WILC3000 evaluation board via SDIO for Wi-Fi and USART for Bluetooth in SAMA5D4EK Android platform and SAMA5D3 Xplained Linux platforms. In addition, this document shows how to run ATWILC3000 with prebuilt images provided from Atmel SmartConnect GitHub. The following links also are available to get more information on Atmel wireless drivers, Android/Linux kernel and prebuilt images.

- SAMA5D4EK board: http://www.atmel.com/tools/sama5d4-ek.aspx.
- SAMA5D3 Xplained board: http://www.atmel.com/tools/atsama5d3-xpld.aspx
- Linux4sam: http://www.at91.com/linux4sam
- Android4sam: http://www.at91.com/android4sam/bin/view/Android4SAM/
- ATWILC3000 official GitHub: https://github.com/atwilc3000

Table of Contents

1. ATWILC3000 Linux Software Package

The ATWILC3000 resources are available on the Atmel SmartConnect ATWILC3000 GitHub [1], which hosts official releases of ATWILC3000 kernel driver and firmware. In addition, there are patch files for several Atmel AT91 SAM Android and Linux platform. Download the latest ATWILC3000 driver and firmware by issuing the following commands.

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

For more information on WLAN and Bluetooth, visit the GitHub wiki pages [2].

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 [3]. There is also available the SAMA5D4EK user guide [4]. 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
```

When successfully done, **system_ubifs-SAMA5D4-ANDROID-4.4.2_r2.img** and **userdata_ubifs-SAMA5D4-ANDROID-4.4.2_r2.img** are generated in the Android root directory.

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 /android4sam_v4.4_rc2/out/target/product/sama5d4/root ../linix-at91
```

Then, issue the following commands to build the kernel, modules and Device Tree Binaries.

```
$ 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
```

2.6 Flash images

This section briefly introduces how to flash the SAMA5D4EK board with prebuilt and engineering images. Refer to FlashPreBuildDemo [5] in the Android4sam [3] to flash prebuilt images and engineering images built with the Android and kernel source codes in the corresponding board.

- Prebuilt images

  The prebuilt images are provided in the Android4sam [3]. Download the following demo package for NAND boot for the SAMA5D4EK Android platform.

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

  Then, push PB4 (RESET) while pushing PB3. The board goes 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**.

- Engineering images

  If Android and kernel are successfully built, the following outputs are generated.

  - */android4sam_v4.4_rc2/system_ubifs-SAMA5D4-ANDROID-4.4.2_r2.img*

  - */android4sam_v4.4_rc2/userdata_ubifs-SAMA5D4-ANDROID-4.4.2_r2.img*

  - */linux-at91/arch/arm/boot/zImage*

  - */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.

3. SAMA5D3 Xplained Board Software Package

This section introduces how to get the SAMA5D3 Xplained board [6] Linux software and build the Buildroot-at91. For more information on Atmel Linux platform, visit the Linux4sam [7] which is the main starting point for Linux OS and Atmel SAM products. There is also useful information on SAMA5D3 Xplained Linux platform, for example, downloading the Buildroot, building and flashing images to the target board.

3.1 Download Buildroot

Download the Buildroot for SAMA5D3 Xplained Linux board by issuing the following commands.

```
$ git clone git://github.com/linux4sam/buildroot-at91.git
$ cd buildroot-at91
$ git checkout origin/buildroot-2013.11-at91 -b buildroot-2013.11-at91
```

3.2 Build rootfs images

Build the Buildroot-at91 to make new rootfs images by issuing the following command.

```
$ make sama5d3_xplained_defconfig
$ make
```

The Linux kernel source is also being downloaded while building the Buildroot-at91. For example, the kernel source is available at */buildroot-at91/output/build/linux-xxxxx* after a compile is completed. The ATWILC3000 programming guide describes how to integrate the ATWILC3000 driver into the kernel source downloaded while compiling a Buildroot-at91.

3.3 Flash images

This section briefly introduces how to flash the SAMA5D3 Xplained board with prebuilt and engineering images. Install the SAM-BA tool [8] if not available in Linux or Windows machine.

- Prebuilt images

  The prebuilt images are provided by the Linux4sam [7]. Download the following demo package for SAMA5D3 Xplained board: ftp://www.at91.com/pub/demo/linux4sam_4.3/linux4sam-poky-sama5d3_xplained-4.3.zip. The followings are the procedures to flash images for Windows and Linux users.
  - Open JP5 to disable NAND flash memory access
  - Press BP2 reset button to boot from on-chip Boot ROM
  - For Windows users, verify that the USB connection is well established. AT91 USB to Serial converter should appear in Device Manager. Then, launch the **demo_linux_nandflash.bat** file in the demo archive.
  - For Linux users, check /dev/ttyACMx by issuing the following command. Then, launch the **demo_linux_nandflash.sh** file. If the /dev/ttyACMx that appear is different from /dev/ttyACM0, modify the **demo_linux_nandflash.sh** before launching the script.

  ```
  $ ls /dev/ttyACM*
  ```

  - When the logfile.log appears, check that = Done. = is written at the end of file.
  - Press BP2 reset button to boot on the NAND flash memory and play the demo.

- Engineering images

  The following outputs are generated when Buildroot id completely compiled. All of outputs should be downloaded to the target board.

  - /buildroot-at91/output/build/linux-xxxxx/arch/arm/boot/zImage

  - /buildroot-at91/output/build/linux-xxxxx/arch/arm/boot/dts/at91-sama5d3_xplained.dtb

  - /buildroot-at91/output/images/rootfs.ubi

  Copy all of them into the demo archive directory where the demo_linux_nandflash.sh or batch file exists. Then, replace the existing files by new outputs. Modify the followings described in the **demo_linux_nandflash.tcl** file in the demo archive directory like the followings and run the batch or script file as done with prebuilt image.

  ```
  set kernelFile "zImage"
  set rootfsFile "rootfs.ubi"
  ```

4. Hardware Consideration

This section shows how to connect the ATWILC3000 EVB to SAMA5D4EK and SAMA5D3 Xplained boards for WiFi and Bluetooth. The ATWILC3000 should be connected to them via SDIO for WLAN and USART for Bluetooth.

4.1 ATWILC3000 EVB

Make sure the following checklists before starting hardware consideration to bring up the ATWILC3000 with SAMA5D4EK and SAMA5D3 Xplained boards.

- Check if R220, R217, R218 and R219 are installed in the EVB for the SDIO interface

- Check if SDIO/SPI pin in a SW500 switch is 0 for the SDIO interface

- Check if J227, J805 and J901 open for the USART interface

## 4.2 SAMA5D4EK to ATWILC3000 EVB Bluetooth H/W connection

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 ATWILC3000 programming guide. Figure 1 is I/O expansion, J15 on SAMA5D4EK board. 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
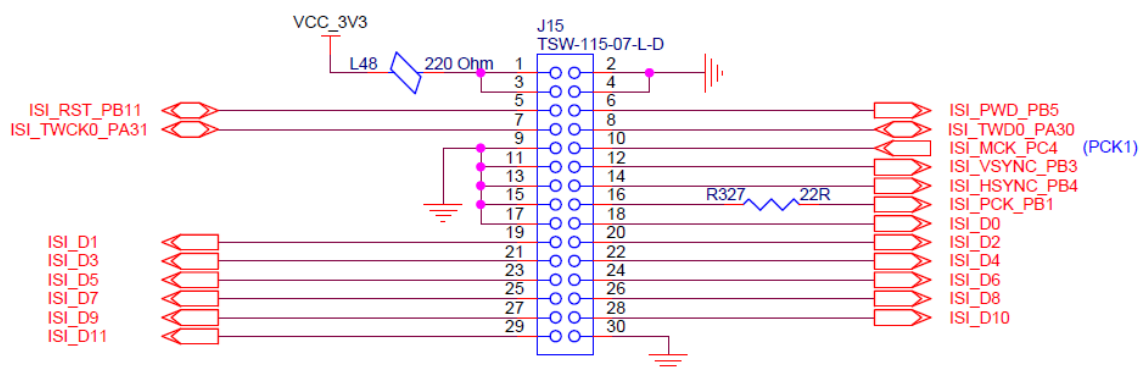
The TXD2 on the SAMA5D4EK should be connected to the RXD on the ATWILC3000 EVB, J216 pin18. In contrast, the RXD2 on the SAMA5D4EK should be connected to the TXD on the ATWILC3000 EVB, J216 pin19. The RTS2 on the SAMA5D4EK should be connected to the CTS on the ATWILC3000 EVB, J216 pin16 and the CTS2 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 3 is J216 connector on the ATWILC3000 EVB.

4.3 SAMA5D3 Xplained to ATWILC3000 Bluetooth H/W connection

For Bluetooth, the USART2 of SAMA5D3 Xplained is connected to the ATWILC3000 EVB. Figure 2 is I/O expansion, J19 on the SAMA5D3 Xplained board. The expansion, J19 has the TXD2 on pin E26, RXD2 on pin E25, RTS2 on pin PE24 and CTS2 on pin PE23.

The TXD2 on the SAMA5D3 Xplained should be connected to the RXD on the ATWILC3000 EVB, J216 pin18. In contrast, the RXD2 on the SAMA5D3 Xplained should be connected to the TXD on the ATWILC3000 EVB, J216 pin19. The RTS2 on the SAMA5D3 Xplained 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. The Figure 3 shows ATWILC3000 EVB J216 connector.

Figure 2 SAMA5D3 Xplained I/O Expansion
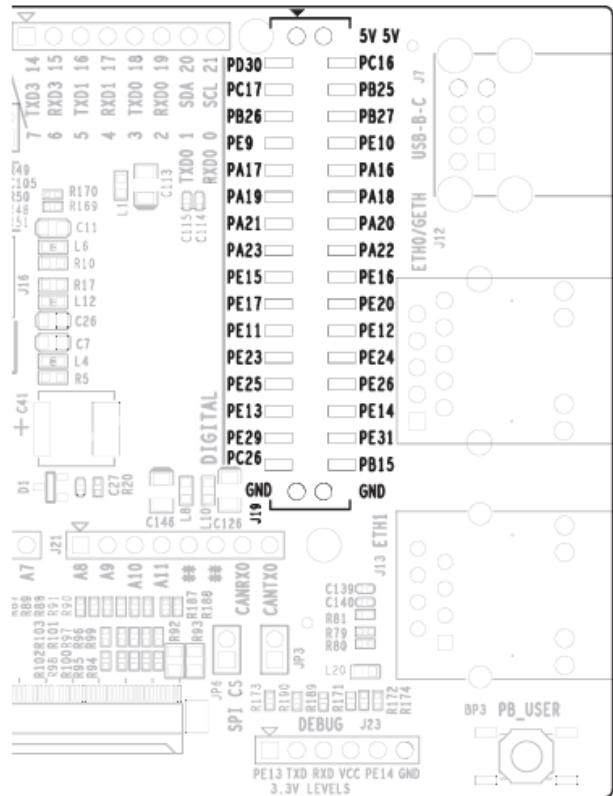
Figure 3 ATWILC3000 EVB J216
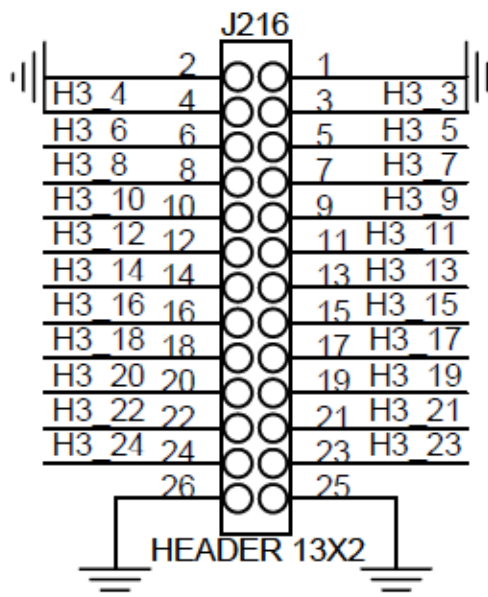
4.4 SAMA5D4EK to ATWILC3000 EVB WLAN H/W connection

For the WLAN, the MCI1 on the SAMA5D4EK should be connected to the SDIO_J1 on the ATWILC3000 EVB. Figure 4 is I/O expansion, J19 on SAMA5D4EK. The expansion, J19 has the DAT0 to DAT3 on pin 53 to 56 respectively, SD_CMD on pin 52, SD_CLK on pin 50, GND on pin 51 and VCC3V3 on pin 21.

For the card detection, MC1_CD on pin 38 should be connected to GND on pin 31.
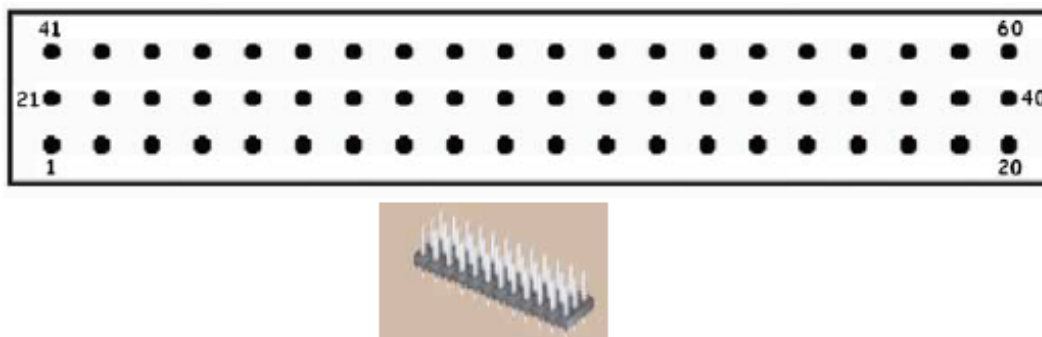
Figure 4 SAMA5D4EK IO Expansion socket J19

Table 4-20.   IO Expansion Socket J19 Signal Descriptions

Signal	Pin Number	Signal	Pin Number	Signal	Pin Number
PB0	1	VCC3V3	21	PE9	41
PB2	2	VCC3V3	22	PE10	42
PB6	3	PE31	23	PE11	43
PB7	4	PA24	24	PE12	44
PB8	5	PA25	25	PE13	45
PB9	6	PC4	26	PE14	46
PB10	7	PD8	27	PE15	47
PB12	8	PD30	28	PE16	48
PB13	9	PD31	29	PE17	49
PB15	10	-	30	PE18	50
GND	11	GND	31	GND	51
PB16	12	PE0	32	PE19	52
PB17	13	PE1	33	PE20	53
PB21	14	PE2	34	PE21	54
PB22	15	PE3	35	PE22	55
PB24	16	PE4	36	PE23	56
PB25	17	PE5	37	PE24	57
PB27	18	PE6	38	PE25	58
PB28	19	PE7	39	PE26	59
PB31	20	PE8	40	PE27	60

Figure 5 is the SDIO_J1 pin on the ATWILC3000. The SD_DAT0 to SD_DATA3 should be connected to support SDIO 4-bit mode. In addition, SD_CMD, SD_CLK, 3V3 and GND should be connected to the SAMA5D4EK board.

4.5 SAMA5D3 Xplained to ATWILC3000 WLAN H/W connection

The MCI0 SD/MMC card slot J10 at the bottom on the SAMA5D3 Xplained should be connected to the SDIO_J1 on the ATWILC3000 EVB. Figure 5 is the SDIO_J1 pin on the ATWILC3000. 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 SAMA5D3 Xplained board.

Figure 5 ATWILC3000 SDIO_J1 Pin
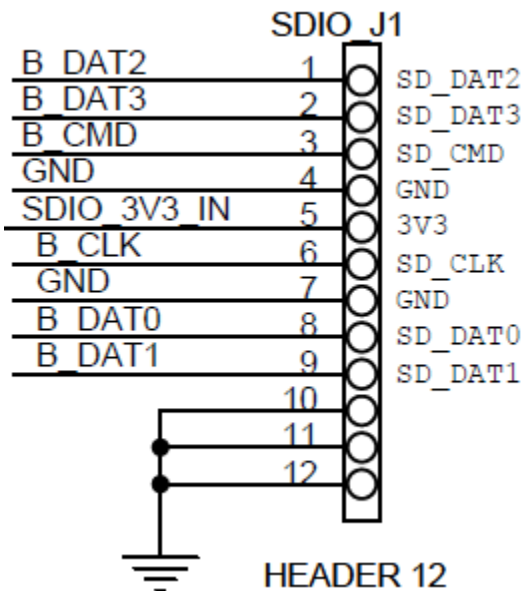
5. ATWILC3000 demo

This section introduces the ATWILC3000 prebuilt images running on SAMA5D4EK Android and SAMA5D3 Xplained Linux platforms. The prebuilt images for Atmel AT91 SAM series are available on the Atmel demo4sc GitHub [9]. These prebuilt images are provided quickly to bring up the Atmel platform with ATWILC3000. In addition, this section briefly introduces the wireless tools in Linux platform to run ATWILC3000 WLAN.

5.1 Play the demo on SAMA5D4EK Android platform

Download the prebuilt image of SAMA5D4EK Android platform from the following link.

https://github.com/demo4sc/wilc3000_with_sama5d4_ek_android/releases.

Refer to the section 2.6 to flash the image to the target board. Then, play the WLAN and Bluetooth with Android GUI.

5.2 Play the demo on SAMA5D3 Xplained Linux platform

The prebuilt image of SAMA5D3 Xplained Linux board is also downloadable from the following link.

https://github.com/demo4sc/wilc3000_with_sama5d3_xplained_linux

Refer to the section 3.3 to flash the image to the target board. There is no GUI for the SAMA5D3 Xplained Linux platform to run ATWILC300. So, issuing the command that wireless tools provide is required to run the ATWILC3000 WLAN and Bluetooth. For more information on the wireless tools, refer to the ATWILC3000 Linux User Guide, ATWILC3000 Programming Guide [10] or ATWILC3000 GitHub wiki [2].

6. Conclusion

This Getting Started Guide described how to quickly bring up the Atmel WILC3000 Combo driver with ATWILC3000 EVB in the SAMA5D4EK Android and SAMA5D3 Xplained platforms.

7. Revision history

Doc. Rev.	Date	Comments
XXXXXA	03/2015	Initial document release

8. References

[1] Atmel SmartConnect ATWILC3000 GitHub: https://github.com/atwilc3000

[2] Atmel SmartConnect ATWILC3000 wiki page: https://github.com/atwilc3000/driver/wiki

[3] Atmel Android for SAM: http://www.at91.com/android4sam/bin/view/Android4SAM/WebHome

[4] SAMA5D4 Evaluation Kit User Guide: http://www.atmel.com/Images/Atmel-11294-32-bit-Cortex-A5-Microcontroller-SAMA5D4-Evaluation-Kit_UserGuide.pdf

[5] FlashPreBuildDemo on Android4sam: http://www.at91.com/android4sam/bin/view/Android4SAM/FlashPreBuildDemo

[6] SAMA5D3 Xplained board User Guide: http://www.atmel.com/Images/Atmel_11269_32-bit-Cortex-A5-Microcontroller_SAMA5D3-Xplained_User-Guide.pdf

[7] Atmel Linux for SAM: http://www.at91.com/linux4sam/bin/view/Linux4SAM/

[8] Atmel SAM-BA tool: http://www.at91.com/linux4sam/bin/view/Linux4SAM/SoftwareTools#SAM_BA

[9] Atmel prebuilt image GitHub: https://github.com/demo4sc

[10] ATWILC3000 docs GitHub: https://github.com/atwilc3000/driver/tree/master/docs

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