WILC Linux applications User Guide

This user guide describes how to run application on the SAMA5D27-SOM1-EK1 with ATWILC1000 SD card or ATWILC3000 Shield board.

To start evaluation of Microchip Wi-Fi solutions with a Microchip MPU running Linux, following hardware is recommended:

1. *SAMA5D27-SOM1-EK1*

2. *Wi-Fi connectivity*

User can connect SAMA5D27-SOM1-EK1 MPU with WILC1000 SD or WILC3000 SD card for Wi-Fi application development.

a. *WILC1000 SD*

The ATWILC1000 SD is a Secure Digital (SD) card interface board that supports IEEE 802.11 b/g/n Wi-Fi standard.

Microchip Website Link:
https://www.microchip.com/developmenttools/ProductDetails/atwilc1000-sd

b. *WILC3000 SD*

The ATWILC3000 SD is a Secure Digital (SD) card interface board that supports IEEE 802.11 b/g/n Wi-Fi standard and Bluetooth Low Energy (BLE) 5.0.

Microchip Website Link:
https://www.microchip.com/DevelopmentTools/ProductDetails/AC164158

Software getting started

User needs to have Linux system setup for WILC application development.

Step 1: Download and build buildroot

Follow the below steps for downloading and building the buildroot:

- Clone buildroot using below command

  git clone [https://github.com/linux4sam/buildroot-at91.git -b linux4sam_6.0](https://github.com/linux4sam/buildroot-at91.git)

- Unzip the buildroot_external.zip available at the Github page
- *cd buildroot-at91*
- Run below command to set the configuration
  BR2_EXTERNAL=../buildroot_external/ make sama5d27_som1_ek_wilc_defconfig
- Run the below command to start the build
  make
  This command downloads necessary package required for buildroot.
- **Updating kernel with WILC driver and firmware:**
  - Remove old driver files

    rm -rf *buildroot-at91/output/build/linux-linux4sam_6.0/drivers/staging/wilc1000/**

  - Clone the WILC driver code from github using below command

    git clone git://github.com/linux4wilc/driver
    User can clone this driver to any temporary folder.

  - Copy WILC cloned "driver/wilc" files to "*buildroot-at91/output/build/linux-linux4sam_6.0/drivers/staging/wilc1000*" folder.

    cp -rf <cloned_dir>/ driver/wilc/* *buildroot-at91/output/build/linux-linux4sam_6.0/drivers/staging/wilc1000/*

  - Modify the following line in "*buildroot-at91/output/build/linux-linux4sam_6.0/drivers/staging/Makefile*" so that the build finds the correct directory

    FROM: obj-$(CONFIG_WILC1000) += wilc1000/
    TO: obj-$(CONFIG_WILC) += wilc1000/

  - Updating WILC firmware:
    - WILC Firmware desired version can be updated in the "buildroot_external\package\wilc-firmware\wilc-firmware.mk" file. Current

buildroot external is using WILC 15.2.User can change to any stable version of WILC firmware. Released WILC firmware information is available at: https://github.com/linux4wilc/firmware/releases

  - Run the below command to rebuild the kernel
    make linux-rebuild
  - Run the below command to update kernel in buildroot.
    make
  - Note: WILC driver and firmware should of the same version number (example: WILC 15.2). This release is tested with WILC 15.2 driver and firmware. User should follow above steps to upgrade to any WILC driver or firmware version.

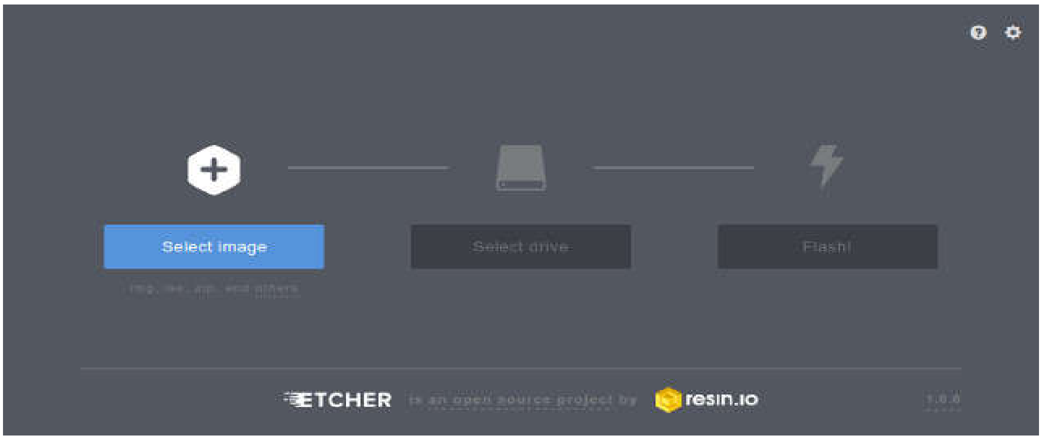
- On successful compilation an “images” folder is created under “buildroot-at91/output” folder.
- This folder will have the final image “sdcard.img” which needs to be programmed in a SD card. More details are in STEP 2.

Step 2: Install Etcher

User needs a 1 GB SD card (or more) to download final image "sdcard.img" of the demo. The final image "sdcard.img" is compressed to reduce the amount of data to download. This image contains:

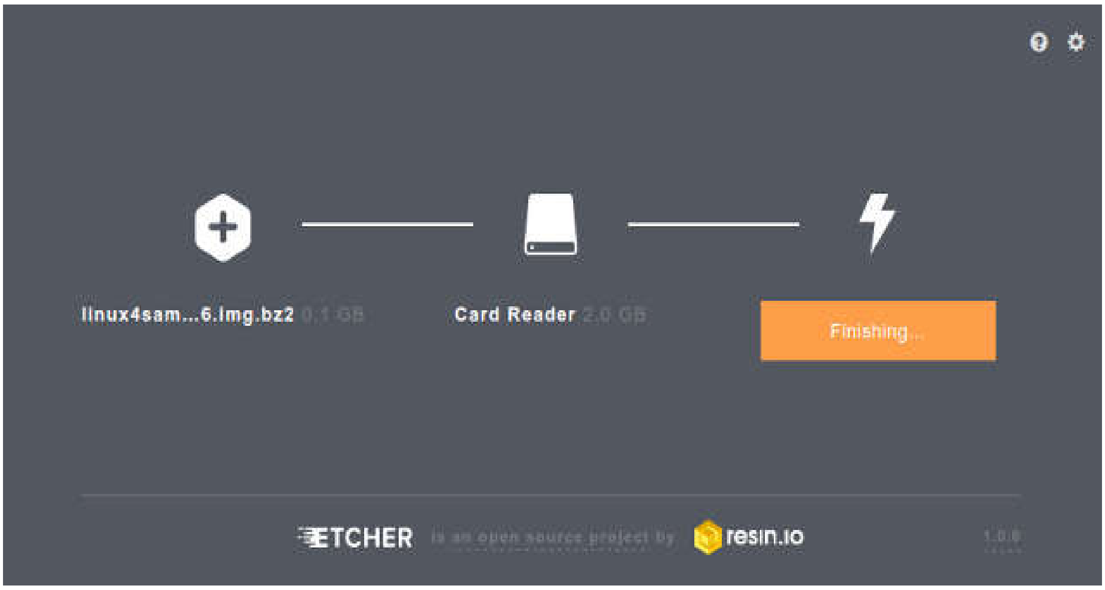
- A FAT32 partition with the AT91Bootstrap, U-Boot and the Linux Kernel (zImage and dtb).
- An EXT4 partition for the rootfs.

1. Download etcher from *https://www.balena.io/etcher/*
   This tool, which is an Open Source software, is useful since it allows to take a compressed image as input. More information and extra help is available on the Etcher website.



Snapshot of the Etcher

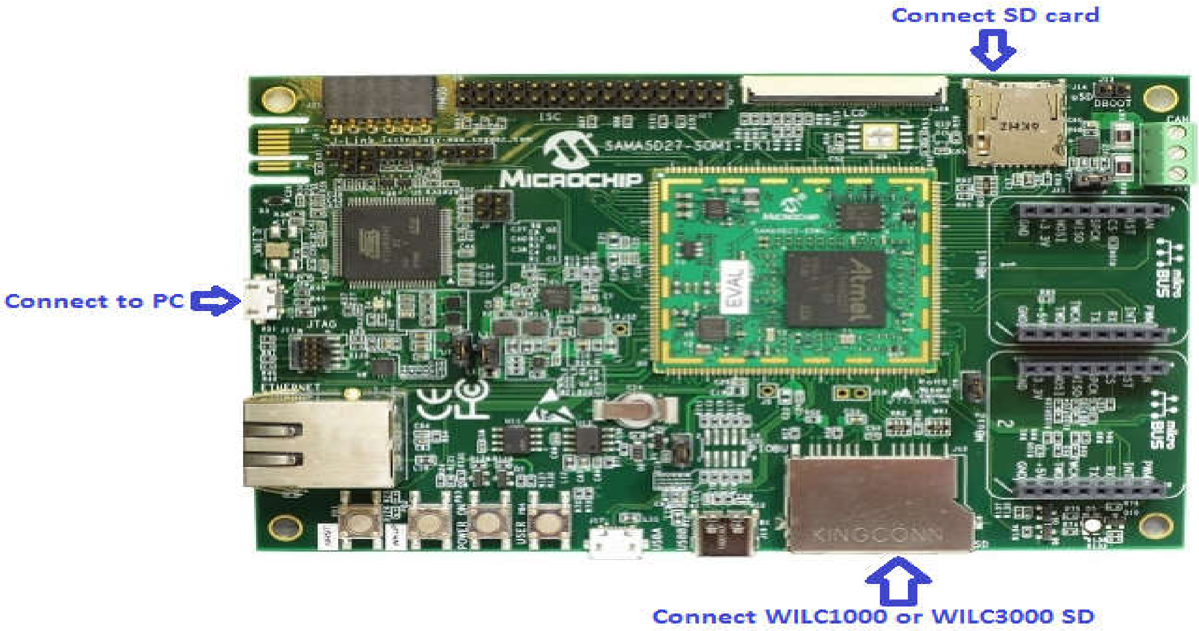
- Select the compiled image in STEP 1. It's marked as "SD Card image" in the tool.
- Select the device corresponding to your SD card (Etcher gives user advise on the devices that are removable to avoid erasing your system disk. Please doubly check the selected SD card before moving to next step).
- Click on the Flash! Button


linux4sam...6.img.bz2 0.1 GB
Card Reader 2.0 GB
Finishing...
ETCHER is an open source project by resin.io
1.0.0

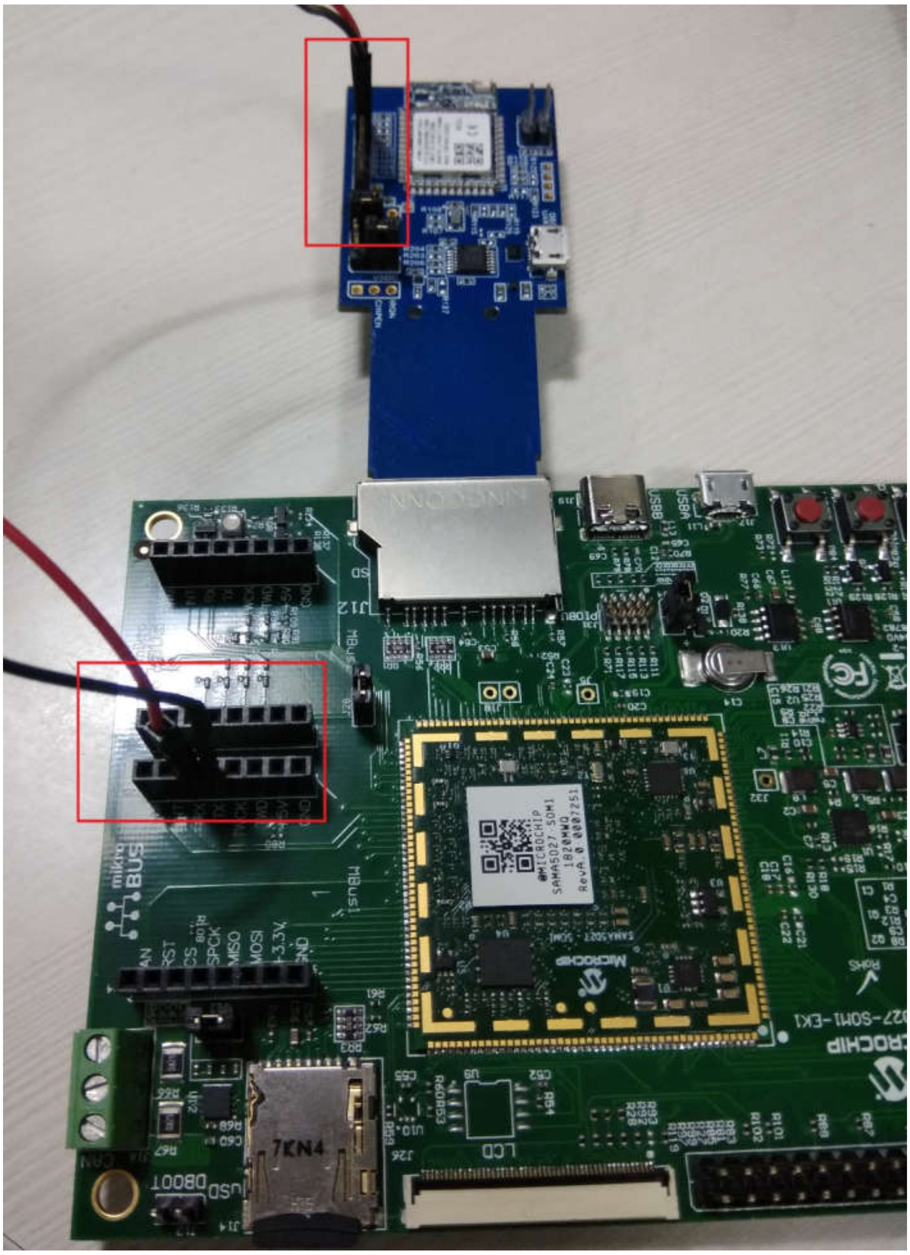
After writing is complete, user can observe “Flash complete!” message.

Step 3: Hardware Setup

SAMA5D27-SOM1-EK1 hardware connection details are given in the below pictures:



1. Plug the WILC1000 SD or WILC3000 SD to the SAMA5D27-SOM1-EK1 on J12 connector, and connect the SAMA5D2 Xplained board to the host PC through EDBG-USB port (J10)
2. Plug the STEP 2 program SD card to the SAMA5D27-SOM1-EK1 on J14 connector.
3. Please see below picture on how to connect the WILC3000 SD board with SAMA5D27-SOM1-EK1 with loose wires. This is required to enable BLE usage. It is optional if only Wi-Fi needs to be used.



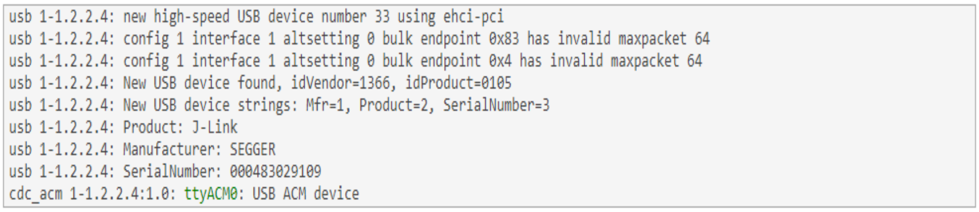
Pin mapping details between SAMA5D27 and WILC3000 SD for enabling BLE are given in the below table.

SAMA5D27-SOM1-EK1 J25 MBus1 connector	WILC3000 SD J101 connector
Rx (3rd pin on the connector)	BTTxD (5th pin on the connector)
Rx (4th pin on the connector)	BTRxD (4th pin on the connector)

4. Open a serial terminal, setup the correct COM port (EDBG Virtual COM) as follows:

Baud rate	115200
Data	8-bit
Parity	None
Stop	1-bit
Flow Control	None

- In Linux system, user can use “dmesg” command to know COM port connected device as shown in below snapshot.



```
usb 1-1.2.2.4: new high-speed USB device number 33 using ehci-pci
usb 1-1.2.2.4: config 1 interface 1 altsetting 0 bulk endpoint 0x83 has invalid maxpacket 64
usb 1-1.2.2.4: config 1 interface 1 altsetting 0 bulk endpoint 0x4 has invalid maxpacket 64
usb 1-1.2.2.4: New USB device found, idVendor=1366, idProduct=0105
usb 1-1.2.2.4: New USB device strings: Mfr=1, Product=2, SerialNumber=3
usb 1-1.2.2.4: Product: J-Link
usb 1-1.2.2.4: Manufacturer: SEGGER
usb 1-1.2.2.4: SerialNumber: 000483029109
cdc_acm 1-1.2.2.4:1.0: ttyACM0: USB ACM device
```

- In Linux system, User can use “minicom” tool for capturing COM port log.
  - Open COM port with “minicom -D /dev/ttyACM0” command on terminal
  - Press CTRL-A Z for menu.
  - Press “P” and “E” for selecting 115200 baudrate.

  To install minicom tool on linux, enter “apt-get install minicom” on terminal.

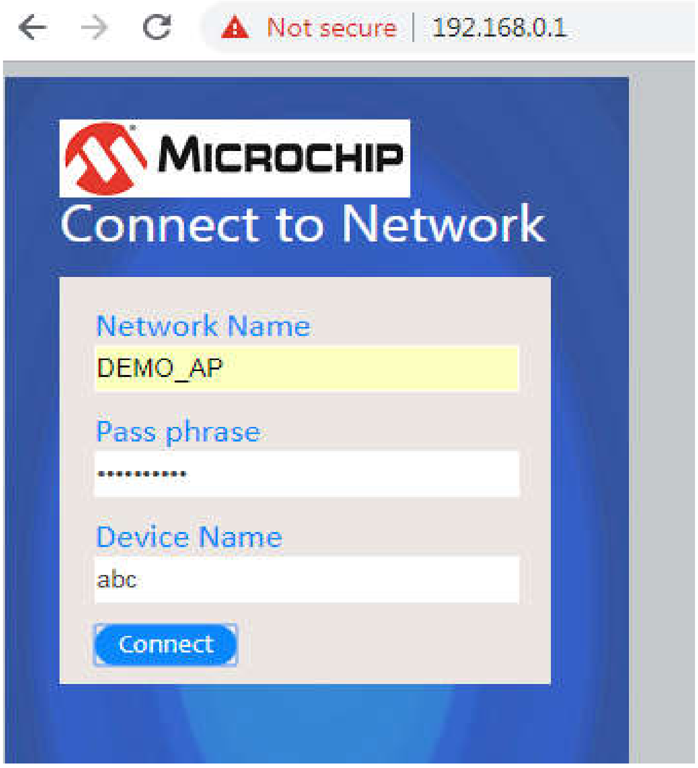
Step 4: Run a demo application

1. Wi-Fi provision application

This demo application provides out of box experience to user for Wi-Fi provisioning. Demo will help user to configure desired AP configuration like SSID, Passphrase using a webpage.

1. On power on reset, the DUT will enter in softAP mode.
2. SoftAP SSID is “wilc_SoftAP” and it’s in open network.
3. User can connect any Wi-Fi device (laptop/mobile) to DUT for provisioning.
4. SoftAP is configured with IP address as “192.168.0.1”. User can check with “ifconfig” command on serial terminal.
5. User needs to open web browser in 3rd party device and enter DUT IP address on URL.
6. The webpage provides three option to user for provision,
   - Network Name: Desired AP SSID configuration
   - Pass Phrase: Desired AP Passphrase configuration. This field need to be empty for open network configuration.
   - Device Name: DUT name (user choice)

   User needs to fill the details based on home AP configuration and press connect.



Connect to Network

Network Name
DEMO_AP

Pass phrase
••••••••••

Device Name
abc

Snapshot of the Webpage

7. On pressing Connect, DUT will reboot and switch to STA mode automatically.
8. On bootup, DUT enters in STA mode and will make a connection with configured home AP.

<u>Forcing Device to Provision mode:</u>

In case user wants to switch from STA mode to SoftAP for provisioning purpose again, user needs to run “Start_Provision.sh” from DUT shell. The DUT will switch to provisioning mode and user can configure the desired AP setting from webpage.

This application is applicable with WILC3000 and WILC1000 SD board.

2. Bluetooth Low Energy (BLE) Heart Rate profile

This demo application provides experience to user for heart rate profile of Bluetooth Low Energy (BLE). Application sends the data over BLE to mobile using Bluetooth heart rate profile.

After building a generic buildroot by following steps in the Section "**Step 1: Download and build buildroot"**, below steps need to be followed for compiling a heart rate application

- To compile source files, follow the below steps:
  - Go to package/apps folder inside buildroot_external
  - run "sh setup.sh"
  - This command enables compilation of Heart Rate application as part of buildroot and generates the final image "sdcard.img" under "buildroot-at91/output" folder.
  - Follow step 2 (**Step 2: Install Etcher)** for downloading image into SD card.
  - Follow below section for running the demo application

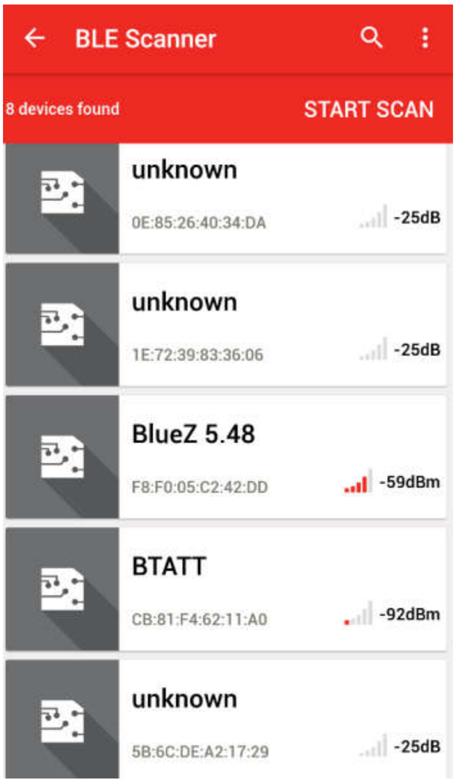
1. Before running this demo, the DUT should be in the STA mode.
   a. Kindly follow the steps 1 to 8 in "WiFi Provision Application" section of "Run a demo application" to ensure the DUT is provisioned with the Wi-Fi credentials of the home AP.
2. Enter login as "root" and password as "r"
3. Enter below command to start the BLE heart rate profile functionality.

   ```
   sh /root/Start_BT.sh 1
   ```

   User will observe below console output:

   ```
   hci0:   Type: Primary  Bus: UART

           BD Address: F8:F0:05:C2:42:DD  ACL MTU: 27:30  SCO MTU: 0:0

           UP RUNNING

           RX bytes:396 acl:0 sco:0 events:26 errors:0

           TX bytes:200 acl:0 sco:0 commands:26 errors:0

   Started listening on ATT channel. Waiting for connections
   ```

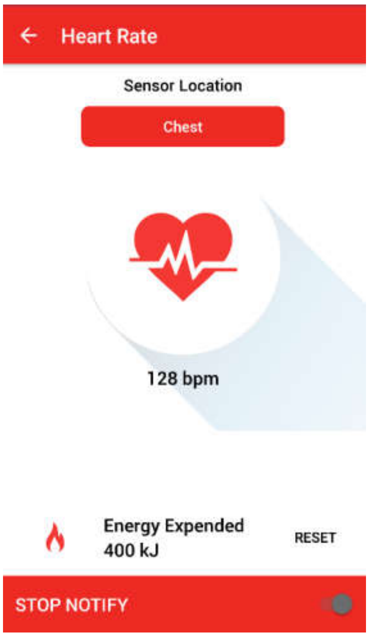
1. In smartphone mobile, download the "Microchip Bluetooth Data" application from play store (Tested with Android OS).
2. Turn-On Smartphone Bluetooth.
3. Goto "Microchip Bluetooth Data" application.
4. Start the "Bluetooth Smart" -> "Start Scan".
5. User will observe the "BlueZ X" in device list.



8 devices found

unknown	0E:85:26:40:34:DA	-25dB
unknown	1E:72:39:83:36:06	-25dB
BlueZ 5.48	F8:F0:05:C2:42:DD	-59dBm
BTATT	CB:81:F4:62:11:A0	-92dBm
unknown	5B:6C:DE:A2:17:29	-25dB

6. Click on “BlueZ X” device for connection.
7. On connecting with device, user will observe data of heart rate.



Sensor Location
Chest

128 bpm

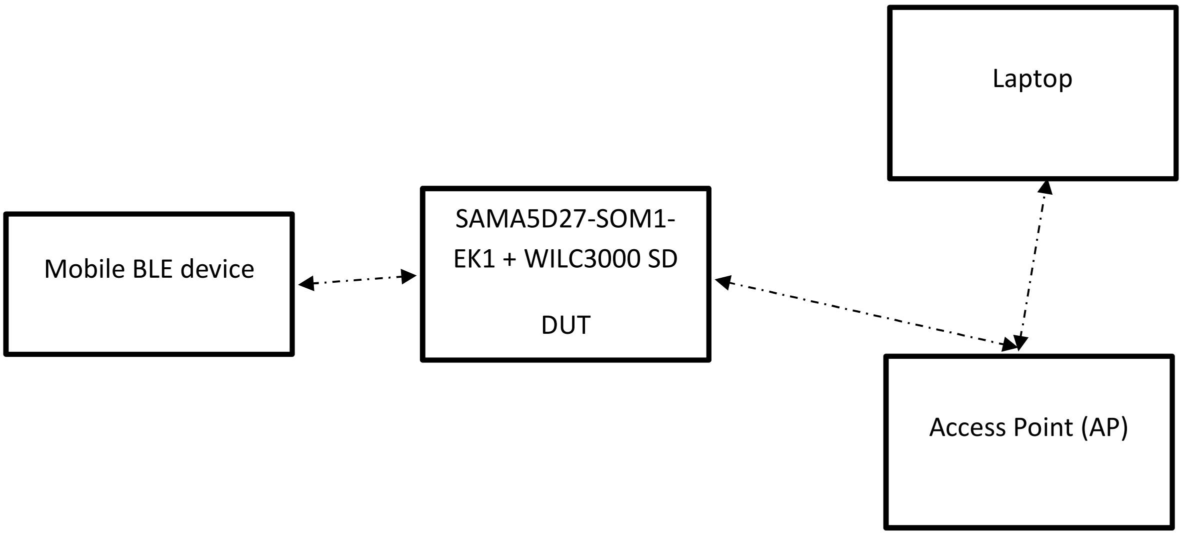
Energy Expended
400 kJ

This application is only applicable with WILC3000 SD board.

3. Wi-Fi and Bluetooth Coexistence

This demo application provides information to user for Wi-Fi and Bluetooth application data transfer at the same time.

User can create below network for this demo application:



1. Before running this demo, the DUT should be in the STA mode.
   a. Kindly follow the steps 1 to 8 in “WiFi Provision Application” section of “Run a demo application” to ensure the DUT is provisioned with the Wi-Fi credentials of the home AP.
2. Enter login as “*root*” and password as “*r*”
3. DUT will connect to AP.
4. Make Laptop Wi-Fi connection to same AP.
5. From Laptop start the ping to DUT
   a. DUT IP address can be found with “ifconfig” command.
6. On DUT, Ping Laptop using below command.

   ```
   ping <Laptop_IP_Address> &
   ```

   User needs to run the ping command in background to allow other command on DUT.

7. Start the Bluetooth application by following steps 3 to 10 in “BLE Heart Rate Profile” section of “Run a demo application”
8. User will observe the below logs on Console,

   ```
   [GATT server]# att: ATT op 0x01

   [GATT server]# att: < 01 08 12 00 0a                    .....

   [GATT server]# att: > 04 0e 00 0e 00                    .....
   ```

```
[GATT server]# att: ATT PDU received: 0x04
[GATT server]# server: Find Info - start: 0x000e end: 0x000e
[GATT server]# att: ATT op 0x05
[GATT server]# att: < 05 01 0e 00 02 29                              .....)
[GATT server]# 64 bytes from 192.168.43.59: seq=41 ttl=128 time=36.951 ms
64 bytes from 192.168.43.59: seq=42 ttl=128 time=6.300 ms
64 bytes from 192.168.43.59: seq=43 ttl=128 time=6.833 ms
wilc_sdio mmc0:0001:1 wlan0: INFO [debug_thread]*** Debug Thread Running ***
att: > 0a 10 00                              ...
[GATT server]# att: ATT PDU received: 0x0a
[GATT server]# server: Read Req - handle: 0x0010
[GATT server]# att: ATT op 0x0b
```

It shows that DUT is transferring data to Laptop over Wi-Fi (with ping) and at the same time DUT is also transferring BLE heart rate sensor data to smartphone.

This application is only applicable with WILC3000 SD board.

4. Bluetooth Low Energy (BLE) Transparent service

This demo application demonstrates to user on how to use transparent service of Bluetooth Low Energy (BLE). This is a custom/user defined profile of BLE. Application sends the data over BLE to mobile.

After building a generic buildroot by following steps in the Section "**Step 1: Download and build buildroot"**, below steps need to be followed for compiling a transparent service application

- To compile source files, follow the below steps:
  - Go to package/apps folder inside buildroot_external
  - run "sh setup.sh"
  - This command enables compilation of BLE Transparent Service application as part of buildroot and generates the final image "sdcard.img" under "buildroot-at91/output" folder.
  - Follow step 2 (**Step 2: Install Etcher)** for downloading image into SD card.
  - Follow below section for running the demo application

Running Demo application:

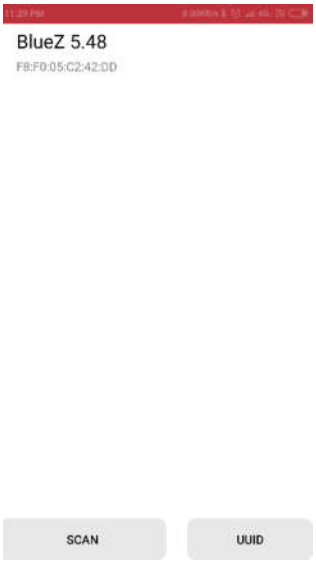
4. Before running this demo, the DUT should be in the STA mode.
   a. Kindly follow the steps 1 to 8 in "WiFi Provision Application" section of "Run a demo application" to ensure the DUT is provisioned with the Wi-Fi credentials of the home AP.
5. Enter login as "*root*" and password as "*r*"
6. Enter Below command to start the transparent service,

   ```
   sh /root/Start_BT.sh 2
   ```

   User will observe below UART console logs,

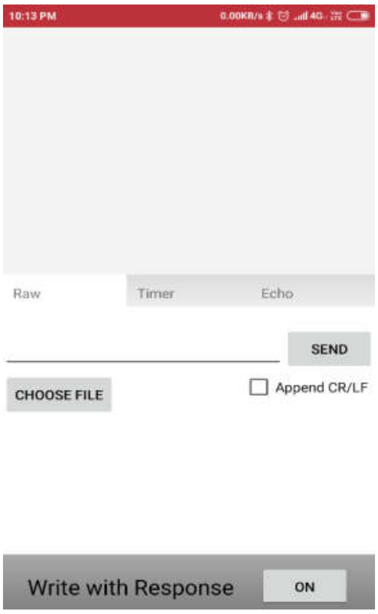
   ```
   transparent service application started
   start of the main
   Started listening on ATT channel. Waiting for connections
   ```

7. In smartphone mobile, download the "Microchip Bluetooth Data" application from play store (Tested with Android OS).
8. Turn-On Smartphone Bluetooth.
9. Goto "Microchip Bluetooth Data" application.
10. Start the "BM70" -> "Start Scan".
11. User will observe the device "Bluez 5.48" in scan list.Select the device for connection.



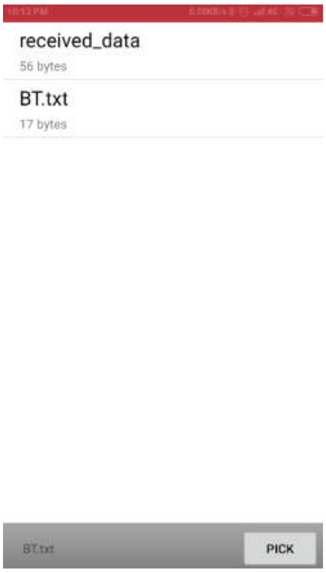
BlueZ 5.48
F8:F0:05:C2:42:DD

12. Enable Write with response to “ON”.



Write with Response ON

13. Write a text on text box and click on “send” button to send a data to WILC device.
14. User can observe received data on UART console.
15. User can transfer file from mobile phone to WILC device using below method.
    a. In mobile “mchp” folder, place the file user want to transfer.
    b. Select “choose file” option from APP.
    c. Select the file user want to transfer.



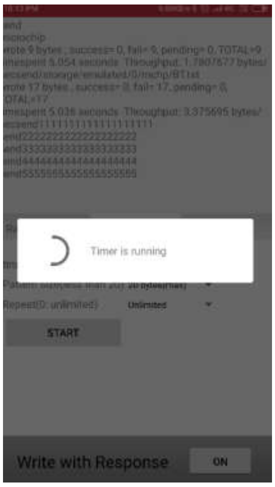
received_data
56 bytes

BT.txt
17 bytes

d. The file data will be transferred to WILC and UART console will display the file contents.

16. Similarly, there is an option for sending predefined data from mobile phone to WILC device.

    a. User can enable “timer” menu.



```
end
icrochip
rote 9 bytes , success= 0, fail= 9, pending= 0, TOTAL=9
imespent 5.054 seconds  Throughput: 1.7807677 bytes/
ecsend/storage/emulated/0/mchp/BT.txt
rote 17 bytes , success= 0, fail= 17, pending= 0,
OTAL=17
imespent 5.036 seconds  Throughput: 3.375695 bytes/
ecsend11111111111111111111
end22222222222222222222
end33333333333333333333
end44444444444444444444
end55555555555555555555
```

Timer is running

Repeat(0: unlimited) Unlimited

    b. This feature will keep sending the predefined data to WILC for the defined time duration.

5. Wi-Fi provisioning over BLE

This demo application provides demo to user on how to provision Wi-Fi credentials of the Home AP via the Bluetooth low energy (BLE). This is a custom/user defined profile of BLE. Application sends the data over BLE for provisioning Wi-Fi device.

After building a generic buildroot by following steps in the Section “**Step 1: Download and build buildroot”**, below steps need to be followed for compiling a transparent service application

- To compile source files, follow the below steps:
  - Go to package/apps folder inside buildroot_external
  - run “sh setup.sh”
  - This command enables compilation of Wi-Fi provisioning application as part of buildroot and generates the final image “sdcard.img” under “buildroot-at91/output” folder.
  - Follow step 2 (**Step 2: Install Etcher)** for downloading image into SD card.
  - Follow below section for running the demo application

<u>Running Demo application:</u>

1. On power on reset, the DUT should in the STA mode.

   - To know device mode, follow below Note 4 details.
   - In case device is not in STA mode, kindly follow the steps 1 to 8 in “WiFi Provision Application” to force the device into STA mode.

2. Enter login as “*root*” and password as “*r*”
3. Before running this demo, the DUT should be in the STA mode.
   a. Enter Below command to start the STA mode:
      i. sh /root/Start_STA.sh
4. Reboot the DUT
5. Enter login as “*root*” and password as “*r*”
6. Enter Below command to start the transparent service,

   ```
   sh /root/Start_BT.sh 3
   ```

   User will observe below UART console logs,

   ```
   WiFi provisioning service application started
   start of the main
   Started listening on ATT channel. Waiting for connections In smartphone mobile,
   download
   ```

7. In smartphone mobile, download the “Microchip Bluetooth Data” application from play store (Tested with Android OS).
8. Turn-On Smartphone Bluetooth.
9. Goto “Microchip Bluetooth Data” application.
10. Start the “Ble provisioner”

11. Select the listed device “bluez 5.48”
12. Enter Wi-Fi SSID and passphrase based on Home AP credentials and click on “provision”.
13. DUT will connect to configured AP.

    Note: Mobile Application Scan functionality is not available with this application.

Note:

1. Device configured login is “root” and password: “r”.
2. In case user wants to make a change in SoftAP security, then user needs to modify “/root/Start_AP.sh” file “hostapd /etc/wilc_hostapd_open.conf -B &” configuration with desired configuration of /etc/wilc_hostapd_web.conf or /etc/wilc_hostapd_wpa.conf.
3. Recompiling a buildroot single package,
   - In case user has modified one specific package of buildroot and wants to recompile only one package then follow below steps,
     - Goto wilc compilation folder
     - Run a command “make <package_name>-reconfigure”. For example “make websocket-reconfigure” where websocket is package name.
     - Then run “make” command.
4. Device mode Information

   To know device boot mode is STA or SoftAP, user needs to check the “/etc/init.d/S85start_wlan” file ‘start’ case configured details where:

   - sh /root/Start_AP.sh   : Device will bootup in softAP mode
   - sh /root/Start_STA.sh : Device will bootup in STA mode
5. Modification of “bluez5_utils-5.48” version change

   Current build root package has Bluetooth “bluez5_utils-5.48” utility, any future modification in bluez package number needs to be updated in the “buildroot_external/package/apps/setup.sh” file for inclusion of the Bluetooth application.

6. This document doesn’t cover information about installing a linux package for buildroot. User can use “apt-get install <package_name> “ to install package in setup linux system.
7. BT application can’t be run in SoftAP mode. This limitation will be resolved in future release.

Known issue: Sometime system soft-reboot doesn’t reboot WILC3000. In that case, user needs to hard reboot the DUT to switch mode from SoftAP to STA or STA to SoftAP.

Related document:

- ATWILC1000/3000 Linux User Guide
- ATWILC1000 Wi-Fi Link Controller RTOS Driver Release Notes
- ATWILC1000-SD Evaluation Kit Design Files
- ATWILC3000-SD Evaluation kit design Files
- ATSAMA5D27 SOM1 Kit1 User Guide

For other related documents, please visit the product webpage:

- ATWINC1000
- ATWINC3000
- SAMA5D27-SOM1-EK1