After the hardware setup (refer section:[Getting Started](#_Getting_Started)), follow the subsequent steps to run the AWS KINESIS video streaming application. Use any serial port terminal application such as GTKTerm to issue the commands on T31z INP3201 Host.

**Note**: The video streaming set-up supports streaming from a single web connection/browser at a given time.

**Method 1: Manual Setup**

1. Insert the kernel modules using the following command:

|  |
| --- |
| insmod mmc\_detect\_test.ko  insmod sdio-wlan.ko |

1. Start the tunadapter in the background.

|  |
| --- |
| ./tunadapter board\_conf=t31\_innophase.conf & |

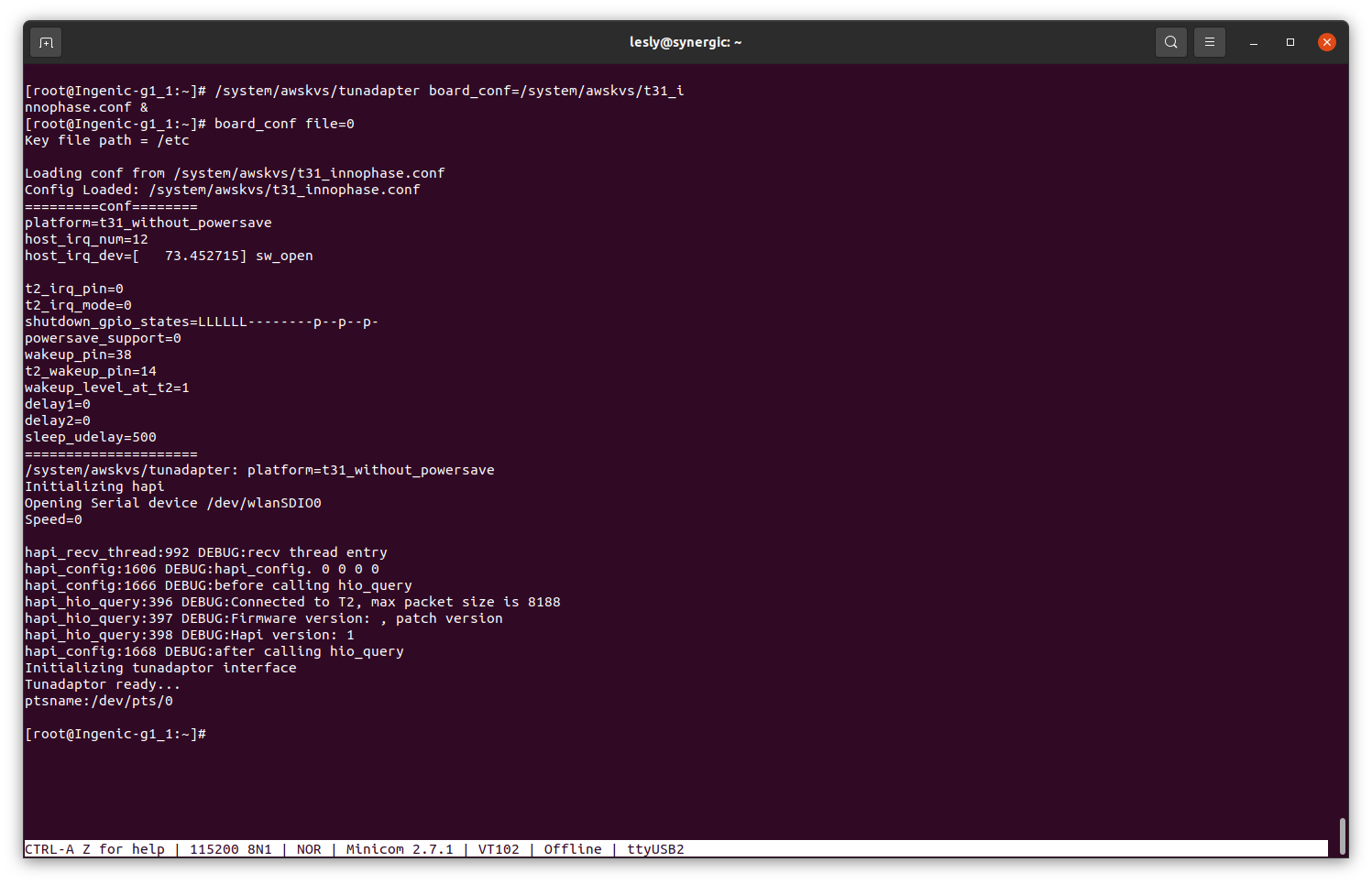


Figure 1: Start tunadapter

1. Connect Talaria TWO to an Access Point. This can be achieved using either one of the following methods:
   1. Execute the following set of commands:
      1. Start provisioning:

|  |
| --- |
| ./conmgr provstart <device\_name> <manufacturer\_name> |

* + 1. Get provisioning information (SSID and passphrase of the Access Point):

|  |
| --- |
| ./conmgr waitforprovinfo |

* + 1. Stop provisioning:

|  |
| --- |
| ./conmgr provstop |

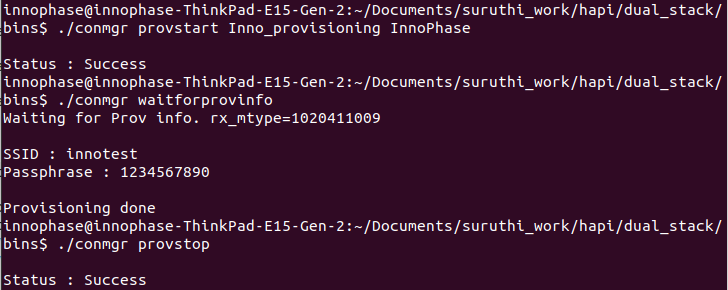


Figure 2: Enabling Wi-Fi provisioning over BLE

**Note**: <device\_name> will be visible in the BLE scan list on the mobile app.

* 1. Enable Wi-Fi provisioning over BLE using the [AWS\_Streaming APP](#_AWS_Streaming_APP). It can be used to enable Wi-Fi provisioning over BLE and stream video.

For more details on passing the SSID and passphrase through the AWS\_Sreaming mobile app, refer section: *Running the Application using Android or iOS app* from *Example\_using\_Provisioning.pdf* (*freertos\_sdk\_x.y\examples\prov\doc*).

**Note**: x and y in freertos\_sdk\_x.y refer to the SDK package release version.

* 1. **Alternate method**: Pass the SSID and passphrase of the appropriate AP by executing the following command:

|  |
| --- |
| ./conmgr connect <ssid> <password> |

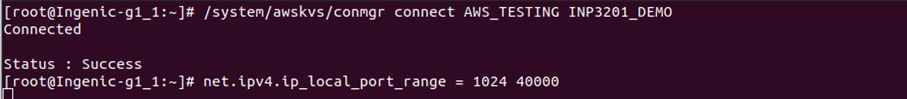


Figure 3: Connect to AP

1. Run the NTP service from T31z host:

|  |
| --- |
| [root@Ingenic-g1\_1:ces\_master]# ntpd -nqp pool.ntp.org |



Figure 4: Run NTP service

1. To start the video streaming, insert kernel modules and start RTSP server on T31z host:

|  |
| --- |
| insmod /lib/modules/tx-isp-t31.ko  insmod /lib/modules/sensor\_gc2053\_t31.ko data\_interface=1  insmod /lib/modules/avpu.ko clk\_name='vpll' avpu\_clk=400000000  insmod /lib/modules/audio.ko |

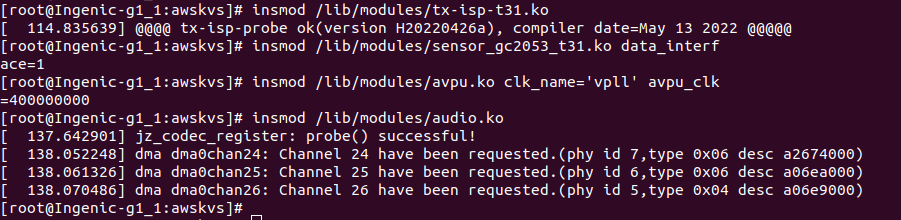


Figure 5: Insert ko modules

1. The following script configures the AWS access key, secret access key, region and starts streaming:

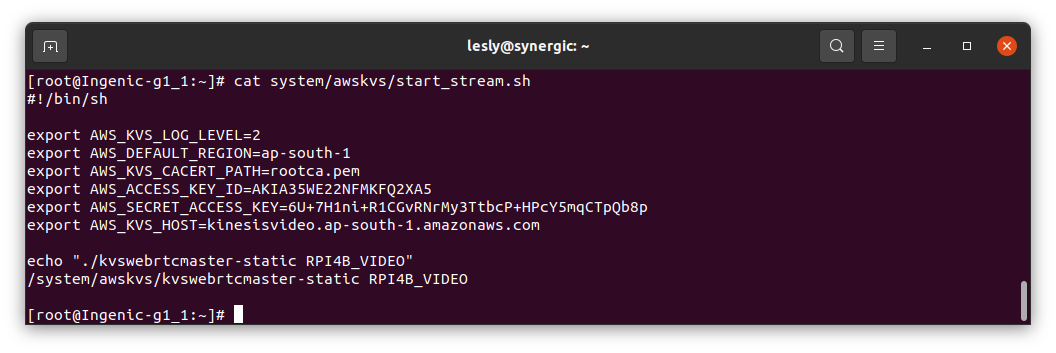


Figure 6: Start kinesis video streaming

**Note**: For information on creating the AWS security credentials (access key, secret access key), refer [Create AWS Security Credentials](#_Create_AWS_Security).

**Method 2 – Automated Shell Scripts**

1. Execute the init script file to automatically load kernel drivers, connect Wi-Fi, and start the AWS KINESIS Video Streaming after booting.

**Note**:

* + 1. Change the SSID and passphrase in app\_init.sh to the required configuration.
    2. Steaming duration is configurable. Modify the same in app\_init.sh for the required duration.

Use Delay=0 for uninterrupted streaming without entering the low-power mode.

|  |
| --- |
| ./T31\_mcu\_app <Delay 0 to 300 sec> <Trun OFF devices 1= T31, 3= T31 & T2 > |

* + 1. Configure Talaria TWO Wi-Fi wakeup from MCU.

|  |
| --- |
| ./conmgr wakeupconfig <T2 wakeup GPIO pin> <GPIO level> <GPIO direction 0=output / 1= input> |

* + 1. Configure MCU wakeup from Talaria TWO.

|  |
| --- |
| ./conmgr wakeupconfig <MCU wakeup GPIO pin> <GPIO level> |

* + 1. Initialize the MQTT connection.

|  |
| --- |
| ./sc scinit <server name> <port number> <tranport> <mqtt id> <usr name> <password> [ca cert] [client cert] [client key] |

* + 1. Close the MQTT connection.

|  |
| --- |
| ./sc scdeinit |

* + 1. Update the streaming status.

|  |
| --- |
| ./sc scstatusupdate < status : 1/0 > |

where,

1- ON, 0 - OFF

app\_init.sh script for INP3201 board with MCU loads the required kernel driver modules, starts the MQTT connection and turns the camera ON. Kinesis video streaming is then started for a specified time. The device is put to sleep mode in case of inactivity on the PIR sensor connected to the MCU.

**Note**: To put the INP3201 board with MCU in low-power mode, execute the following steps:

* + 1. Start tunadapter in the background and connect to the AP:

|  |
| --- |
| ./tunadapter board\_conf =./t31.conf &  ./conmgr connect <ssid> <Pass> |

* + 1. Initiate Host shutdown:
       1. Provide the number of seconds for which streaming needs to be enabled:

|  |
| --- |
| ./T31\_mcu\_app <Delay> 1 |

* + - 1. Configure the Host wakeup PIN:

|  |
| --- |
| ./conmgr wakeupconfig 14 1 1  ./conmgr wakeupconfig 21 0 |

**Note**: For more information on configuring the Host wakeup PIN, refer section: *wakeupconfig – Configures Host Wakeup PIN* of UG\_Dual-Stack.pdf (*\talaria\_two\_dual\_stack\talaria\_two\_dual\_stack\_vx.y\doc*)

* + - 1. Get the previous wakeup reason:

|  |
| --- |
| ./conmgr getwakeupreason |

* + - 1. Clear wakeup reason before initiating shutdown:

|  |
| --- |
| ./conmgr wakeupreasoncls |

* + - 1. Trigger Host shutdown:

|  |
| --- |
| ./conmgr shutdown |

1. The following script configures the AWS access key, secret access key, region and starts streaming:

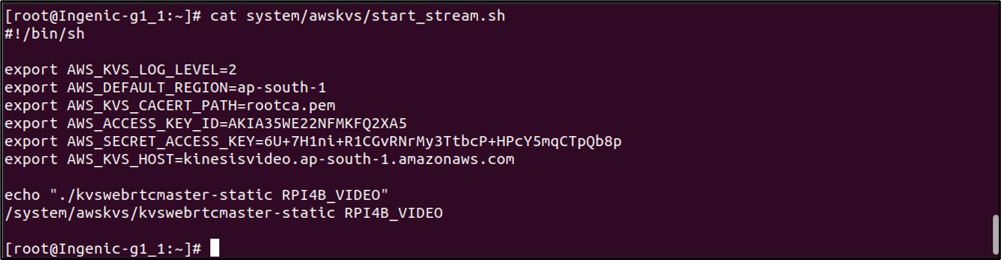


Figure 7: Script to configure AWS access parameters

**Note**: For information on creating the AWS security credentials (access key, secret access key), refer [Create AWS Security Credentials](#_Create_AWS_Security).