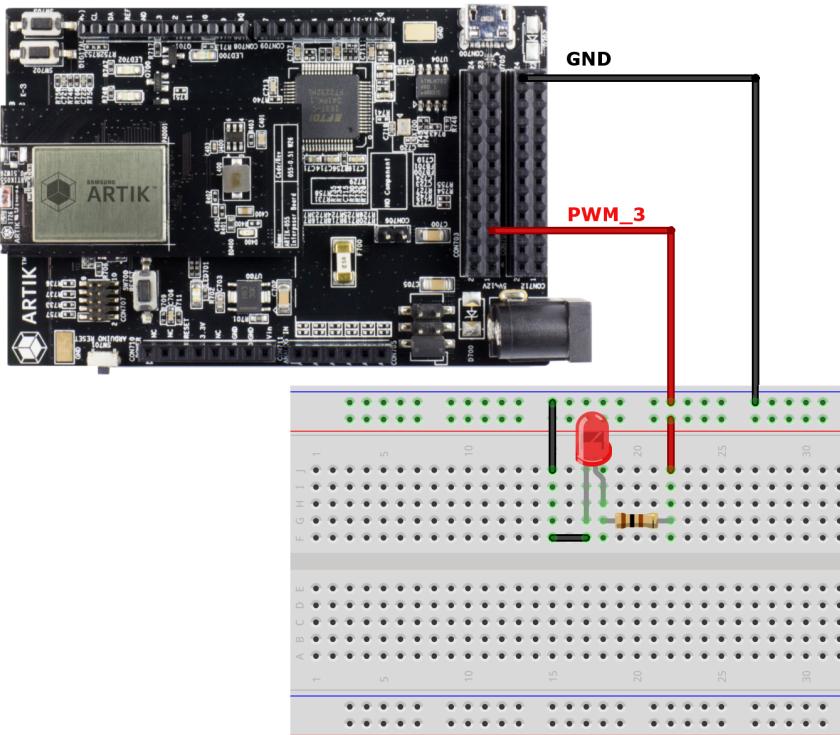


# Develop IoT with Samsung ARTIK platform

July 19th, 2018

Samsung Strategic and Innovation Center



fritzing

## Table of Contents

<b>Introduction</b>	<b>2</b>
<b>Before you begin</b>	<b>3</b>
<b>Setup</b>	<b>4</b>
<b>Exercise 1: Monitor sensor data and set up real time sensor dashboard (Node-RED)</b>	<b>5</b>
<b>Exercise 2: Communicating with ARTIK cloud</b>	<b>8</b>
<b>Exercise 3: Get Email notification</b>	<b>18</b>
<b>Exercise 4: Trigger cross-device action by using ARTIK Rules Engine (ARTIK Cloud, ARTIK 05x)</b>	<b>23</b>
<b>Bonus Exercises</b>	<b>30</b>

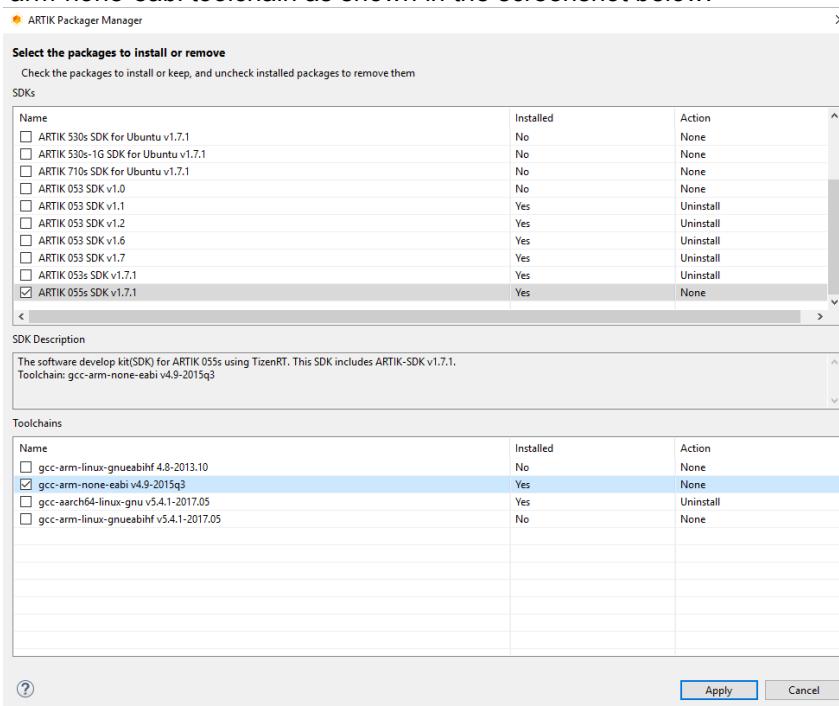
## Introduction

Welcome to the Samsung ARTIK workshop! This workshop is your introduction to developing IoT with the Samsung ARTIK platform. In this session, you will learn how to:

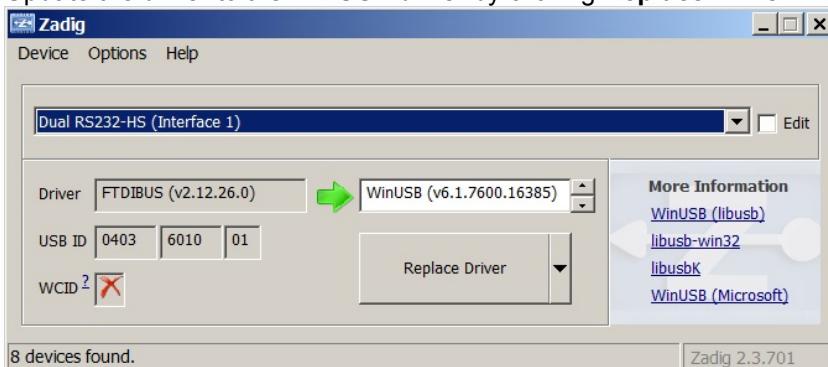
- Access your target board from the serial console
- Build program flows using Node-RED
- Collect sensor data
- Stream data to ARTIK Cloud
- Enable MQTT so that your ARTIK boards can communicate with each other
- Use the ARTIK Cloud Rules Engine to trigger cross-device actions

## Before you begin

1. Bring your own laptop. For Windows users, pre-install PuTTY.  
PuTTY (<http://www.putty.org/>) – SSH and Telnet client, for serial console access
2. Have a Gmail account: ARTIK will send emails to your Gmail account. In order to do this, we need to turn on “Access for less secure apps” setting in your Gmail.  
If you are using a corporate Gmail account or if you want to keep your project emails separate from your personal emails, please create a new account for the workshop.
3. Establish an ARTIK Cloud user portal account: We will stream data to ARTIK Cloud service. Create an account at ARTIK Cloud user portal (<https://artik.cloud/>).
4. Install ARTIK IDE and ARTIK 055s SDK v1.7.1(or v1.8 if you use Mac) on your laptop. Instruction is at: <https://developer.artik.io/documentation/artik-05x/getting-started/prepare-ide.html>.  
After you install ARTIK IDE, please go to Help-> ARTIK Package Manager, and install ARTIK 055s SDK v1.7.1 and gcc-arm-none-eabi toolchain as shown in the screenshot below.



5. For Windows users, Please
  - a. Browse to the ARTIK\_IDE\_Installation\_Directory\tools directory.
  - b. Run the zadig-2.3.exe application.
  - c. Under **Options**, select **List All Devices**.
  - d. Under the devices drop-down list, select Interface **1** of the dual COM port.
  - e. Update the driver to the WinUSB driver by clicking **Replace Driver**.



- f. Now select Interface **0** of the FTDI device, and change it to a 'libusb' driver.

## Setup

1. Access Kitra from your serial console. Instructions can be found here:

<https://github.com/rushup/kitra520/wiki/Getting-started-with-Kitra520>

To power up Kitra, press the power button until you hear 2 beeps, then connect its serial USB to your host machine. Follow the instruction in the link below to access the board from your serial console.

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

2. Connect Kitra to the network.

2.1 Input WiFi credentials as shown below.

```
[root@localhost ~]# wpa_passphrase "ARTIK2" "artik123" >> /etc/wpa_supplicant/wpa_supplicant.conf
```

2.2 Restart wpa\_supplicant & run dhclient to obtain an IP address.

```
[root@localhost ~]# systemctl restart wpa_supplicant  
[root@localhost ~]# dhclient wlan0
```

2.3 Do a 'ping' test from your console and take note of your board IP address. If you are not able to obtain an IP address, please go to step 3.

```
[root@localhost ~]# ping www.google.com  
PING www.google.com (74.125.28.103) 56(84) bytes of data.  
64 bytes from nuq04s18-in-f4.1e100.net (74.125.28.103): icmp_seq=1 ttl=52 time=8.83 ms  
64 bytes from nuq04s18-in-f4.1e100.net (74.125.28.103): icmp_seq=2 ttl=52 time=59.9 ms  
64 bytes from nuq04s18-in-f4.1e100.net (74.125.28.103): icmp_seq=3 ttl=52 time=17.2 ms
```

(Ctrl-C to terminate)

```
[root@localhost ~]# ifconfig wlan0  
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 10.0.0.2 netmask 255.255.255.0 broadcast 10.0.0.255  
        inet 2601:647:4e01:7b45:489d:21ff:fe85:6c27 prefixlen 64 scopeid 0x0<global>  
        inet6 fe80::489d:21ff:fe85:6c27 prefixlen 64 scopeid 0x20<link>  
    ether 4a:9d:21:85:6c:27 txqueuelen 1000 (Ethernet)  
        RX packets 14769 bytes 20317291 (19.3 MiB)  
        RX errors 0 dropped 0 overruns 0 frame 0  
        TX packets 0 bytes 311790 (304.4 Kib)  
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

3. In case you are not able to obtain an IP address, please restart dhclient by following the steps below, then repeat step 2.3.

```
[root@localhost ~]# dhclient wlan0 -r  
Killed old client process  
[root@localhost ~]# dhclient wlan0
```

## Exercise 1: Monitor sensor data and set up real time sensor dashboard (Node-RED)

In this exercise, we can utilize sensors on Kitra to monitor body temperature, count number of steps and detect fall incident, set up real-time visualization chart and trigger actions based on targets achieved

1. Start Node-RED in the background from your serial console.

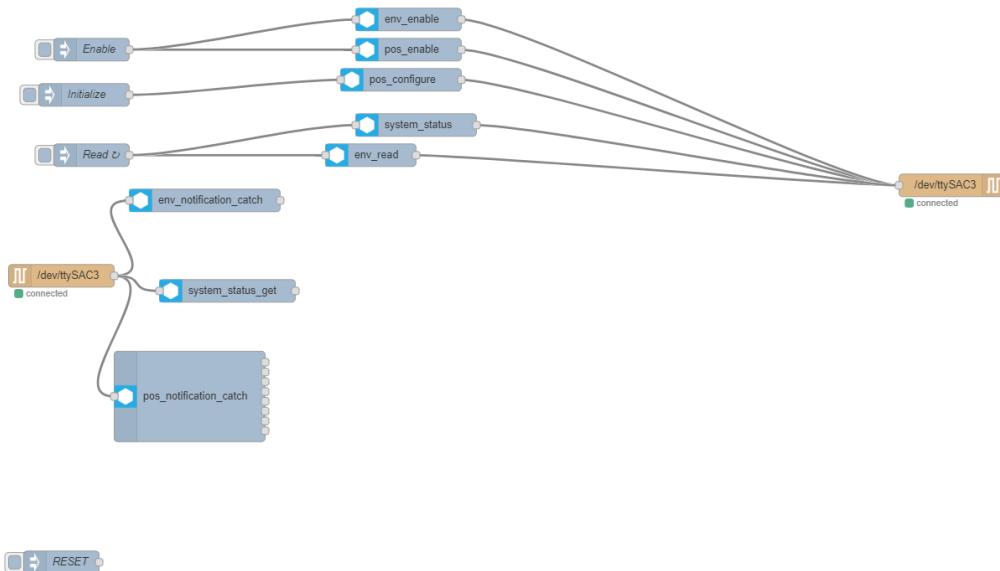
Get the node-red template from github using the command

```
[root@localhost ~]# wget https://github.com/SamsungARTIK/Training/blob/master/workshop.json
```

Start Node-RED

```
[root@localhost ~]# node-red workshop.json &
...
Welcome to Node-RED
=====
29 Jan 13:53:43 - [info] Node-RED version: v0.16.2
29 Jan 13:53:43 - [info] Node.js version: v4.7.3
...
29 Jan 13:53:48 - [info] Server now running at http://127.0.0.1:1880/
```

2. Open a browser on your laptop, and launch [http://<your\\_board\\_ip\\_address>:1880/](http://<your_board_ip_address>:1880/). <your\_board\_ip\_address> should be replaced by the IP address you obtained in Setup step earlier.

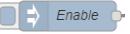


3. Design your project flow.

### 3.1 Extend the ENABLE flow on the Node-RED canvas.



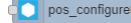
- Enable Kitra's environmental sensors and accelerometer/gyroscope/magnetometer (Already done in the template provided).

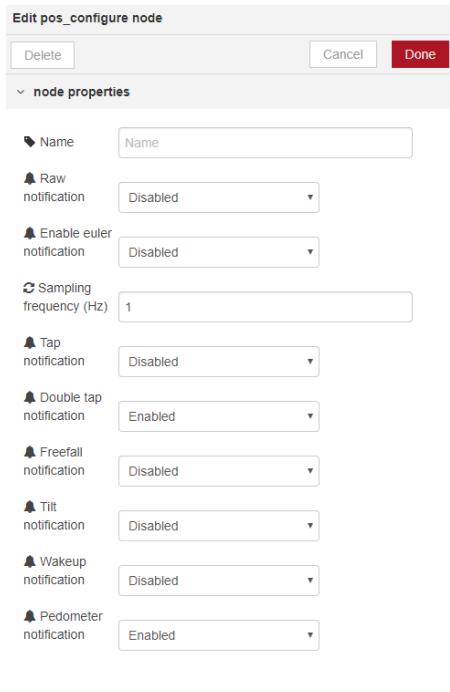
3.2 Click  at the upper right corner to make your flow live (Click the switch on  to enable the sensors).

#### 4. Configure sensors.

##### 4.1 Configure Position/Orientation sensors to reset the pedometer count and enable to count steps.



- Double click on  node and enable “Double tap notification” and “Pedometer notification” to get notified on corresponding events and then click on “Done” (Already done in the template provided).



##### 4.2 Initialize global variables.

- Drag and drop a  node as shown below, and connect its input to the output of “Initialize” node.



- Double click on the  node. Provide the name “INIT” and input the following code snippet in the function window and click on “Done”.

```

flow.set('steps',0);
flow.set('old_steps',0);
flow.set('temp', 0);
flow.set('LED', 0);
flow.set('Charging', 0);
flow.set('battery', 0);
  
```



4.3 Click at the upper right corner to make your flow live (Click the switch on to initialize the step counter).

## 5. Read all the sensors enabled.

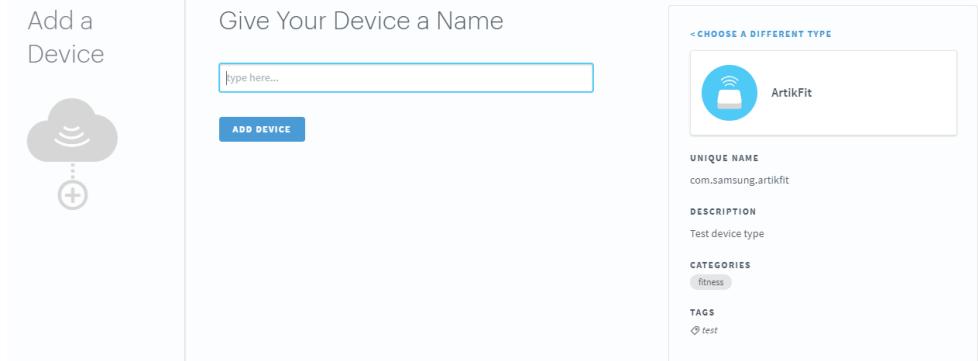
5.1 Drag and drop the node to get the enabled sensors' reading (Already done in the template provided).



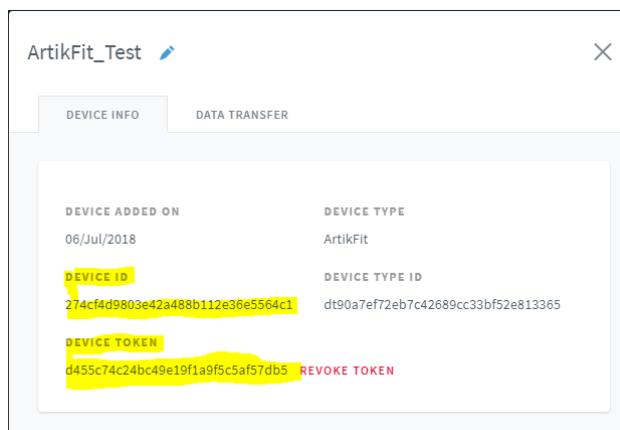
5.2 Click at the upper right corner to make your flow live (Click the switch on to start reading sensor data).

## Exercise 2: Communicating with ARTIK cloud

1. Set up device on ARTIK cloud.
  - 1.1 Log in to ARTIK Cloud portal (<https://my.artik.cloud/>).
  - 1.2 Under “Devices” tab, click on “+ Add Another Device...”.
  - 1.3 In device types, search for “ArtikFit” and select the same.



- 1.4 Provide a name for your device and click on “ADD DEVICE”, the device is added and is listed under “Devices” tab.
- 1.5 Click on the device you created and click on “GENERATE DEVICE TOKEN”. Make a note of “DEVICE ID” (not DEVICE TYPE ID) and “DEVICE TOKEN” as they are required in further steps.



2. Connecting to ARTIK cloud.

- 2.1 Transmit the temperature value obtained to Artik cloud.

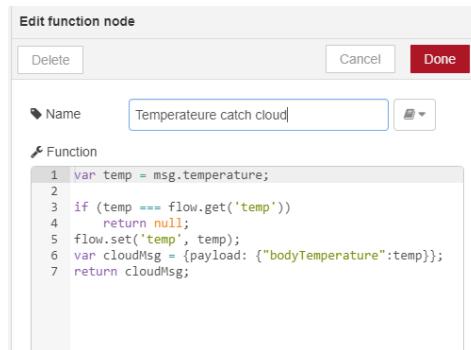
- Drag and drop a node to format the data obtained from the sensor.



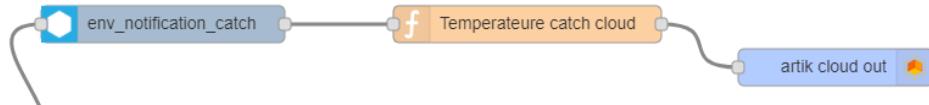
- Double click on the function node, input the following code snippet and update the name as "Temperature catch cloud".

```
var temp = msg.temperature;

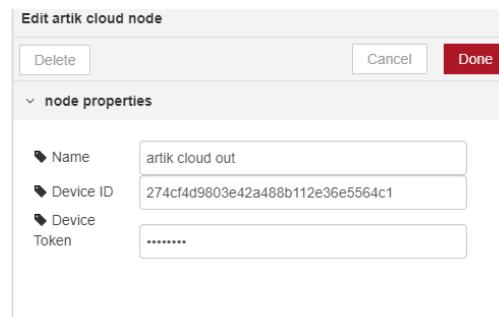
if (temp === flow.get('temp'))
    return null;
flow.set('temp', temp);
var cloudMsg = {payload: {"bodyTemperature":temp}};
return cloudMsg;
```



- 2.2 Drag and drop and connect the output of the above function to the input of ARTIK cloud node as shown below.



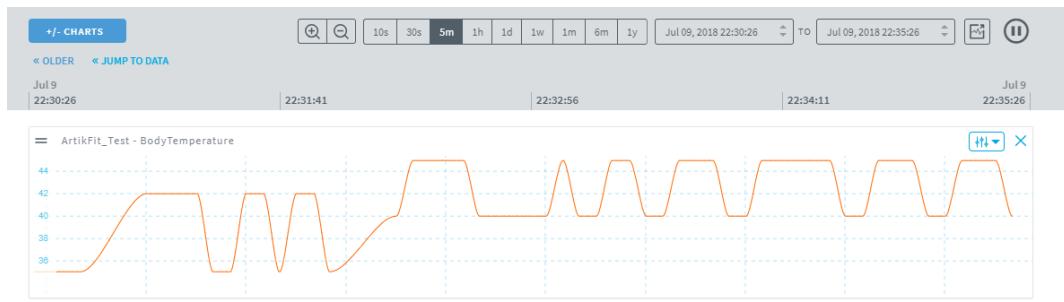
Double click on the node and update the device id and device token obtained from the ARTIK cloud dashboard for your device then click on "Done".



- 2.3 Click at the upper right corner to make your flow live.

- 2.4 On deploying the flow and enabling, initializing the sensors, the temperature value is transmitted to ARTIK cloud and the same can be visualized in the cloud portal.

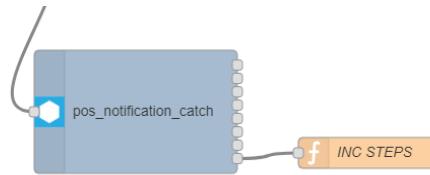
Click on "Charts" → "+/- CHARTS" → "BodyTemperature" & "StepCount" to enable visualization.



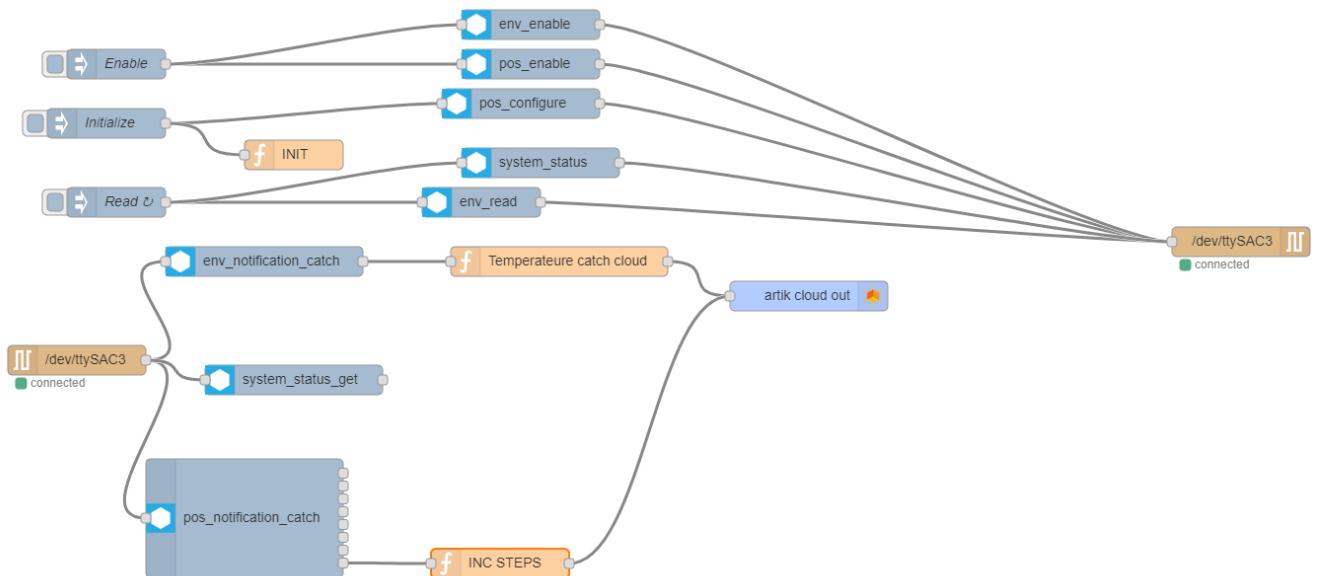
## 2.5 Transmit the step counter value to ARTIK cloud.

- Drag and drop a **f function** node and connect it with the bottom most output of **pos\_notification** that outputs the cumulative step counter value. Double click on it, update the name as “INC STEPS” and input the following code snippet to update the global variable ‘steps’ on receiving the notification for the updated step counter value (given by msg.extra).

```
flow.set('steps', msg.extra);
node.log(flow.get('steps'));
var steps_count = flow.get('steps') || 0;
var cloudMsg = {payload: {"stepCount":steps_count}};
return cloudMsg;
```



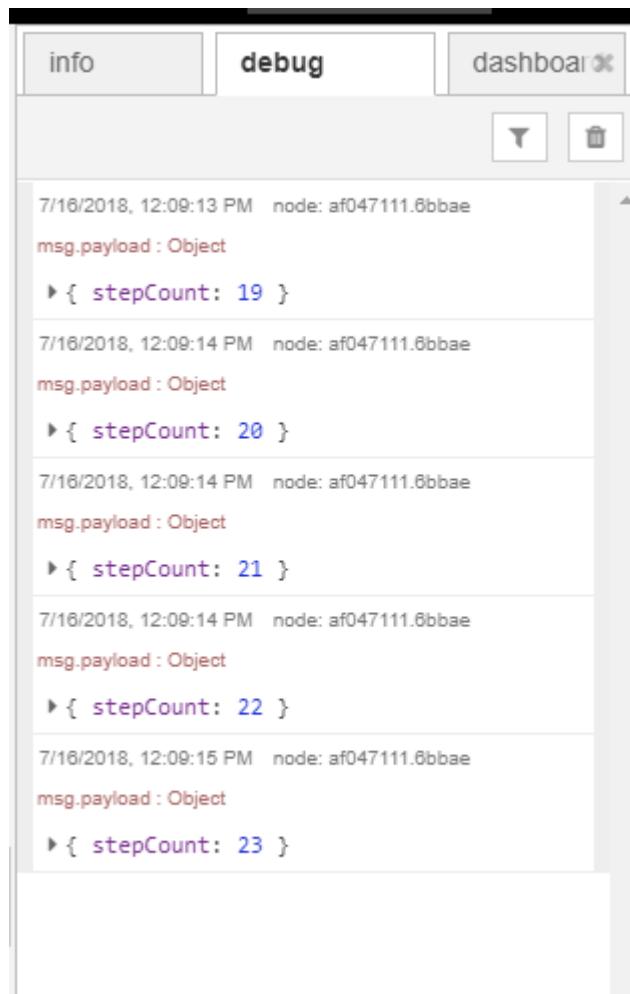
- Connect the output of “INC STEPS” function node to the input of **artik cloud** as shown below.



- Add  node to monitor the step counter increments in the debug window of node-red.



- Click  at the upper right corner to make your flow live.
- Choose “debug” window in the right side window of node-red to view the debug messages received by the  node.



### 3. Trigger actions from Artik cloud.

#### 3.1 Log in to Artik cloud portal and click on “Rules” → “+NEW RULE”.

- Under “Choose device activity to monitor”, select your device already registered with the Artik cloud and then the field to monitor found in the dropdown list in “IF” section.
- Select the condition → “more than or equal to” and then specify the value of the step counter which after reaches this target or more than this triggers the specified action (Ex: 20).

### Choose device activity to monitor

IF

Actual Value ▾ ArlikFit\_Test stepCount X

Number of steps covered by the ArlikFit device user

more than or equal to 20

**ADD DURATION**

- Under “Send actions to your devices”, select your registered device from the drop down list and then the action you want to trigger (Ex: setColorRGB) and provide the RGB values (0-255) and click on “SAVE RULE”.

Send actions to your devices

THEN

ArlikFit\_Test setColorRGB X

PARAMETER VALUES CAN BE ENTERED BELOW OR TAKEN FROM ONE OF YOUR CONNECTED DEVICES.

COLORRGB.BLUE 60	COLORRGB.GREEN 35
COLORRGB.RED 103	

+ NEW ACTION

Describe your rule

RULE TITLE  
Send setColorRGB to ArlikFit\_Test

DESCRIPTION  
IF ArlikFit\_Test.stepCount is more than or equal to 20  
THEN send to ArlikFit\_Test the action setColorRGB with colorRGB.blue = 60,  
colorRGB.green = 35, colorRGB.red = 103

SAVE RULE CANCEL

- You will see the new rule listed under the “Rules” tab (and it is activated).

### 3.2 Receive actions in Node-red.

- Drag and drop  and  node and connect them together.
- Double click on  node and update the device id and device token (You can find this in the Artik cloud portal when you click on the registered device).



- Double click on the  node and update its name to “Receive Action” and input the following code snippet then click on “Done”.

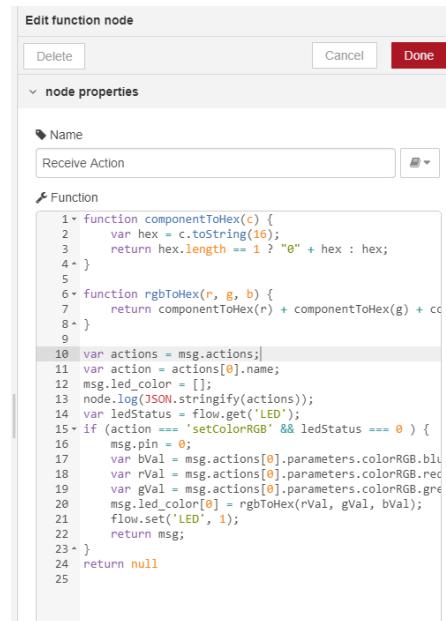
```

function componentToHex(c) {
    var hex = c.toString(16);
    return hex.length == 1 ? "0" + hex : hex;
}

function rgbToHex(r, g, b) {
    return componentToHex(r) + componentToHex(g) + componentToHex(b);
}

var actions = msg.actions;
var action = actions[0].name;
msg.led_color = [];
node.log(JSON.stringify(actions));
var ledStatus = flow.get('LED');
if (action === 'setColorRGB' && ledStatus === 0 ) {
    msg.pin = 0;
    var bVal = msg.actions[0].parameters.colorRGB.blue;
    var rVal = msg.actions[0].parameters.colorRGB.red;
    var gVal = msg.actions[0].parameters.colorRGB.green;
    msg.led_color[0] = rgbToHex(rVal, gVal, bVal);
    flow.set('LED', 1);
    return msg;
}
return null;
}

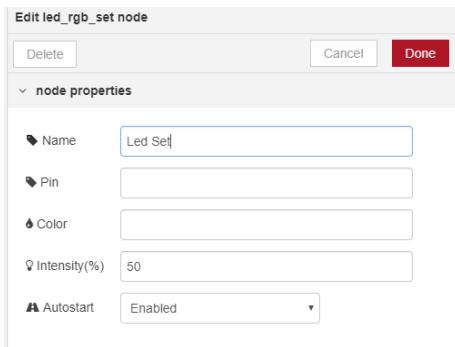
```



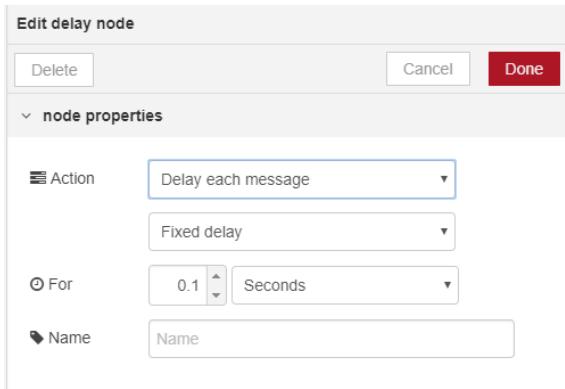
- To turn on LEDs on receiving the action, drag and drop and nodes found under kitra output.
- Double click on and update the fields as shown below then click on “Done”.

	<input type="text" value="Name"/>
	<input type="text" value="0"/>
	<input checked="" type="checkbox" value="Enabled"/> Enabled

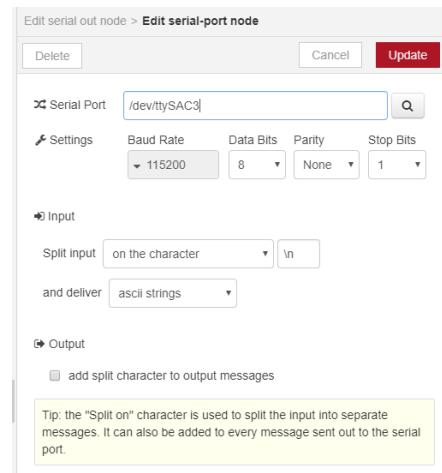
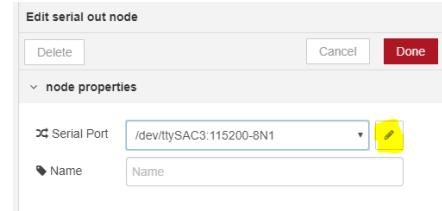
- Double click on node and update the fields as shown below then click on “Done”.



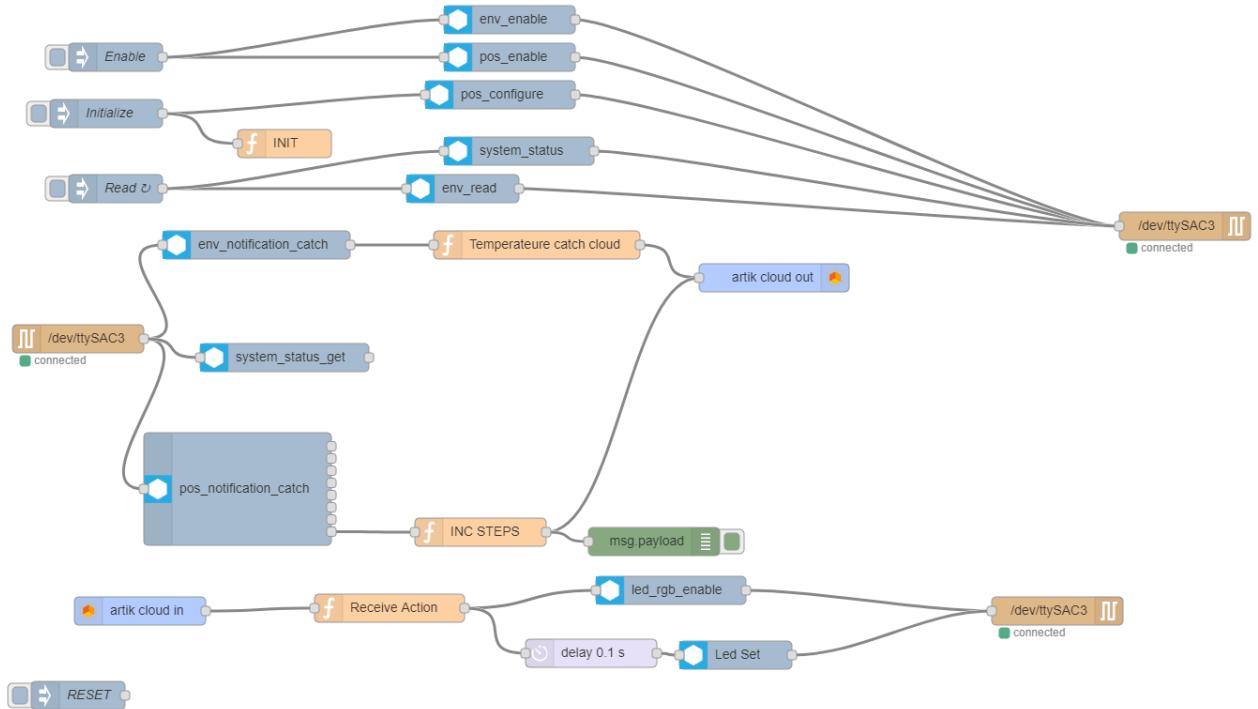
- Drag and drop node and double click and update the fields as shown below then click on "Done".



- Drag and drop node and update its values as shown below.



- Click on "Update" and then "Done". Connect the nodes as shown below.

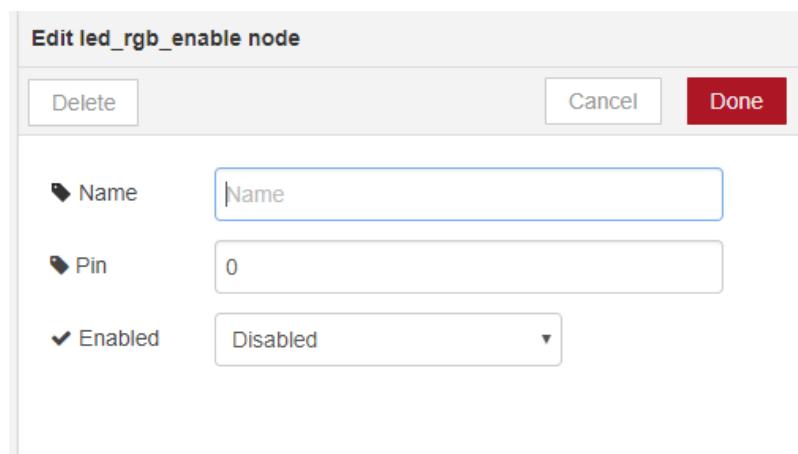


Click at the upper right corner to make your flow live.

3.3 Shake the Kitra device to increase the step counter value and once the counter reaches the mentioned target in the rule stated in Artik cloud, the action is triggered and the LEDs on Kitra board turn on and are colored as per the mentioned RGB color code.

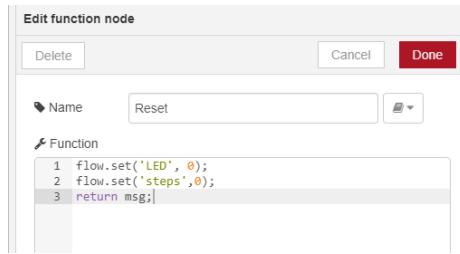
4. Implement Reset feature to turn off the LEDs and reset the step counter.

4.1 Drag and drop node and place it next to input node. Double click on the node and disable the LEDs then click on "Done".

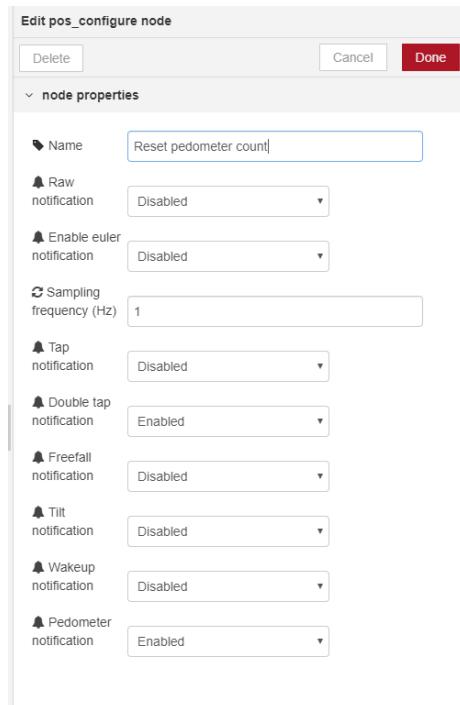


4.2 Drag and drop node, double click on it and input the following code snippet to reset the global variable that keep track of step count, update the name as "Reset" then click on "Done".

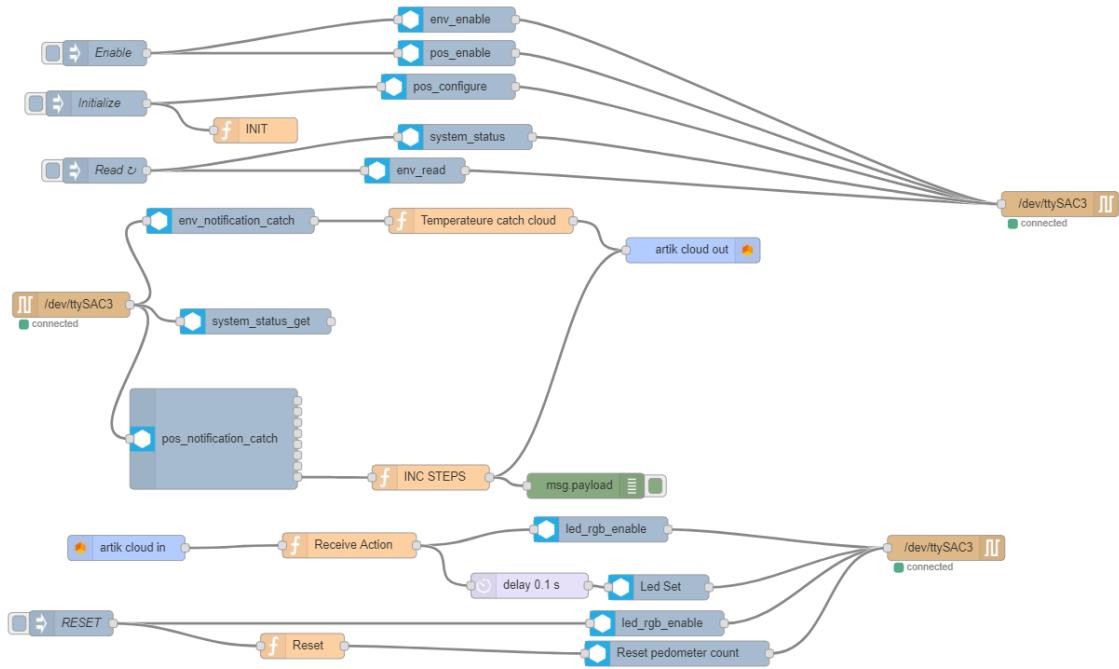
```
flow.set('LED', 0);
flow.set('steps',0);
return msg;
```



4.3 Drag and drop node, double click on it and update the name as “Reset pedometer count” and enable “Pedometer notification” and “Double tap notification”.



4.4 Connect the nodes as shown below and Click at the upper right corner to make your flow live.



4.5 Clicking the switch on will turn off the LEDs and reset the step counter value back to 0.

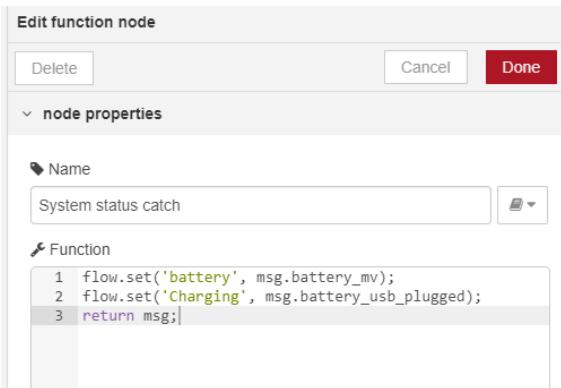
## Exercise 3: Get Email notification

To the current ArtikFit module, we will add SoS kind of feature where a double tap on Kitra board will send an Email notification about the current readings of the sensors, battery level and also trigger the onboard buzzer.

- Obtain the system status from node (battery level and charging status).

- Drag and drop node and connect it with the output of node and double click on it and input the following code snippet, update its name as "System status catch" then click on "Done".

```
flow.set('battery', msg.battery_mv);
flow.set('Charging', msg.battery_usb_plugged);
return msg;
```

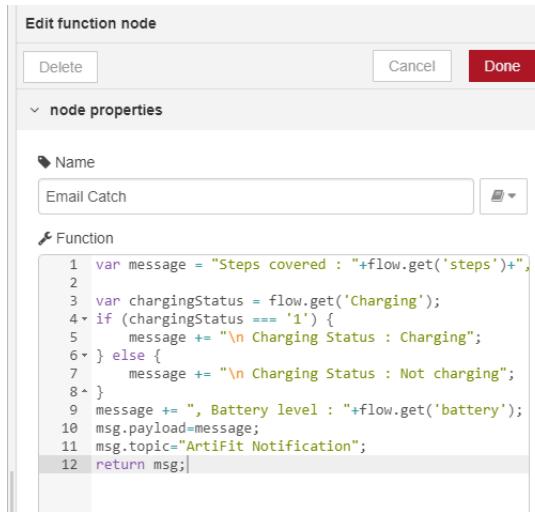


- Catch "Double tap notification". The node outputs signal in the 5th pin on receiving double tap.

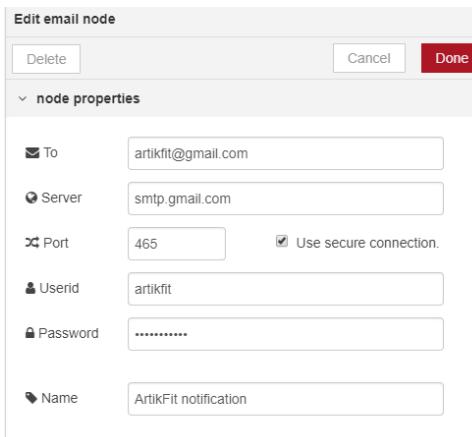
- Drag and drop node and connect it with the 5th pin of node. Double click on it and input the following code snippet.

```
var message = "Steps covered : "+flow.get('steps')+", temperature : "+flow.get('temp');

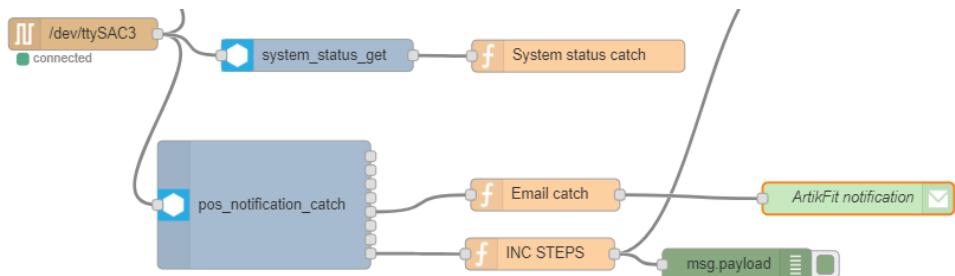
var chargingStatus = flow.get('Charging');
if (chargingStatus === '1') {
    message += "\n Charging Status : Charging";
} else {
    message += "\n Charging Status : Not charging";
}
message += ", Battery level : "+flow.get('battery');
msg.payload=message;
msg.topic="ArtiFit Notification";
return msg;
```



2.2 Drag and drop node (found under “social”) and double click and update the fields as shown below (Create a new email id in Gmail for this purpose and update the same on both “To” and Userid fields).

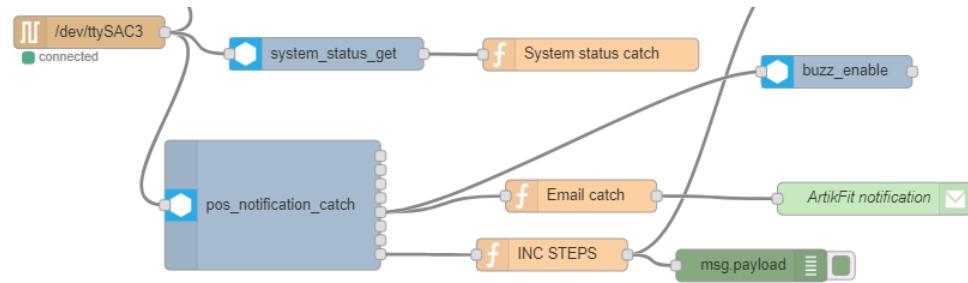


2.3 Connect the nodes as shown below.

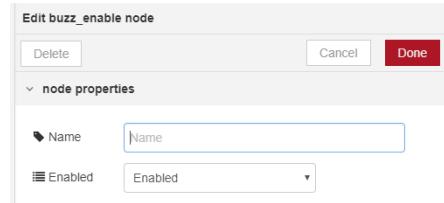


3. Trigger onboard buzzer on double tap event.

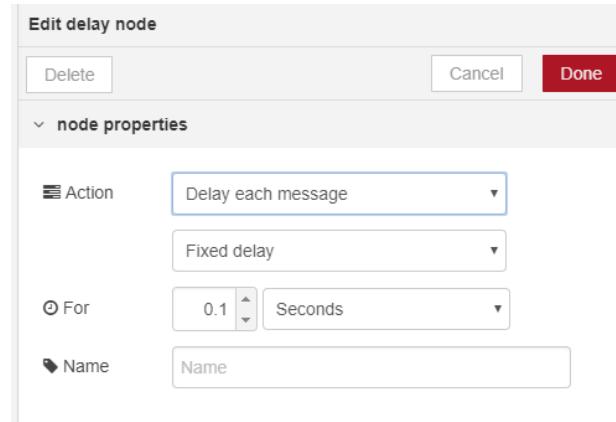
3.1 Enable the onboard buzzer – drag and drop node and connect it with the as shown below.



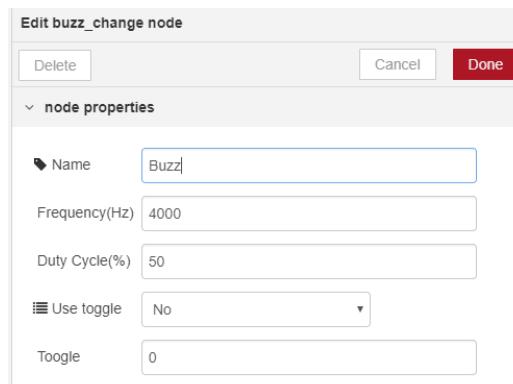
3.2 Double click on and update the fields as shown below.



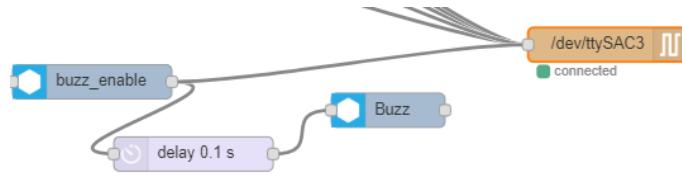
3.3 Drag and drop node to induce a delay for enabling the buzzer before sending the trigger signal to it. Double click on and update the fields as shown below then click on “Done”.



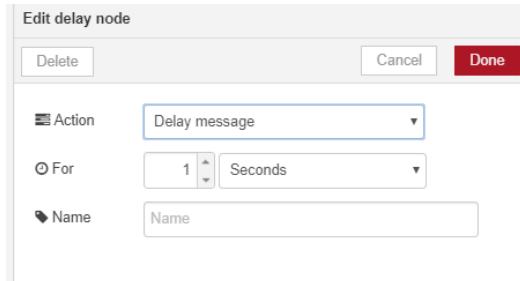
3.4 Turn on buzzer – drag and drop node and double click on it and update the fields as shown below then click on “Done”.



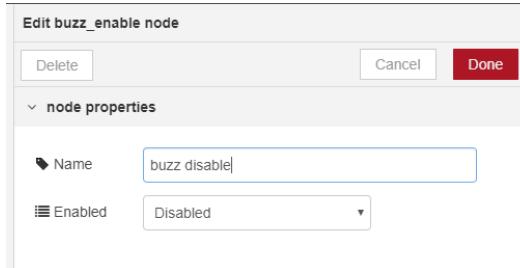
3.5 Connect the nodes as shown below.



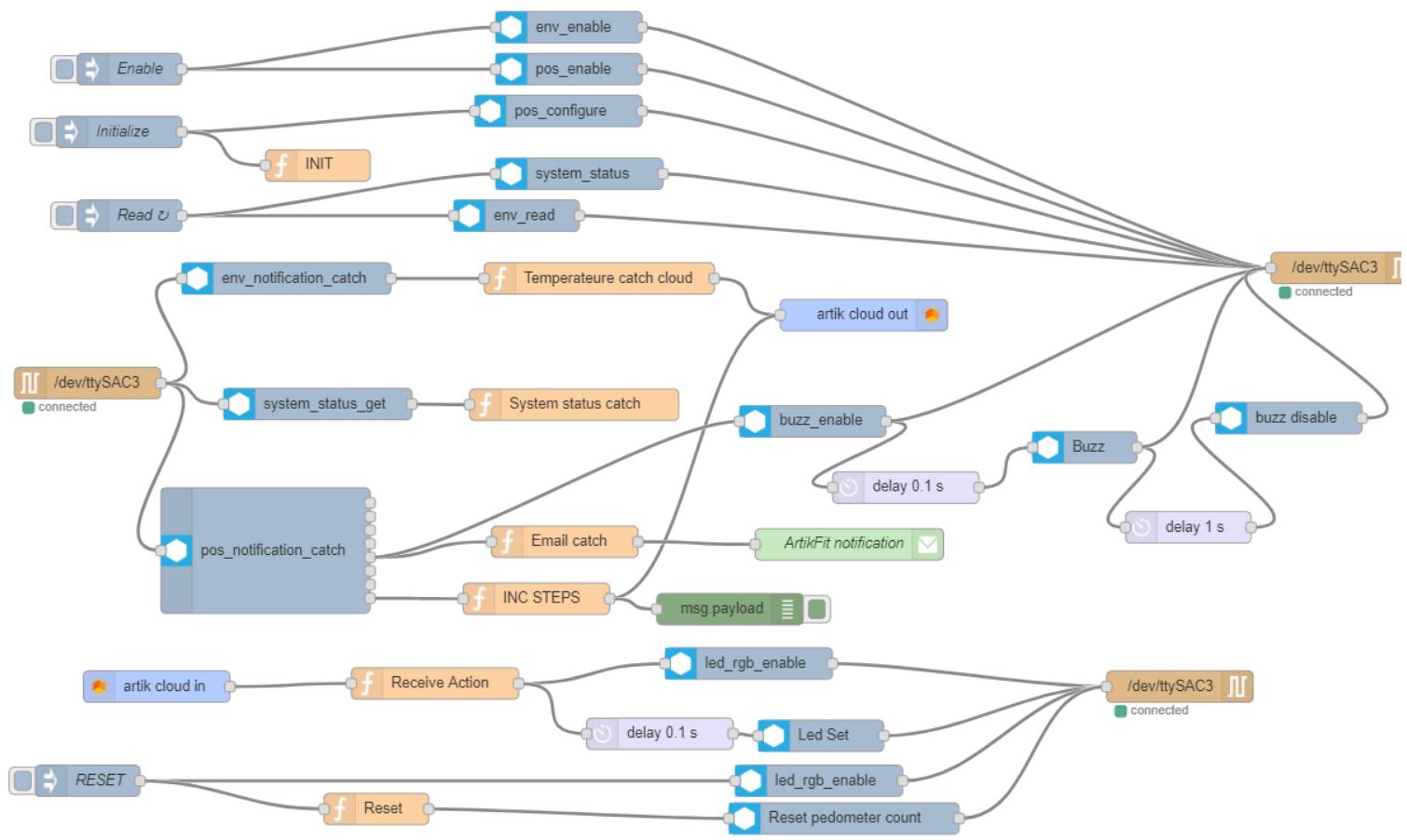
3.6 Turn off the buzzer – drag and drop node and double click on it, update the fields as shown below.



3.7 Drag and drop node and double click on it, update the fields as shown below (Select “Disabled” option under Enabled drop down).



3.8 Connect the nodes as shown below and Click  Deploy at the upper right corner to make your flow live.



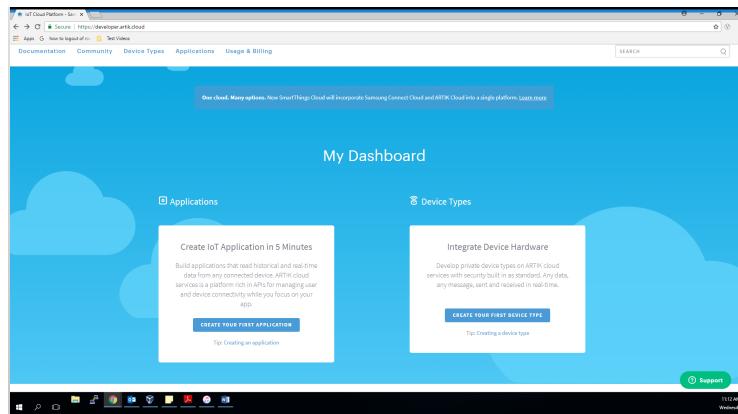
## Exercise 4: Trigger cross-device action by using ARTIK Rules Engine (ARTIK Cloud, ARTIK 05x)

Now that your personal fitness band is working, it'd be cool if you could control other devices with it. Maybe your fan turns on when you are feeling warm. To trigger such actions, we are going to again make use of the rules engine. The ARTIK cloud, using the rules engine sends text messages to an IOT Device. The IOT device then processes the text message, understands what needs to be done, and triggers the specified action. In this tutorial, we'll use an ARTIK 05x to process the action messages and trigger a fan!

### 1. Create a Device Type on ARTIK Cloud.

#### 1.1 Login on ARTIK Cloud [Developer Dashboard](#).

#### 1.2 Click on CREATE YOUR FIRST DEVICE TYPE (+ New Device) if you already have devices in associated with your account) and name your device. (You can use "A055 Light and Fan Controller" for the **DEVICE DISPLAY NAME** and "com.samsung.a05xLFC" for the **UNIQUE NAME**).

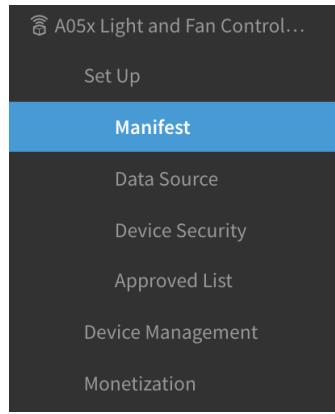


#### 1.3 Create the device type.

<b>DEVICE DISPLAY NAME</b>	A05x Light and Fan Controller	35
<b>UNIQUE NAME</b>	com.samsung.a05xLFC	236
<b>CREATE DEVICE TYPE</b>		<b>CANCEL</b>

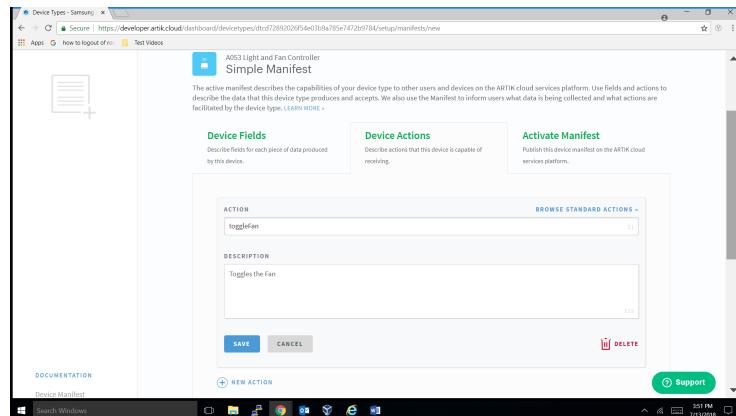
### 2 Creating a Manifest.

#### 2.1 Go to Set Up under your device name and then click on Manifest. Here we'll create a new Manifest for an ARTIK 05x device that toggles a fan.



A Manifest specifies what kind of data is coming from the device you created and what kind of actions it can perform. Since our ARTIK 05x device is not collecting any kind of data, we only need to specify the actions the device can listen to.

**2.2 Click on + NEW MANIFEST and go to Device Actions.** Add an action for toggling the fan and name it **toggleFan**. Save.



**2.3 Click on NEXT: ACTIVATE MANIFEST. Click on ACTIVE MANIFEST.**

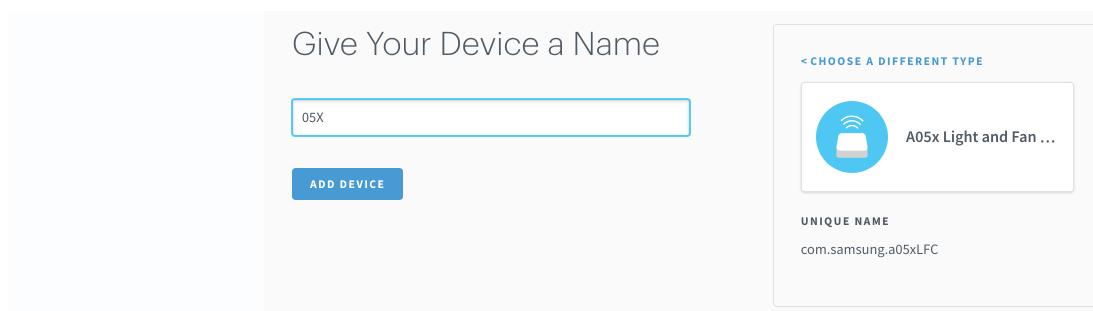
**3.** Create a Device instance and generate a Device Token.

**3.1** What you just created was a device type but not a device. Any devices that you create under a given device type, would have same properties and similar behavior. For example any device under the aforementioned device type would be a simple light and fan controller that can toggle the fan by receiving actions from the ARTIK Cloud.

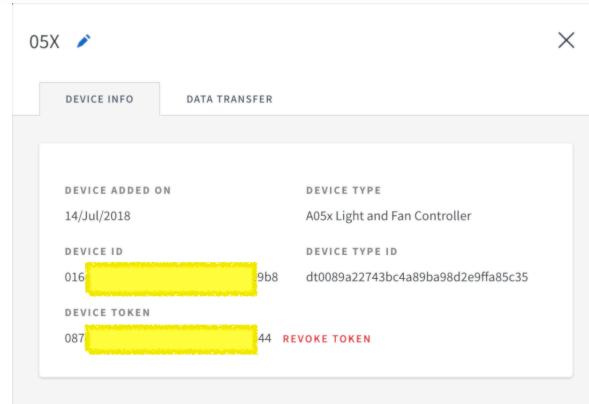
Now to create an ARTIK 05x device as a “A05x Light and Fan Controller”, go to My ARTIK Cloud.

A screenshot of the My ARTIK Cloud search interface. At the top, a question 'What kind of device are you looking for?' is followed by a search bar containing 'A05x'. To the right of the search bar is a grid of icons representing different device types. Below the search bar is a section labeled 'Search Result' containing a card with a blue icon of a light and fan, and the text 'A05x Light and Fa...'. There is also a small 'Support' button in the bottom right corner of the search interface.

**3.2** Search for “A05x Light and Fan Controller” device type and add a new Device by giving it an arbitrary name. Here we simply used “05x”.

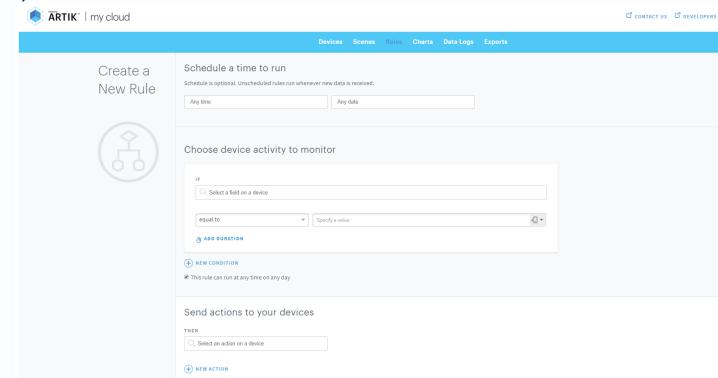


3.3 In order to interact with the Cloud, one needs a device token. Click on the newly created device and then click on **GENERATE DEVICE TOKEN...**. Once the token is generated, copy the device token and the device ID for later use.

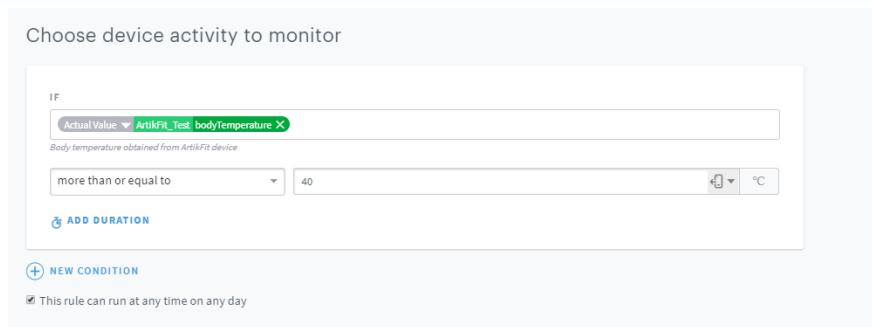


#### 4. Configure Rules Engine to trigger actions.

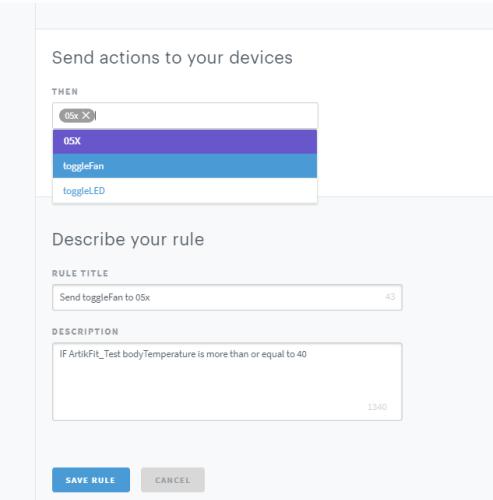
4.1 In ARTIK cloud portal, select “Rules” tab and then “+NEW RULE”.



4.2 Under “Choose device activity to monitor”, select your ArtikFit device name and corresponding parameter to monitor (Ex: bodyTemperature).



4.3 Under “Send actions to your devices”, select the ARTIK 05x device name, and select the action you want to send for this rule (Ex: Toggle fan once bodyTemperature is more than 40).

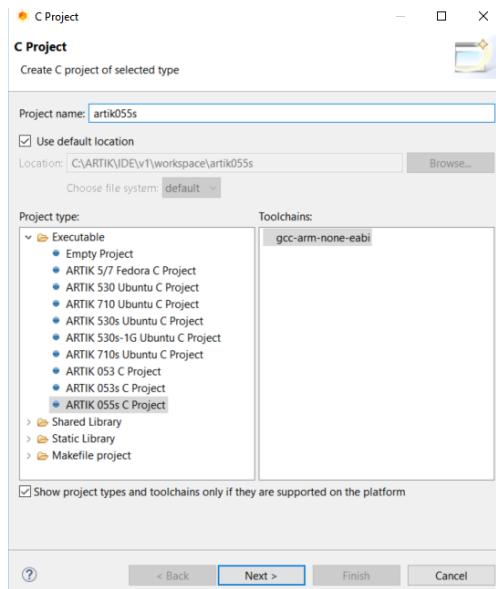


4.4 Click on “SAVE RULE” and it should be listed under the “Rules” tab along with the rule specified for Kitra device.

## 5. Set up a project in ARTIK IDE for ARTIK 05X.

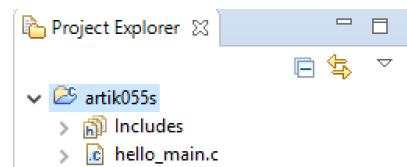
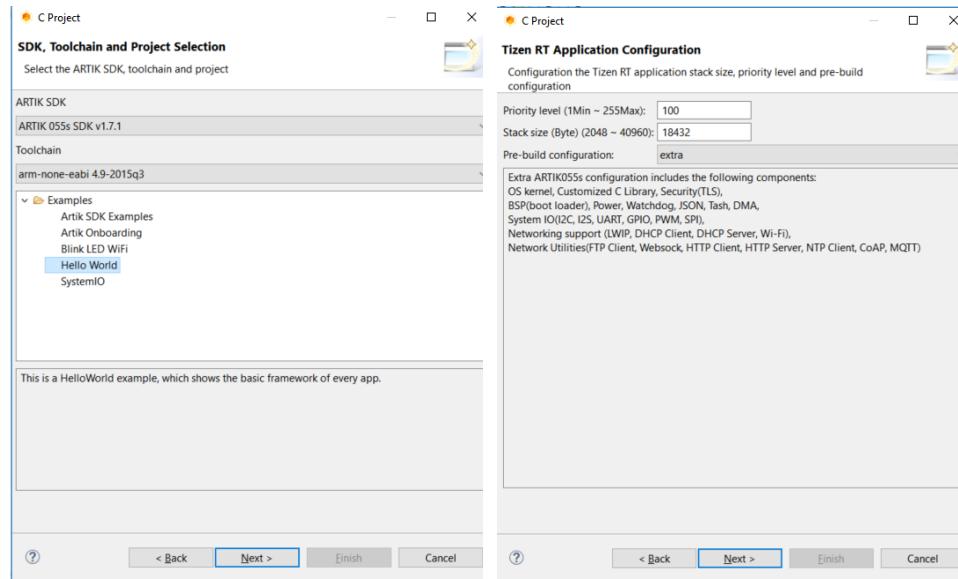
5.1 Go to File → New Project → C Project.

5.2 Give an arbitrary name to the project and select “ARTIK 055 C Project” under Project type. Click on Next.



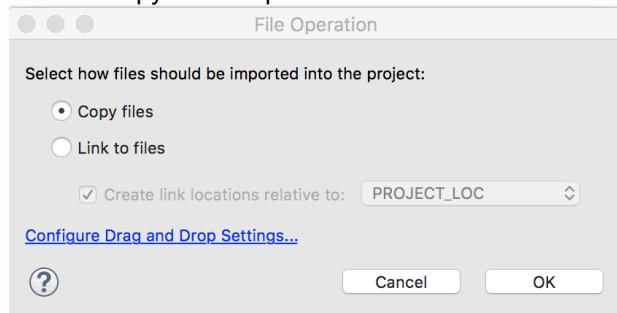
5.3 Select the “Hello World” example and proceed to Next.

5.4 In the Tizen RT Application Configuration, choose “extra” under “Pre-build configurations.” Click on next, followed by Finish to complete the set up.



In the Project Explorer, a project is created.

6. Right Click `hello_main.c`, and select Delete.
7. Drag and drop all the provided files in ARTIK05x Workshop Code folder into the newly created project. The dialog below will pop up. Select “Copy files” to proceed.



8. Configure Device ID & Token in `info.h`

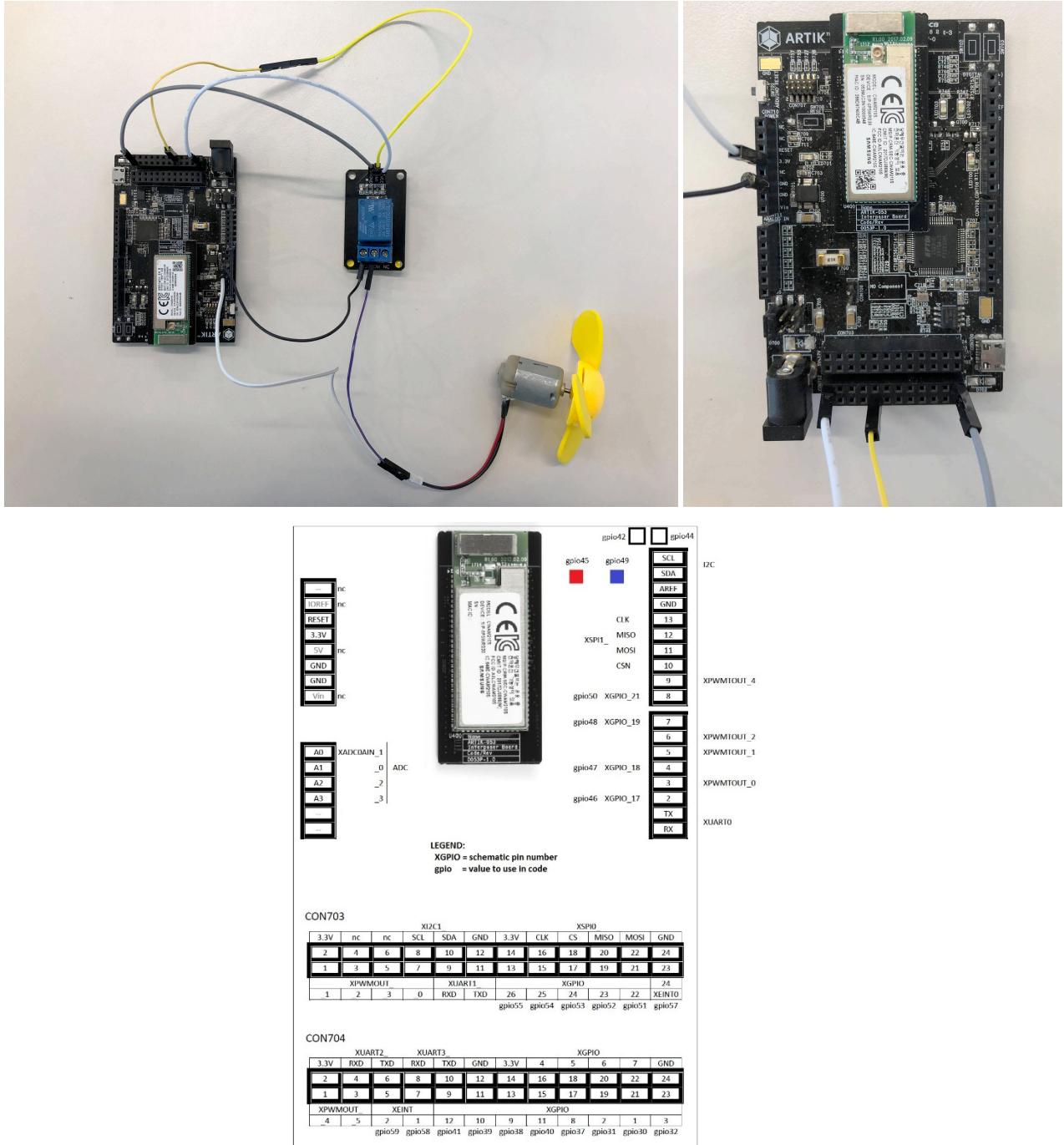
Enter your Device Token in the `DEVICE_TOKEN` field, Device ID in the `DEVICE_ID`, `ACTION_TOPIC`, and `MESSAGE_TOPIC` fields. These will be used to get actions from the cloud.

```
#define DEVICE_TOKEN "_your_DEVICE_TOKEN_goes_here_"
#define DEVICE_ID "_your_DEVICE_ID_goes_here_"
#define ACTION_TOPIC "/v1.1/actions/_your_DEVICE_ID_goes_here_"
#define MESSAGE_TOPIC "/v1.1/messages/_your_DEVICE_ID_goes_here_"
```

**NOTE:** The Device ID and Device Token should be of the device that you created for the “A05x Light and Fan Controller”.

## 9. Complete the Circuit.

9.1 Follow the given circuit to connect the fan to your ARTIK 055s board. In the diagram we are controlling the fan using GPIO Pin 41. However, feel free to connect the control wire to any other GPIO port (Refer to the Pin Map). Take note of the number suffixed to “gpio” in the pin map corresponding to the port you chose for controlling your fan and replace 41 after FAN\_PORT in main.c with the new number if you decide to use some other port.



## 10. Complete the Code.

10.1 The code uses a protocol known as MQTT wherein the ARTIK 055s subscribes to the cloud for actions. Every time the ARTIK 055s receives a message from the cloud, it calls the “onMessage” function in “main.c”. This is where the message is decoded and the magic happens. It’s left incomplete. Follow the next instruction to complete it.

10.2 Since, there are a lot of input and output components on the board, there is a unique number associated with each one of them, which functions exactly like an address. This number is called port number. For the fan we chose 41 in the previous step. To turn on the fan, because of the relay unit, one needs to write “0” to port 41 and to turn it off one needs to write “1” to port 41. The provided gpio\_write() function writes 0 to a port number if the port number is currently outputting 1. Toggle the fan by calling the gpio\_write() function as follows:

- gpio\_write(FAN\_PORT);

Calling this function will only turn on the fan. We have included the code to turn off the fan using the onboard button at port 42. Note if you call the gpio\_write function twice with the same port number, it'll write 1 to the specified port only once. You have to reset the port back to 0 in order to re-write 1. That's exactly what the button at port 42 does: resets the fan port back to 0 or in simple terms, turns off the fan so it can be switched on again.

11. Build your code: Click on the small hammer icon  to build your code.
12. Running your code: With your project highlighted, click on a small green hexagon with a lightning bolt  to upload your code to the board. After uploading or “flashing” the code, it'll run automatically.
13. Reboot your ARTIK055s device, and you should be able to see the messages below from the console.

```
TASH>>Current value of the gpio port45 is 0
Starting supplicant in foreground...
CIDHAR.
Current value of the gpio port45 is 1
Connect to Wi-Fi success
Get IP address
IP address 10.0.1.18
MAC: 28:6d:97:40:94:78
Registering mqtt client with id = 286d97409478
mqtt client connected to the server
mqtt client Subscribed to the topic successfully
mqtt client on message received
```

Now, if your body temperature reaches 40 degree, you can see the **toggleFan** action is triggered.

```
Topic - /v1.1/actions/016d42284feb48f7b56d60d20544e9b8 , payload -
{"actions":[{"name":"toggleFan","parameters":{}}]}
Toggling the Fan
mqtt client on message received
Topic - /v1.1/actions/016d42284feb48f7b56d60d20544e9b8 , payload -
{"actions":[{"name":"toggleFan","parameters":{}}]}
Toggling the Fan
```

## Bonus Exercises

1. Control the Fan (Toggle On/Off) connected with Artik 055 module using Node-red flow and Artik cloud Rules engine.
2. Send an Email notification on charging-usb disconnect from the Kitra board using Node-red.