Talaria TWO™ (INP2045)

Ultra-Low Power Multi-Protocol Wireless Platform SoC

IEEE 802.11 b/g/n, BLE 5.0

Quick Start Guide – T23

Set-up of Talaria TWO Dual-Stack Demo using the T23 INP2201 Reference Kit

Release: 12-13-2024

InnoPhase IoT, Inc. | 2870 Zanker Road, Suite 200, San Jose, CA 95134 | https://innophaseiot.com/ | sales@innophaseiot.com

Revision History

Version	Date	Comments
1.0	06-05-2024	First release.
1.1	08-22-2024	Updated with steps to program Talaria TWO using the SSBL method.
1.2	12-13-2024	Updated steps to download packages.

Contents

Figures

Terms & Definitions

AP	Access Point
ELF	Executable Linkable Format
FAT	File Allocation Table
MCU	Microcontroller Unit
OpenOCD	Open On-Chip Debugger
SDK	Software Development Kit
SPI	Serial Peripheral Interface
SSID	Service Set Identifier
UART	Universal Asynchronous Receiver/Transmitter
UDP	User Datagram Protocol
WLAN	Wireless Local Area Network

Introduction

The T23 INP2201 Low Power Video Camera Reference Kit empowers embedded designers with IoT enabled Wi-Fi camera design for multiple video applications like video doorbell, IP camera, security camera, drones with camera, video conferencing devices and so on.

The reference kit has an Ingenic processor (Host), Talaria TWO (Wi-Fi module) and STM32 Low Power Module (MCU). The integrated T23 processor on the INP2201 reference kit acts as the Host processor. It is used as an Image Processing Unit to send image data to Talaria TWO for Wi-Fi transmission. It can be implemented as a solution with/without MCU.

This Quick Start Guide provides a brief introduction to setup and run the Talaria TWO Dual-Stack Solution demo using the T23 INP2201 Reference Kit.

Prerequisites

1. GTKTerm or similar application
2. STM32 programmer
3. INP3000 programmer board
4. Bootable SD card
5. Install the following dependencies to program Talaria TWO using the Download Tool:
    a. OpenOCD
    b. Python3 and dependencies

    For steps to install the same, refer `UG_Environment_Setup_for_Linux.pdf` (*sdk_x.y/doc/user_guides/ug_env_setup_linux*)

Package and Contents

To download any InnoPhase IoT offered software development kits or evaluation kits, register on the customer portal.

1. Go to the InnoPhase website (www.innophaseiot.com/Register) and click on Register.

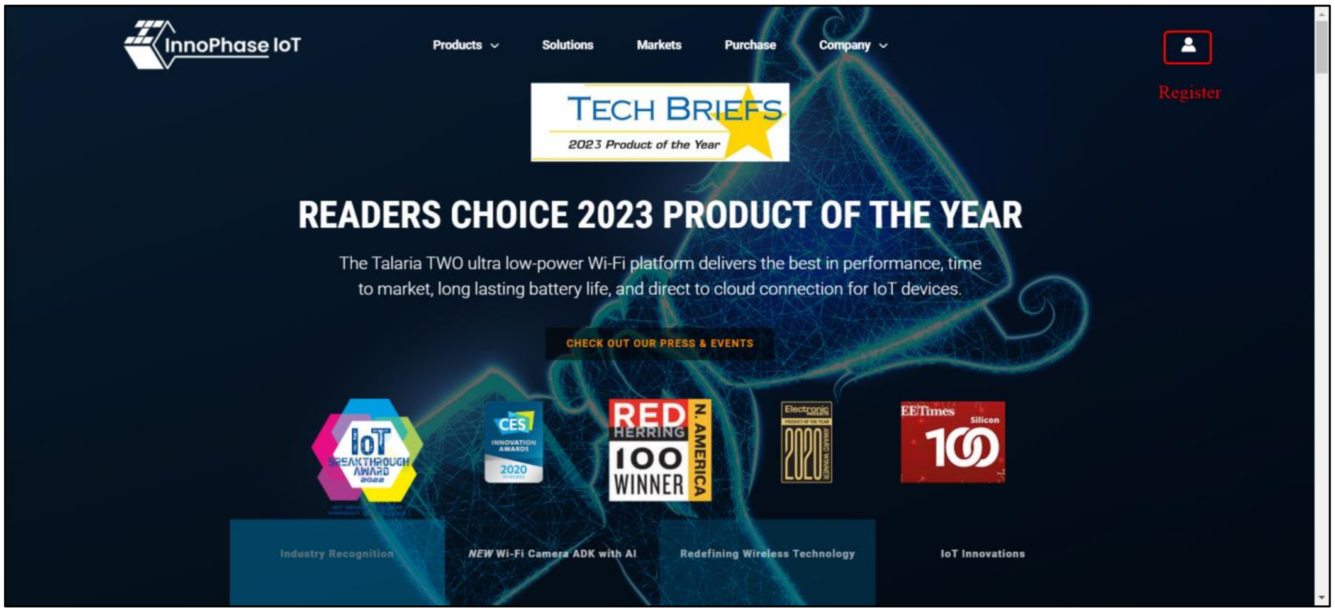
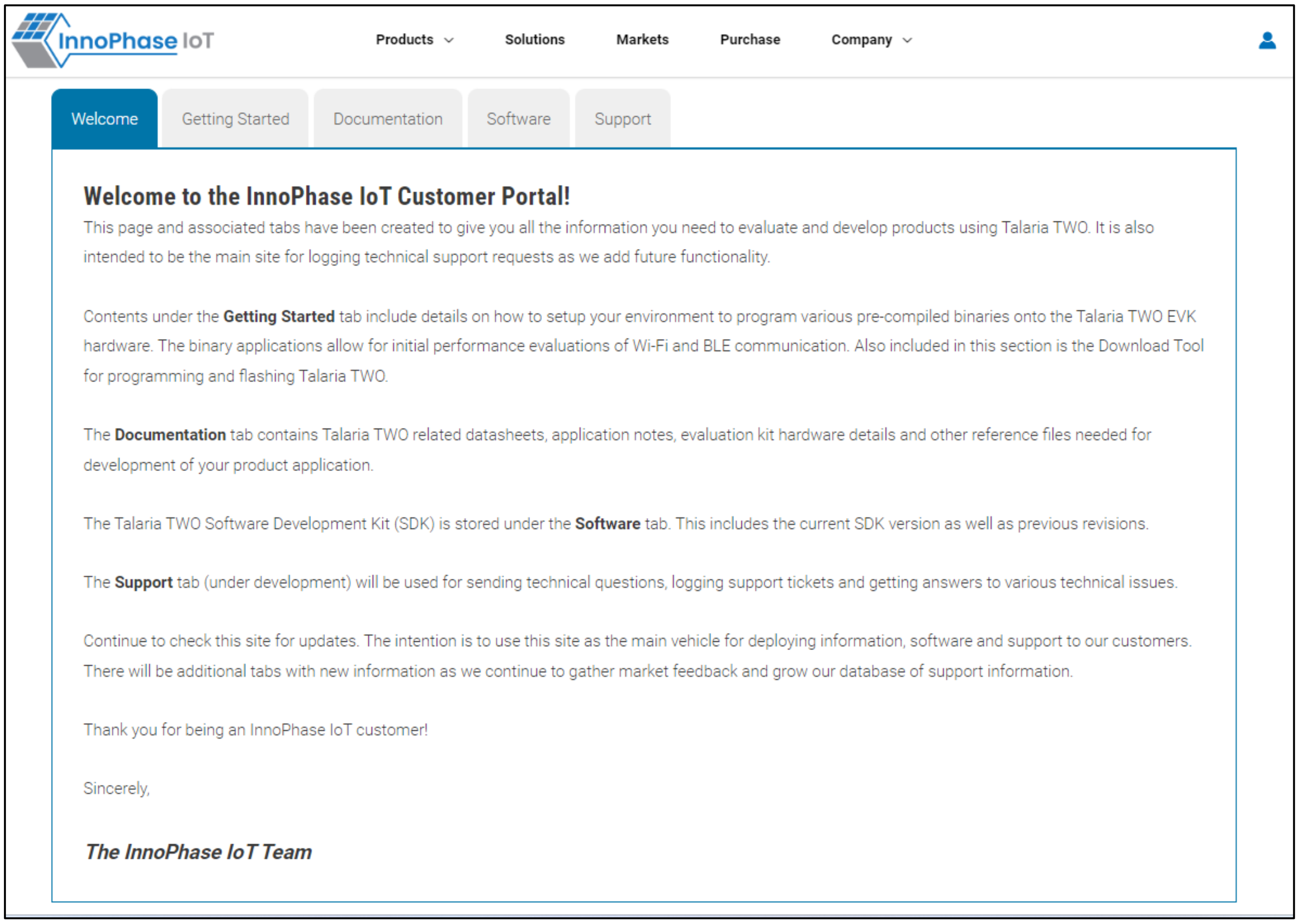

InnoPhase IoT
Products
Solutions
Markets
Purchase
Company
Register
TECH BRIEFS
2023 Product of the Year
READERS CHOICE 2023 PRODUCT OF THE YEAR
The Talaria TWO ultra low-power Wi-Fi platform delivers the best in performance, time to market, long lasting battery life, and direct to cloud connection for IoT devices.
CHECK OUT OUR PRESS & EVENTS
Industry Recognition
NEW Wi-Fi Camera ADK with AI
Redefining Wireless Technology
IoT Innovations

Figure 1: InnoPhase website

2. Provide the appropriate details to register onto the InnoPhase Customer Portal.
   Note: InnoPhase requires the Mutual Non-Disclosure Agreement (MNDA) and Development Tool License Agreement (DTLA) to be signed prior to granting access to the Customer Portal.

3. On successfully registering for Customer Portal, the following screen will appear:



Welcome to the InnoPhase IoT Customer Portal!

This page and associated tabs have been created to give you all the information you need to evaluate and develop products using Talaria TWO. It is also intended to be the main site for logging technical support requests as we add future functionality.

Contents under the **Getting Started** tab include details on how to setup your environment to program various pre-compiled binaries onto the Talaria TWO EVK hardware. The binary applications allow for initial performance evaluations of Wi-Fi and BLE communication. Also included in this section is the Download Tool for programming and flashing Talaria TWO.

The **Documentation** tab contains Talaria TWO related datasheets, application notes, evaluation kit hardware details and other reference files needed for development of your product application.

The Talaria TWO Software Development Kit (SDK) is stored under the **Software** tab. This includes the current SDK version as well as previous revisions.

The **Support** tab (under development) will be used for sending technical questions, logging support tickets and getting answers to various technical issues.

Continue to check this site for updates. The intention is to use this site as the main vehicle for deploying information, software and support to our customers. There will be additional tabs with new information as we continue to gather market feedback and grow our database of support information.

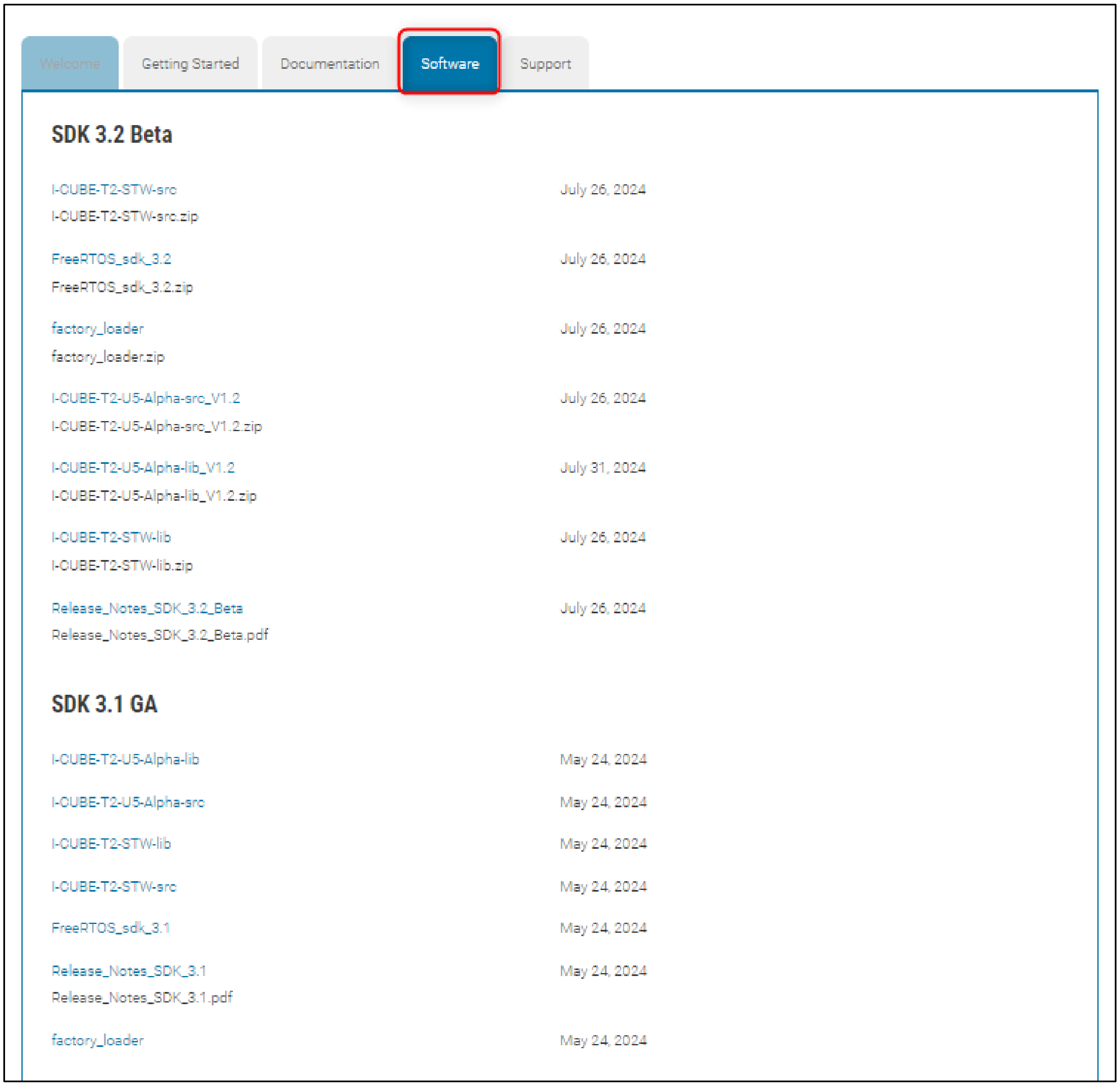
Thank you for being an InnoPhase IoT customer!

Sincerely,

The InnoPhase IoT Team

Figure 2: Customer portal registration

4. Navigate to the Software Tab and download the appropriate software package(s):



SDK 3.2 Beta

I-CUBE-T2-STW-src I-CUBE-T2-STW-src.zip	July 26, 2024
FreeRTOS_sdk_3.2 FreeRTOS_sdk_3.2.zip	July 26, 2024
factory_loader factory_loader.zip	July 26, 2024
I-CUBE-T2-U5-Alpha-src_V1.2 I-CUBE-T2-U5-Alpha-src_V1.2.zip	July 26, 2024
I-CUBE-T2-U5-Alpha-lib_V1.2 I-CUBE-T2-U5-Alpha-lib_V1.2.zip	July 31, 2024
I-CUBE-T2-STW-lib I-CUBE-T2-STW-lib.zip	July 26, 2024
Release_Notes_SDK_3.2_Beta Release_Notes_SDK_3.2_Beta.pdf	July 26, 2024

SDK 3.1 GA

I-CUBE-T2-U5-Alpha-lib	May 24, 2024
I-CUBE-T2-U5-Alpha-src	May 24, 2024
I-CUBE-T2-STW-lib	May 24, 2024
I-CUBE-T2-STW-src	May 24, 2024
FreeRTOS_sdk_3.1	May 24, 2024
Release_Notes_SDK_3.1 Release_Notes_SDK_3.1.pdf	May 24, 2024
factory_loader	May 24, 2024

Figure 3: Software tab

Talaria TWO SDK Package

Package Contents:

The package includes `dual_stack` folder (Path: *sdk_x.y\solutions\dual_stack*).

Note: x and y in *sdk_x.y* refers to the SDK release version.

The `dual_stack` folder contains the following:

1. `docs`:
   Readme which includes details of Files and folders of Talaria TWO Dual-Stack solution.
2. `bin`:
   Pre-built binaries (`dualstack.elf` and `dualstack_sdio.elf`).
3. `src`:
   Source code of Dual-Stack, and custom groups.
4. `lib`:
   Dual-Stack library/source.
5. `fs`:
   Contains json files.
6. `firmware_upgrade_images`:
   Root files.

Host Package

Download `talaria_two_dual_stack host` package provided by InnoPhase IoT.

Package Contents:

The package includes `talaria_two_dual_stack_vx.y` folder (Path: *talaria_two_dual_stack\talaria_two_dual_stack_vx.y*).

Note: x and y in *vx.y* refers to the package release version.

The `talaria_two_dual_stack_vx.y` folder contains the following:

1. `Quick Start Guide`: This document
2. `apps`
   Contains the optional binary images of common applications like iPerf. These applications can be executed once the Dual-Stack solution is up and running.
3. `doc`
   Contains documents which can be referred to for evaluating and working with the Dual-Stack solution:
   a. firmware-upgrade-via-serial: Contains the Readme to flash the firmware upgrade related files to Talaria TWO
   b. Talaria TWO Host API Reference Guide
   c. Dual-Stack User Guide
   d. Example applications for Dual-Stack
   e. MCU Firmware Flashing
   f. AWS Kinesis Video Streaming Quick Start Guide
4. `dual-stack`
   Contains the Dual-Stack solutions' software components. It also includes an example code to demonstrate custom logic on top of Dual-Stack solution to communicate with the Talaria TWO Wi-Fi module.
5. `patch`
   Contains the patch files for the kernel and/or U-boot if it exists. These patch files should be applied to the kernel of the platform before building the kernel (optional in case of a pre-built kernel).
6. `mcu`
   Contains the MCU firmware binary and application designed to run on T23 to communicate with MCU over UART to enable low-power mode while using video streaming capabilities.
7. `t23_kernel`
   Contains the binary file with bootloader, Linux kernel and root filesystem for T23 CPU.
8. `wlan-sdio`
   Contains the SDIO host driver for the Talaria TWO Wi-Fi module (optional in case of SPI).
9. `AWSIOT_certificates`
   Contains the AWS IoT certificates for MQTT connection authentication.
10. `readme`:
   Refer the Readme file for `talaria_two_dual_stack_vx.y` package folder structure. Follow individual readme files present in the sub-folders for more information.

Test Setup Topology

Figure 4 depicts the interaction between T23 INP2201 and Talaria TWO module.

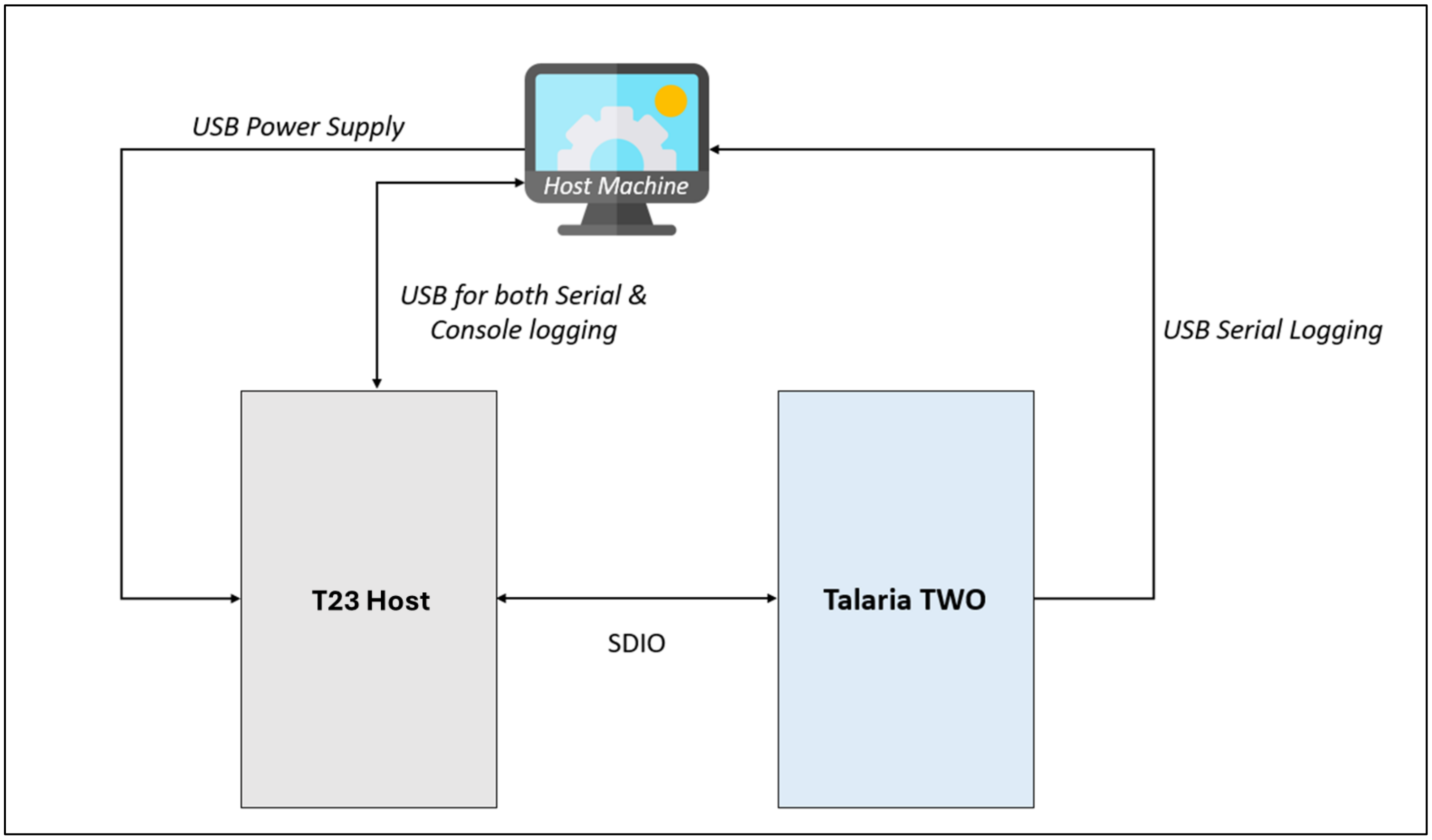


Figure 4: Block diagram

Evaluation Set-up & Usage

This section describes the set-up and usage of Talaria TWO module with T23 INP2201 for evaluation.

It provides:

1. Pre-built firmware binary for Talaria TWO (*sdk_x.y\solutions\dual_stack*)
2. Quick start guide *(sdk_x.y, documentation, programming tools (sdk_x.y\pc_tools\Download_Tool*))
3. Dual-Stack Host app binary and utilities (*talaria_two_dual_stack_vx.y\host\T23\dual-stack*)

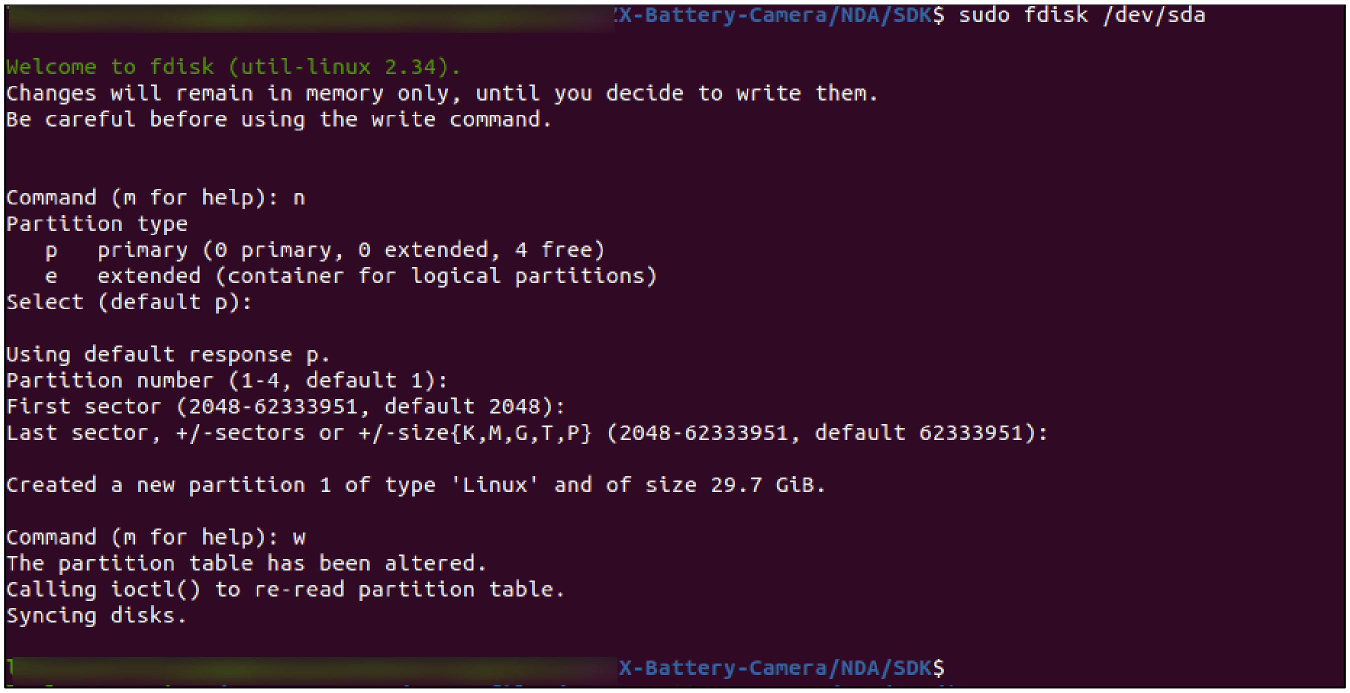
Preparing the SD Card and Booting T23 INP2201 Board

Note: The following steps need to be executed when booting the board for the very first time.

1. Using a 16GB or 32GB SDCARD, connect to a Linux PC/LAPTOP using a card reader.

2. Create a new FAT partition in SDCARD – sdcard is enumerated as `/dev/sda`

```
sudo fdisk /dev/sda
//For creating a new partition type n
Command (m for help): n <enter>

//Select default primary partition
Select (p(default) - primary (1 primary, 0 extended, 3 free),
e – extended(container for logical partitions)): <enter>

//Use default partition number 1
Partition number (1 - 4, default 1): <enter>

First sector (2048 – 62333951, default 2048): <enter>
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048 – 62333951, default 62333951):
<enter>
//Created a new partition 1 of ‘Linux’ and size of 29.7 GiB.
//For writing the changes to SDCARD type w
Command (m for help): w <enter>
```

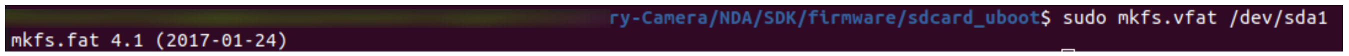


```
[redacted]X-Battery-Camera/NDA/SDK$ sudo fdisk /dev/sda

Welcome to fdisk (util-linux 2.34).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p):

Using default response p.
Partition number (1-4, default 1):
First sector (2048-62333951, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-62333951, default 62333951):

Created a new partition 1 of type 'Linux' and of size 29.7 GiB.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

[redacted]X-Battery-Camera/NDA/SDK$
```

Figure 5: Create a new FAT partition

3. Create FAT filesystem on this new partition

```
sudo mkfs.vfat /dev/sda1
```

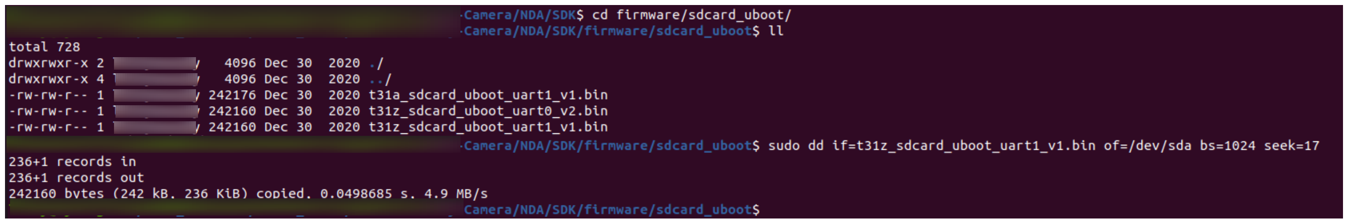


```
[redacted]ry-Camera/NDA/SDK/firmware/sdcard_uboot$ sudo mkfs.vfat /dev/sda1
mkfs.fat 4.1 (2017-01-24)
```

Figure 6: Create FAT filesystem

4. Copy the pre-built MMC u-boot image given in the T23 SDK to SDCARD
   SDK u-boot path: *T23_files/T23-Battery-Camera/NDA/SDK/firmware/sdcard_uboot*

```
sudo dd if=t23_sdcard_uboot_uart1_v1.bin of=/dev/sda bs=1024 seek=17
```



```
[redacted]Camera/NDA/SDK$ cd firmware/sdcard_uboot/
[redacted]Camera/NDA/SDK/firmware/sdcard_uboot$ ll
total 728
drwxrwxr-x 2 [redacted]/   4096 Dec 30  2020 ./
drwxrwxr-x 4 [redacted]/   4096 Dec 30  2020 ../
-rw-rw-r-- 1 [redacted]/ 242176 Dec 30  2020 t31a_sdcard_uboot_uart1_v1.bin
-rw-rw-r-- 1 [redacted]/ 242160 Dec 30  2020 t31z_sdcard_uboot_uart0_v2.bin
-rw-rw-r-- 1 [redacted]/ 242160 Dec 30  2020 t31z_sdcard_uboot_uart1_v1.bin
[redacted]Camera/NDA/SDK/firmware/sdcard_uboot$ sudo dd if=t31z_sdcard_uboot_uart1_v1.bin of=/dev/sda bs=1024 seek=17
236+1 records in
236+1 records out
242160 bytes (242 kB, 236 KiB) copied, 0.0498685 s, 4.9 MB/s
[redacted]Camera/NDA/SDK/firmware/sdcard_uboot$
```

Figure 7: Pre-build MMC u-boot image

5. Insert the SD card into the SD card slot of the T23 INP2201 board. Power ON the T23 INP2201 board by pressing the boot-select button to boot the uboot

Programming T23 Host

Execute the following steps to program T23 Host with `T23_INP2201.bin` (*talaria_two_dual_stack_vx.y\host\T23\t23_kernel*):

1. `T23_INP2201.bin` contains the bootloader, Linux kernel and root filesystem for T23
2. Copy `T23_INP2201.bin` to the Micro SD card and insert the Micro SD card in the SD card slot on the T23 INP2201.
3. Power on the T23 INP2201 board and press any key to halt at the u-boot prompt.
4. Run the following commands on the u-boot prompt to flash the SDK build image file `T23_IN2201.bin` to SPI flash.
5. Reboot the board to start booting Linux kernel from SPI flash.

```
//Filling the 16MB RAM with 0xff
mw.b 0x80600000 0xff 0x1000000
//List SD card contents
fatls mmc 0
//Loading the image from SDCRAD to RAM
fatload mmc 0 0x80600000 T23_INP2201.bin
//initialize the SPI flash
sf probe
//Erase the 16MB flash
sf erase 0x0 0x1000000
//Write the image from RAM to SPI flash
sf write 0x80600000 0x0 0x1000000
//remove the SDCARD and Reboot the board
reset
Now the T23 INP2201 board will boot the Linux kernel.
Zeratul login: root
```

Programming Talaria TWO

Hardware Set-up

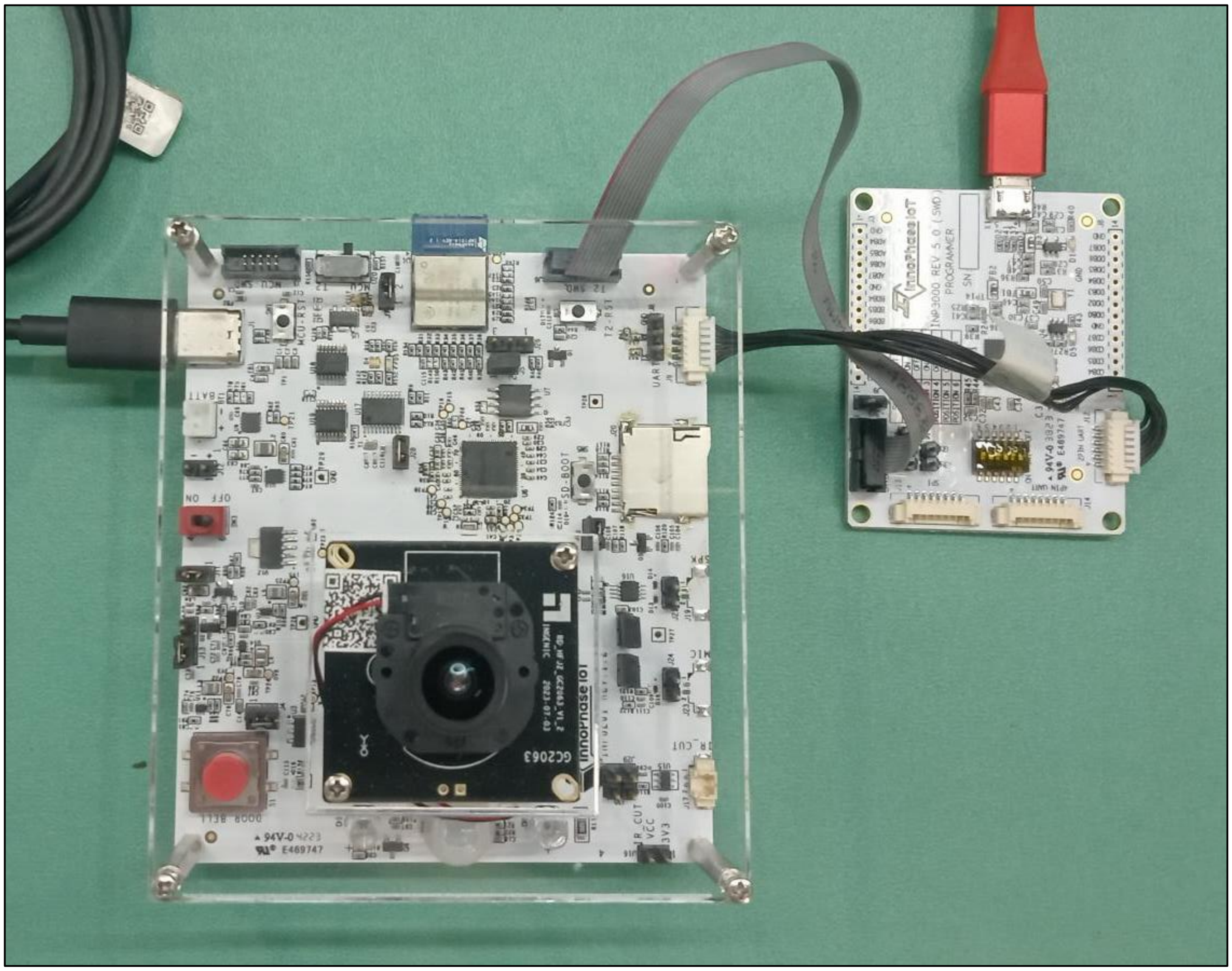


Figure 8: Hardware set-up

Note:

1. For more details on the hardware requirements, refer section: *Hardware Requirements* of the document: `UG_Dual-Stack.pdf` (*talaria_two_dual_stack_vx.y\host\T23\doc*).
2. For pin-outs details on INP3000 programmer board, refer `UG_Programming_using_INP3000.pdf` (*sdk_x.y\doc\user_guides\ug_programming_using_INP3000*).

Procedure

Program the Dual-Stack ELF (`<dual_stack_sdio>.elf`) onto Talaria TWO using the Download Tool (`T2DownloadTool_Windows.exe`) available in the SDK package (*sdk_x.y\pc_tools\Download_Tool\bin*).

This section describes the step-wise procedure only. For more details on using the Download Tool, refer: `UG_Download_Tool.pdf` (*sdk_x.y\pc_tools\Download_Tool\doc)*.

1. Launch the Download Tool and ensure to use the `SSBL` tab.

2. SSBL image: Select the SSBL image from: *sdk_x.y\apps\ssbl\fast_ssbl.img*

3. ELF file: Load the Dual-Stack ELF (*sdk_x.y\solutions\dual_stack\out\dual_stack.elf.strip*) with the appropriate boot arguments

4. Boot argument: Use the following boot arguments:

```
hio.min_heap_for_burst_tx=24000 hio.transport=sdio hio.maxsize=8192
hio.sdio_mhz=10 krn.coredump=1 wifi.listen_interval=1 wifi.traffic_timeout=0
wifi.outq_max=32 krn.gpio=--------------P----p-p
```

   Note:
   a. The `ds.mcu_uart_en=1` boot argument is used to enable the MCU UART on Talaria TWO. This boot argument needs to be enabled only for a Host platform with MCU (this is disabled by default for all Host platforms).
   b. For video streaming, use the following bootargs:

```
hio.min_heap_for_burst_tx to 32768
```

   The minimum heap required for burst mechanism is 32768.

5. Root image:
   Open a terminal on console machine and execute the following command from SDK root folder (*sdk_x.y*) to generate the root image:

```
For Non-SDIO:
python3 ./script/build_rootfs_generic.py --folder_path ./solutions/dual_stack/


For SDIO:
python3 ./script/build_rootfs_generic.py --folder_path ./solutions/dual_stack/ --
dualstack_option sdio
```

   Load the generated root image (`root.img`) from: *./solutions/dual_stack/*

6. Partition file: Load the appropriate partition file - `ssbl_part_table.json` (*sdk_x.y/ tools/partition_files*)

7. Flash the root filesystem:
   `Write Files` feature is used to write required files to Talaria TWO DATA partition. Select the appropriate folder (`data`) containing files/folders to be written onto Talaria TWO DATA partition and click on `Write Files`.
   Ensure that the selected folder contains the AWS certificate files: `aws_device_cert.crt`, `aws_device_pkey.key` & `aws_root_ca.crt`.
   For more details on writing files onto Talaria TWO, refer: `Write Files` section of the document: `UG_Download_Tool.pdf` (*sdk_x.y\pc_tools\Download_Tool\doc*).

   Note:
   a. The Download Tool has the capability to check and validate the data partition onto Talaria TWO. In case the current partition is not a SSBL partition, a pop-up message notifies the user to change the partition to the default SSBL partition (`ssbl_part_table.json`).
      Pop-up message options:
      i. **Yes:** Download Tool changes the partition to default SSBL partition.
      ii. **No**: Download Tool terminates the action and asks the user to correct the partition and try again.

8. Click on Prog Flash to flash to selected ELF. In this case, the Dual-Stack ELF.

9. Once flashing is complete, the Download Tool will automatically reset the device. The `Reset` button in the Download Tool can also be used to reset the device.

Programming ST MCU Binary

For information on programming the MCU binary, refer: `MCU_Firmware_Flashing.pdf` (*\talaria_two_dual_stack_vx.y\host\T23\doc*).

Testing for Basic Operations

There are three application `init` scripts in `/system/init` folder:

1. `app_init.sh`: Basic functionality (using connection manager commands)
2. `app_init_t2.sh`: Talaria TWO used for power management (copy this script to `app_init.sh`)
3. `app_init_mcu.sh`: STM32 MCU used for power management (copy this script to `app_init.sh`)

When T23 booting is completed, the `init` script loads the `sdio-wlan.ko` driver module and runs the tunadapter application as an initial step to use the Talaria TWO Dual-Stack commands.

For more details on the tunadapter, refer: `UG_Dual_Stack.pdf` (*talaria_two_dual_stack_vx.y\host\T23\doc*).

Use Case 1: Station Mode Wi-Fi Connection

This use case is to demonstrate the station mode configuration and connecting to the Wi-Fi Access Point.

Execute the following operations on the Talaria TWO:

Step 1: Scan the network.

Step 2: Connect to the desired network by providing SSID and passphrase

Step 3: Get the IP address of the Talaria TWO module

Step 4: Get the WLAN status of the Talaria TWO module

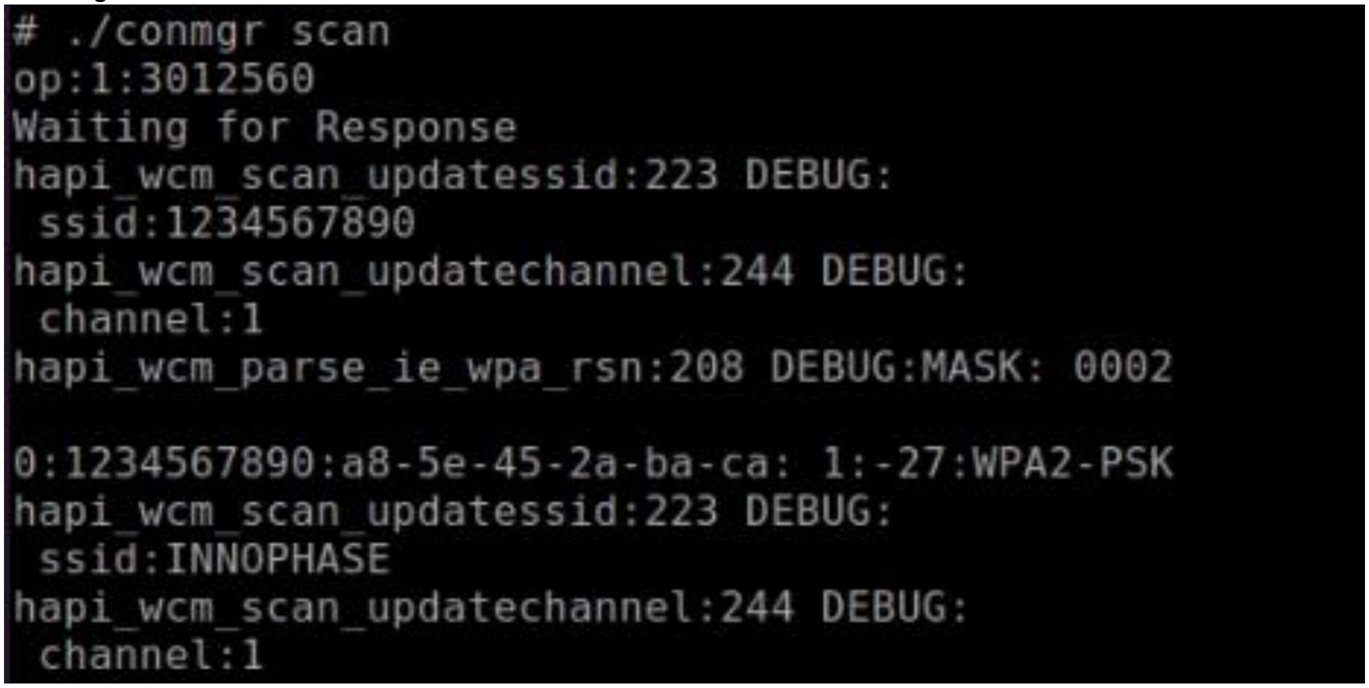
Step 5: Disconnect from the connected network

Command	Description
./conmgr scan	Scan
./conmgr connect <SSID><AP PWD>	Connect
./conmgr ip	Get IP address
./conmgr status	Get WLAN status
./conmgr disconnect	Disconnect

Table 1: Station Mode Wi-Fi Connection – Commands

Console outputs:

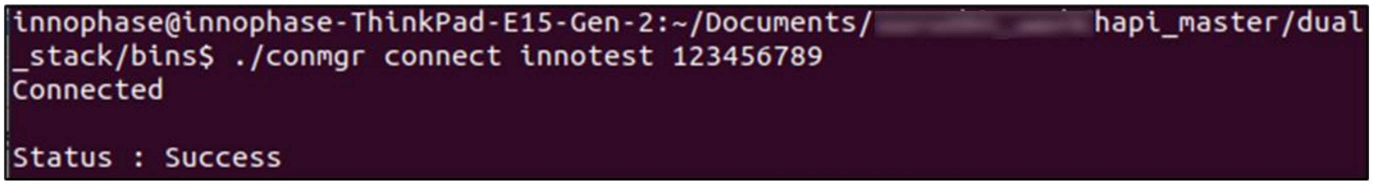
1. ./conmgr scan



```
# ./conmgr scan
op:1:3012560
Waiting for Response
hapi_wcm_scan_updatessid:223 DEBUG:
 ssid:1234567890
hapi_wcm_scan_updatechannel:244 DEBUG:
 channel:1
hapi_wcm_parse_ie_wpa_rsn:208 DEBUG:MASK: 0002

0:1234567890:a8-5e-45-2a-ba-ca: 1:-27:WPA2-PSK
hapi_wcm_scan_updatessid:223 DEBUG:
 ssid:INNOPHASE
hapi_wcm_scan_updatechannel:244 DEBUG:
 channel:1
```

Figure 9: ./conmgr scan – output

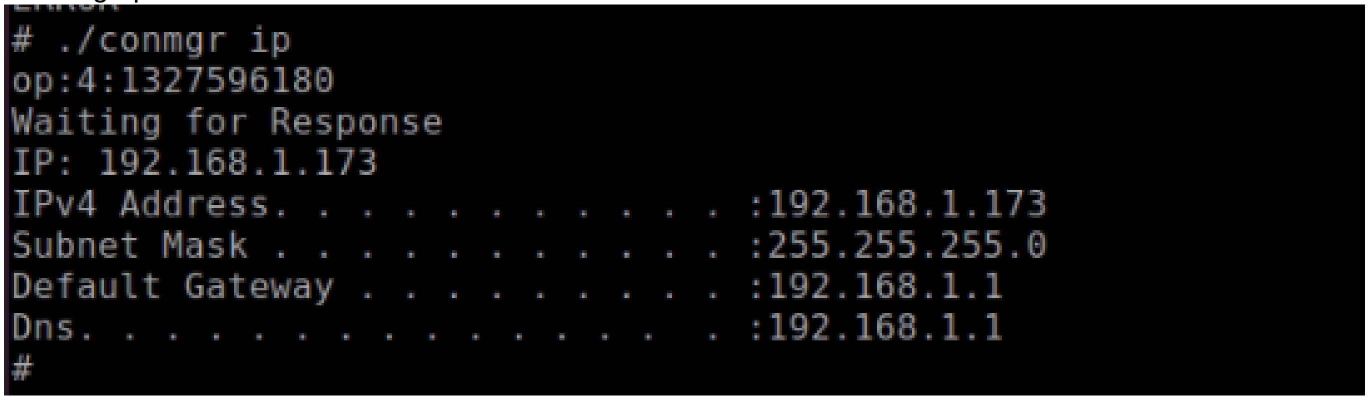
2. ./conmgr connect `innotest 123456789`



```
innophase@innophase-ThinkPad-E15-Gen-2:~/Documents/[redacted]hapi_master/dual
_stack/bins$ ./conmgr connect innotest 123456789
Connected

Status : Success
```

Figure 10: ./conmgr connect - output

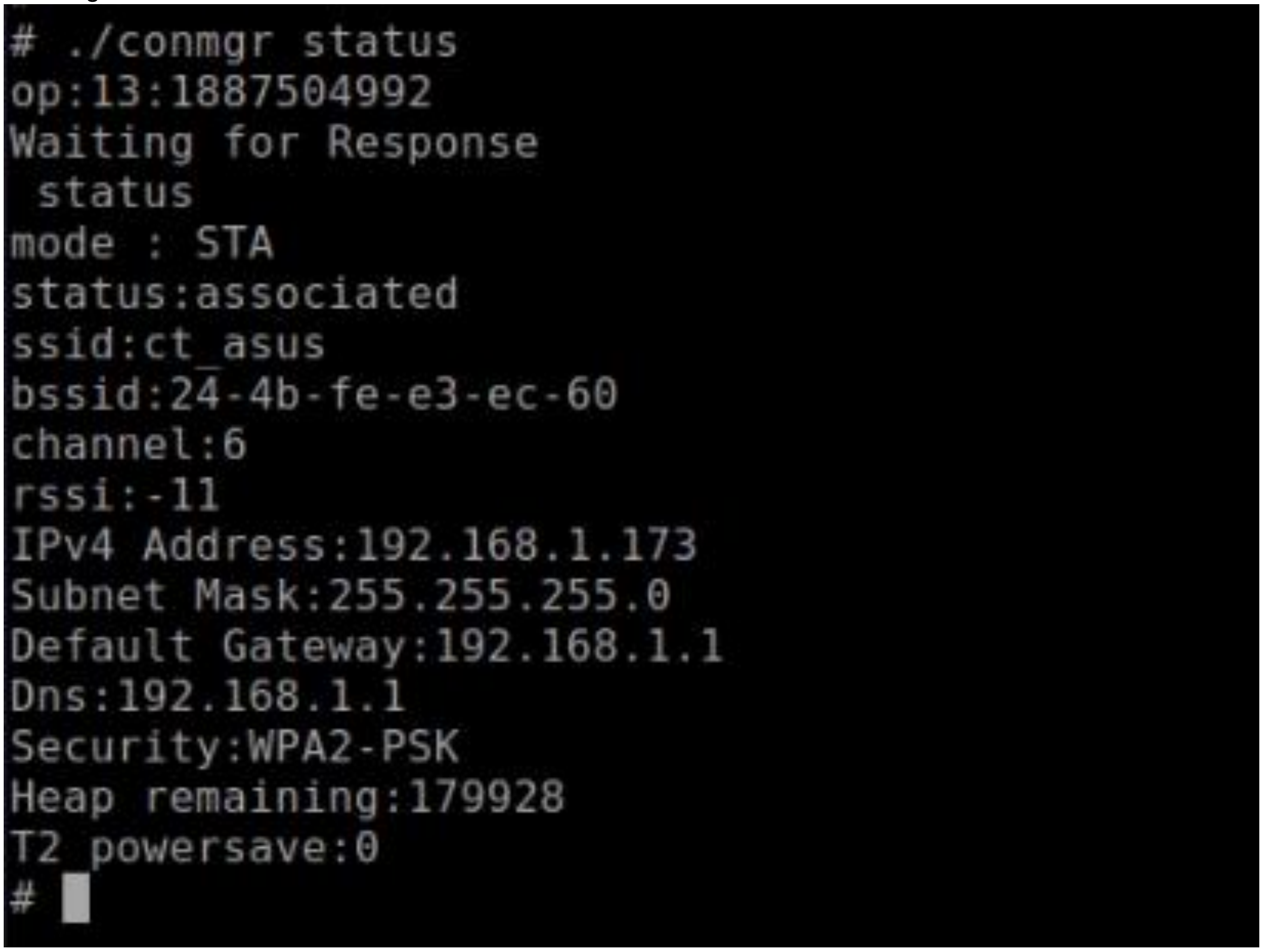
3. ./conmgr ip



```
# ./conmgr ip
op:4:1327596180
Waiting for Response
IP: 192.168.1.173
IPv4 Address. . . . . . . . . . . . . :192.168.1.173
Subnet Mask . . . . . . . . . . . . . :255.255.255.0
Default Gateway . . . . . . . . . . . :192.168.1.1
Dns. . . . . . . . . . . . . . . .  . :192.168.1.1
#
```

Figure 11: ./conmgr ip – output

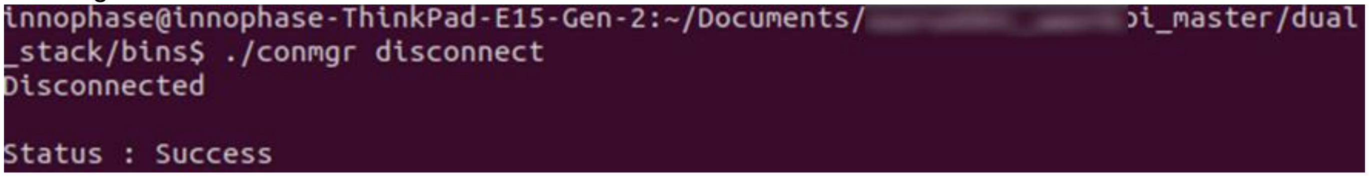
4. ./conmgr status



```
# ./conmgr status
op:13:1887504992
Waiting for Response
 status
mode : STA
status:associated
ssid:ct_asus
bssid:24-4b-fe-e3-ec-60
channel:6
rssi:-11
IPv4 Address:192.168.1.173
Subnet Mask:255.255.255.0
Default Gateway:192.168.1.1
Dns:192.168.1.1
Security:WPA2-PSK
Heap remaining:179928
T2 powersave:0
# 
```

Figure 12: ./conmgr status – output

5. ./conmgr disconnect



```
innophase@innophase-ThinkPad-E15-Gen-2:~/Documents/[redacted]pi_master/dual
_stack/bins$ ./conmgr disconnect
Disconnected

Status : Success
```

Figure 13: ./conmgr disconnect – output

Use Case 2: Performance Test using iPerf Application

This use case is to test the performance using iPerf application. Execute the following steps:

Step 1:

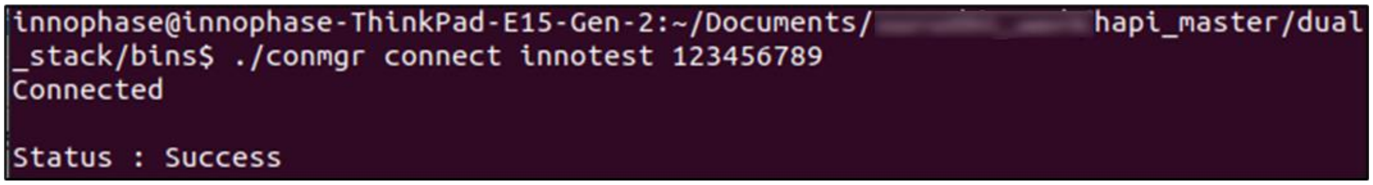
1. Scan the network
2. Connect to the network of SSID `ct_asus` and passphrase `12345678`
3. Get the IP address of the Talaria TWO module
4. Get the WLAN status of the Talaria TWO module
5. Start the UDP server and send data once the client connects

Command	Description
./conmgr scan	Scan
./conmgr connect <SSID><AP PWD>	Connect
./conmgr ip	Get IP address
./conmgr status	Status
iperf3 -s -i 1	Iperf traffic for UDP UL

Table 2: Performance Test using iPerf Application – Commands

Console outputs:

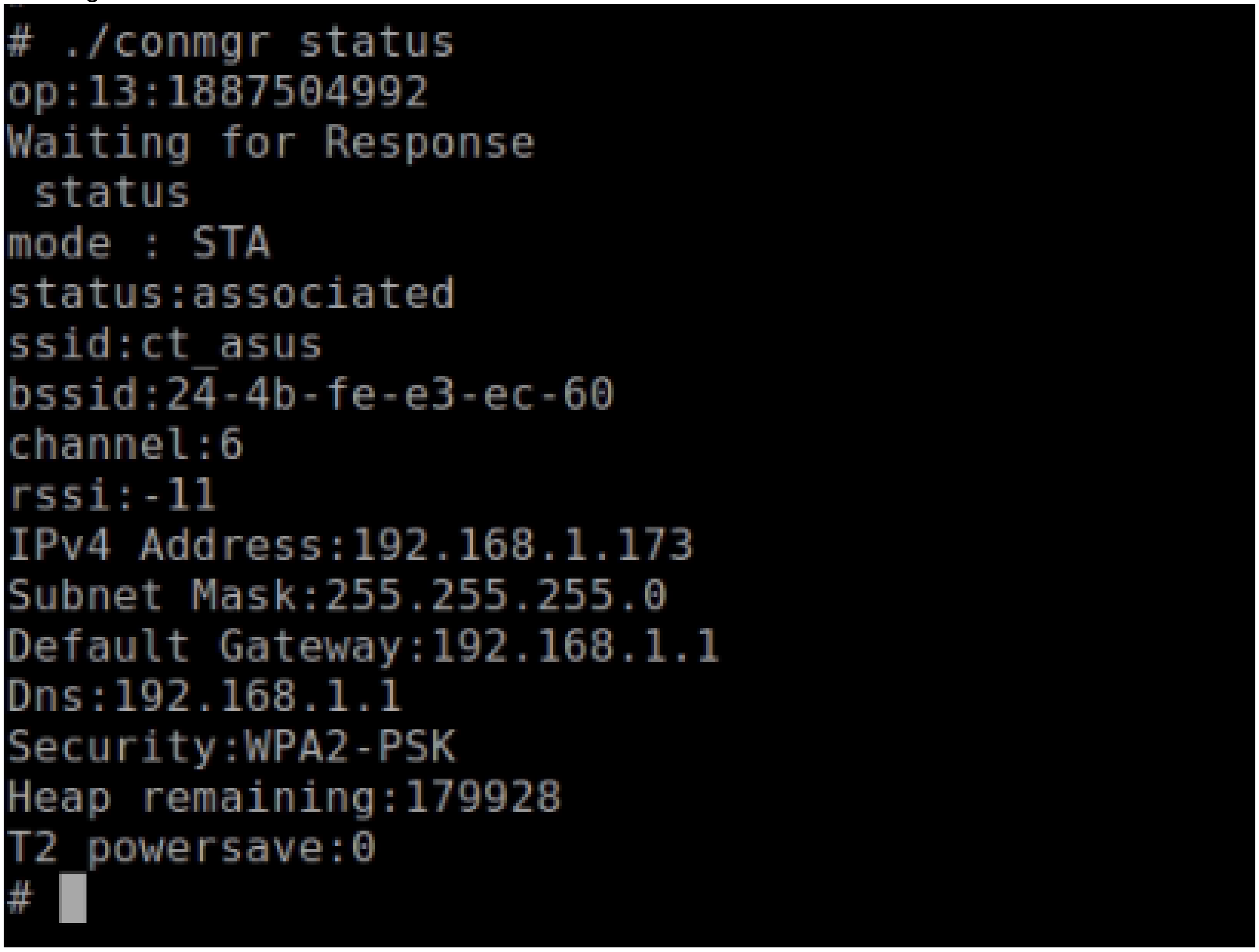
1. ./conmgr connect `innotest 123456789`



```
innophase@innophase-ThinkPad-E15-Gen-2:~/Documents/[redacted]hapi_master/dual
_stack/bins$ ./conmgr connect innotest 123456789
Connected

Status : Success
```

Figure 14: ./conmgr connect - output

2. ./conmgr status



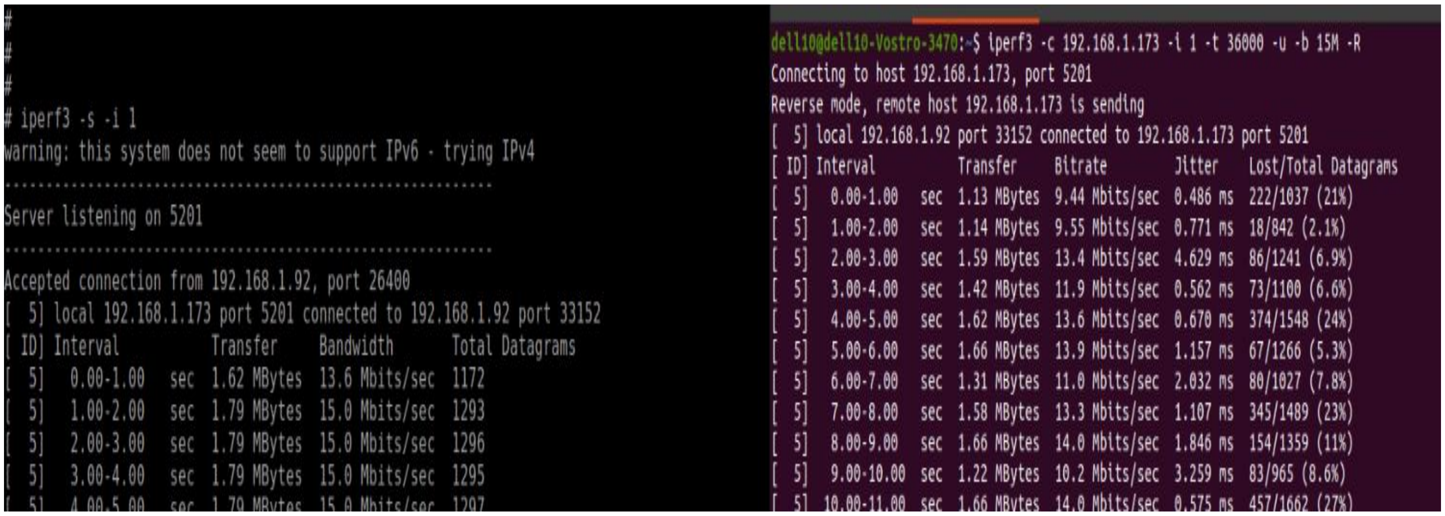
```
# ./conmgr status
op:13:1887504992
Waiting for Response
 status
mode : STA
status:associated
ssid:ct_asus
bssid:24-4b-fe-e3-ec-60
channel:6
rssi:-11
IPv4 Address:192.168.1.173
Subnet Mask:255.255.255.0
Default Gateway:192.168.1.1
Dns:192.168.1.1
Security:WPA2-PSK
Heap remaining:179928
T2 powersave:0
#
```

Figure 15: ./conmgr status – output

Step 2: From the Linux host machine, start the UDP client using the following command, connect to the Talaria TWO UDP server of IP address 192.168.1.173 and port 5201. Once the client connects, Talaria TWO will start sending the data over UDP socket:

```
iperf3 -c <Ipaddress> -i 1 -t 36000 -u -b 15M -R
```

Output:



```
#
#
#
# iperf3 -s -i 1
warning: this system does not seem to support IPv6 - trying IPv4
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
Accepted connection from 192.168.1.92, port 26400
[  5] local 192.168.1.173 port 5201 connected to 192.168.1.92 port 33152
[ ID] Interval           Transfer     Bandwidth       Total Datagrams
[  5]   0.00-1.00   sec  1.62 MBytes  13.6 Mbits/sec  1172
[  5]   1.00-2.00   sec  1.79 MBytes  15.0 Mbits/sec  1293
[  5]   2.00-3.00   sec  1.79 MBytes  15.0 Mbits/sec  1296
[  5]   3.00-4.00   sec  1.79 MBytes  15.0 Mbits/sec  1295
[  5]   4.00-5.00   sec  1.79 MBytes  15.0 Mbits/sec  1297
```

```
dell10@dell10-Vostro-3470:~$ iperf3 -c 192.168.1.173 -i 1 -t 36000 -u -b 15M -R
Connecting to host 192.168.1.173, port 5201
Reverse mode, remote host 192.168.1.173 is sending
[  5] local 192.168.1.92 port 33152 connected to 192.168.1.173 port 5201
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total Datagrams
[  5]   0.00-1.00   sec  1.13 MBytes  9.44 Mbits/sec  0.486 ms  222/1037 (21%)
[  5]   1.00-2.00   sec  1.14 MBytes  9.55 Mbits/sec  0.771 ms  18/842 (2.1%)
[  5]   2.00-3.00   sec  1.59 MBytes  13.4 Mbits/sec  4.629 ms  86/1241 (6.9%)
[  5]   3.00-4.00   sec  1.42 MBytes  11.9 Mbits/sec  0.562 ms  73/1100 (6.6%)
[  5]   4.00-5.00   sec  1.62 MBytes  13.6 Mbits/sec  0.670 ms  374/1548 (24%)
[  5]   5.00-6.00   sec  1.66 MBytes  13.9 Mbits/sec  1.157 ms  67/1266 (5.3%)
[  5]   6.00-7.00   sec  1.31 MBytes  11.0 Mbits/sec  2.032 ms  80/1027 (7.8%)
[  5]   7.00-8.00   sec  1.58 MBytes  13.3 Mbits/sec  1.107 ms  345/1489 (23%)
[  5]   8.00-9.00   sec  1.66 MBytes  14.0 Mbits/sec  1.846 ms  154/1359 (11%)
[  5]   9.00-10.00  sec  1.22 MBytes  10.2 Mbits/sec  3.259 ms  83/965 (8.6%)
[  5]  10.00-11.00  sec  1.66 MBytes  14.0 Mbits/sec  0.575 ms  457/1662 (27%)
```

Figure 16: Starting UDP Client

Support

1. Sales Support: Contact an InnoPhase sales representative via email – sales@innophaseiot.com
2. Technical Support:
   a. Visit: https://innophaseiot.com/contact/
   b. Also Visit: https://innophaseiot.com/talaria-two-modules/
   c. Contact: support@innophaseiot.com

InnoPhase is working diligently to provide customers outstanding support to all customers.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, InnoPhase IoT Incorporated does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and assumes no liability associated with the use of such information. InnoPhase IoT Incorporated takes no responsibility for the content in this document if provided by an information source outside of InnoPhase IoT Incorporated.

InnoPhase IoT Incorporated disclaims liability for any indirect, incidental, punitive, special or consequential damages associated with the use of this document, applications and any products associated with information in this document, whether or not such damages are based on tort (including negligence), warranty, including warranty of merchantability, warranty of fitness for a particular purpose, breach of contract or any other legal theory. Further, InnoPhase IoT Incorporated accepts no liability and makes no warranty, express or implied, for any assistance given with respect to any applications described herein or customer product design, or the application or use by any customer's third-party customer(s).

Notwithstanding any damages that a customer might incur for any reason whatsoever, InnoPhase IoT Incorporated' aggregate and cumulative liability for the products described herein shall be limited in accordance with the Terms and Conditions of identified in the commercial sale documentation for such InnoPhase IoT Incorporated products.

Right to make changes — InnoPhase IoT Incorporated reserves the right to make changes to information published in this document, including, without limitation, changes to any specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — InnoPhase IoT Incorporated products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an InnoPhase IoT Incorporated product can reasonably be expected to result in personal injury, death or severe property or environmental damage. InnoPhase IoT Incorporated and its suppliers accept no liability for inclusion and/or use of InnoPhase IoT Incorporated products in such equipment or applications and such inclusion and/or use is at the customer's own risk.

All trademarks, trade names and registered trademarks mentioned in this document are property of their respective owners and are hereby acknowledged.