



SAMSUNG
ARTIK™ Modules

Samsung Training Lab
ARTIK Gateway Modules

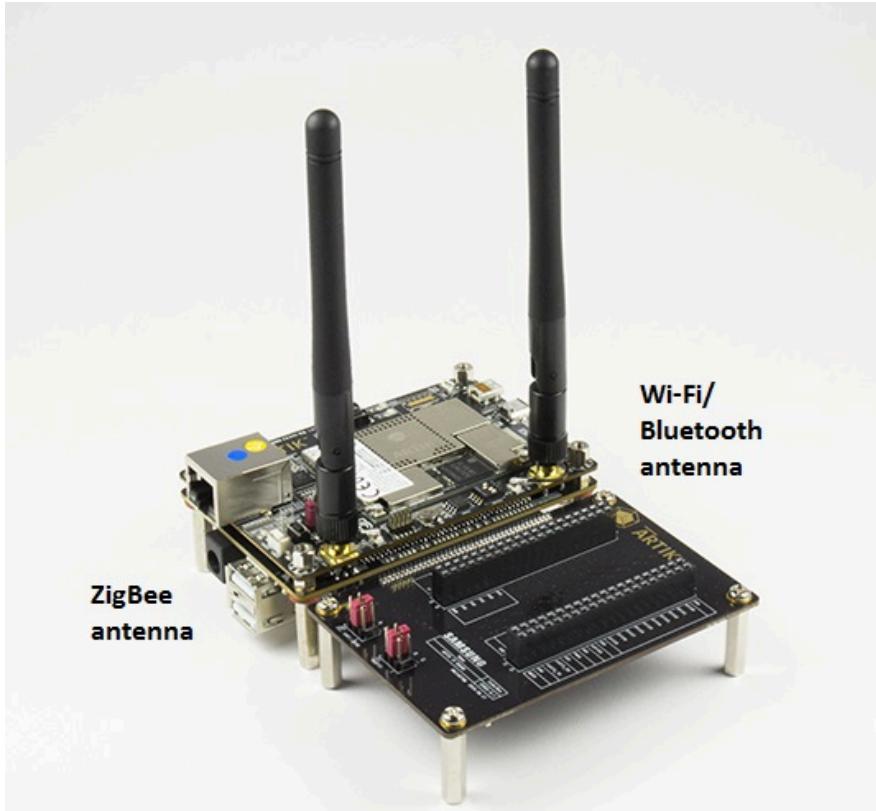
TABLE OF CONTENTS

ARTIK530s Set Up	2
ARTIK530s Hardware setup	2
Setting up the Terminal Console.....	3
ARTIK530s Activation	3
Configure WiFi on ARTIK 530s.....	4
Lab 6: MQTT (C/JavaScript).....	6
6.1 Publish MQTT message from ARTIK055s.....	6
6.2 Subscribe to MQTT message from ARTIK530s.....	8
Lab 7: Running Google Assistant on ARTIK 530 (Python).....	10
7.1 Configure and Test the Audio on ARTIK 530.....	10
7.2 Configure a Google Developer Project	10
Configure a Google Developer Project	10
Set activity controls for your account	12
7.3 Install the SDK and Sample Code	13
7.4 Register the Device Model.....	14
7.5 Run the sample app.....	14
Lab 8: Build a ZigBee network by using ARTIK SDK	16

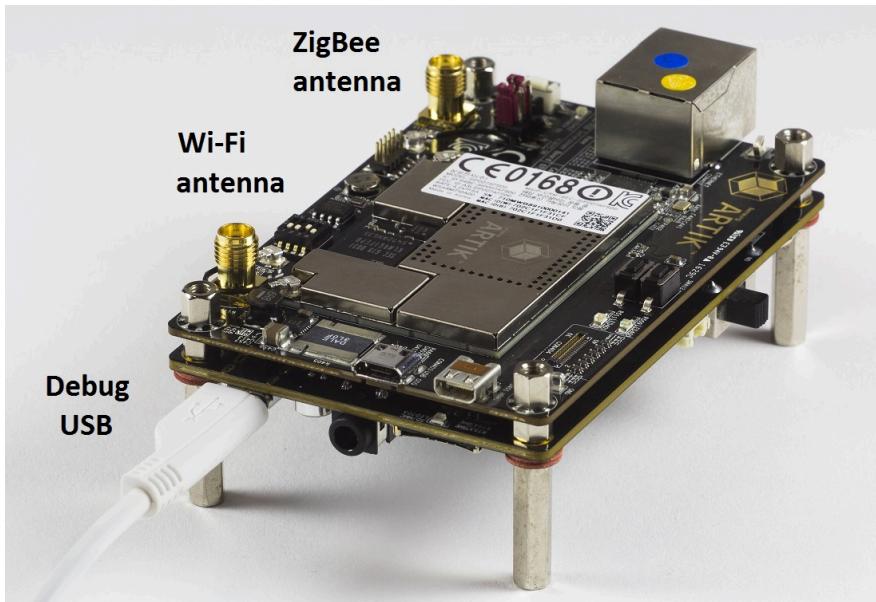
ARTIK530s Set Up

ARTIK530s Hardware setup

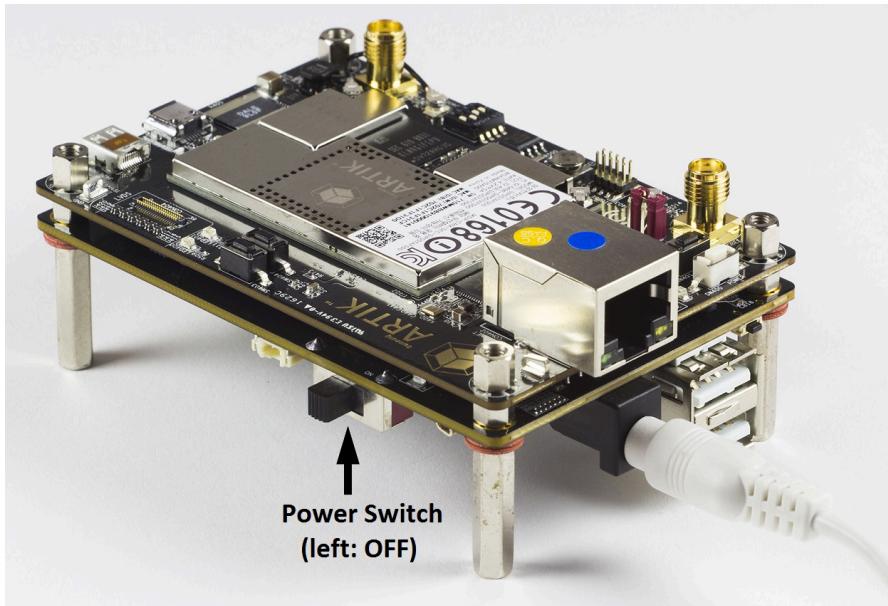
1. We provide two antennas for ARTIK530s developer kit, one for ZigBee and one for WiFi. They are identical and interchangeable. Please connect the interface board and mount your antennas as shown below:



2. Plug in USB cable for serial access. Connect the other end of the USB cable to your laptop.



3. Make sure the power switch on the development board is set to the off position. Plug in the power supply to the development board.



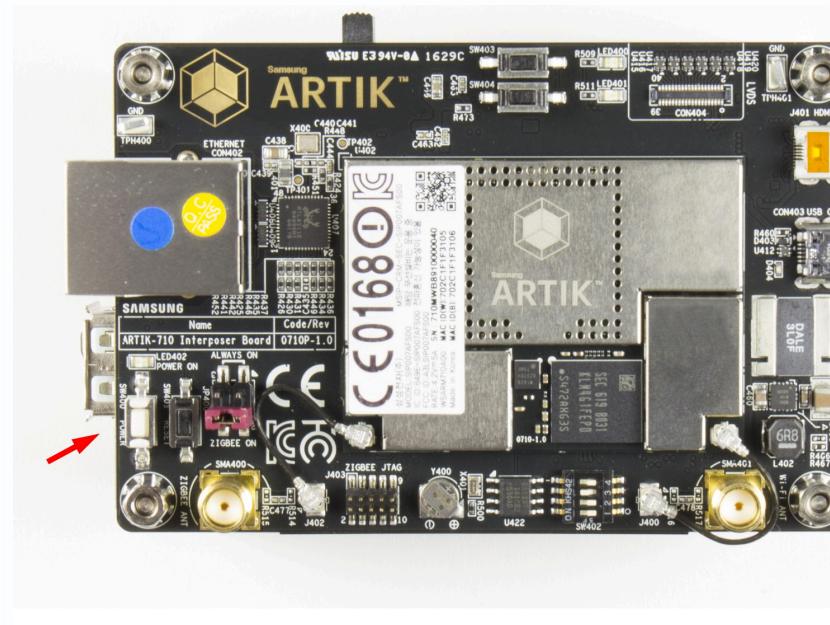
Setting up the Terminal Console

Please follow this tutorial to set up your terminal console.

<https://developer.artik.io/documentation/developer-guide/artik-ide/serial-debug.html>

ARTIK530s Activation

After you first apply power, you'll need to activate the board with a 1-second reset pulse using the POWER push-button switch, located as shown for ARTIK530 board.



Configure WiFi on ARTIK 530s

In this step, we will configure the `wpa_supplicant.conf` file to include your WiFi router settings. Please replace SSID and PASSWORD in the first command below by using your own configuration.

Use `'cat /etc/wpa_supplicant/wpa_supplicant.conf'` to make sure your settings have been included into the configuration file properly.

```
[root@artik ~]# wpa_passphrase "SSID" "PASSWORD" >>
/etc/wpa_supplicant/wpa_supplicant.conf

[root@artik ~]# cat /etc/wpa_supplicant/wpa_supplicant.conf
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=netdev

network={
    ssid="SSID"
    #psk="PASSWORD"
    psk=c2161655c6ba444d8df94cbbf4e9c5c4c61fc37702b9c66ed37aee1
545a5a333
}
```

Restart `wpa_supplicant` and use `dhclient` to obtain an IP address

```
[root@artik ~]# systemctl start wpa_supplicant
[root@artik ~]# dhclient wlan0
```

Now, your ARTIK530s board should have an IP address associated with `wlan0` interface. Take a note of your ARTIK530s `wlan0` IP address.

```
[root@artik ~]# ifconfig wlan
wlan0      Link encap:Ethernet  HWaddr 00:50:43:02:fe:01
            inet addr:10.0.1.51  Bcast:10.0.1.255
Mask:255.255.255.0
            inet6 addr: fe80::250:43ff:fe02:fe01/64 Scope:Link
            inet6 addr: 2601:646:8881:4923:250:43ff:fe02:fe01/64
Scope:Global
            inet6 addr: 2601:646:8881:4923:99af:2c8:df37:ceca/64
Scope:Global
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:51 errors:0 dropped:0 overruns:0 frame:0
            TX packets:79 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:7177 (7.1 KB)  TX bytes:11851 (11.8 KB)
```

Run ping command to make sure your WiFi connection is live.

```
[root@artik ~]# ping www.google.com
PING www.google.com (172.217.11.68) 56(84) bytes of data.
64 bytes from lax17s34-in-f4.1e100.net (172.217.11.68):
icmp_seq=1 ttl=53 time=21.2 ms
64 bytes from lax17s34-in-f4.1e100.net (172.217.11.68):
icmp_seq=2 ttl=53 time=20.6 ms
64 bytes from lax17s34-in-f4.1e100.net (172.217.11.68):
icmp_seq=3 ttl=53 time=23.1 ms
64 bytes from lax17s34-in-f4.1e100.net (172.217.11.68):
icmp_seq=4 ttl=53 time=19.6 ms
64 bytes from lax17s34-in-f4.1e100.net (172.217.11.68):
icmp_seq=5 ttl=53 time=19.8 ms
```

Lab 6: MQTT (C/JavaScript)

In this lab, we are going to use ARTIK055s as the MQTT client to publish messages to ARTIK 530s gateway. ARTIK 530s has a MQTT broker running on it listening for incoming messages, at the same time, it streams the sensor data to ARTIK Cloud and receives actions triggered from rules engine.

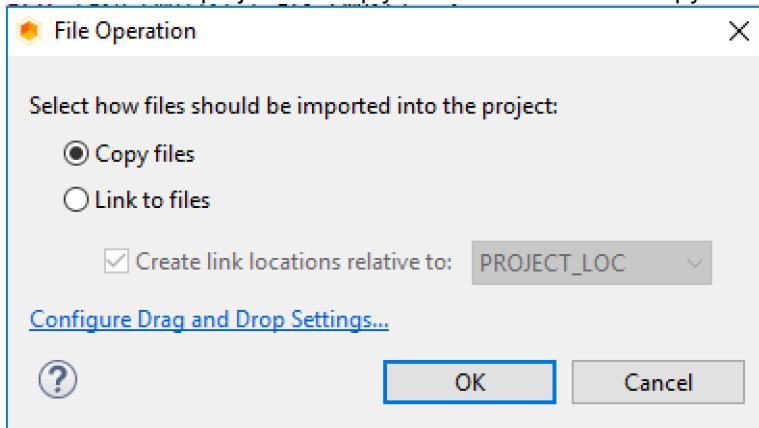
Mosquitto should be running on ARTIK530s already when the boots up.

6.1 Publish MQTT message from ARTIK055s

1. Add the 2 attached files to artik055s app project from the IDE. The 2 files are:

- wifi.c
- mqtt.c

You can simply drag and drop the files to your artik055s project. A message will pop up asking if you want to copy the files into the project or simply link to them. Select 'Copy files' and click OK.



2. Extend example-api.c to include 2 additional TASH shell commands wifi and mqtt for WiFi configuration and MQTT publication.

```
extern int artik_onboarding_main(int argc, char *argv[]);
extern int security_main(int argc, char *argv[]);
extern int artik_wifi_main(int argc, char *argv[]);
extern int artik_mqtt_main(int argc, char *argv[]);

static task_cmdlist_t atk_cmds[ ] = {
    {"onboard", artik_onboarding_main, TASH_EXECMD_SYNC},
    {"security", security_main, TASH_EXECMD_SYNC},
    {"wifi", artik_wifi_main, TASH_EXECMD_SYNC},
    {"mqtt", artik_mqtt_main, TASH_EXECMD_SYNC},
    {NULL, NULL, 0}
};
```

- Configure WiFi SSID and password. In wifi.c, replace WiFi SSID and password with your own.

```
/*
 *-----*/
/*User Required Defines
 *-----*/
-----*/
#define SSID "YOUR_SSID_GOES_HERE"
#define PSK "YOUR_WIFI_PASSWORD_GOES_HERE"
```

- Compile the project and re-flash your board.
- Reboot ARTIK055s, use wifi TASH command to automatically configure WiFi.

```
TASH>>wifi
Starting supplicant in foreground...
Connect Wi-Fi success

TASH>>ifconfig w1
w1      Link encap: 28:6d:97:40:94:78      RUNNING: UP          MTU: 1500
        inet addr: 10.0.1.22    Mask: 255.255.255.0

lo0     Link encap: 00:00:00:00:00:00      RUNNING: DOWN        MTU: 0
        inet addr: 127.0.0.1    Mask: 255.0.0.0
        inet6 addr: ::1
```

- Use MQTT APIs to connect to MQTT broker.

MQTT command line options are as below:

```
TASH>> mqtt
usage:
    mqtt connect - connect <host> <port> <user> <password>
    mqtt disconnect - disconnect
    mqtt publish - publish
    mqtt subscribe - subscribe <topic>
    mqtt unsubscribe - unsubscribe <topic>
```

First, we need to establish the connection to MQTT broker. Replace the IP address with the wlan0 address of your ARTIK530s board. Default port number is 1883. Username and password are NULL.

```
TASH>>mqtt connect mqtt://10.0.1.51 1883 NULL NULL
Client mosq/_bHDlMgXavuA@5\mN? sending CONNECT
TASH>>Client mosq/_bHDlMgXavuA@5\mN? received CONNACK
MQTT connection result: OK
```

- Publish telemetry data to MQTT broker by entering 'mqtt publish'

6.2 Subscribe to MQTT message from ARTIK530s (JavaScript)

On your ARTIK530s /root directory, there is an mqtt.js file.

```
[root@artik ~]# cd /home/
[root@artik ~]# nodejs mqtt.js YOUR_DEVICE_INSTANCE_DEVICE_ID
YOUR_DEVICE_INSTANCE_DEVICE_TOKEN

Device ID:7a1fb71xxxxxxxxxxxxxx
Device Token: 72c660c6363xxxxxxxxx34908
Registered
```

6.3 Publish telemetry data from ARTIK 055s to ARTIK 530s

On ARTIK055s, publish telemetry data to MQTT broker by entering ‘mqtt publish’. It will collect distance sensor data at a 10 seconds interval and publishes to the broker.

```
TASH>> mqtt publish
ADC0=1596
MQTT: publish 1596 on ARTIKTraining
Client mosq/_bHDlMgXavuA@5\mN? sending PUBLISH (d0, q0, r0,
m1, 'ARTIKTraining', ... (4 bytes))
MQTT message published: 1
ADC0=714
MQTT: publish 714 on ARTIKTraining
Client mosq/_bHDlMgXavuA@5\mN? sending PUBLISH (d0, q0, r0,
m2, 'ARTIKTraining', ... (3 bytes))
MQTT message published: 2
ADC0=1716
MQTT: publish 1716 on ARTIKTraining
Client mosq/_bHDlMgXavuA@5\mN? sending PUBLISH (d0, q0, r0,
m3, 'ARTIKTraining', ... (4 bytes))
MQTT message published: 3
ADC0=820
MQTT: publish 820 on ARTIKTraining
Client mosq/_bHDlMgXavuA@5\mN? sending PUBLISH (d0, q0, r0,
m4, 'ARTIKTraining', ... (3 bytes))
MQTT message published: 4
ADC0=1021
MQTT: publish 1021 on ARTIKTraining
Client mosq/_bHDlMgXavuA@5\mN? sending PUBLISH (d0, q0, r0,
m5, 'ARTIKTraining', ... (4 bytes))
MQTT message published: 5
```

On ARTIK530s console, you should be able to see the log as below:

```
[root@artik ~]# cd /home/  
[root@artik ~]# nodejs mqtt.js YOUR_DEVICE_INSTANCE_DEVICE_ID  
YOUR_DEVICE_INSTANCE_DEVICE_TOKEN  
  
Device ID:7a1fb71xxxxxxxxxxxxxx  
Device Token: 72c660c6363xxxxxxxxx34908  
Registered  
Subscribed to MQTT broker  
publish sensor data 516 to ARTIK Cloud  
publish sensor data 468 to ARTIK Cloud  
publish sensor data 636 to ARTIK Cloud  
publish sensor data 608 to ARTIK Cloud  
publish sensor data 2543 to ARTIK Cloud  
publish sensor data 777 to ARTIK Cloud
```

Lab 7: Running Google Assistant on ARTIK 530 (Python)

7.1 Configure and Test the Audio on ARTIK 530

As the 1st step, let's verify that recording and playback work on ARTIK 530:

1. Plug in your earplugs into ARTIK530 and play a test sound. From your terminal window, run

```
[root@artik ~]# speaker-test -t wav
```

Press the Ctrl + C when done.

If the volume is too low, you can adjust it by using alsamixer. Press the up arrow key to set the volume. Press ESC when you are done.

```
[root@artik ~]# alsamixer
```

2. Record a short audio clip and play it back.

```
[root@artik~]#arecord --format=S16_LE --duration=5 --rate=16000 --file-type=raw out.raw
[root@artik~]#aplay --format=S16_LE --rate=16000 out.raw
```

7.2 Configure a Google Developer Project

Configure a Google Developer Project

1. In the Cloud Platform Console, go to the Projects page <https://console.cloud.google.com/cloud-resource-manager>. Select an existing project or create a new one.

The screenshot shows the Google Cloud Platform interface with the URL <https://console.cloud.google.com/cloud-resource-manager> in the address bar. The main header says "Google Cloud Platform". Below it, there are buttons for "Manage resources", "+ CREATE PROJECT", and "DELETE". A search bar is also present. The main content area displays a table of projects. The table has columns for "Project name" and "Project ID". There are checkboxes next to each project name. The first project listed is "ARTIK Sensor Dashboard" with "Project ID" "artik-sensor-dashboard". The second project listed is "GoogleAssistant" with "Project ID" "basic-tube-170418". This row is highlighted with a red rectangle. The third project listed is "SmartTrashCan" with "Project ID" "smartrashcan-1261". There are three vertical ellipsis buttons on the far right of the table.

Project name	Project ID
ARTIK Sensor Dashboard	artik-sensor-dashboard
GoogleAssistant	basic-tube-170418
SmartTrashCan	smartrashcan-1261

2. Enable Google Assistant API.

Google Assistant API
Google
Google Assistant API
MANAGE API enabled

Type	Overview
APIs & services	Google Assistant API
Last updated	6/29/17, 4:01 PM
Service name	embeddedassistant.googleapis.com

Google's mission is to organize the world's information and make it universally accessible and useful. Through products and platforms like Search, Maps, Gmail, Android, Google Play, Chrome and YouTube, Google plays a meaningful role in the daily lives of billions of people.

3. Create an OAuth Client ID.

Credentials OAuth consent screen Domain verification

Create credentials Delete

- API key** Identifies your project using a simple API key to check quota and access
- OAuth client ID** Requests user consent so your app can access the user's data
- Service account key** Enables server-to-server, app-level authentication using robot accounts
- Help me choose** Asks a few questions to help you decide which type of credential to use

3.2 Select Other as the Application type, and give your client a name. Select 'Create'.

The screenshot shows the Google APIs console interface. On the left, there's a sidebar with icons for Application type, Name, and Create. The 'Application type' section is expanded, showing options like Web application, Android, Chrome App, iOS, PlayStation 4, and Other. 'Other' is selected. Below that is a 'Name' field containing 'Google Assistant Client'. At the bottom are 'Create' and 'Cancel' buttons.

A dialog box pops up that shows you a client ID and secret. You don't have to remember them. Simply close the dialog.

3.3 Click (at the far right of screen) for the client ID to download the client secret JSON file (client_secret_<client-id>.json). Do not rename the file.

The screenshot shows the 'Credentials' tab in the Google APIs console. It lists a single OAuth 2.0 client ID named 'Google Assistant Client' with a creation date of Mar 7, 2018. The Client ID is 228723809066-reijdmh6bqddsv2t0gae9mequ4h2d4kc.apps.googleusercontent.com. The 'Download' icon (a small arrow pointing down) is highlighted with a red box.

3.4 Copy the file over to your ARTIK /home directory. For Windows users, you can use a SCP client like FileZilla. For Mac users, you can simply run scp command from your host machine. Replace ARTIK-IP-ADDRESS by using the IP address of your ARTIK board.

```
[root@artik~]#scp client_secret_<client-id>.json root@ARTIK-IP-ADDRESS:/home
```

ADDRESS by using the IP address of your ARTIK board.

Set activity controls for your account

Open the Activity Controls page of your Google Account, and enable the options below.

- Web & App Activity: In addition, be sure to select the **Include Chrome browsing history and activity from websites and apps that use Google services** checkbox.
- Device Information

- Voice & Audio Activity

7.3 Install the SDK and Sample Code

1. Install a Python virtual environment to isolate the SDK and its dependencies from the system Python packages.

```
[root@artik home]#apt-get install python-dev python-virtualenv virtualenv
[root@artik home]#virtualenv env --no-site-packages
[root@artik home]#env/bin/python -m pip install --upgrade pip setuptools
wheel
[root@artik home]#source env/bin/activate
(env) [root@artik home]#
```

2. Install package dependencies and the latest version of the Python package in the virtual environment.

The last step will take 15 minutes or so. Be patient!

```
(env) [root@artik home]# sudo apt-get install portaudio19-dev libffi-dev libssl-dev
(env) [root@artik home]# python -m pip install --upgrade google-assistant-library
(env)[root@artik home]# python -m pip install --upgrade google-assistant-
sdk[samples]
```

3. Generate credentials

```
(env) [root@artik home]# python -m pip install --upgrade google-auth-
oauthlib[tool]

(env) [root@artik home]# google-oauthlib-tool --scope
https://www.googleapis.com/auth/assistant-sdk-prototype --save --headless -
-client-secrets /home/client_secret_client_id.json

Please visit this URL to authorize this application:
https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=..

Enter the authorization code: 4/Md1uiKkfkqTKmb0xRY-O_3Ade8X6S5pIAdjMTk8ZHyy
credentials saved: /root/.config/google-oauthlib-tool/credentials.json
```

You should see a URL displayed. Open the URL in a browser, then copy and past the authorization code to your terminal. If authorization was successful, you will see a response saying credentials saved.

```
Enter the authorization code: 4/Md1uiKkfkqTKmb0xRY-O_3Ade8X6S5pIAdjMTk8ZHyy
credentials saved: /root/.config/google-oauthlib-tool/credentials.json
```

7.4 Register the Device Model

1. Register your device by using googlesamples-assistant-devicetool.

```
(env) [root@artik home]# googlesamples-assistant-devicetool register-model --manufacturer my-manufacturer --product-name my-product-name [--description my-product-description] --type device-type [--trait supported-trait] --model my-model
```

An example of this is:

```
(env) [root@artik home]# googlesamples-assistant-devicetool register-model --manufacturer "Samsung" --product-name "ARTIK" --type SWITCH --model "ARTIK530"  
Creating new device model  
Model ARTIK530 successfully registered
```

2. You can query your device model by using the command below:

```
(env) [root@artik home]# googlesamples-assistant-devicetool get --model ARTIK530
```

7.5 Run the sample app



```
(env) [root@artik home]# googlesamples-assistant-pushtotalk --device-model-id ARTIK530
INFO:root:Connecting to embeddedassistant.googleapis.com
Press Enter to send a new request...
INFO:root:Recording audio request.
INFO:root:Recording audio request.
INFO:root:Transcript of user request: "hey".
INFO:root:Playing assistant response.
INFO:root:Transcript of user request: "hey goo".
INFO:root:Playing assistant response.
INFO:root:Transcript of user request: "hey Google".
INFO:root:Playing assistant response.
INFO:root:Transcript of user request: "hey Google".
INFO:root:Playing assistant response.
INFO:root:Transcript of user request: "hey Google what".
INFO:root:Playing assistant response.
INFO:root:Transcript of user request: "hey Google what time".
INFO:root:Playing assistant response.
INFO:root:Transcript of user request: "hey Google what time is".
INFO:root:Playing assistant response.
INFO:root:Transcript of user request: "hey Google what time is it".
INFO:root:Playing assistant response.
INFO:root:Transcript of user request: "hey Google what time is it".
INFO:root:Playing assistant response.
INFO:root:Transcript of user request: "hey Google what time is it in".
INFO:root:Playing assistant response.
INFO:root:Transcript of user request: "hey Google what time is it now".
INFO:root:Playing assistant response.
INFO:root:Transcript of user request: "hey Google what time is it now".
INFO:root:Playing assistant response.
INFO:root:Transcript of user request: "hey Google what time is it now".
INFO:root:Playing assistant response.
INFO:root:End of audio request detected
INFO:root:Transcript of user request: "hey Google what time is it now".
INFO:root:Playing assistant response.
INFO:root:Finished playing assistant response.
```



Bonus Question

Use voice commands to control the GPIOs on ARTIK055s or trigger other actions.

Hints: On ARTIK 055s, you can use ‘mqtt subscribe’ to subscribe to the messages from the MQTT broker running on ARTIK530s.

Lab 8: Build a ZigBee network by using ARTIK SDK

In this exercise, we are going to use ARTIK SDK to establish a ZigBee network. One device acts as a ZigBee coordinators and other devices will act as ZigBee routers.

1. Compile the ZigBee sample code, provided as part of ARTIK SDK

```
[root@localhost ~]# gcc zigbee_test.c zigbee_test_common.c -o zb_test -I/usr/include/zigbee -I/usr/include/glib-2.0 -I/usr/lib/arm-linux-gnueabihf/glib-2.0/include -lzigbee -lgobject-2.0 -lglib-2.0 -lm
```

2. Add ZigBee devices

2.1 ZigBee coordinators, please follow the instructions in 3.1.

ZigBee routers, please skip to step 3.2.

You will see a menu as shown above. Enter **1** to add an On/Off Switch, press <Enter> to accept the default endpoint id. Once "On/Off Switch" device is added, enter **0** to finish adding devices.

```
[root@localhost ~]# ./zb_test
=====
1: On/Off Switch
2: Level Control Switch
3: On/Off Light
4: Dimmable Light
5: Light Sensor
6: Occupancy Sensor
7: Temperature Sensor
8: Remote Control
9: IAS Control and Indication Equipment
10: Thermostat client
11: Thermostat server
0: ADD DEVICE DONE
Add device: 1
Set endpoint id (1): (Press <Enter> to use the default value)
Added device "On/Off Switch" with ep(1)
```

```
Add device: 0
```

2.2 ZigBee coordinators, please stop here.

```
[root@localhost ~]# ./zb_test=====
1: On/Off Switch
2: Level Control Switch
3: On/Off Light
4: Dimmable Light
5: Light Sensor
6: Occupancy Sensor
7: Temperature Sensor
8: Remote Control
9: IAS Control and Indication Equipment
10: Thermostat client
11: Thermostat server
0: ADD DEVICE DONE
Add device: 3
Set endpoint id (19): 20 (Every router has to use different endpoint id.)
Added device "On/Off Light" with ep(20)

Add device: 0
```

Enter **3** to add an On/Off Light device, press <Enter> to accept the default endpoint id. Once “On/Off Light” device is added, enter **0** to finish adding devices.

3. Manually form a ZigBee network (**Only ZigBee coordinators need to do this. Routers please wait coordinators' permission to join a network, then go to step 5.2**)

From 2.1 above, when coordinators hit <Enter> after adding device, coordinator will see the menu below. Initial state shows ZIGBEE_NO_NETWORK. Enter **2** to manually form a ZigBee network.

```
Add device: 0
State: ZIGBEE_NO_NETWORK
Network: Non Exist
=====
1: Form network
2: Form network (advanced)
3: Join network
4: Join network (advanced)
5: Network find and join
6: Network stop scan
7: Leave network
8: Network permit join
9: Discover device
10: Get discovered device
11: Commissioning operation
12: Start testing without network
e: Exit (Quit with calling clean and reset device)
q: Quit (Quit without calling clean and reset device)

Please select operation: 2
Preferred channel (11, 14, 15, 19, 20, 24, 25)
Set channel(25): 11 (Please use the channel number that is assigned to your table)
Preferred TX (-9 ~ 8)
Set TX(2): (Press <Enter> to use the default TX value)
Set PAN ID(0x1234): (Press <Enter> to use the default PAN ID value)
Form network channel(11) TX(2) PAN ID(0x1234):
Manually form network success
Done
```

In case coordinators see error message like below, please press **e** to leave the network, and press **2** to manually form the network again.

```
03-04 16:16:53.651 2558 2558 E <network.c:61> NULL or Wrong response type
03-04 16:16:53.652 2558 2558 E <artik_zigbee_coordinator.c:1540> Manually form network failed: No zigbee message
```

4. Join ZigBee network

4.1 ZigBee coordinators, please follow the instructions in 5.1.

ZigBee routers, please skip to step 5.2.

Coordinators can press <Enter> at anytime to show the menu.

Enter **8** to permit join a network, after this, all routers have up to **60 seconds** to join the newly formed ZigBee

```
=====
State: ZIGBEE_JOINED_NETWORK
```

```
Type: ZIGBEE_COORDINATOR
```

```
=====
1: Form network
```

```
2: Form network (advanced)
```

```
3: Join network
```

```
4: Join network (advanced)
```

```
5: Network find and join
```

```
6: Network stop scan
```

```
7: Leave network
```

```
8: Network permit join
```

```
9: Discover device
```

```
10: Get discovered device
```

```
11: Commissioning operation
```

```
12: Start testing without network
```

```
e: Exit (Quit with calling clean and reset device)
```

```
q: Quit (Quit without calling clean and reset device)Please select operation: 8
```

```
Permit join for 60 seconds
```

```
Done
```

networks.

4.2 ZigBee routers only

Routers can press <Enter> anytime to show the menu.

Enter **4** to join a network, then follow the instruction below. In case the network is not formed successfully, immediately press **e** to leave the network, and start zp_test program again and press **4** to re-join. You only have **60 seconds** to join the network unless coordinators grant permissions again.

```
1: Form network
2: Form network (advanced)
3: Join network
4: Join network (advanced)
5: Network find and join
6: Network stop scan
7: Leave network
8: Network permit join
9: Discover device
10: Get discovered device
11: Commissioning operation
12: Start testing without network
e: Exit (Quit with calling clean and reset device)
q: Quit (Quit without calling clean and reset device)
```

Please select operation: **4**

Set channel(25): **11** *(Please use the channel number that is assigned to your table)*

Preferred TX (-9 ~ 8)

Set TX(2): *(Press <Enter> to use the default TX value)*

Set PAN ID(0x1234): *(Press <Enter> to use the default PAN ID value)*

Join network channel(11) TX(2) PAN ID(0x1234):

Every time you re-enter the application it will ask you to **Resume device** or Add new device. In our exercise please

```
=====
0: Resume device
1: Add new device
Please select operation: 0
```

choose Resume device (**0**).

4.3 **(ZigBee coordinators only)** If the router is successfully joined, coordinators will see callback messages below.

```
In callback, response type : 3307  
NETWORK_NOTIFICATION: JOIN  
[testzigbee] callback end
```

4.4 (ZigBee coordinators only) Run 9 to discover all of the joined devices.

```
1: Form network
2: Form network (advanced)
3: Join network
4: Join network (advanced)
5: Network find and join
6: Network stop scan
7: Leave network
8: Network permit join
9: Discover device
10: Get discovered device
11: Commissioning operation
12: Start testing
e: Exit (Quit with calling clean and reset device)
q: Quit (Quit without calling clean and reset device)
Please select operation: 9
Wait response
In callback, response type : 3309
Device discovery start
[testzigbee] callback end
In callback, response type : 3309
Device discovery found:
Node id:0x1647 eui:0x0EAE5EFF570B00
Endpoint 20 Nodeid 0x1647
Device id(0x0100) name(On/Off Light)
Cluster id 0x0000, SERVER
Cluster id 0x0003, SERVER
Cluster id 0x0004, SERVER
Cluster id 0x0005, SERVER
Cluster id 0x0006, SERVER
Cluster id 0x0406, CLIENT
[testzigbee] callback end
In callback, response type : 3309
Done
[testzigbee] callback end
```



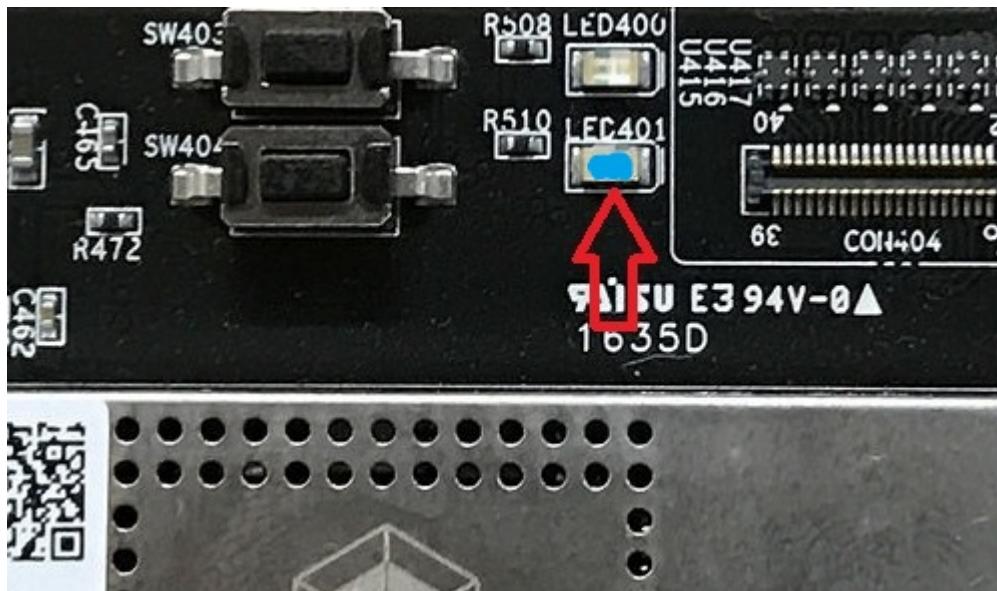
5. Test ZigBee network(ZigBee coordinator only). After the ZigBee network is formed, **coordinators** choose **12** to test

```
=====
1: Form network
2: Form network (advanced)
3: Join network
4: Join network (advanced)
5: Network find and join
6: Network stop scan
7: Leave network
8: Network permit join
9: Discover device
10: Get discovered device
11: Commissioning operation
12: Start testing
e: Exit (Quit with calling clean and reset device)
q: Quit (Quit without calling clean and reset device)
Please select operation: 12
=====
1: Get node information
2: Auto test
3: Identify
4: On/off
5: Level control
6: Illuminance measurement
7: Occupancy Sensor
8: Temperature Sensor
9: Humidity
10: Bind api test
11: Remote Control
12: Thermostat
13: Get network topology
14: Request reporting test
15: Groups test
16: General api test
17: ZDO test
18: ZCL raw command api test
0: Setup network
e: Exit (Quit with calling clean and reset device)
q: Quit (Quit without calling clean and reset device)
Please select operation: 4
```

the network. Then select **4** to turn on and off router ZigBee devices' LED.

At this point, routers should see the LEDs on their ARTIK 533s turned on and off by coordinators.

In case the coordinator side shows **Test failed: Not found**, just press **0** and return to Setup network page and press **9** to discover devices again and run the testing.



Bonus Question

Use ARTIK ZigBee API to control a ZigBee Plug or Sensor.