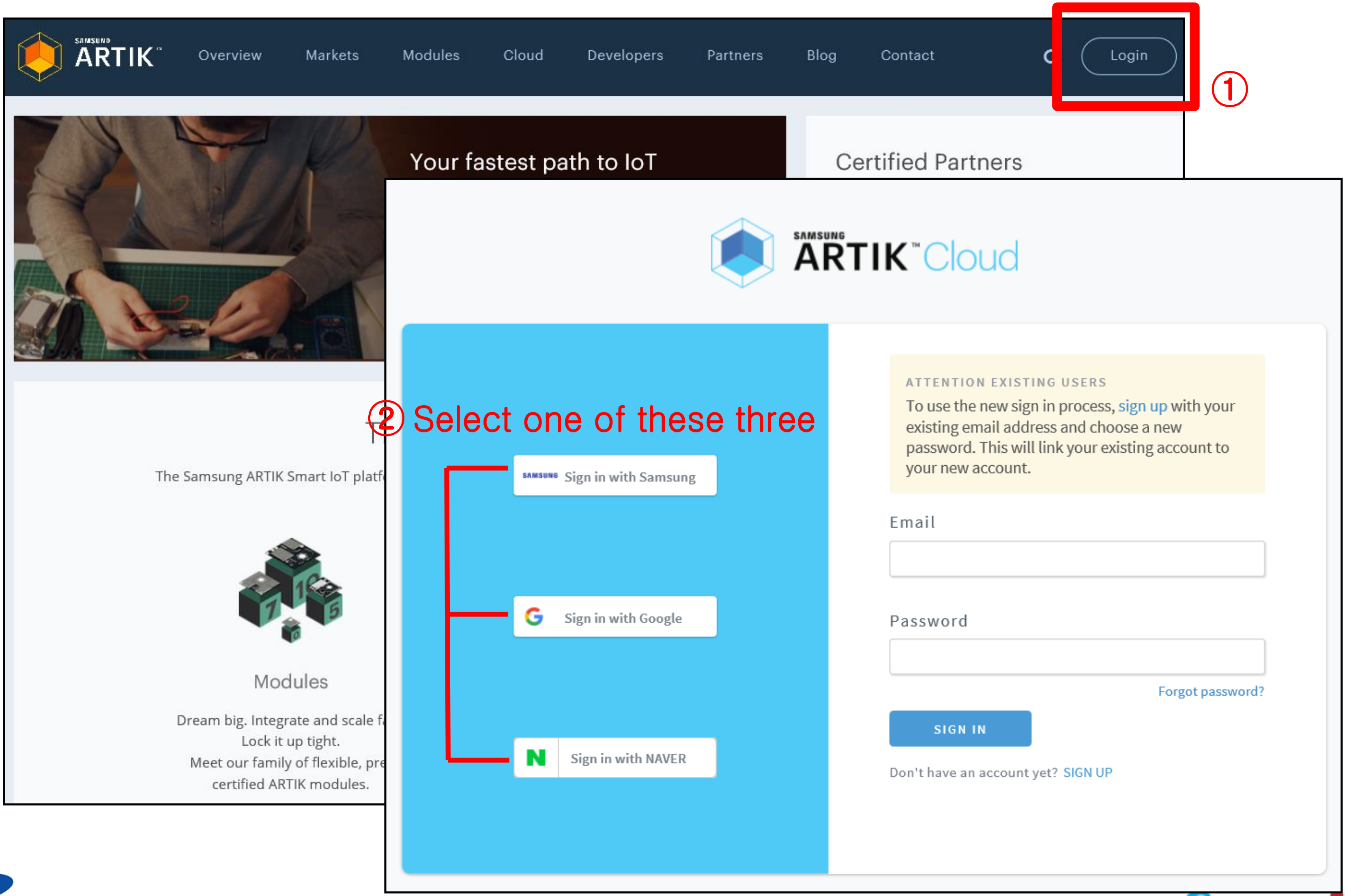
Ⅱ. ARTIK 기술 교육

8. ARTIK Cloud (1)

ARTIK Cloud

ARTIK Cloud Sign up



①

② Select one of these three

ATTENTION EXISTING USERS

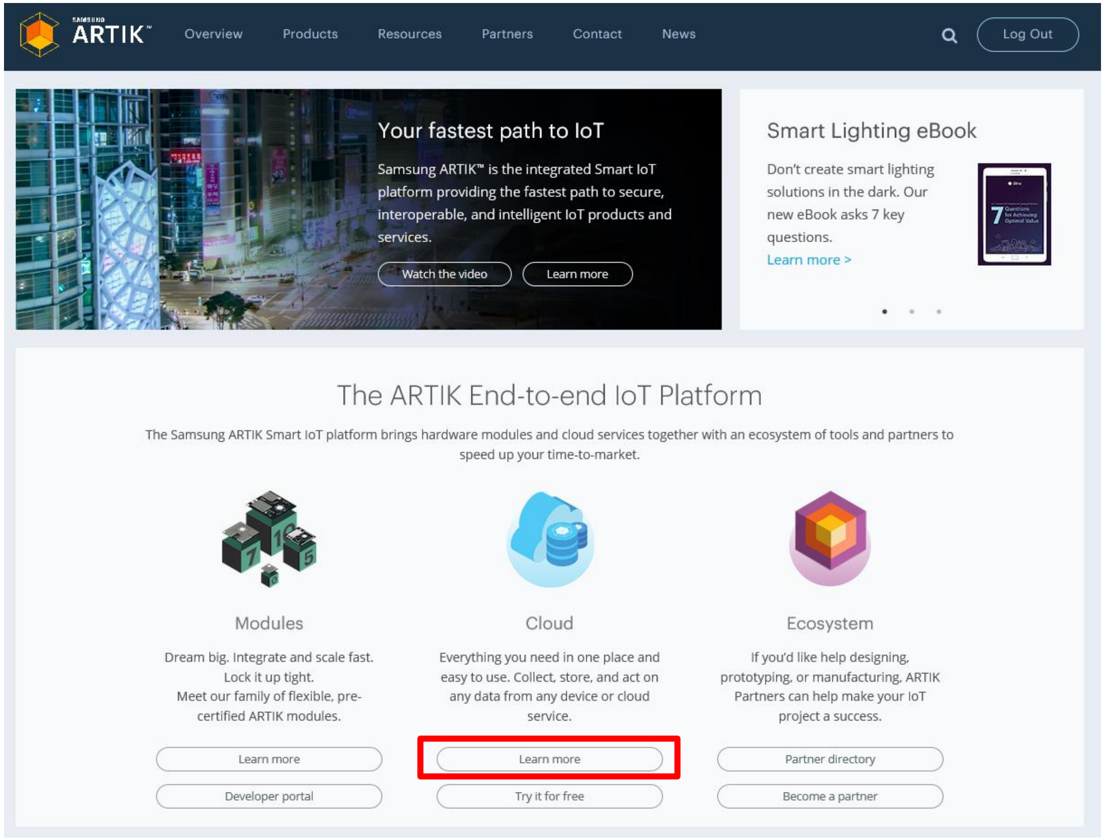
To use the new sign in process, sign up with your existing email address and choose a new password. This will link your existing account to your new account.

Don't have an account yet? SIGN UP

ARTIK Cloud

ARTIK Cloud

- ARTIK Cloud Menu



Your fastest path to IoT

Samsung ARTIK™ is the integrated Smart IoT platform providing the fastest path to secure, interoperable, and intelligent IoT products and services.

Smart Lighting eBook

Don't create smart lighting solutions in the dark. Our new eBook asks 7 key questions.

The ARTIK End-to-end IoT Platform

The Samsung ARTIK Smart IoT platform brings hardware modules and cloud services together with an ecosystem of tools and partners to speed up your time-to-market.

Modules

Dream big. Integrate and scale fast. Lock it up tight. Meet our family of flexible, pre-certified ARTIK modules.

Cloud

Everything you need in one place and easy to use. Collect, store, and act on any data from any device or cloud service.

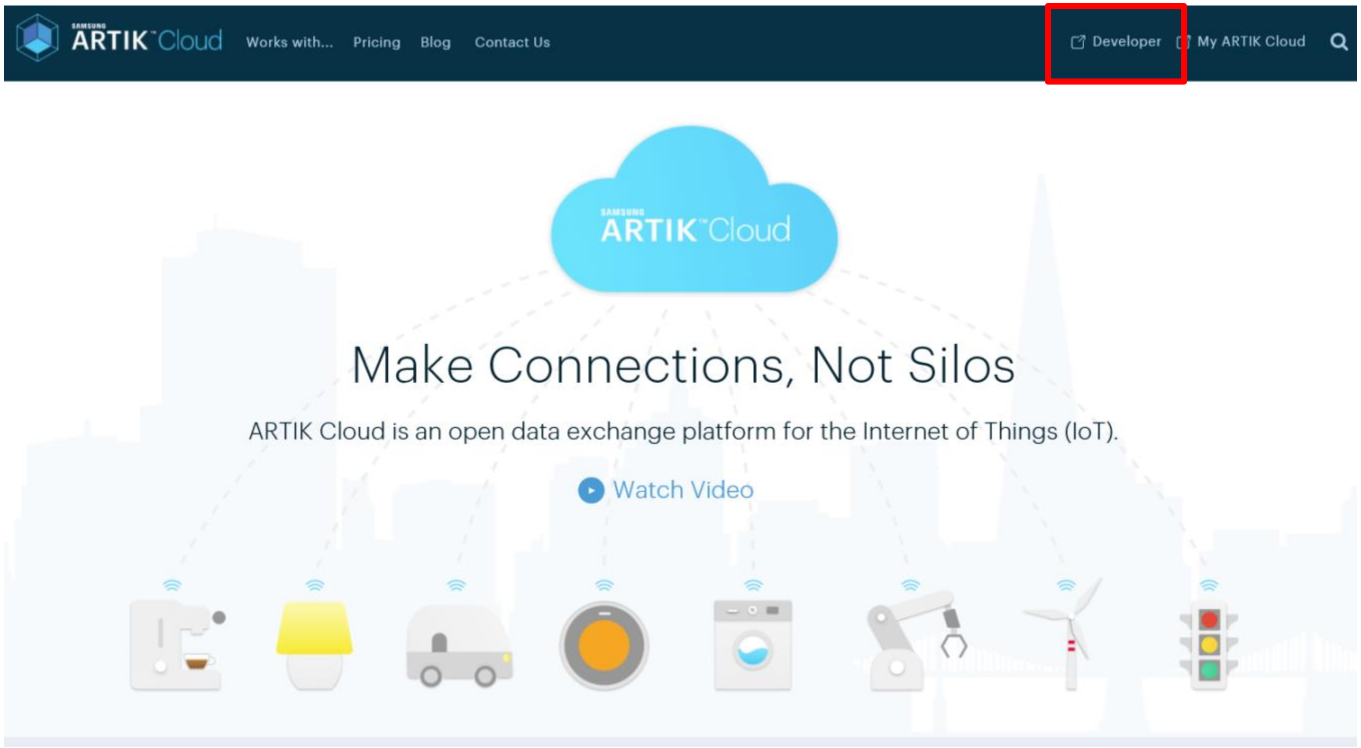
Ecosystem

If you'd like help designing, prototyping, or manufacturing, ARTIK Partners can help make your IoT project a success.

ARTIK Cloud

■ ARTIK Cloud

- Developers Menu

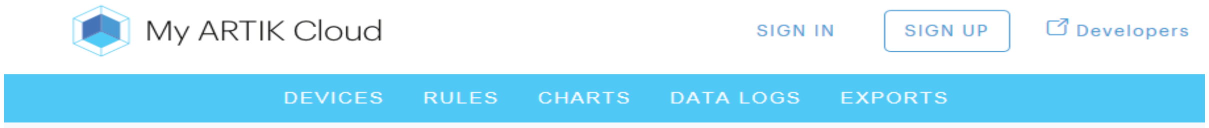


Make Connections, Not Silos

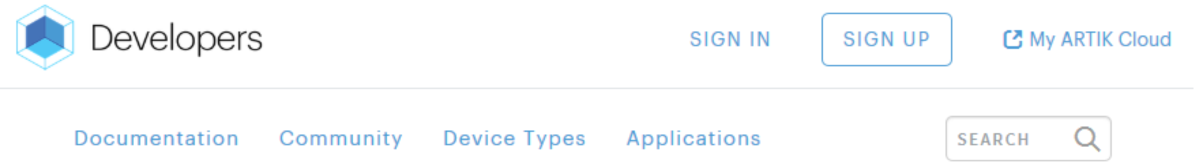
ARTIK Cloud is an open data exchange platform for the Internet of Things (IoT).

ARTIK Cloud

■ ARTIK Cloud Menu



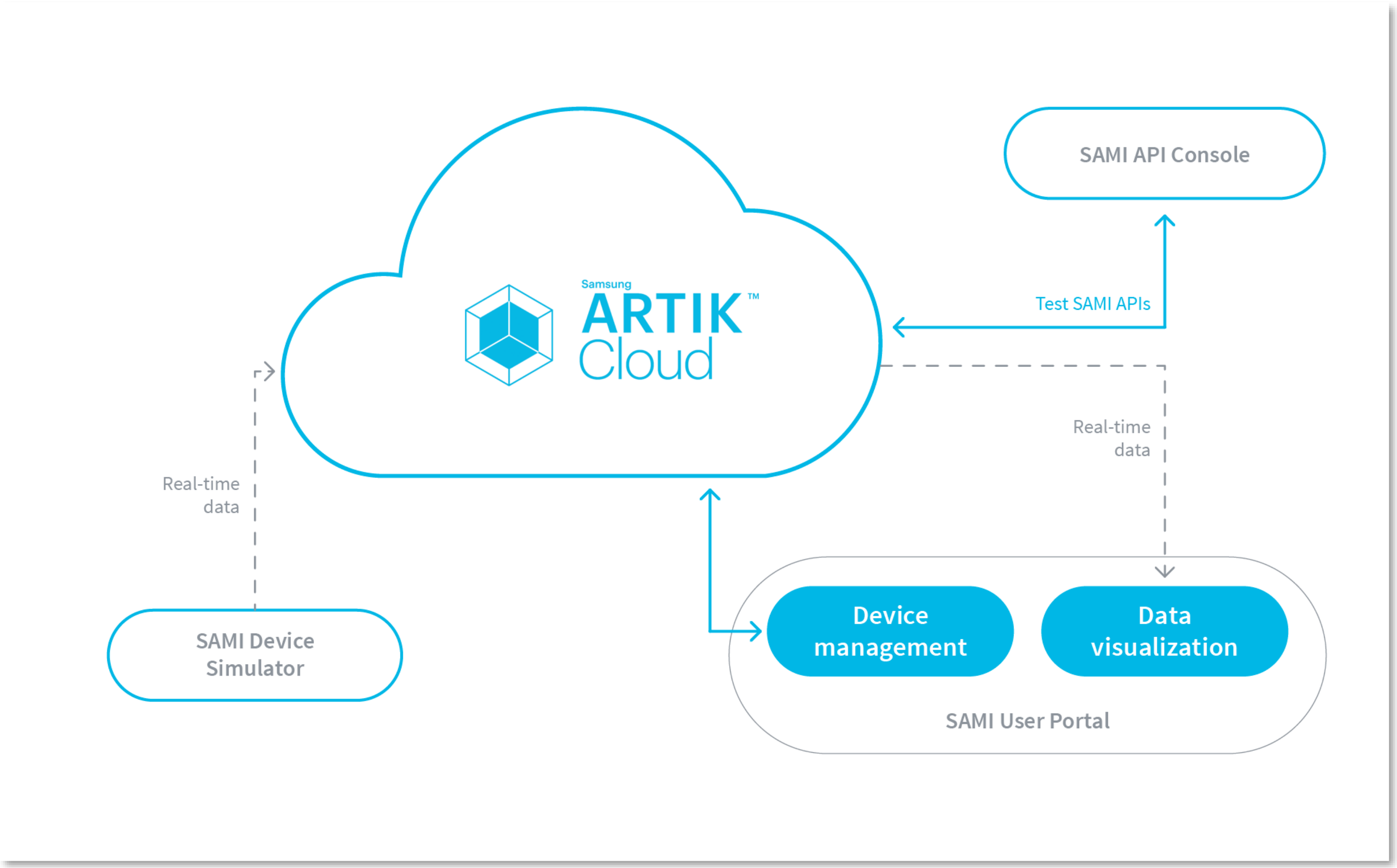
- My cloud : Interface for sending and receiving data to connected devices
  - DEVICES : Add or delete devices and check information about connected devices, Generate token for device
  - RULES : Create rules for sending and receiving data
  - CHARTS : Display data exchanged with device
  - DATA LOGS : Logs on data sent and received with the device
  - EXPORTS : Transfer data to the device



- Developer cloud : Archives Artik Cloud Development
  - Documentation : explanations for using artik cloud
  - Community : forums and blogs about artik cloud
  - Device Types : Create device type
  - Applications : Create an app to use the artik cloud

ARTIK Cloud

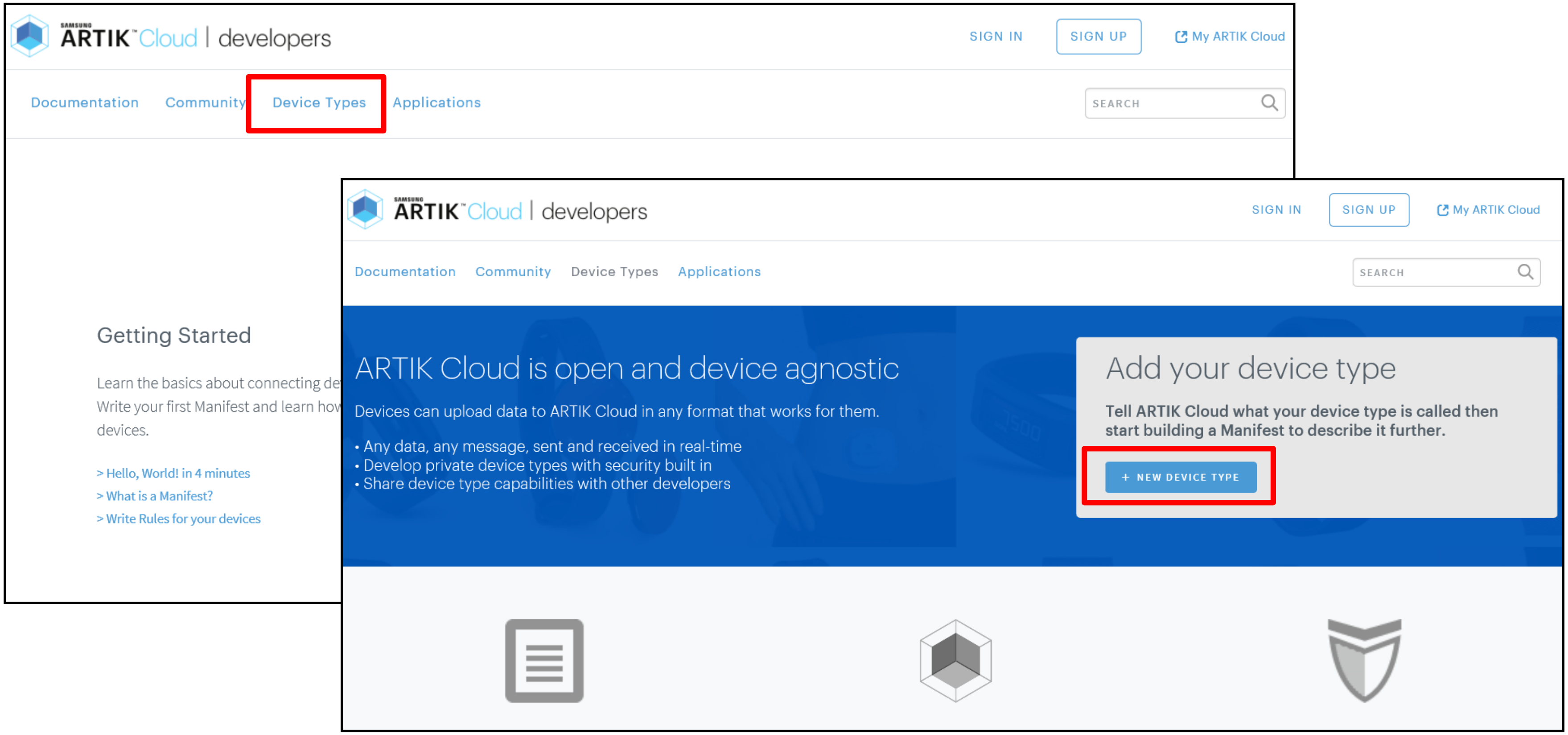
■ ARTIK Cloud



ARTIK Cloud

ARTIK Cloud tutorial

- Step 1 : Make a device type
  - Open the developers menu
  - Dashboard – Device types click
  - Click [+New device type]



SAMSUNG ARTIK Cloud | developers

SIGN IN SIGN UP My ARTIK Cloud

Documentation Community Device Types Applications SEARCH

Getting Started

Learn the basics about connecting de
Write your first Manifest and learn how
devices.

> Hello, World! in 4 minutes
> What is a Manifest?
> Write Rules for your devices

SAMSUNG ARTIK Cloud | developers

SIGN IN SIGN UP My ARTIK Cloud

Documentation Community Device Types Applications SEARCH

ARTIK Cloud is open and device agnostic

Devices can upload data to ARTIK Cloud in any format that works for them.

- Any data, any message, sent and received in real-time
- Develop private device types with security built in
- Share device type capabilities with other developers

Add your device type

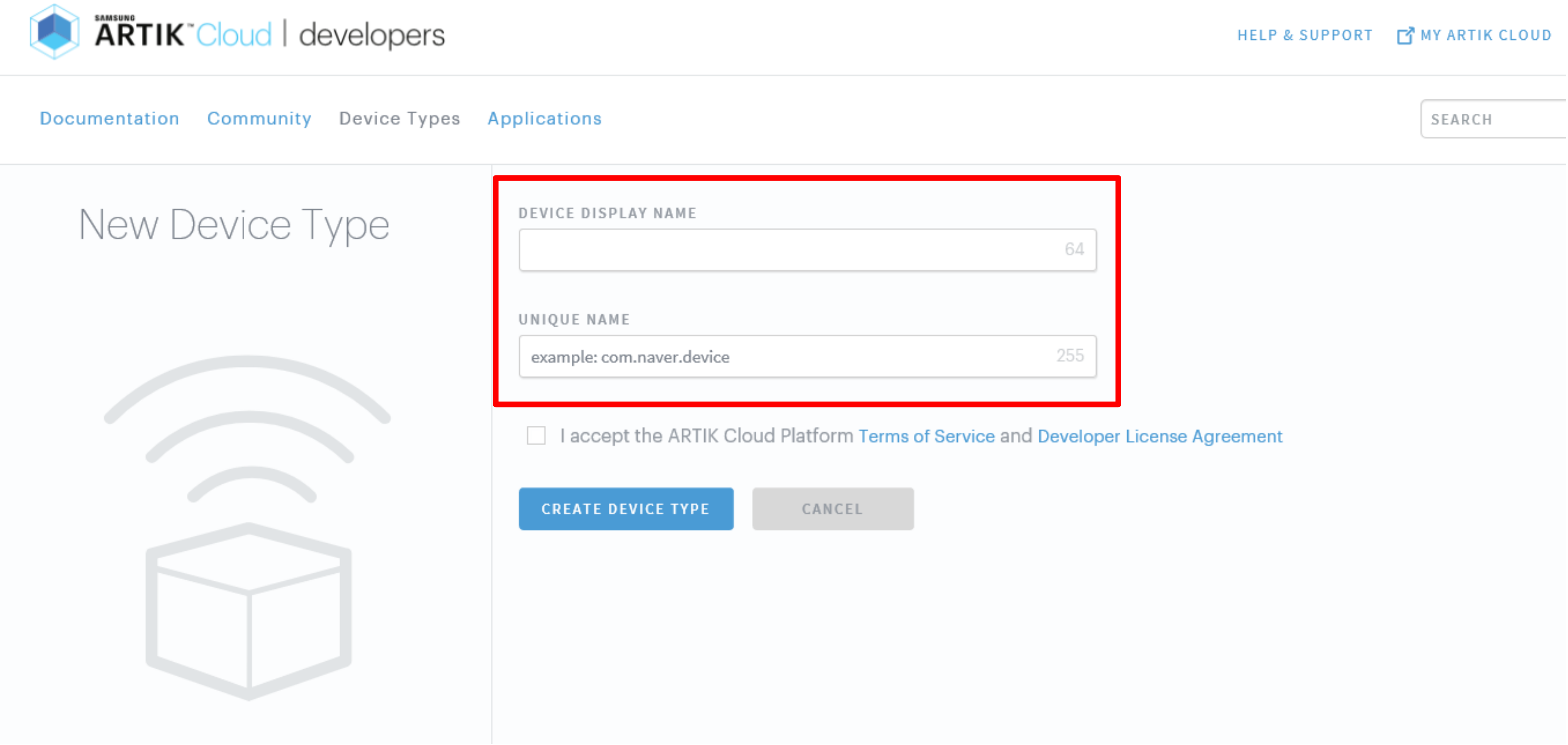
Tell ARTIK Cloud what your device type is called then start building a Manifest to describe it further.

+ NEW DEVICE TYPE

ARTIK Cloud

ARTIK Cloud tutorial

- Step 1 : Make a device type
  - Input Device Display Name (ex) ARTIK new type
  - Input Unique Name (ex) artik.skku.test
  - Click [Create Device Type]



New Device Type

DEVICE DISPLAY NAME
64

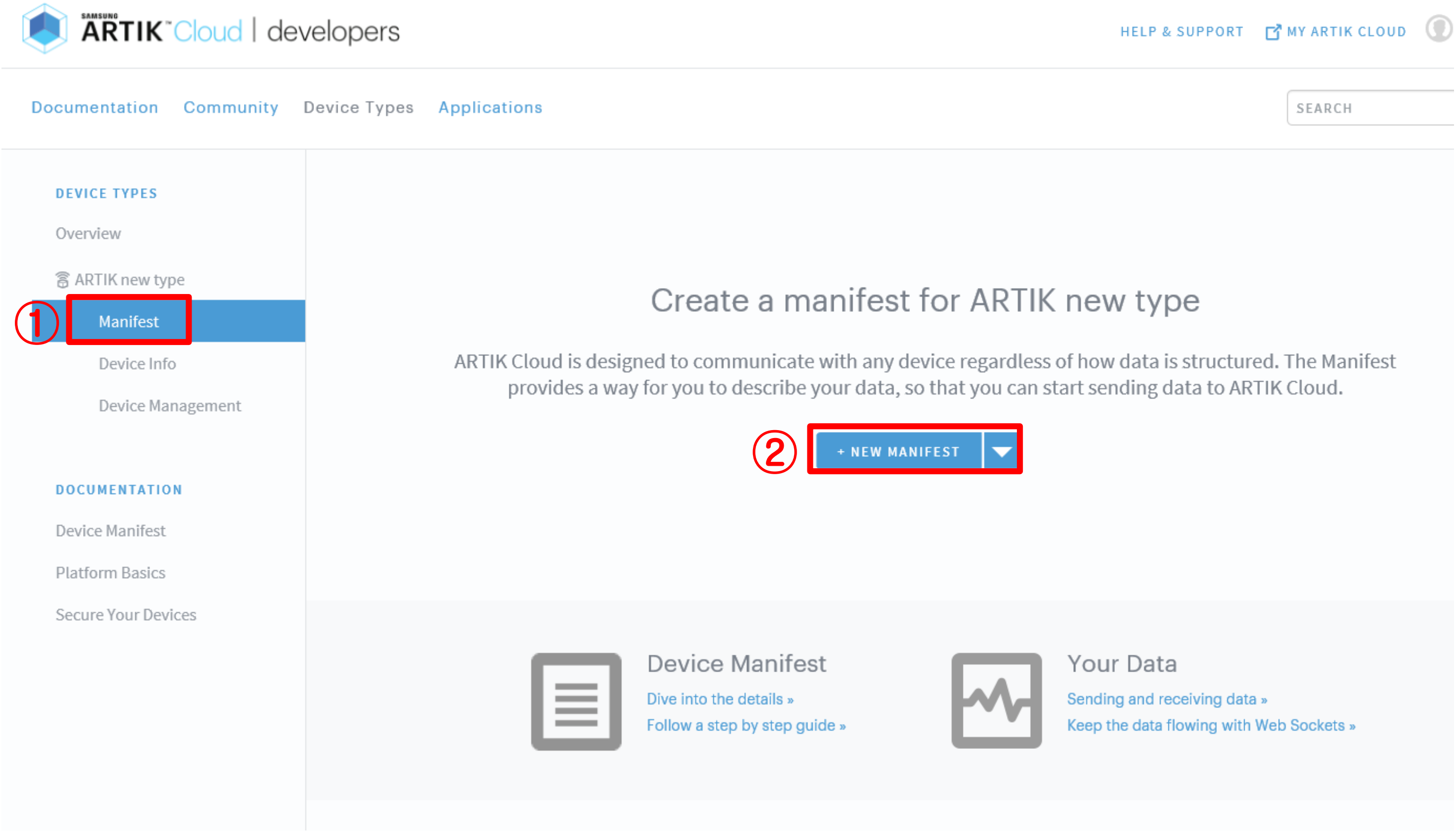
UNIQUE NAME
example: com.naver.device
255

□ I accept the ARTIK Cloud Platform Terms of Service and Developer License Agreement

ARTIK Cloud

■ ARTIK Cloud tutorial

- Step 2 : Make a manifest
  - Check 'ARTIK new type'
  - Click [+ New Manifest]



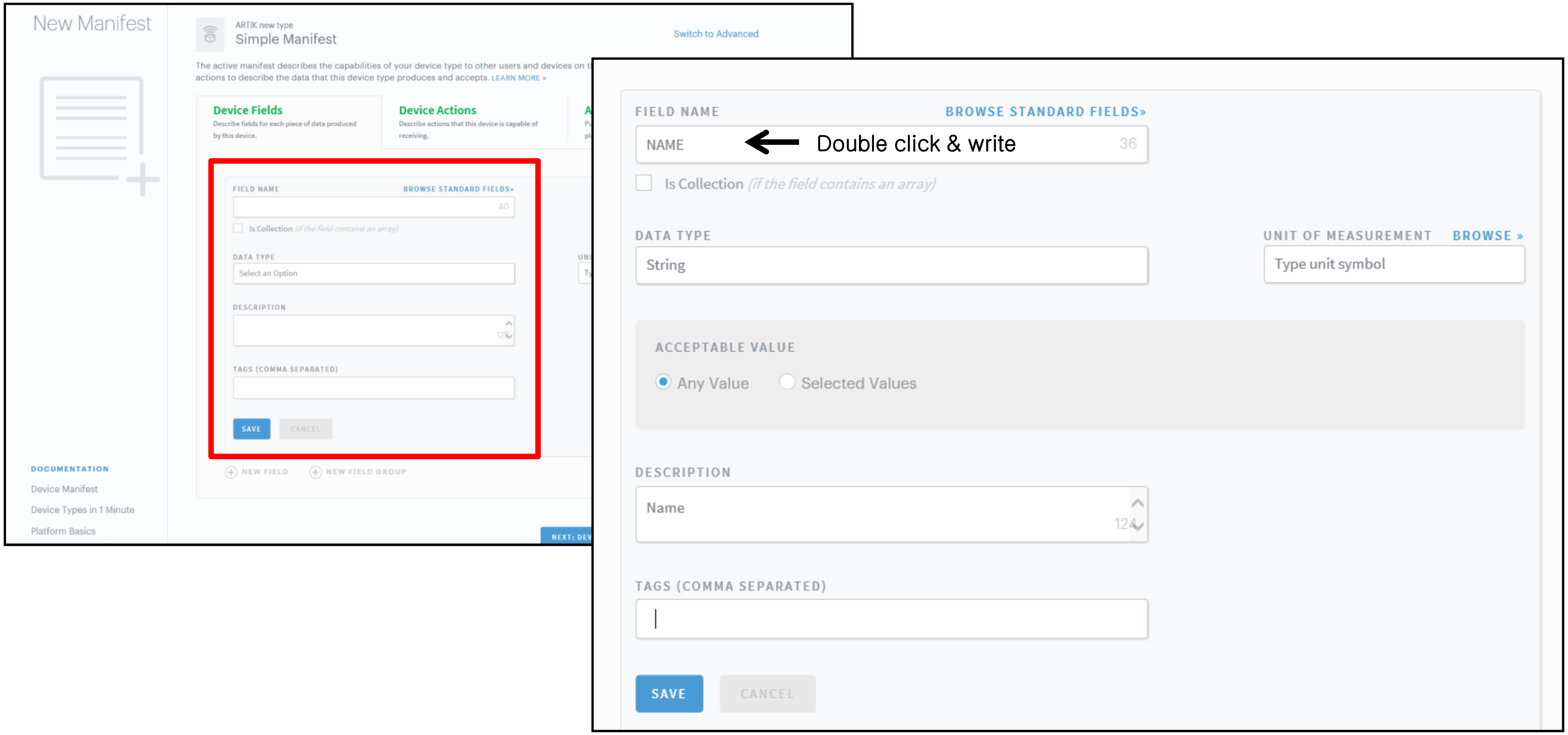
Create a manifest for ARTIK new type

ARTIK Cloud is designed to communicate with any device regardless of how data is structured. The Manifest provides a way for you to describe your data, so that you can start sending data to ARTIK Cloud.

ARTIK Cloud

■ ARTIK Cloud tutorial

- Step 2 : Make a manifest
  - Make messages (from ARTIK to ARTIK cloud)
  - Type 'NAME' in 'Field Name' and select 'string' in 'Data Type'
  - Description is option



New Manifest

ARTIK new type
Simple Manifest

Switch to Advanced

The active manifest describes the capabilities of your device type to other users and devices on t
actions to describe the data that this device type produces and accepts. LEARN MORE »

Device Fields
Describe fields for each piece of data produced by this device.

Device Actions
Describe actions that this device is capable of receiving.

FIELD NAME — BROWSE STANDARD FIELDS»
40
□ Is Collection *(if the field contains an array)*
DATA TYPE
Select an Option
DESCRIPTION
128
TAGS (COMMA SEPARATED)
SAVE CANCEL

NEW FIELD NEW FIELD GROUP

DOCUMENTATION
Device Manifest
Device Types in 1 Minute
Platform Basics

NEXT: DEV

FIELD NAME — BROWSE STANDARD FIELDS»
NAME ← Double click & write 36
□ Is Collection *(if the field contains an array)*

DATA TYPE
String

UNIT OF MEASUREMENT — BROWSE »
Type unit symbol

ACCEPTABLE VALUE
● Any Value ○ Selected Values

DESCRIPTION
Name
124

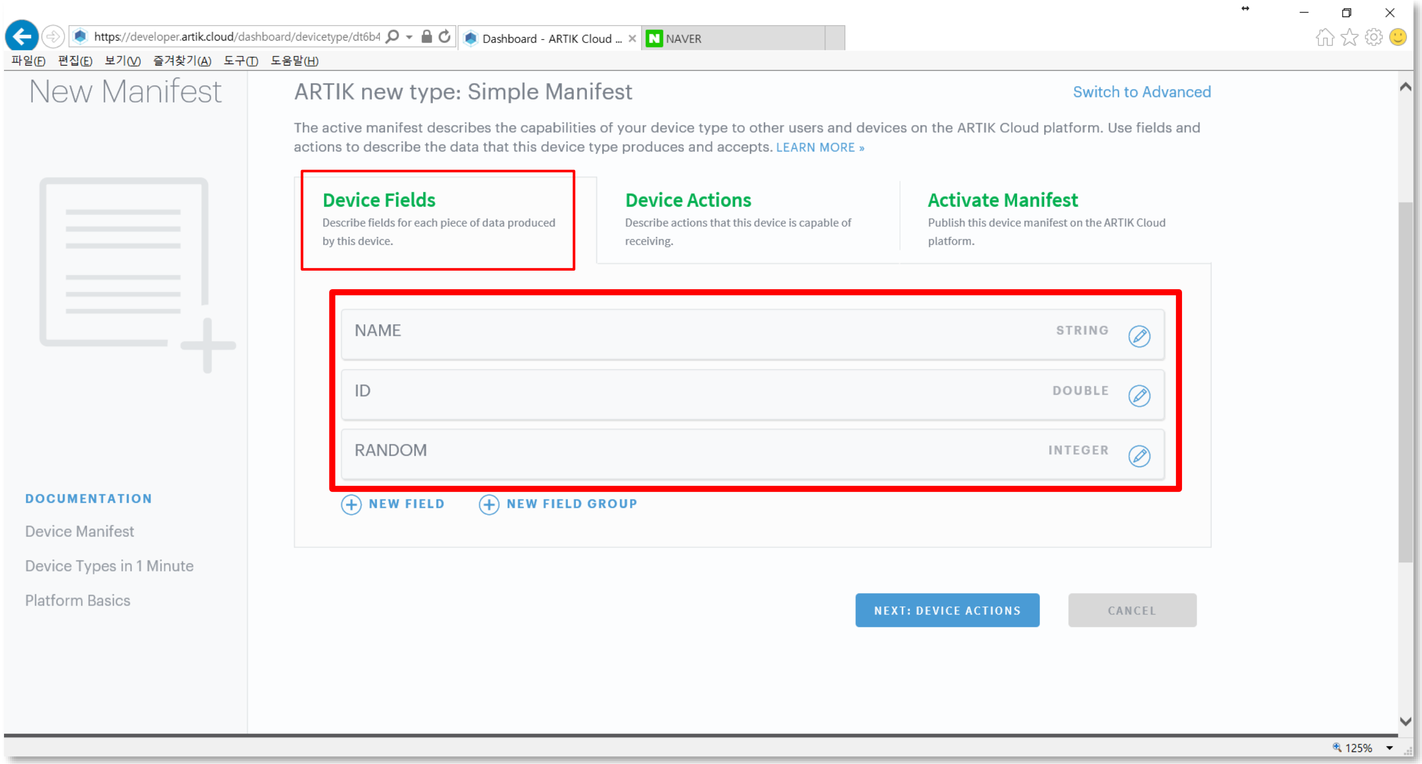
TAGS (COMMA SEPARATED)

SAVE CANCEL

ARTIK Cloud

ARTIK Cloud tutorial

- Step 2 : Make a manifest
  - Click 'NEW FIELD'
  - Type 'ID' in 'Field Name' and select 'double' in 'Data Type'
  - Type 'RAMDOM' in 'Field Name' and select 'Integer' in 'Data Type' as same ways.



https://developer.artik.cloud/dashboard/devicetype/dt6b4
Dashboard - ARTIK Cloud ...
NAVER
파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)
125%

New Manifest

ARTIK new type: Simple Manifest

Switch to Advanced

The active manifest describes the capabilities of your device type to other users and devices on the ARTIK Cloud platform. Use fields and actions to describe the data that this device type produces and accepts. LEARN MORE »

Device Fields
Describe fields for each piece of data produced by this device.

Device Actions
Describe actions that this device is capable of receiving.

Activate Manifest
Publish this device manifest on the ARTIK Cloud platform.

NAME	STRING
ID	DOUBLE
RANDOM	INTEGER

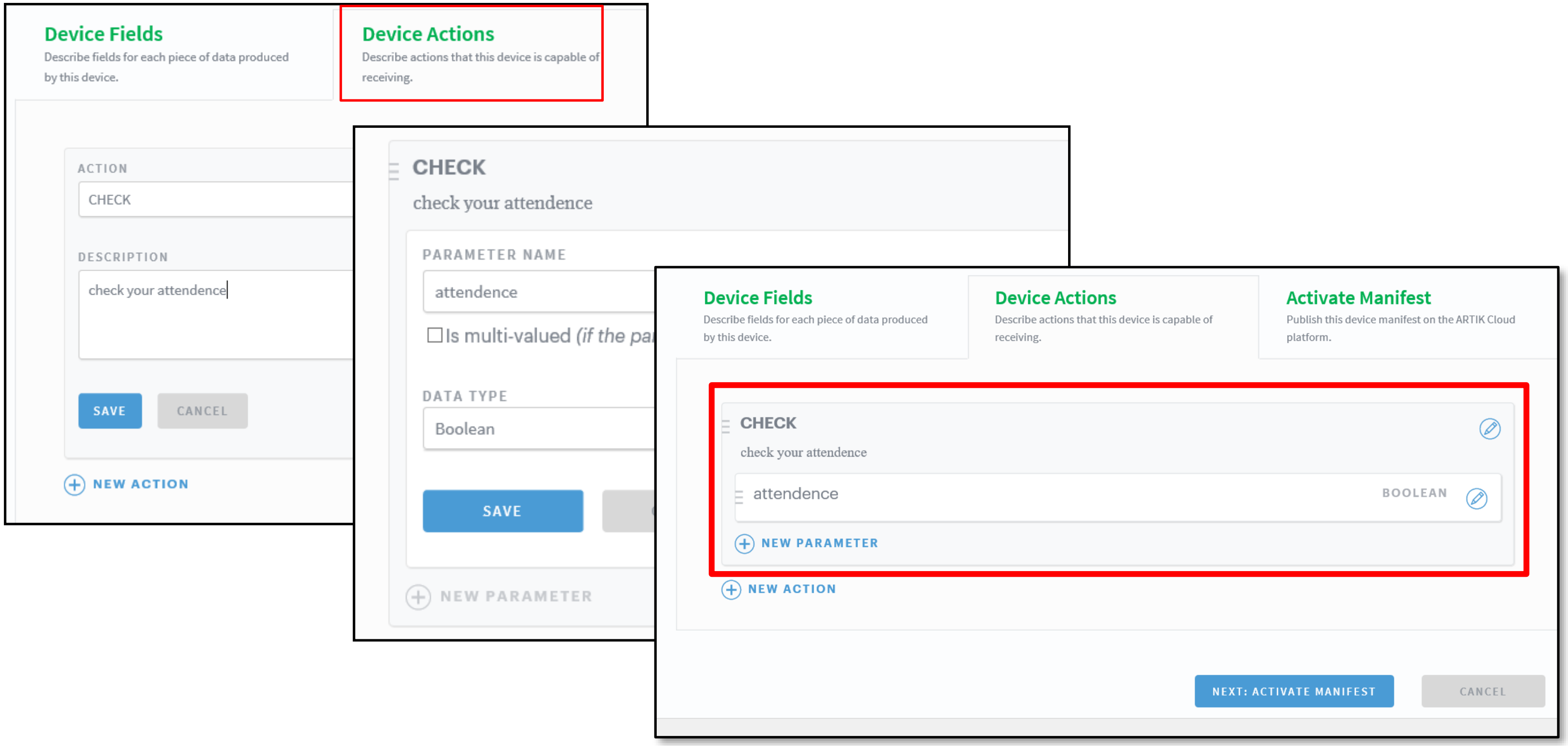
⊕ NEW FIELD ⊕ NEW FIELD GROUP

NEXT: DEVICE ACTIONS CANCEL

DOCUMENTATION
Device Manifest
Device Types in 1 Minute
Platform Basics

ARTIK Cloud tutorial

- Step 2 : Make a manifest
  - Make a action (from ARTIK cloud to ARTIK)
  - Click [NEXT : DEVICE ACTIONS]
  - Make 'CHECK' action
  - Click '+ NEW PARAMETER' and make Boolean type parameter.



Device Fields
Describe fields for each piece of data produced by this device.

Device Actions
Describe actions that this device is capable of receiving.

ACTION: CHECK

DESCRIPTION: check your attendence

CHECK
check your attendence

PARAMETER NAME: attendence

☐ Is multi-valued (if the pa

DATA TYPE: Boolean

Device Fields
Describe fields for each piece of data produced by this device.

Device Actions
Describe actions that this device is capable of receiving.

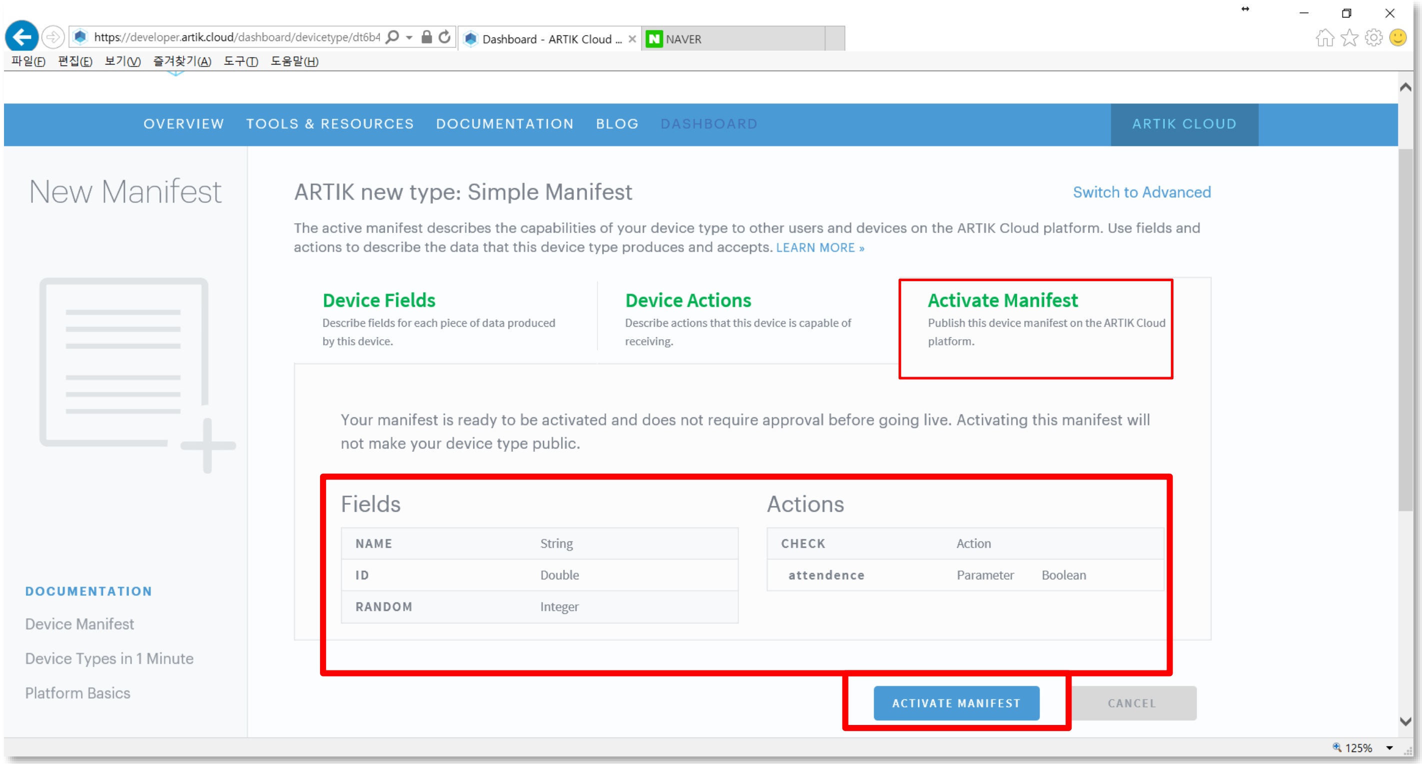
Activate Manifest
Publish this device manifest on the ARTIK Cloud platform.

CHECK
check your attendence

attendence BOOLEAN

■ ARTIK Cloud tutorial

- Step 2 : Make a manifest
  - Check Fields and Actions
  - Click [ACTIVATE MANIFEST]


https://developer.artik.cloud/dashboard/devicetype/dt6b4
Dashboard - ARTIK Cloud ...
NAVER
파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)
OVERVIEW
TOOLS & RESOURCES
DOCUMENTATION
BLOG
DASHBOARD
ARTIK CLOUD
125%

New Manifest

ARTIK new type: Simple Manifest

Switch to Advanced

The active manifest describes the capabilities of your device type to other users and devices on the ARTIK Cloud platform. Use fields and actions to describe the data that this device type produces and accepts. LEARN MORE »

Device Fields
Describe fields for each piece of data produced by this device.

Device Actions
Describe actions that this device is capable of receiving.

Activate Manifest
Publish this device manifest on the ARTIK Cloud platform.

Your manifest is ready to be activated and does not require approval before going live. Activating this manifest will not make your device type public.

Fields

NAME	String
ID	Double
RANDOM	Integer

Actions

CHECK	Action	
attendence	Parameter	Boolean

ACTIVATE MANIFEST   CANCEL

DOCUMENTATION

Device Manifest

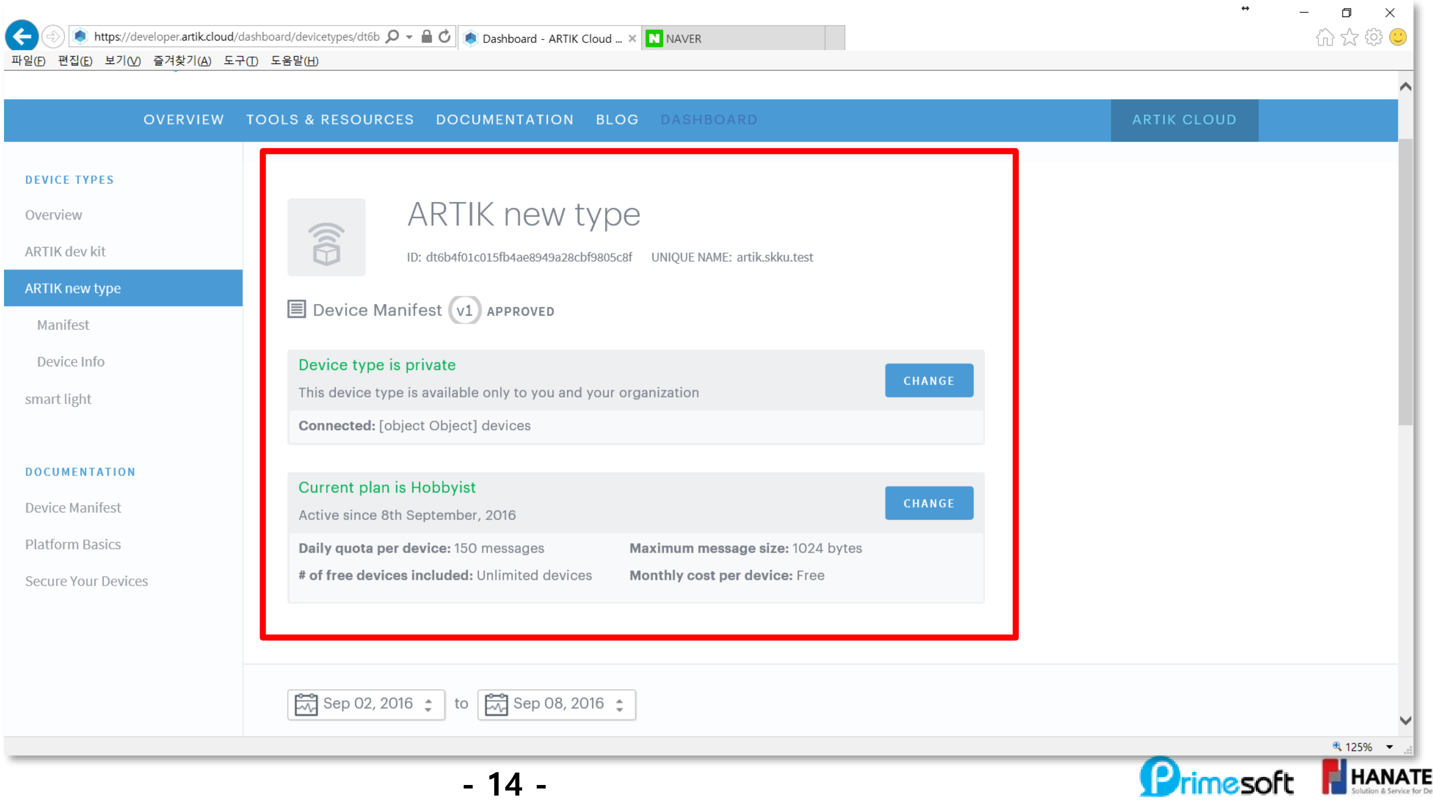
Device Types in 1 Minute

Platform Basics

ARTIK Cloud

■ ARTIK Cloud tutorial

- Step 2 : Make a manifest
  - Check Device Manifest



ARTIK new type

ID: dt6b4f01c015fb4ae8949a28cbf9805c8f UNIQUE NAME: artik.skku.test

Device Manifest v1 APPROVED

Device type is private

This device type is available only to you and your organization

Connected: [object Object] devices

Current plan is Hobbyist

Active since 8th September, 2016

Daily quota per device: 150 messages

Maximum message size: 1024 bytes

# of free devices included: Unlimited devices

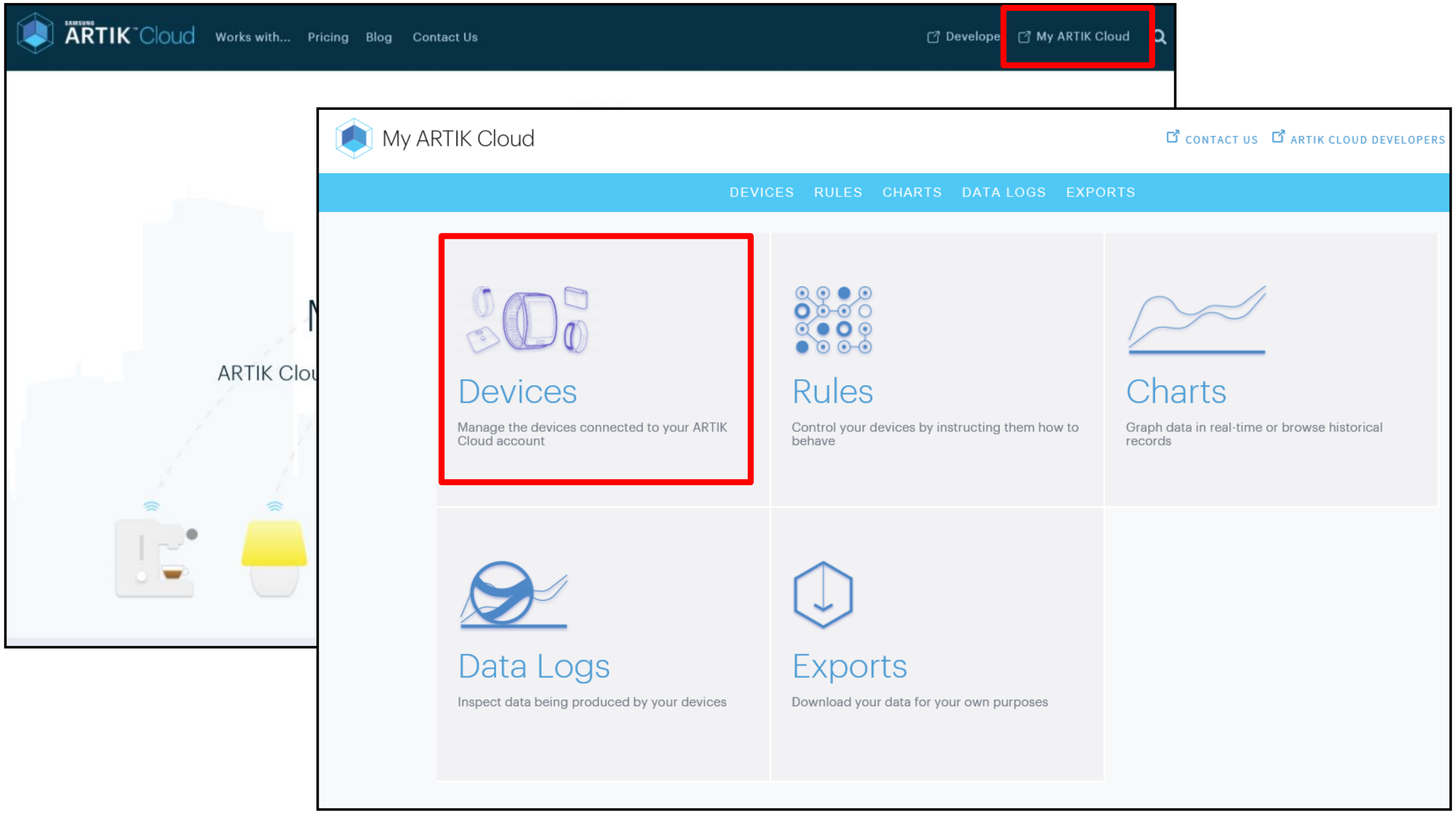
Monthly cost per device: Free

Sep 02, 2016 to Sep 08, 2016

ARTIK Cloud

ARTIK Cloud tutorial

- Step 3 : Make a device
  - Open the ARTIK cloud menu
  - Click ‘My ARTIK Cloud’ – ‘Devices’



Devices

Manage the devices connected to your ARTIK Cloud account

Rules

Control your devices by instructing them how to behave

Charts

Graph data in real-time or browse historical records

Data Logs

Inspect data being produced by your devices

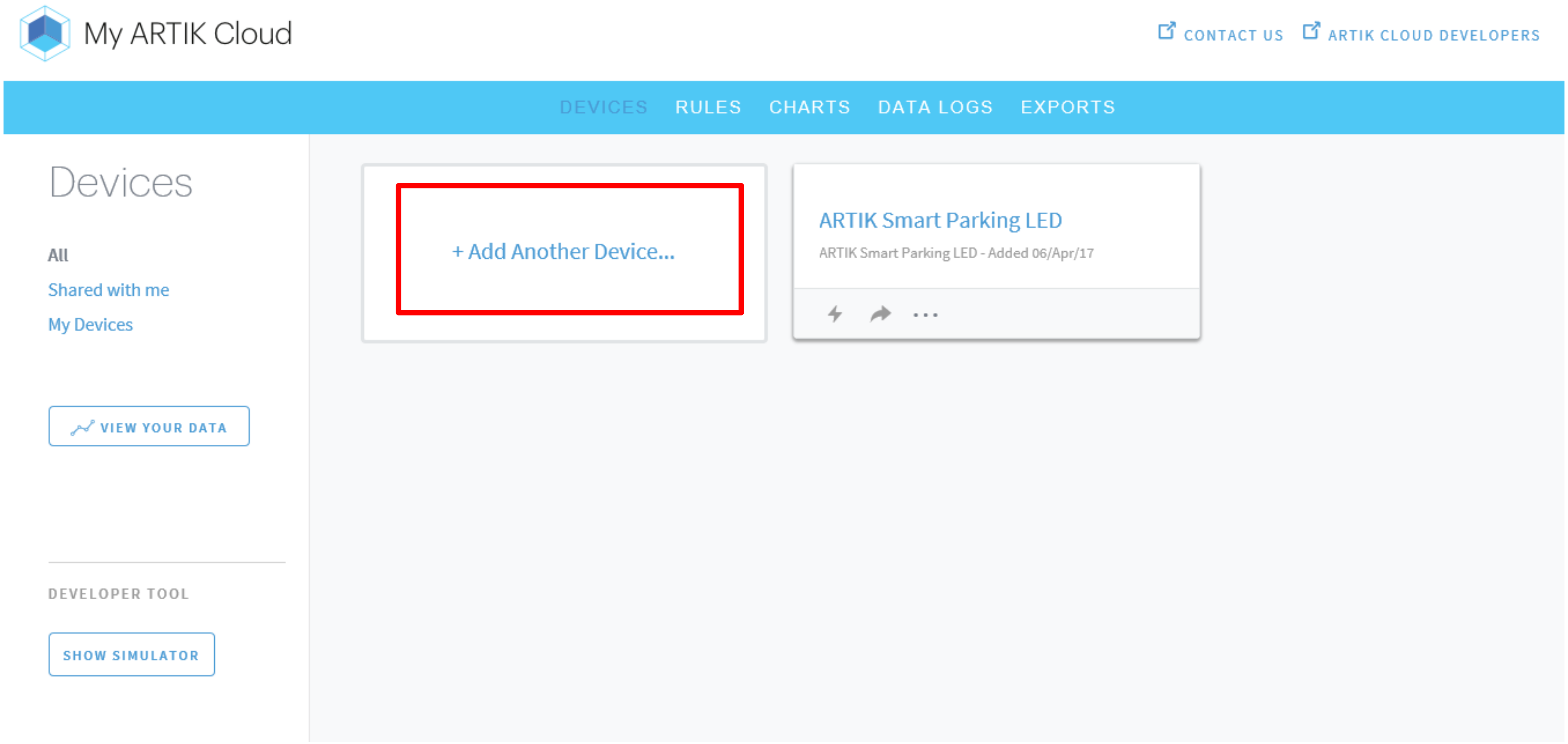
Exports

Download your data for your own purposes

ARTIK Cloud

■ ARTIK Cloud tutorial

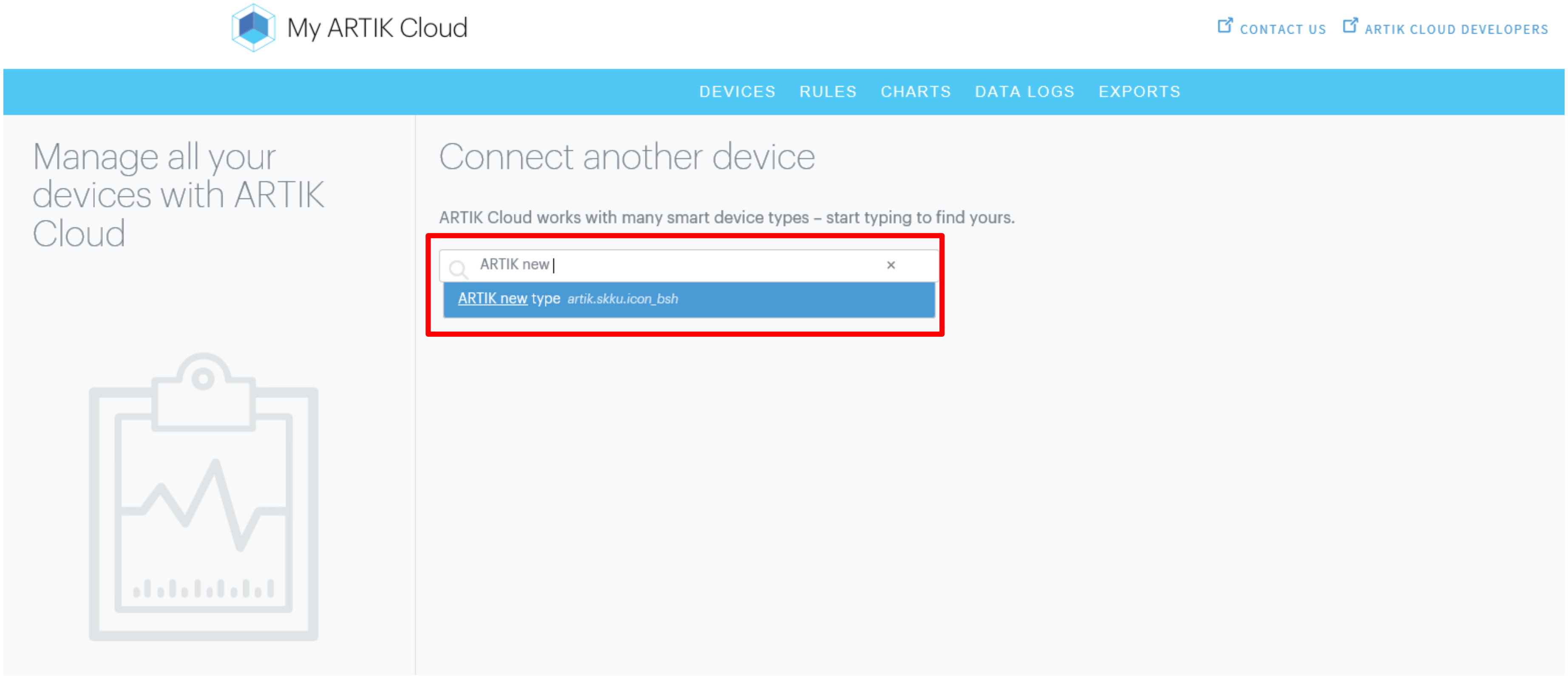
- Step 3 : Make a device
  - You can see all your devices in this page
  - Click [+Add Another Device…]



ARTIK Cloud

ARTIK Cloud tutorial

- Step 3 : Make a device
  - If you type 'ARTIK new type', you will see like that figure.
  - Click 'ARTIK new type'



Manage all your devices with ARTIK Cloud

Connect another device

ARTIK Cloud works with many smart device types – start typing to find yours.

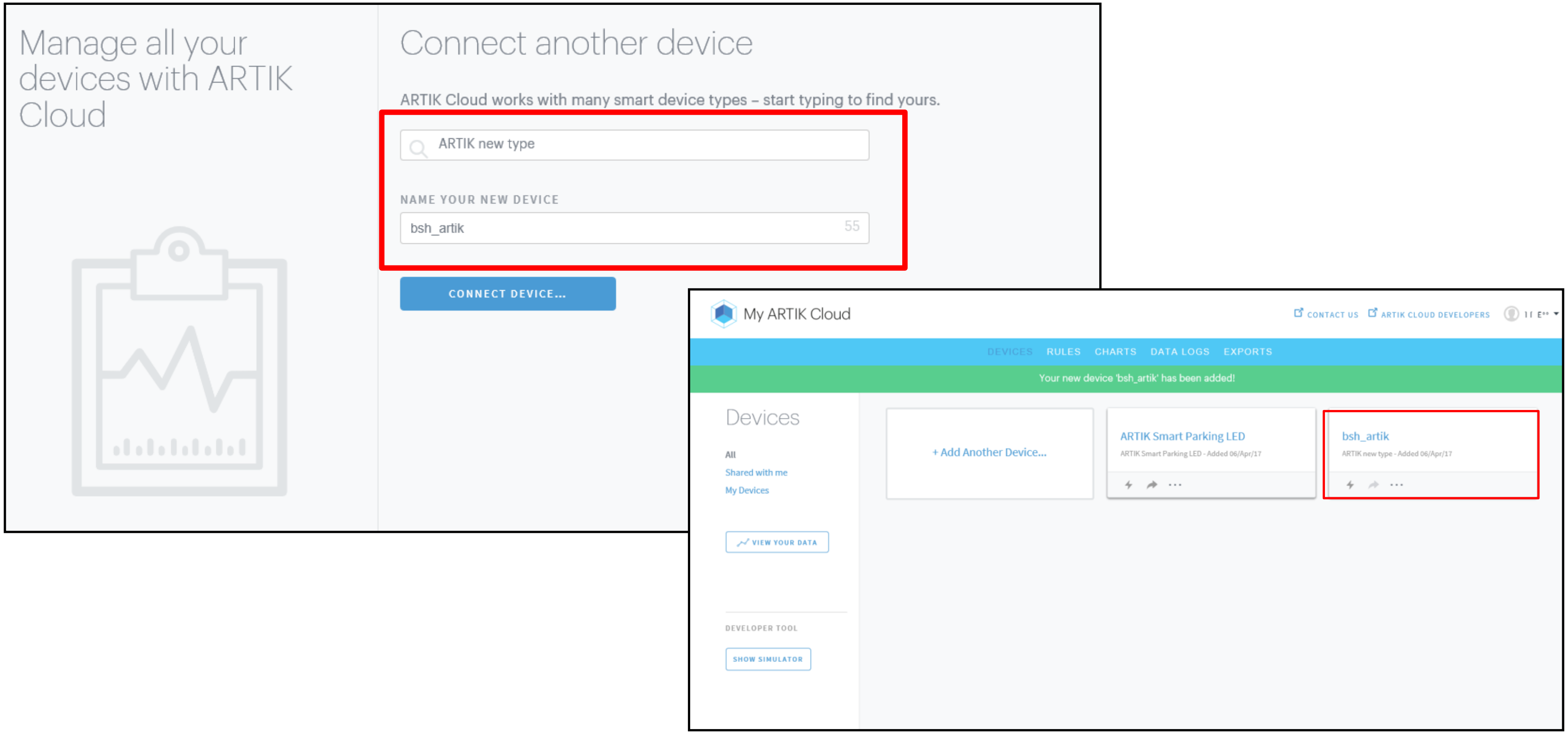
ARTIK new

ARTIK new type artik.skku.icon_bsh

ARTIK Cloud

■ ARTIK Cloud tutorial

- Step 3 : Make a device
  - Type 'Name Your New Device' something. (ex)bsh-artik
  - And you will see this device on 'My ARTIK Cloud' – 'Devices' page.



Manage all your devices with ARTIK Cloud

Connect another device

ARTIK Cloud works with many smart device types – start typing to find yours.

ARTIK new type

NAME YOUR NEW DEVICE

bsh_artik 55

CONNECT DEVICE...

My ARTIK Cloud

Your new device 'bsh_artik' has been added!

Devices

All
Shared with me
My Devices

+ Add Another Device...

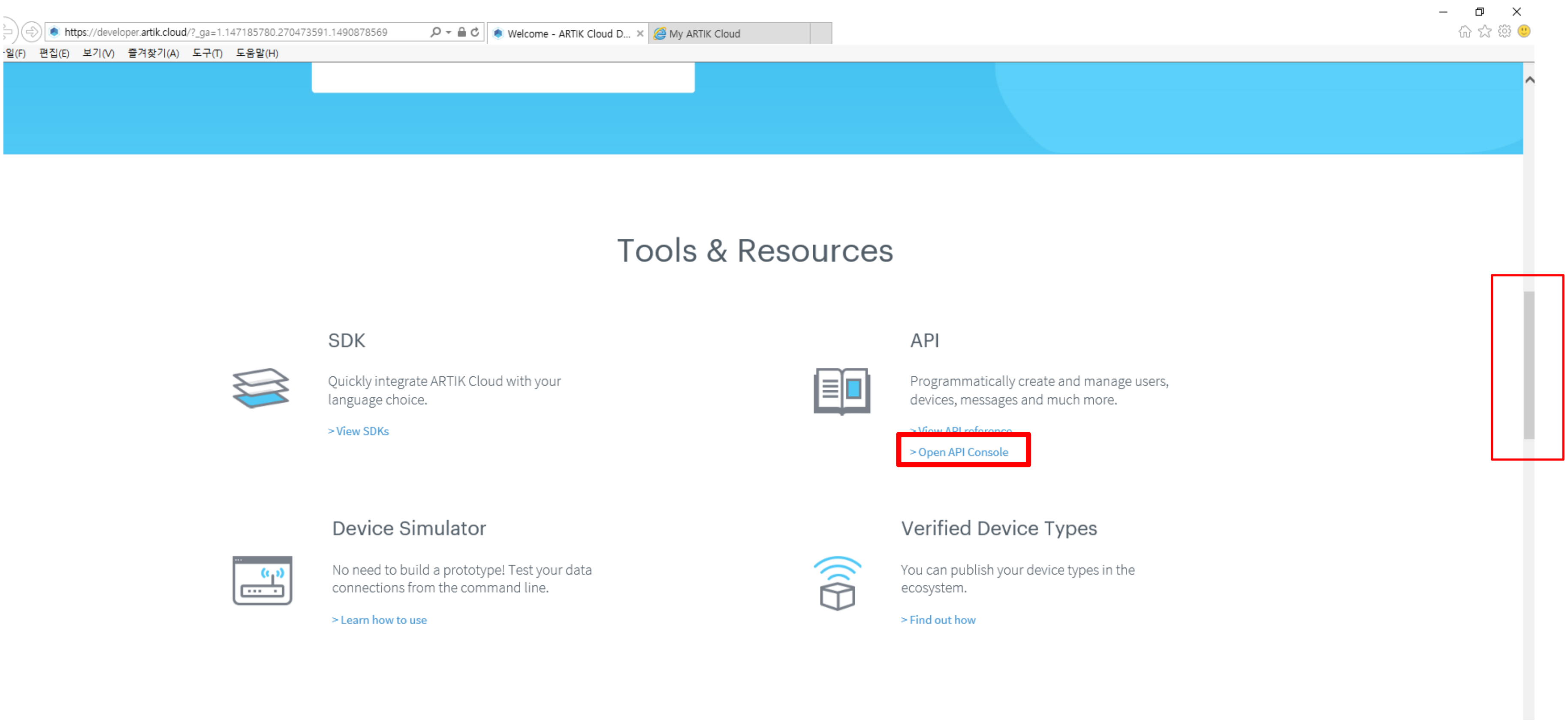
ARTIK Smart Parking LED
ARTIK Smart Parking LED - Added 06/Apr/17

bsh_artik
ARTIK new type - Added 06/Apr/17

ARTIK Cloud

ARTIK Cloud tutorial

- Step 4 : Check your Token
  - Open the developers menu
  - Scroll down
  - Tools & Resources : Click 'Open API Console'


https://developer.artik.cloud/?_ga=1.147185780.270473591.1490878569
Welcome - ARTIK Cloud D...
My ARTIK Cloud
일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)

Tools & Resources

SDK

Quickly integrate ARTIK Cloud with your language choice.

> View SDKs

API

Programmatically create and manage users, devices, messages and much more.

> View API reference

> Open API Console

Device Simulator

No need to build a prototype! Test your data connections from the command line.

> Learn how to use

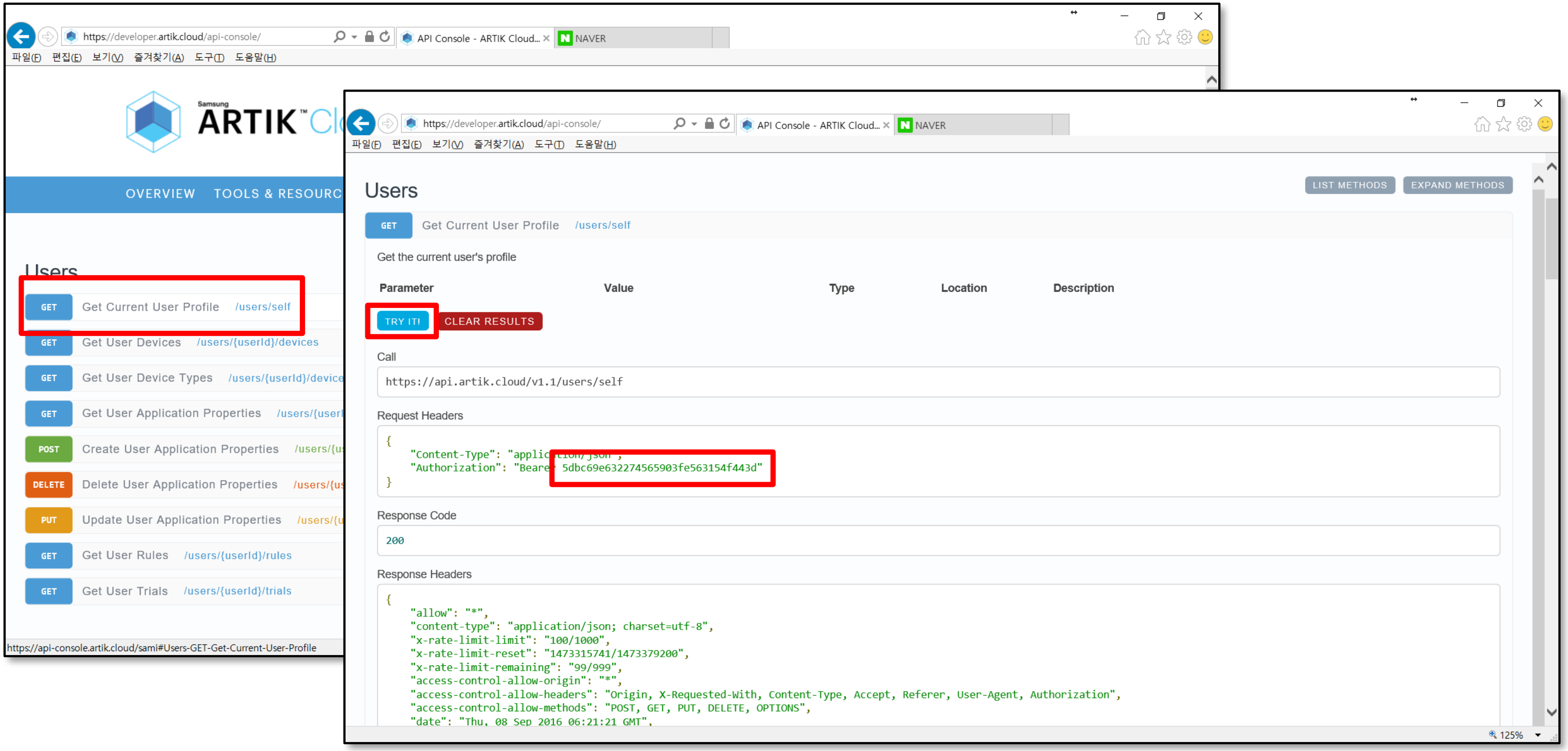
Verified Device Types

You can publish your device types in the ecosystem.

> Find out how

ARTIK Cloud tutorial

- Step 4 : Check your Token
  - Click ‘Get Current User Profile’ and [Try It]
  - You will see your token



https://developer.artik.cloud/api-console/
API Console - ARTIK Cloud...
NAVER
파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)
Samsung ARTIK Cl
OVERVIEW
TOOLS & RESOURC

Users

GET	Get Current User Profile	/users/self
GET	Get User Devices	/users/{userId}/devices
GET	Get User Device Types	/users/{userId}/device
GET	Get User Application Properties	/users/{user
POST	Create User Application Properties	/users/{u
DELETE	Delete User Application Properties	/users/{u
PUT	Update User Application Properties	/users/{u
GET	Get User Rules	/users/{userId}/rules
GET	Get User Trials	/users/{userId}/trials

https://api-console.artik.cloud/sami#Users-GET-Get-Current-User-Profile

https://developer.artik.cloud/api-console/
API Console - ARTIK Cloud...
NAVER
파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)
LIST METHODS
EXPAND METHODS
TRY IT!
CLEAR RESULTS
125%

Users

GET Get Current User Profile /users/self

Get the current user's profile

Parameter	Value	Type	Location	Description

Call

```
https://api.artik.cloud/v1.1/users/self
```

Request Headers

```
{
    "Content-Type": "application/json",
    "Authorization": "Bearer 5dbc69e632274565903fe563154f443d"
}
```

Response Code

```
200
```

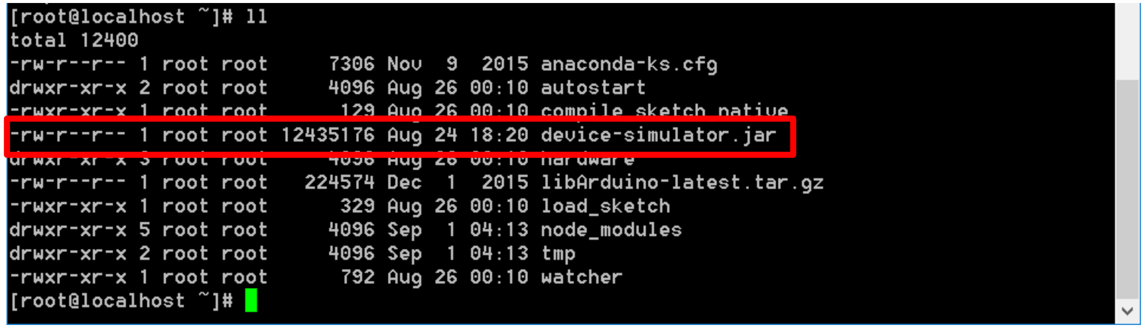
Response Headers

```
{
    "allow": "*",
    "content-type": "application/json; charset=utf-8",
    "x-rate-limit-limit": "100/1000",
    "x-rate-limit-reset": "1473315741/1473379200",
    "x-rate-limit-remaining": "99/999",
    "access-control-allow-origin": "*",
    "access-control-allow-headers": "Origin, X-Requested-With, Content-Type, Accept, Referer, User-Agent, Authorization",
    "access-control-allow-methods": "POST, GET, PUT, DELETE, OPTIONS",
    "date": "Thu, 08 Sep 2016 06:21:21 GMT",
```

ARTIK Cloud tutorial

- Step 5 : Run Device simulator in ARTIK 5 dev. kit
  - Download Device Simulator

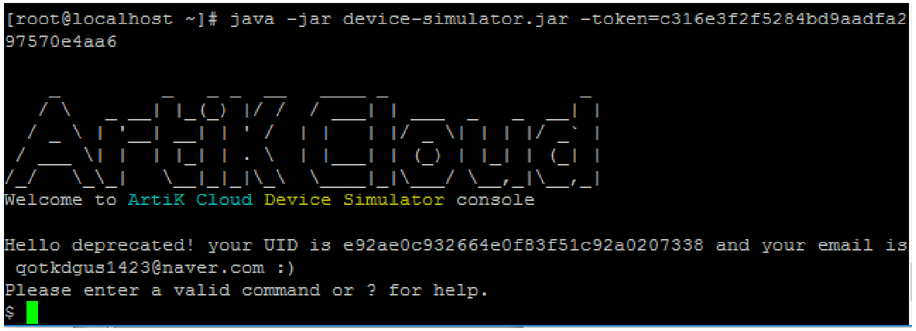
    # wget https://developer.artik.cloud/documentation/downloads/device-simulator.jar
  - You will see that file.



```
[root@localhost ~]# ll
total 12400
-rw-r--r-- 1 root root     7306 Nov  9  2015 anaconda-ks.cfg
drwxr-xr-x 2 root root     4096 Aug 26 00:10 autostart
-rwxr-xr-x 1 root root      129 Aug 26 00:10 compile_sketch_native
-rw-r--r-- 1 root root 12435176 Aug 24 18:20 device-simulator.jar
drwxr-xr-x 3 root root     4096 Aug 26 00:10 hardware
-rw-r--r-- 1 root root   224574 Dec  1  2015 libArduino-latest.tar.gz
-rwxr-xr-x 1 root root      329 Aug 26 00:10 load_sketch
drwxr-xr-x 5 root root     4096 Sep  1 04:13 node_modules
drwxr-xr-x 2 root root     4096 Sep  1 04:13 tmp
-rwxr-xr-x 1 root root      792 Aug 26 00:10 watcher
[root@localhost ~]#
```

  - Execute Device Simulator

    # java –jar device-simulator.jar –token=[your token]

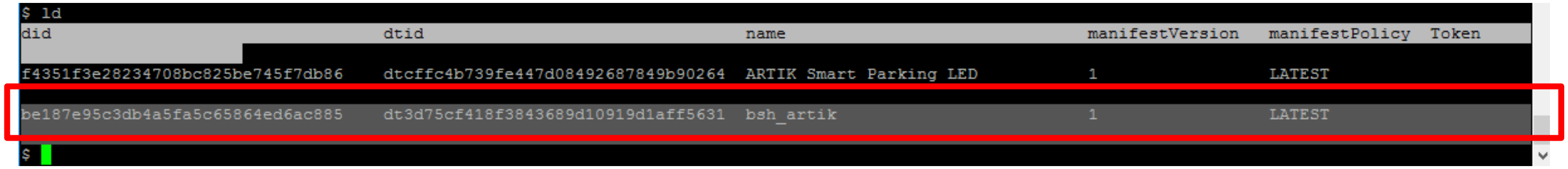


```
[root@localhost ~]# java -jar device-simulator.jar -token=c316e3f2f5284bd9aadfa2
97570e4aa6


    _         _   _ _  __   ____ _                 _
   / \   _ __| |_(_) |/ /  / ___| | ___  _   _  __| |
  / _ \ | '__| __| | ' /  | |   | |/ _ \| | | |/ _` |
 / ___ \| |  | |_| | . \  | |___| | (_) | |_| | (_| |
/_/   \_\_|   \__|_|_|\_\  \____|_|\___/ \__,_|\__,_|
Welcome to ArtiK Cloud Device Simulator console

Hello deprecated! your UID is e92ae0c932664e0f83f51c92a0207338 and your email is
 gotkdgus1423@naver.com :)
Please enter a valid command or ? for help.
$
```

ARTIK Cloud tutorial

Step 6 : Guess Scenario and Run scenario

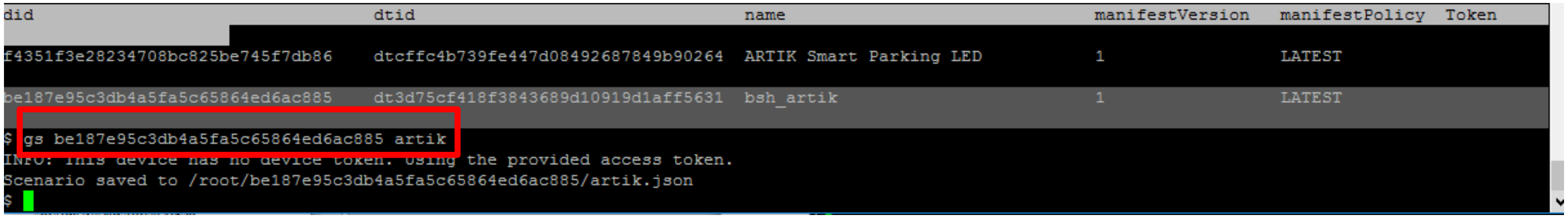
- Using command ‘ld’, you can find your device before you made. (ld : list devices)



```
$ ld
did                                 dtid                                 name                       manifestVersion   manifestPolicy   Token

f4351f3e28234708bc825be745f7db86    dtcffc4b739fe447d08492687849b90264   ARTIK Smart Parking LED    1                 LATEST

be187e95c3db4a5fa5c65864ed6ac885    dt3d75cf418f3843689d10919d1aff5631   bsh_artik                  1                 LATEST

$
```

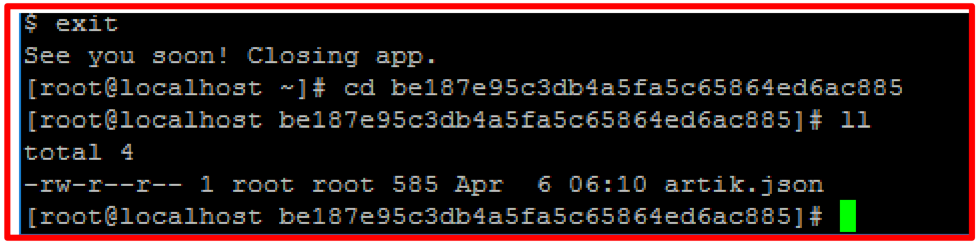
- Using command ‘gs’, you can get scenario sample. (gs : guess scenario)

  $ gs [device ID, did] [Scenario file name]



```
did                                 dtid                                 name                       manifestVersion   manifestPolicy   Token

f4351f3e28234708bc825be745f7db86    dtcffc4b739fe447d08492687849b90264   ARTIK Smart Parking LED    1                 LATEST

be187e95c3db4a5fa5c65864ed6ac885    dt3d75cf418f3843689d10919d1aff5631   bsh_artik                  1                 LATEST

$ gs be187e95c3db4a5fa5c65864ed6ac885 artik
INFO: This device has no device token. Using the provided access token.
Scenario saved to /root/be187e95c3db4a5fa5c65864ed6ac885/artik.json
$
```

- Exit and You can find scenario file in /[did]/artik.json

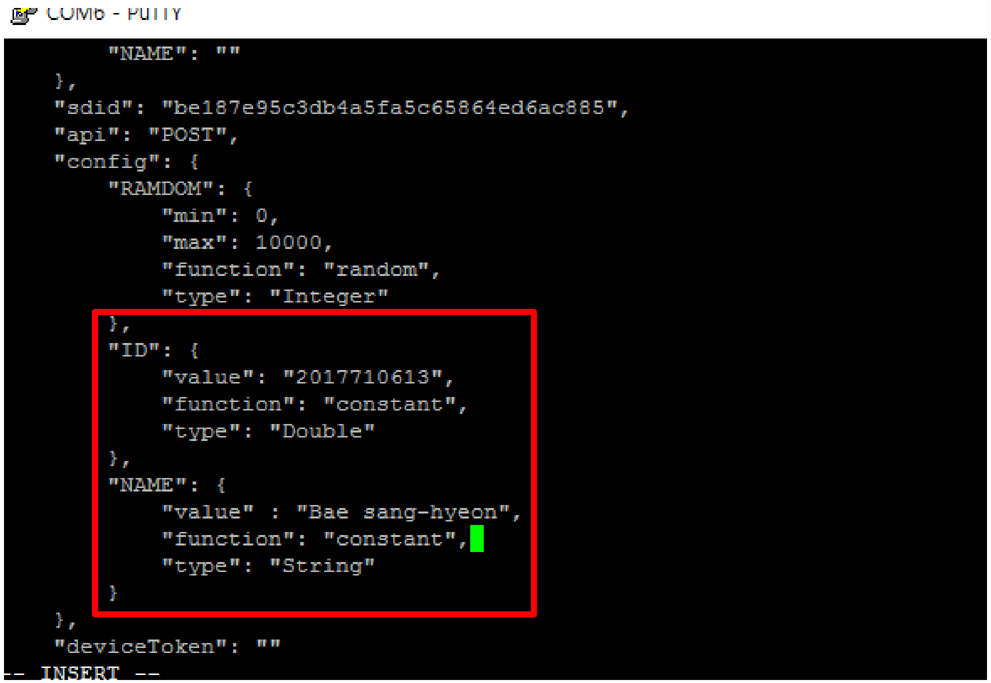


```
$ exit
See you soon! Closing app.
[root@localhost ~]# cd be187e95c3db4a5fa5c65864ed6ac885
[root@localhost be187e95c3db4a5fa5c65864ed6ac885]# ll
total 4
-rw-r--r-- 1 root root 585 Apr  6 06:10 artik.json
[root@localhost be187e95c3db4a5fa5c65864ed6ac885]#
```

ARTIK Cloud

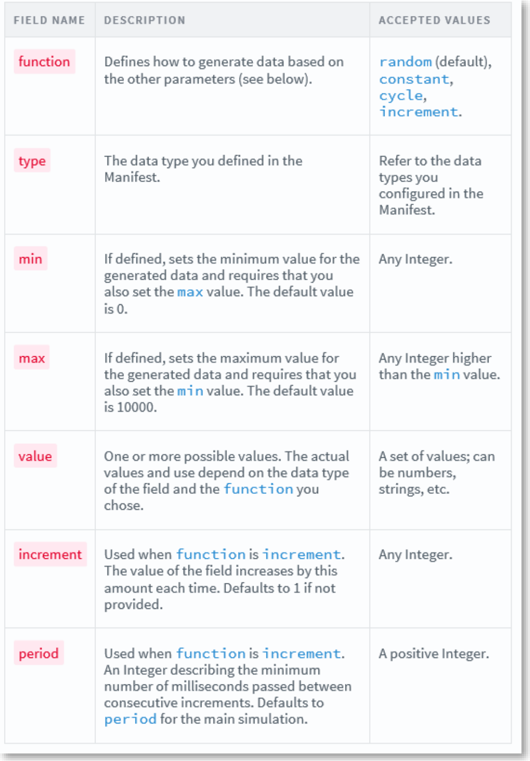
■ ARTIK Cloud tutorial

- Step 6 : Guess Scenario and Run scenario
  - Using vi editor, we will modify ID and NAME in artik.json
  - Modify your ID and your name.



COM6 - PuTTY

```
        "NAME": ""
    },
    "sdid": "be187e95c3db4a5fa5c65864ed6ac885",
    "api": "POST",
    "config": {
        "RAMDOM": {
            "min": 0,
            "max": 10000,
            "function": "random",
            "type": "Integer"
        },
        "ID": {
            "value": "2017710613",
            "function": "constant",
            "type": "Double"
        },
        "NAME": {
            "value" : "Bae sang-hyeon",
            "function": "constant",
            "type": "String"
        }
    },
    "deviceToken": ""
-- INSERT --
```



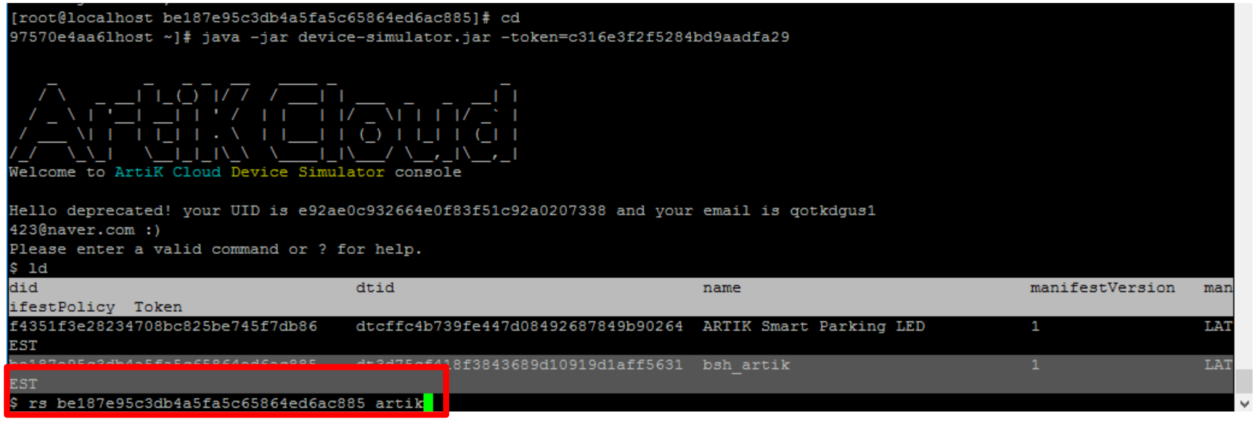
FIELD NAME	DESCRIPTION	ACCEPTED VALUES
function	Defines how to generate data based on the other parameters (see below).	random (default), constant, cycle, increment.
type	The data type you defined in the Manifest.	Refer to the data types you configured in the Manifest.
min	If defined, sets the minimum value for the generated data and requires that you also set the max value. The default value is 0.	Any Integer.
max	If defined, sets the maximum value for the generated data and requires that you also set the min value. The default value is 10000.	Any Integer higher than the min value.
value	One or more possible values. The actual values and use depend on the data type of the field and the function you chose.	A set of values; can be numbers, strings, etc.
increment	Used when function is increment. The value of the field increases by this amount each time. Defaults to 1 if not provided.	Any Integer.
period	Used when function is increment. An Integer describing the minimum number of milliseconds passed between consecutive increments. Defaults to period for the main simulation.	A positive Integer.

ARTIK Cloud

■ ARTIK Cloud tutorial

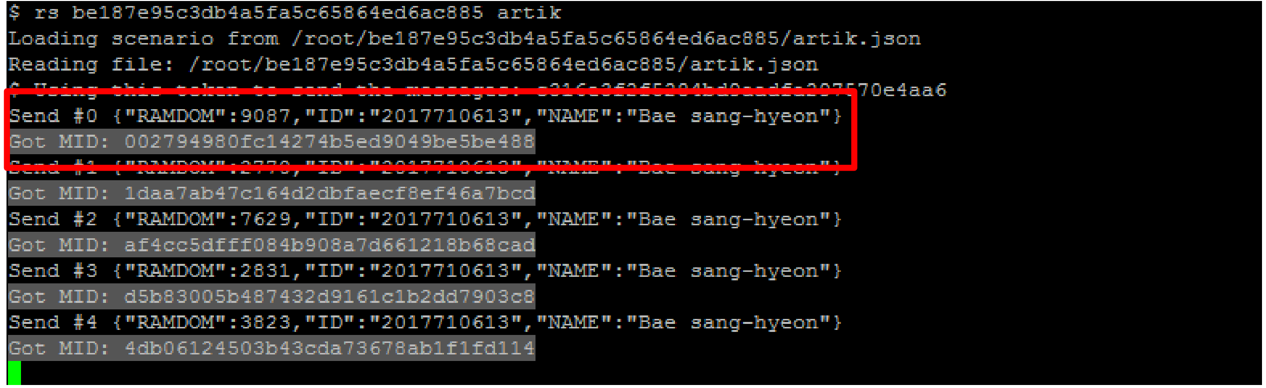
- Step 6 : Guess Scenario and Run scenario
  - Restart Device Simulator and run scenario.

    $ rs [did] [scenario file name]



```
[root@localhost be187e95c3db4a5fa5c65864ed6ac885]# cd
97570e4aa6lhost ~]# java -jar device-simulator.jar -token=c316e3f2f5284bd9aadfa29

    _         _   _ _  __   ____ _                 _
   / \   _ __| |_(_) |/ /  / ___| | ___  _   _  __| |
  / _ \ | '__| __| | ' /  | |   | |/ _ \| | | |/ _` |
 / ___ \| |  | |_| | . \  | |___| | (_) | |_| | (_| |
/_/   \_\_|   \__|_|_|\_\  \____|_|\___/ \__,_|\__,_|
Welcome to ArtiK Cloud Device Simulator console

Hello deprecated! your UID is e92ae0c932664e0f83f51c92a0207338 and your email is qotkdgus1
423@naver.com :)
Please enter a valid command or ? for help.
$ ld
did                                 dtid                                name                          manifestVersion   man
ifestPolicy  Token
f4351f3e28234708bc825be745f7db86    dtcffc4b739fe447d08492687849b90264  ARTIK Smart Parking LED       1                 LAT
EST
be187e95c3db4a5fa5c65864ed6ac885    dt3d75cf418f3843689d10919d1aff5631  bsh_artik                     1                 LAT
EST
$ rs be187e95c3db4a5fa5c65864ed6ac885 artik
```

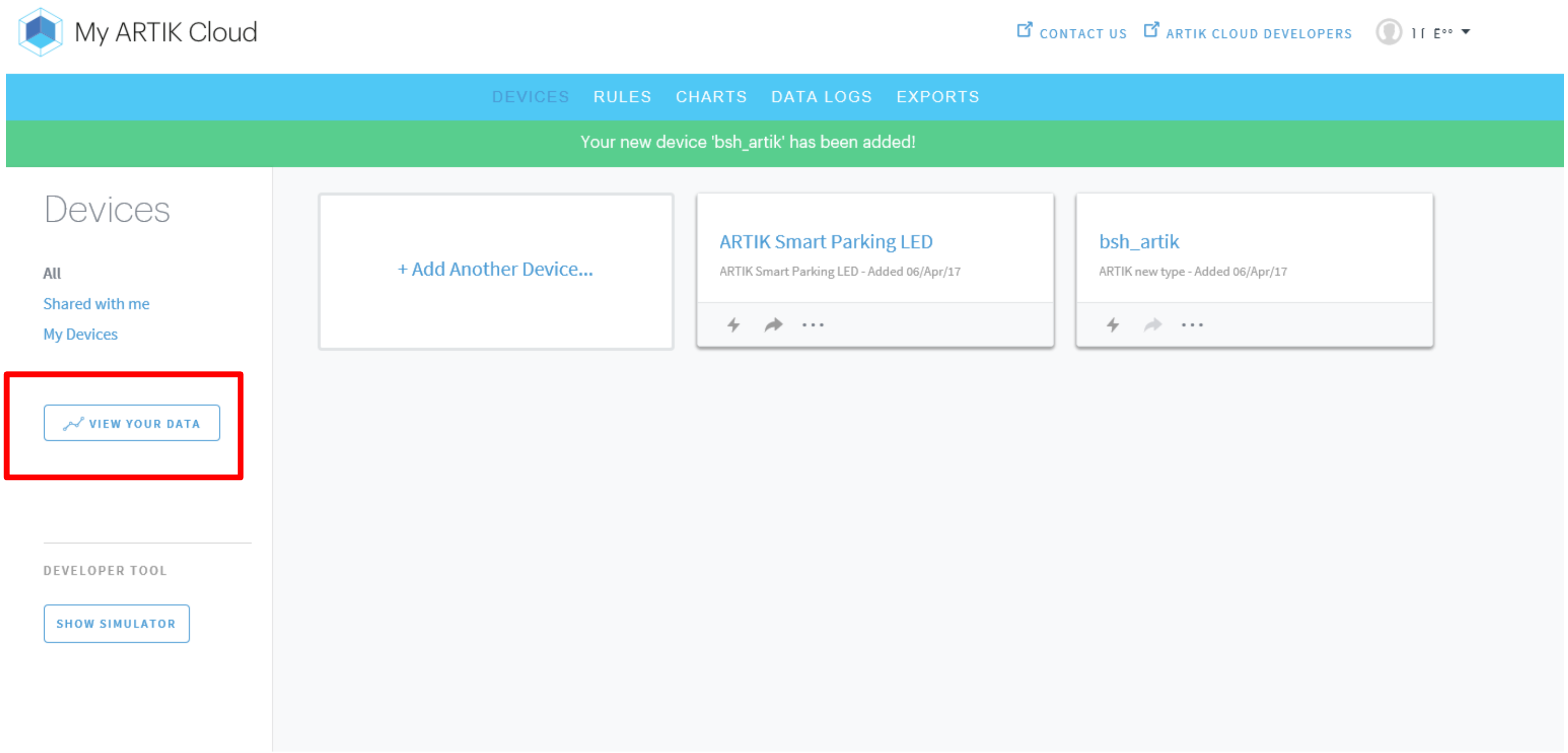
- You can see the sending message



```
$ rs be187e95c3db4a5fa5c65864ed6ac885 artik
Loading scenario from /root/be187e95c3db4a5fa5c65864ed6ac885/artik.json
Reading file: /root/be187e95c3db4a5fa5c65864ed6ac885/artik.json
$ Using this token to send the messages: c316e3f2f5284bd9aadfa2997570e4aa6
Send #0 {"RAMDOM":9087,"ID":"2017710613","NAME":"Bae sang-hyeon"}
Got MID: 002794980fc14274b5ed9049be5be488
Send #1 {"RAMDOM":2770,"ID":"2017710613","NAME":"Bae sang-hyeon"}
Got MID: 1daa7ab47c164d2dbfaecf8ef46a7bcd
Send #2 {"RAMDOM":7629,"ID":"2017710613","NAME":"Bae sang-hyeon"}
Got MID: af4cc5dfff084b908a7d661218b68cad
Send #3 {"RAMDOM":2831,"ID":"2017710613","NAME":"Bae sang-hyeon"}
Got MID: d5b83005b487432d9161c1b2dd7903c8
Send #4 {"RAMDOM":3823,"ID":"2017710613","NAME":"Bae sang-hyeon"}
Got MID: 4db06124503b43cda73678ab1f1fd114
```

ARTIK Cloud tutorial

- Step 7 : Check data in ARTIK Cloud
  - Click ‘VIEW YOUR DATA’



My ARTIK Cloud

Your new device 'bsh_artik' has been added!

Devices

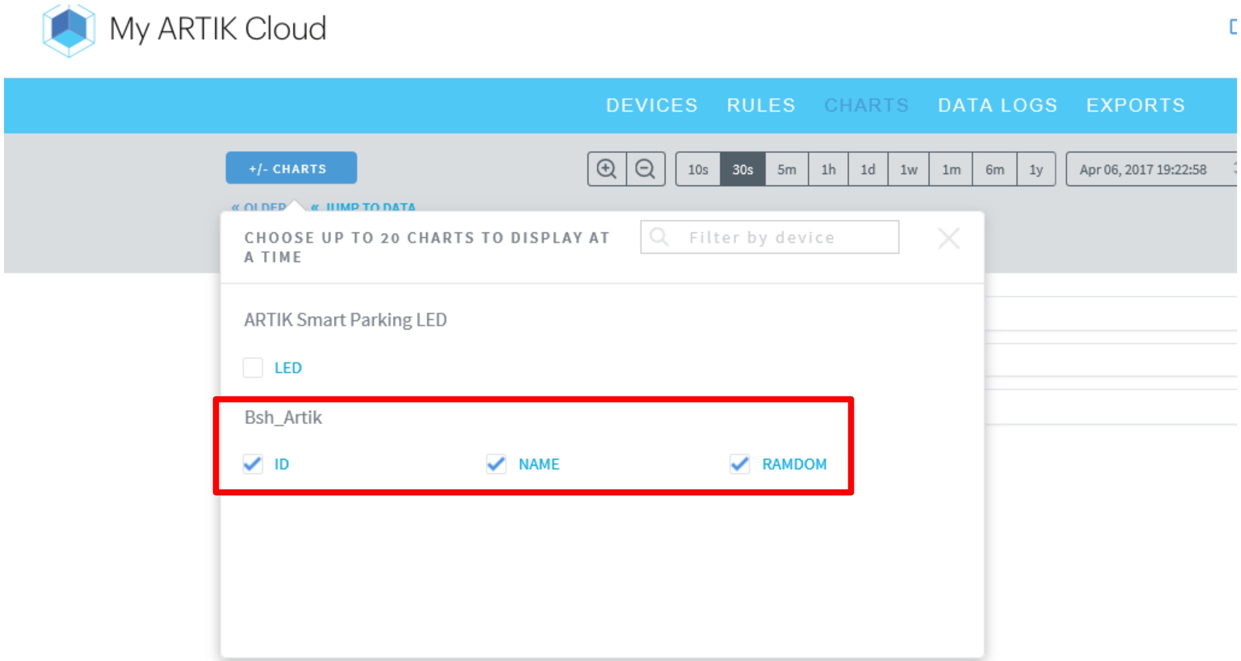
ARTIK Smart Parking LED
ARTIK Smart Parking LED - Added 06/Apr/17

bsh_artik
ARTIK new type - Added 06/Apr/17

ARTIK Cloud

ARTIK Cloud tutorial

- Step 7 : Check data in ARTIK Cloud
  - Check the data that you want



My ARTIK Cloud

CHOOSE UP TO 20 CHARTS TO DISPLAY AT A TIME

ARTIK Smart Parking LED

□ LED

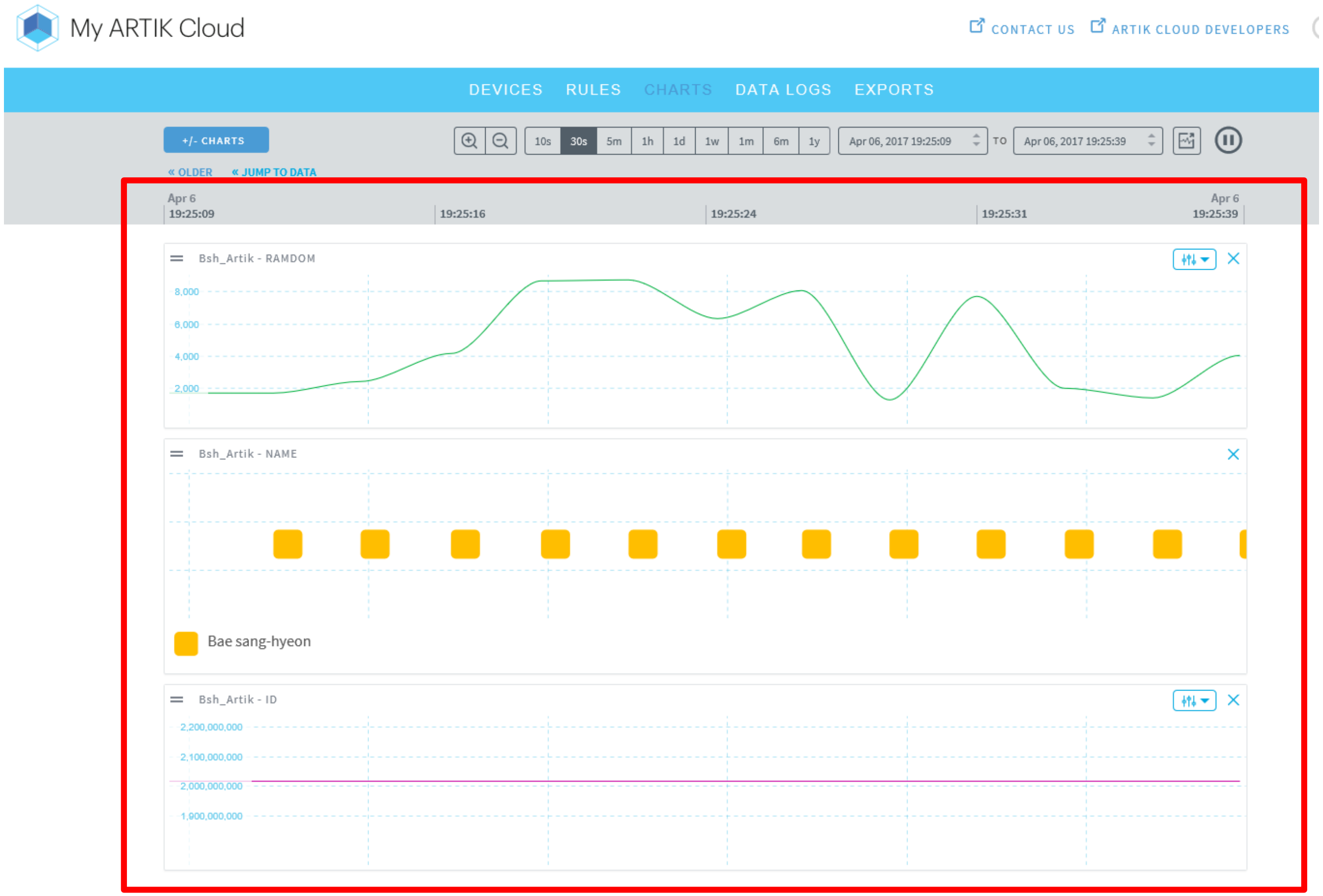
Bsh_Artik

☑ ID ☑ NAME ☑ RAMDOM

ARTIK Cloud

ARTIK Cloud tutorial

- Step 7 : Check data in ARTIK Cloud
  - You can see data like that figure.



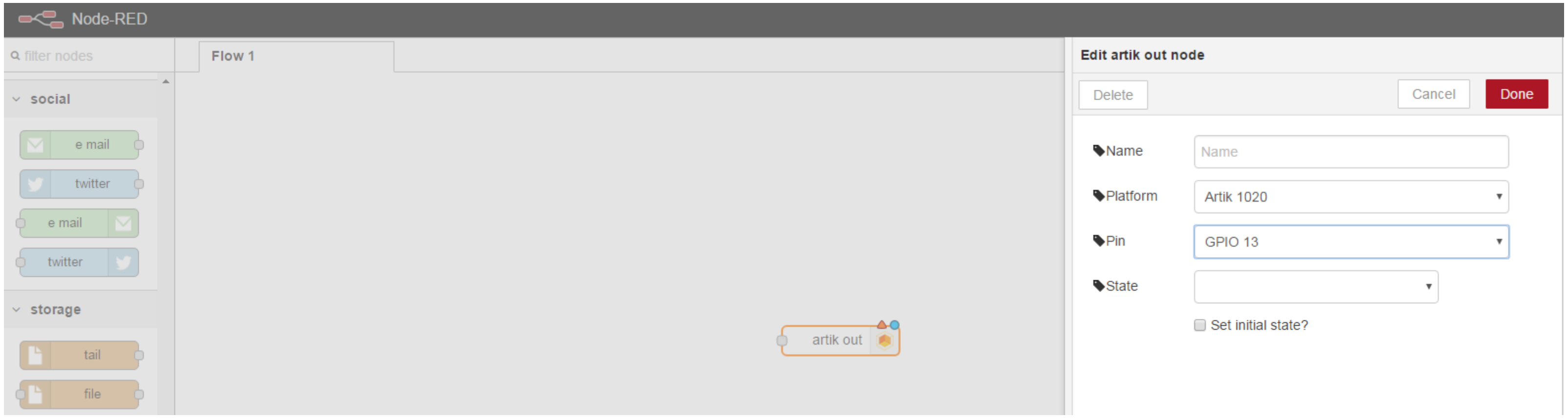
Ⅱ. ARTIK 기술 교육

9. ARTIK Cloud (2)

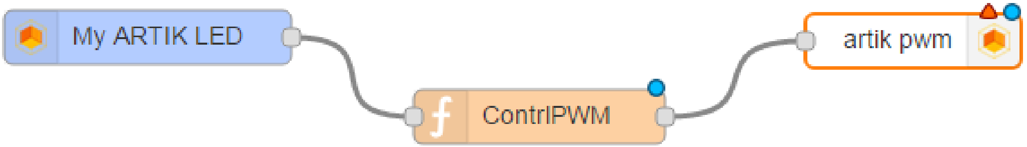
Node-RED

Introduction to Node-RED

- Node.js is a run-time interpreter of JavaScript packages.
- Node-RED is a form of "drag-and-drop" programming, so is emerging as a leading means of developing IoT applications.



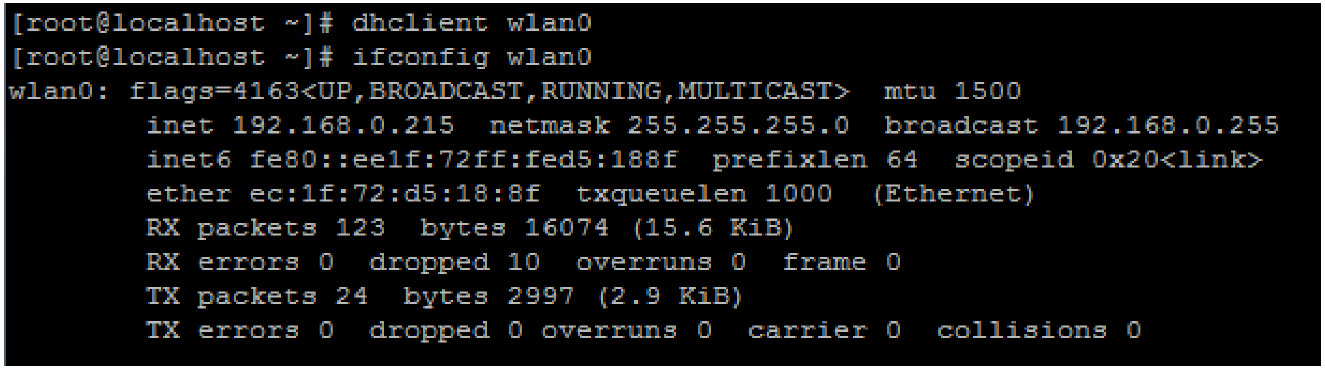
Name: Name
Platform: Artik 1020
Pin: GPIO 13
State:
□ Set initial state?



Node-RED

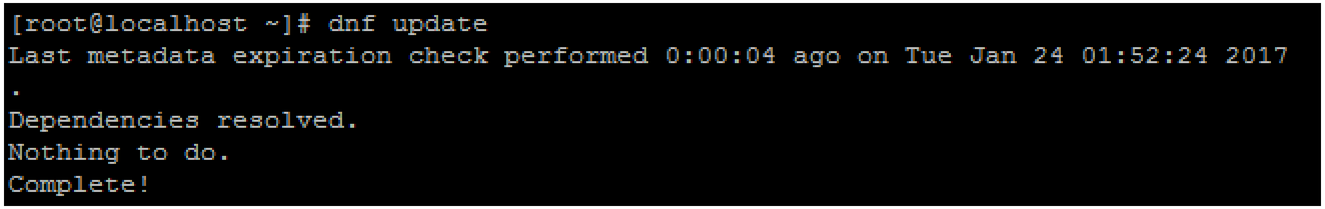
Install Node-RED

- Step 1 : Connect to Wi-Fi
  - Use the DHCP client services program dhclient.
  - Check whether DHCP client succeeded in a assigning an IP address, execute ifconfig.



```
[root@localhost ~]# dhclient wlan0
[root@localhost ~]# ifconfig wlan0
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.215  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::ee1f:72ff:fed5:188f  prefixlen 64  scopeid 0x20<link>
        ether ec:1f:72:d5:18:8f  txqueuelen 1000  (Ethernet)
        RX packets 123  bytes 16074 (15.6 KiB)
        RX errors 0  dropped 10  overruns 0  frame 0
        TX packets 24  bytes 2997 (2.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

- Step 2 : Update firmware image revision



```
[root@localhost ~]# dnf update
Last metadata expiration check performed 0:00:04 ago on Tue Jan 24 01:52:24 2017
.
Dependencies resolved.
Nothing to do.
Complete!
```

Node-RED

■ Install Node-RED

▪ Step 3 : Install NPM tool and Node-RED

- Install the Node Package Management (NPM) tool and the Node-RED visual wiring tool.
- Node is already pre-loaded on your system.
- Install two "contributed" packages, node-red-contrib-artik and node-red-contrib-artik-cloud.

```
# dnf update
# dnf install node
# dnf install npm
# npm –g install node-red
# npm –g install node-red-contrib-artik
# npm –g install node-red-contrib-artik-cloud
```

▪ Step 4 : Starting a Node-RED

- Node-RED is a server running on ARTIK, and a Web browser on host PC will be its client.
- PC must be connected to the same Wi-Fi network as the ARTIK Wi-Fi.

```
# node-red &
```

- Start a browser on PC, point the browser to the localhost Wi-Fi IP address of ARTIK device, on port 1880. Ex) http://10.0.0.5:1880

Node-RED

Update Node.js

- Step 1 : Update Node.js with Wi-Fi
  - Node.js v0.10.42 is an unsupported version, so you should upgrade to the latest release.
  - Before updating, connect to Wi-Fi execute dhclient wlan0.
  - Check the current version of Node.js
  - Install n module using NPM tool
  - Upgrade Node.js to v4.3.2

```
# dhclient wlan0
# node -v
# npm cache clean –f
# npm install –g n
# n 4.3.2
```

- Step 2 : After Updating, check the current version of Node.js
  - Check the current version of Node.js
  - If the current version is not changed, configure file link as following.

```
# ln –sf /usr/local/n/versions/node/4.3.2/bin/node /usr/bin/node
```

Node-RED

■ Update NPM tool

- Step 1 : Update NPM tool
  - Check the current version of npm tool, and install npm using npm tool.
  - After upgrading, check again the current version.

```
# npm –v
# npm install –g npm
# npm –v
```

  - If the version is not changed, execute the following command.

```
# hash –r
```

- Step 2 : Re-install Node-RED
  - After upgrading Node.js, it is better to stop Node-RED, and then re-install as follows.

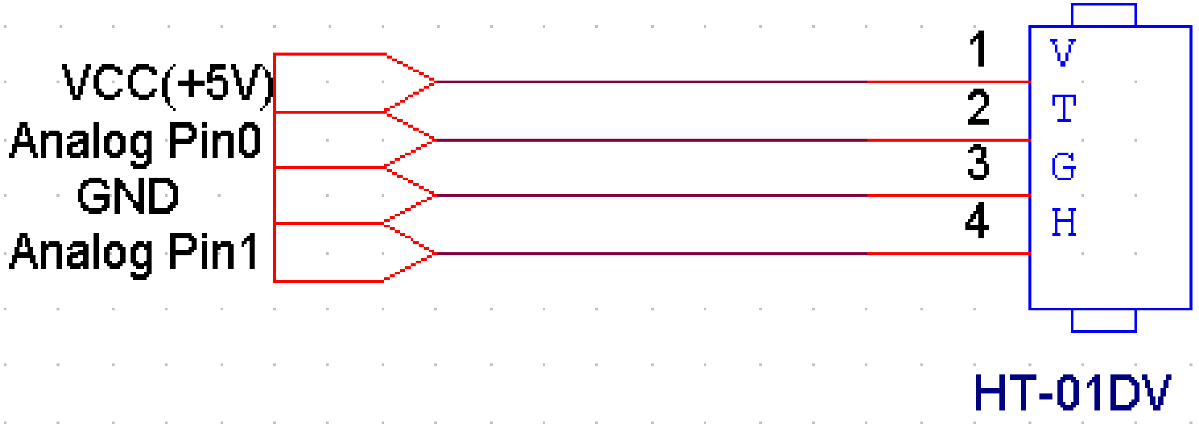
```
# npm cache clean
# npm install –g --unsafe-perm node-red
# cd ~/.node-red
# npm rebuild
```

Node-RED Ex. Read H/T Sensor

Required Hardware

- ARTIK 5 developer kit
- Humidity and Temperature Sensor (ETH-01DV)
- Breadboard
- Connector wires

Circuit Configuration

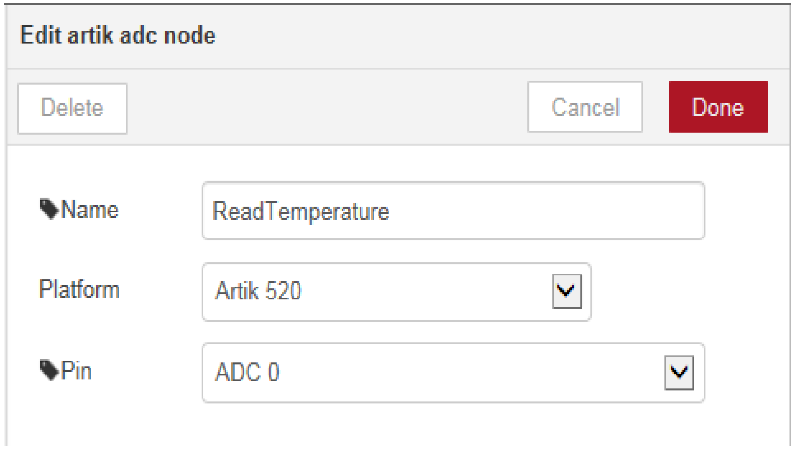


Node-RED Ex. Read H/T Sensor

Developing Node-RED application

- Step 1 : Select an "Inject" node
  - Click and drag an "Inject" input node from the palette in left pane to the convas.
  - Then, "Inject" will change to "timestamp".
  - Configure the inject node to trgger data reading every 5s.

- Step 2 : Select an "Artik ADC" node and double-click to set :
  - "Artik 520" as target platform
  - "ADC 0" as the analog pin form which data will be read.
  - Name as "ReadTemperature"



Edit artik adc node

Name: ReadTemperature

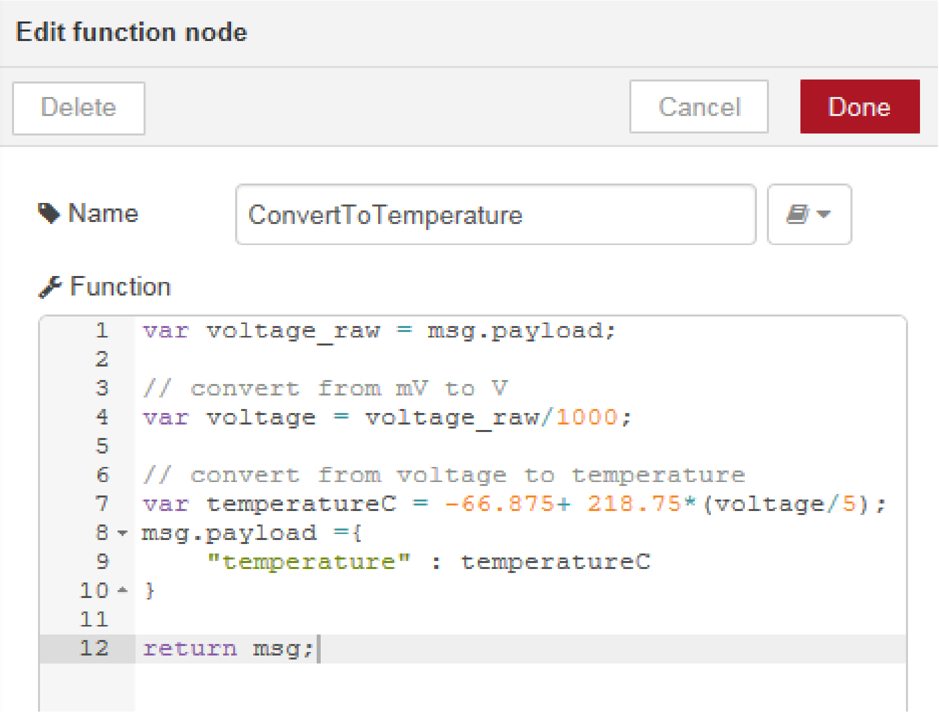
Platform: Artik 520

Pin: ADC 0

Node-RED Ex. Read H/T Sensor

Developing Node-RED application

- Step 3 : Select "function" node
  - Drag a function node to the right of the aritk adc node.
  - It will convert the voltage(mV) reading from ADC 0 pin to a temperature.
  - Double-click to define the function as following.



Name: ConvertToTemperature

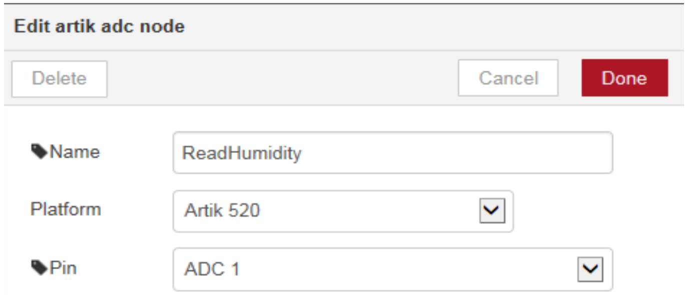
Function

```
var voltage_raw = msg.payload;

// convert from mV to V
var voltage = voltage_raw/1000;

// convert from voltage to temperature
var temperatureC = -66.875+ 218.75*(voltage/5);
msg.payload ={
    "temperature" : temperatureC
}

return msg;
```

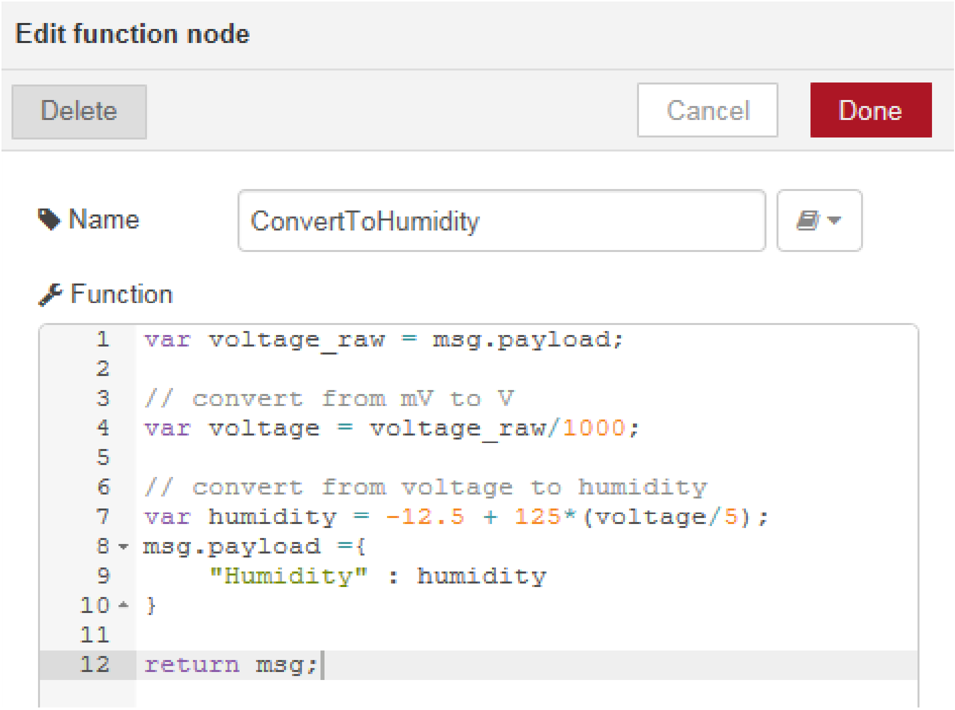
Node-RED Ex. Read H/T Sensor

Developing Node-RED application

- Step 4 : Select "Artik ADC" node and "Function" node
  - Set ADC 1 as a pin of artik adc node.
  - Define the function as following, where will convert voltage(mV) to humidity.



Name: ReadHumidity
Platform: Artik 520
Pin: ADC 1



Name: ConvertToHumidity

Function

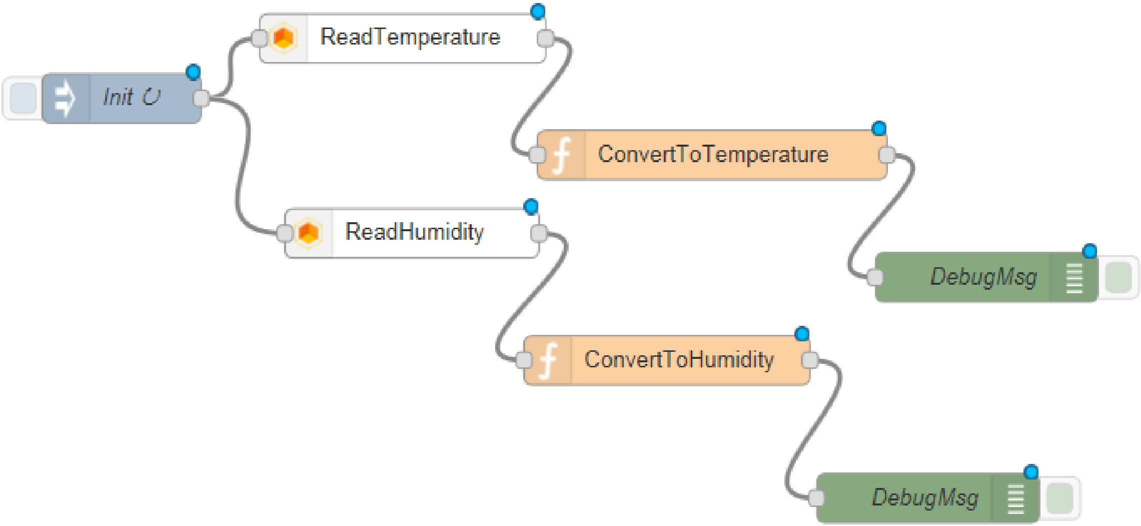
```
var voltage_raw = msg.payload;

// convert from mV to V
var voltage = voltage_raw/1000;

// convert from voltage to humidity
var humidity = -12.5 + 125*(voltage/5);
msg.payload ={
    "Humidity" : humidity
}

return msg;
```

Node-RED Ex. Read H/T Sensor

Developing Node-RED application

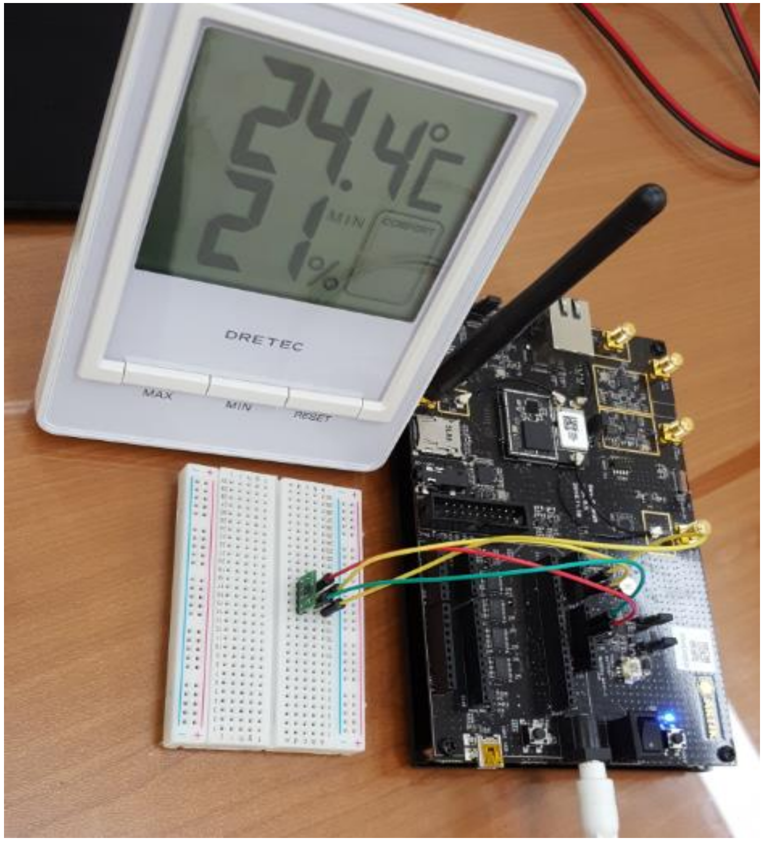
- Step 5 : Select "Debug" output node
  - The debug node shows the msg.payload info on the Node-RED debug tab.

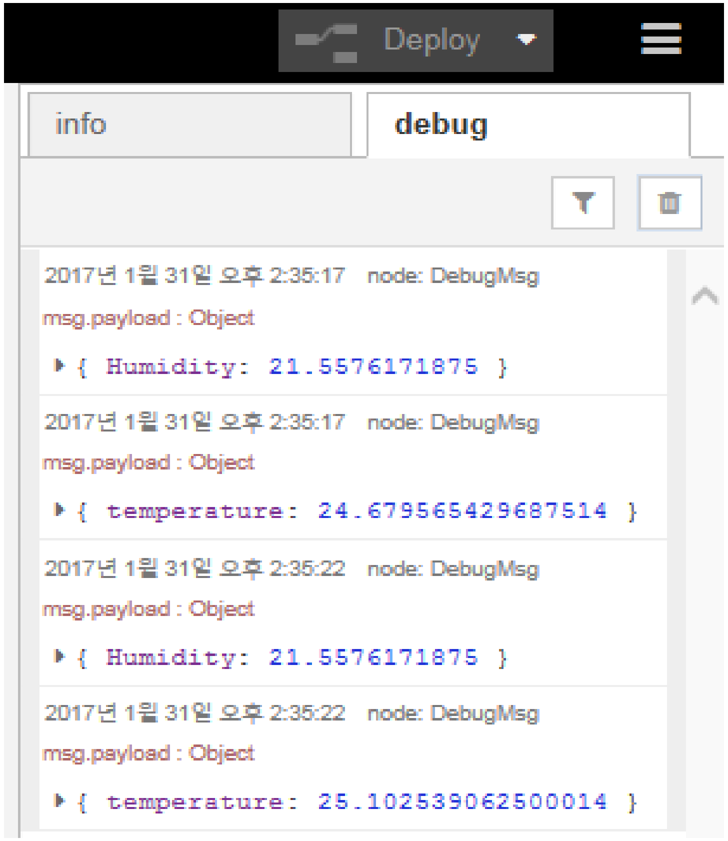
- Step 6 : Wire them altogether and click "Deploy".



Node-RED Ex. Read H/T Sensor

■ Execution Result





```
2017년 1월 31일 오후 2:35:17   node: DebugMsg
msg.payload : Object
▸ { Humidity: 21.5576171875 }
2017년 1월 31일 오후 2:35:17   node: DebugMsg
msg.payload : Object
▸ { temperature: 24.679565429687514 }
2017년 1월 31일 오후 2:35:22   node: DebugMsg
msg.payload : Object
▸ { Humidity: 21.5576171875 }
2017년 1월 31일 오후 2:35:22   node: DebugMsg
msg.payload : Object
▸ { temperature: 25.102539062500014 }
```

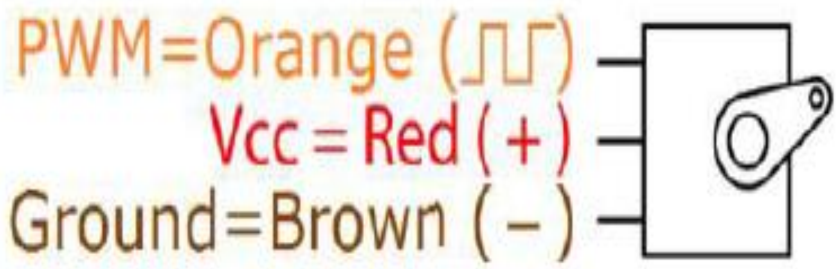
Node-RED Ex. Control Servo Motor

Required Hardware

- ARTIK 5 developer kit
- Servo motor (SG90)
- Connector wires

Circuit Configuration

- PWM – J26[6] (PWM 0)
- VCC – 5V
- Ground -GND

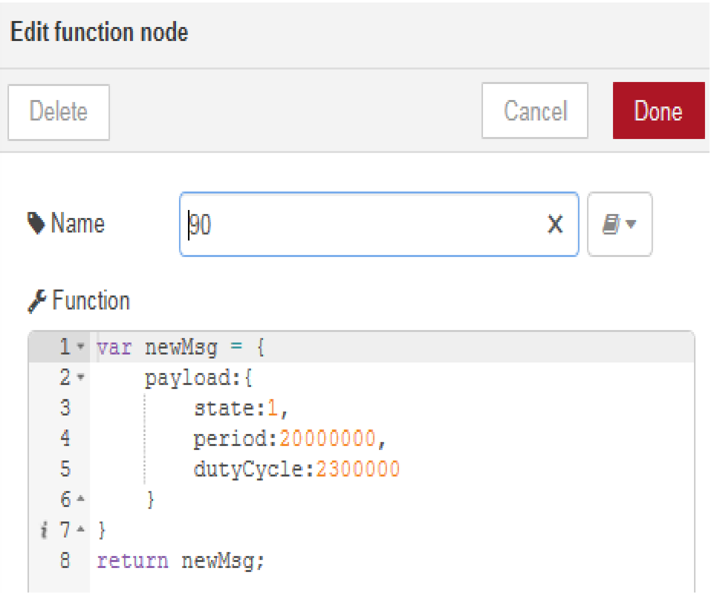


Node-RED Ex. Control Servo Motor

Developing Node-RED application

- Step 1 : Select "Artik pwm" node and Set to :
    - "Artik 520" as a target platform and "PWM0" as a pin.

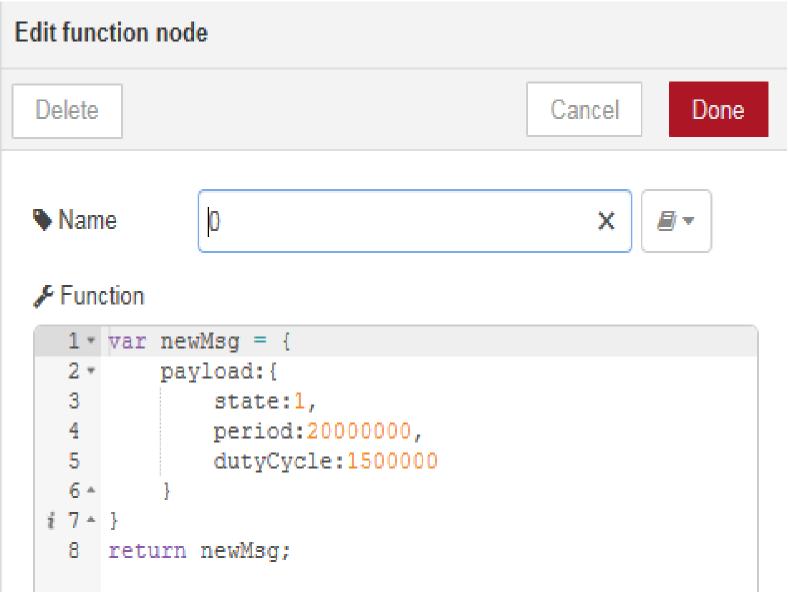
- Step 2 : Select four "function" nodes
    - Which control duty cycle, period and enable of pwm pin.



Name: 90

Function

```
var newMsg = {
    payload:{
        state:1,
        period:20000000,
        dutyCycle:2300000
    }
}
return newMsg;
```



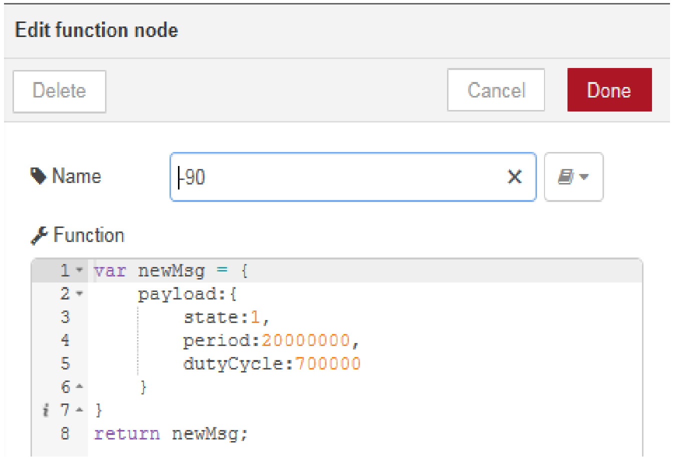
Name: 0

Function

```
var newMsg = {
    payload:{
        state:1,
        period:20000000,
        dutyCycle:1500000
    }
}
return newMsg;
```

Node-RED Ex. Control Servo Motor

■ Developing Node-RED application

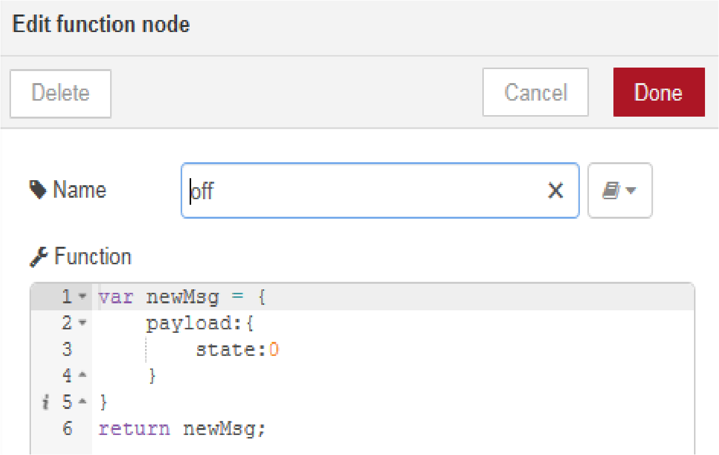


Edit function node

Name: -90

Function

```
var newMsg = {
    payload:{
        state:1,
        period:20000000,
        dutyCycle:700000
    }
}
return newMsg;
```



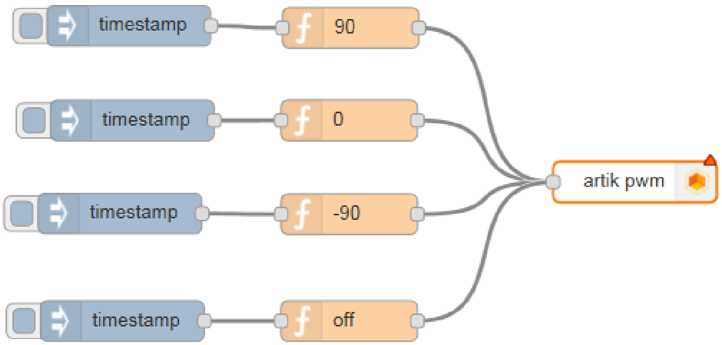
Edit function node

Name: off

Function

```
var newMsg = {
    payload:{
        state:0
    }
}
return newMsg;
```

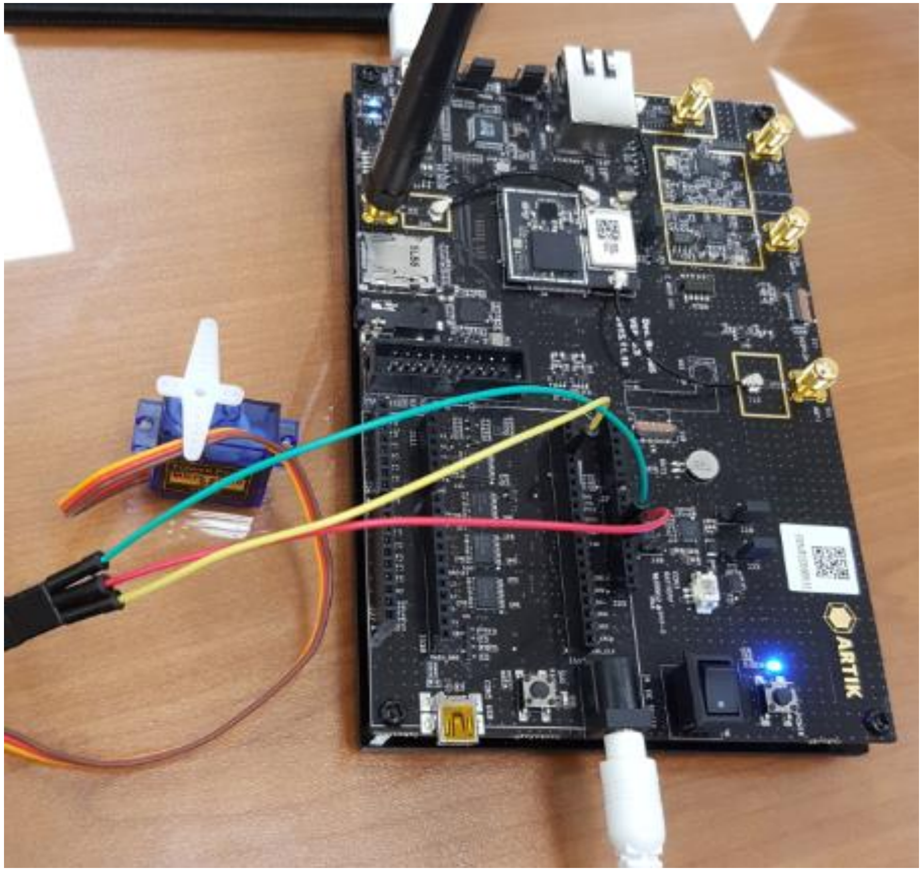
- Step 3 : Wire up four inject nodes to the beginning of the flow and click "Deploy".



Node-RED Ex. Control Servo Motor

Execution Result

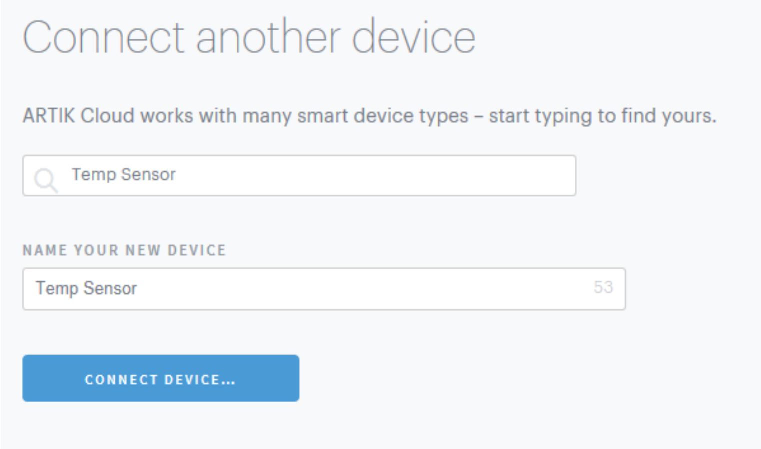
- Watching the servo motor, trigger "timestamp" node one by one.
- As you can see, Node-Red can control the ARTIK PWM pin.



Node-RED and ARTIK Cloud

Sending Node-RED data to ARTIK Cloud

- Step 1 : Log into the My ARTIK Cloud site and connect a new device.
  - Go to Devices and clieck the "Add Another Device" link.
  - Search for "Temp Sensor" and select it as a device type.
- Step 2 : Get the device token.
  - Click on the device of interest, then you can see the Device info pop-up.
  - Click "Generate Device Token" to get a device token.



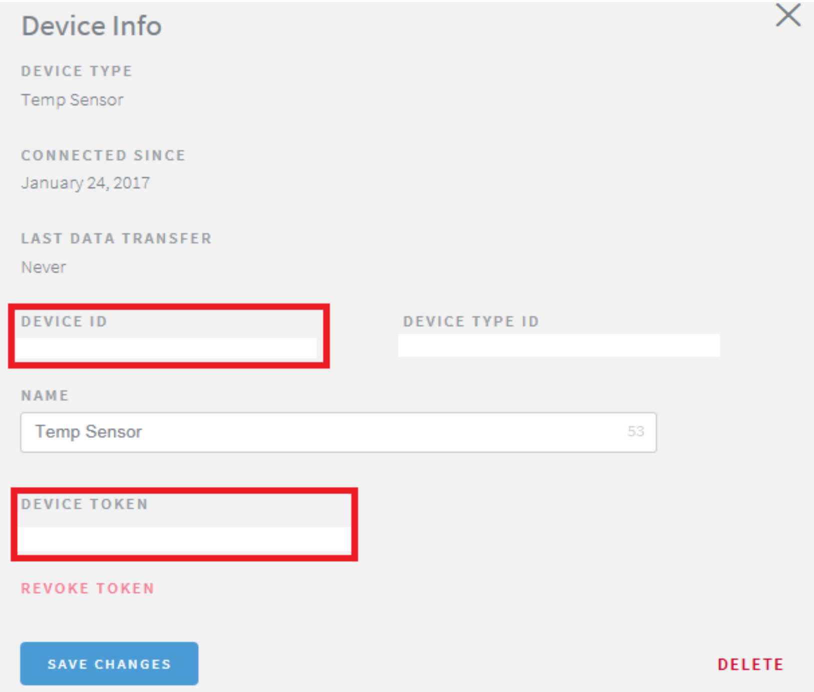
Connect another device

ARTIK Cloud works with many smart device types – start typing to find yours.

Temp Sensor

NAME YOUR NEW DEVICE

Temp Sensor 53



Device Info

DEVICE TYPE
Temp Sensor

CONNECTED SINCE
January 24, 2017

LAST DATA TRANSFER
Never

DEVICE ID

DEVICE TYPE ID

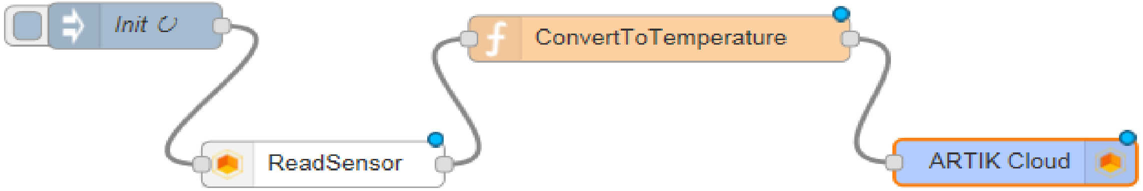
NAME
Temp Sensor 53

DEVICE TOKEN

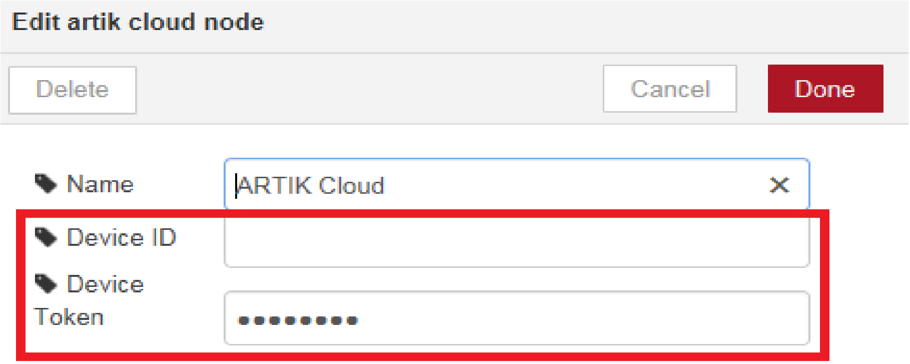
Node-RED and ARTIK Cloud

■ Sending Node-RED data to ARTIK Cloud

- Step 3 : Set up an output node to ARTIK Cloud
  - In "Node-RED ex. Read H/T Sensor" part, delete "ReadHumidity" node and "ConvertToHumidity" node and add an "ARTIK Cloud" output node.



  - Configure "ARTIK Cloud" output node as follows. Enter from the Device Info page.



Edit artik cloud node

Name: ARTIK Cloud

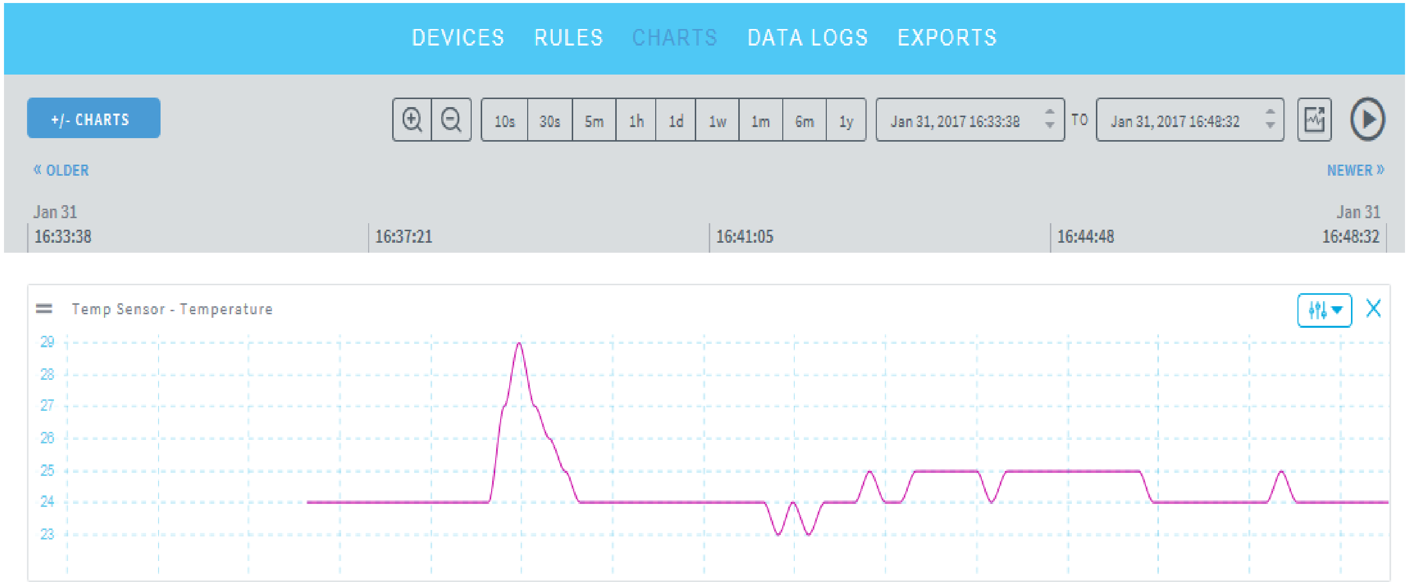
Device ID:

Device Token: ●●●●●●●●●

Node-RED and ARTIK Cloud

Sending Node-RED data to ARTIK Cloud

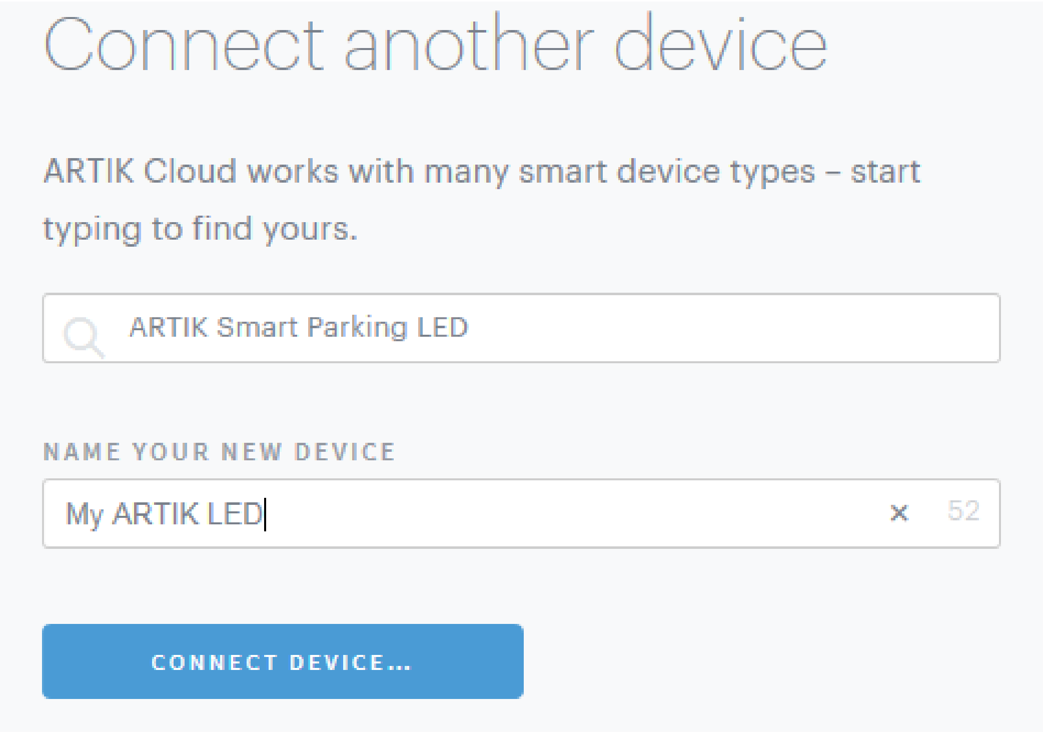
- Step 4 : Click "Deploy"
  - Go back to My ARTIK Cloud and CHARTS, then you can see the streamed data.



Node-RED and ARTIK Cloud

Receiving actions from ARTIK cloud

- Step 1 : Connect another device
  - Select "ARTIK Smart Parking LED" as a device type.
  - Name as "My ARTIK LED" and get a device token from Device Info page.



Connect another device

ARTIK Cloud works with many smart device types – start typing to find yours.

ARTIK Smart Parking LED

NAME YOUR NEW DEVICE

My ARTIK LED

Node-RED and ARTIK Cloud

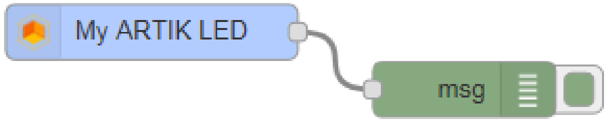
■ Receiving actions from ARTIK cloud

- Step 2 : Write Rules in My ARTIK Cloud
  - By writing Rules, ARTIK Cloud can generate Actions based on Rules.
  - Using the temperature sensor circuit that we built before, let’s write Rules.
  - Go to My ARTIK Cloud and Rules, and write Rules as follows.

✓ **IF Temp Sensor temperature is more than 26 THEN Send to MY ARTIK LED the action setoff**

✓ **IF Temp Sensor temperature is less than or equal to 26 THEN Send to My ARTIK LED the action setOn**

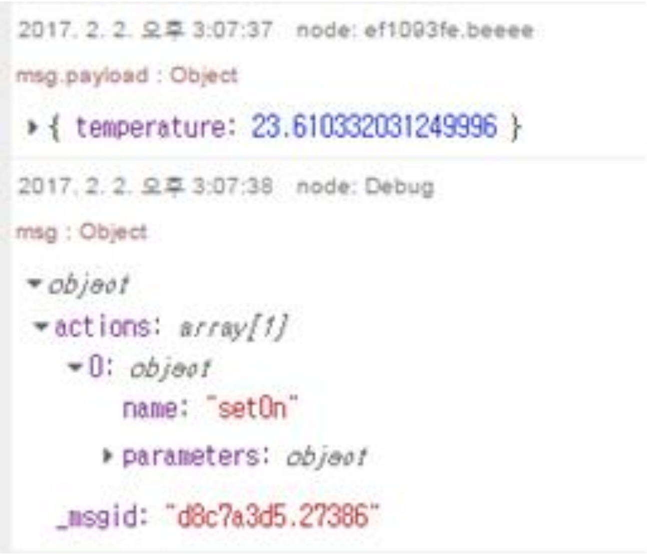
- Step 3 : Go back to Node-RED and add two node.
  - Select "ARTIK Cloud" input node and edit device id and device token.
  - Select "Debug" output node and set its output to show the "complete msg object".
  - Wire them and click "Deploy".



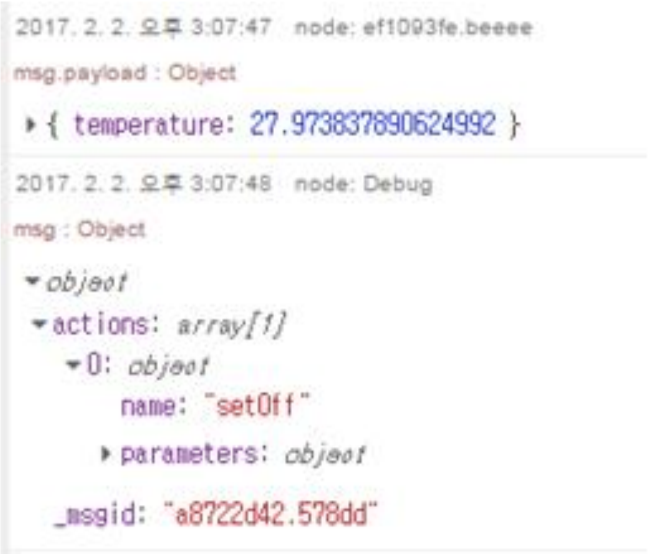
Node-RED and ARTIK Cloud

Receiving actions from ARTIK cloud

- Step 4 : Check debug tap in Node-RED
  - If temperature is less than 26℃, ARTIK Cloud sends actions "setOn".
  - If temperature is more than or equal to 26℃, ARTIK Cloud sends actions "setoff"



```
2017. 2. 2. 오후 3:07:37   node: ef1093fe.beeee
msg.payload : Object
▸{ temperature: 23.610332031249996 }
2017. 2. 2. 오후 3:07:38   node: Debug
msg : Object
▾object
 ▾actions: array[1]
   ▾0: object
      name: "setOn"
     ▸parameters: object
  _msgid: "d8c7a3d5.27386"
```

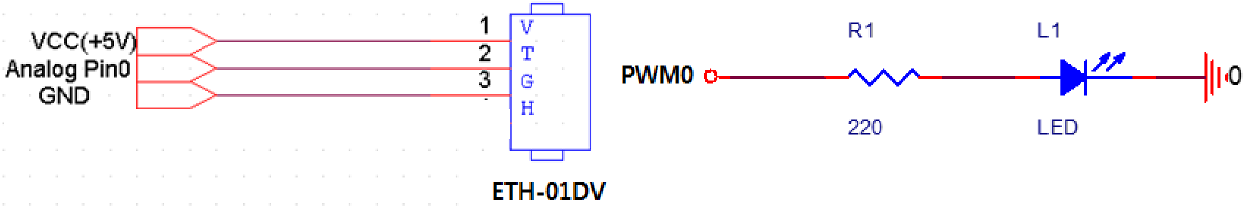


```
2017. 2. 2. 오후 3:07:47   node: ef1093fe.beeee
msg.payload : Object
▸{ temperature: 27.973837890624992 }
2017. 2. 2. 오후 3:07:48   node: Debug
msg : Object
▾object
 ▾actions: array[1]
   ▾0: object
      name: "setOff"
     ▸parameters: object
  _msgid: "a8722d42.578dd"
```

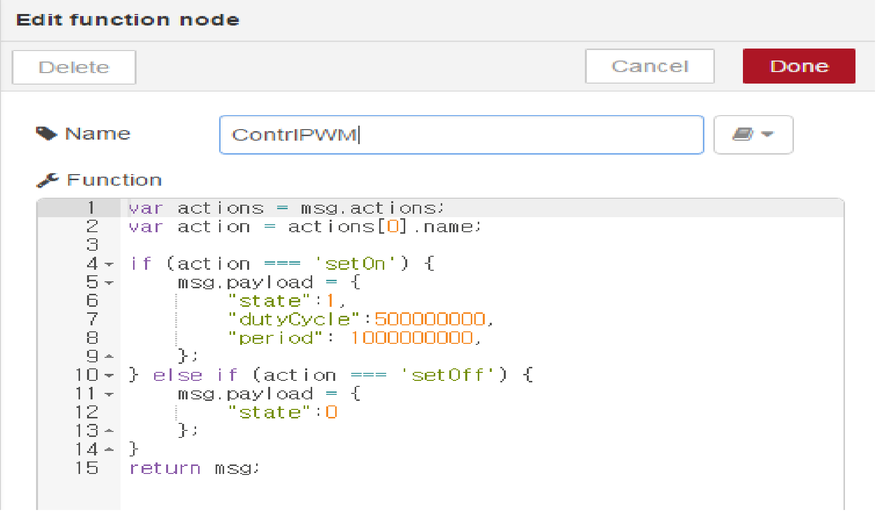
Node-RED and ARTIK Cloud

Using Actions, control LED on/off

- Step 1 : Configure circuit as follows.



- Step 2 : In Node-RED, select "function" node.
  - It will check whether we received a "setOn" or "setoff" action.



Name: ContrlPWM

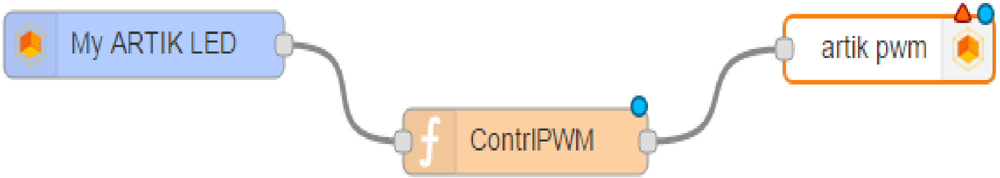
Function

```
var actions = msg.actions;
var action = actions[0].name;

if (action === 'setOn') {
    msg.payload = {
        "state":1,
        "dutyCycle":500000000,
        "period": 1000000000,
    };
} else if (action === 'setOff') {
    msg.payload = {
        "state":0
    };
}
return msg;
```

Node-RED and ARTIK Cloud

■ Using Actions, control LED on/off

- Step 3 : Wire up "Artik pwm" node to the right side of function node.
  - Set "ARTIK 520" as a target platform and "PWM 0" as a pin.



- Step 4 : Click "Deploy"
  - If temperature is less than 26℃, LED blink.
  - If temperature is more than or equal to 26℃, you can see LED turned off.

ARTIK Cloud and REST API Protocol

■ REST API Protocol

- ARTIK Cloud can send and receive messages, using REST API protocol.
- You can also put Actions in messages via REST so that destination devices can perform the specified Actions.
- **End point** : https://api.artik.cloud/v1.1
- **API call** : POST /messages

Request Body Parameters	
sdid	(Send messages) Source device ID
ddid	(Send actions) Destination device ID
type	"message" or "action"
ts	(Optional) Message timestamp. Current time if omitted.
token	(Optional) Device token
data	Data. Can be a simple text field, or a JSON document.

ARTIK Cloud and REST API Protocol

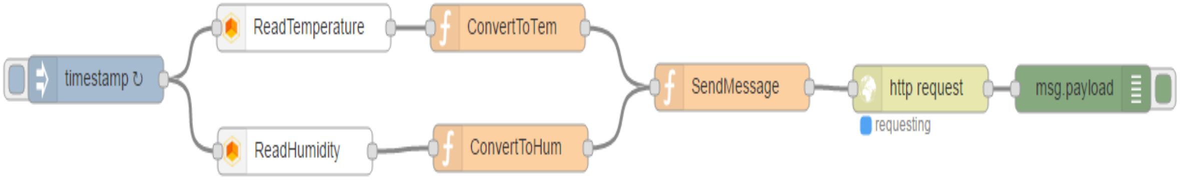
REST API Protocol

Request Header Parameters	
Content-Type	"application/json"
Authrization	"Beare _your_device_token_"

Response Body Parameters	
mid	Message ID.

Using Node-RED, send data to ARTIK Cloud via REST protocol

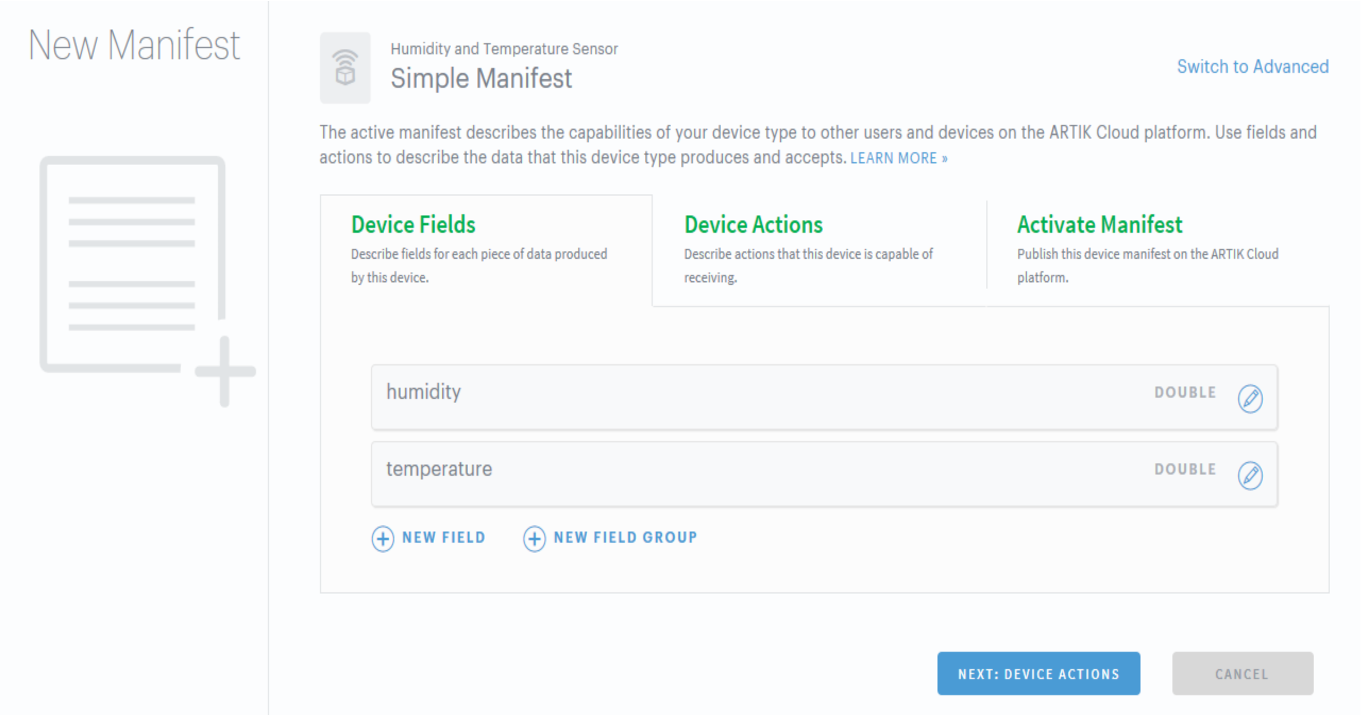
- Step 1 : Wire up nodes.
  - In "Node-RED Ex. Read H/T Sensor", delete two debug nodes and add a "function" node, "http request" node and another "debug" output node as follows.



ARTIK Cloud and REST API Protocol

■ Using Node-RED, send data to ARTIK Cloud via REST protocol

- Step 2 : Create a new Device type and set Manifest.
  - Go to ARTIK Cloud Developer site and create a new device type named as "Humidity and Temperature Sensor".
  - Set Manifest to add two Device Fields as follows.



New Manifest

Humidity and Temperature Sensor

Simple Manifest

Switch to Advanced

The active manifest describes the capabilities of your device type to other users and devices on the ARTIK Cloud platform. Use fields and actions to describe the data that this device type produces and accepts. LEARN MORE »

Device Fields Describe fields for each piece of data produced by this device.

Device Actions Describe actions that this device is capable of receiving.

Activate Manifest Publish this device manifest on the ARTIK Cloud platform.

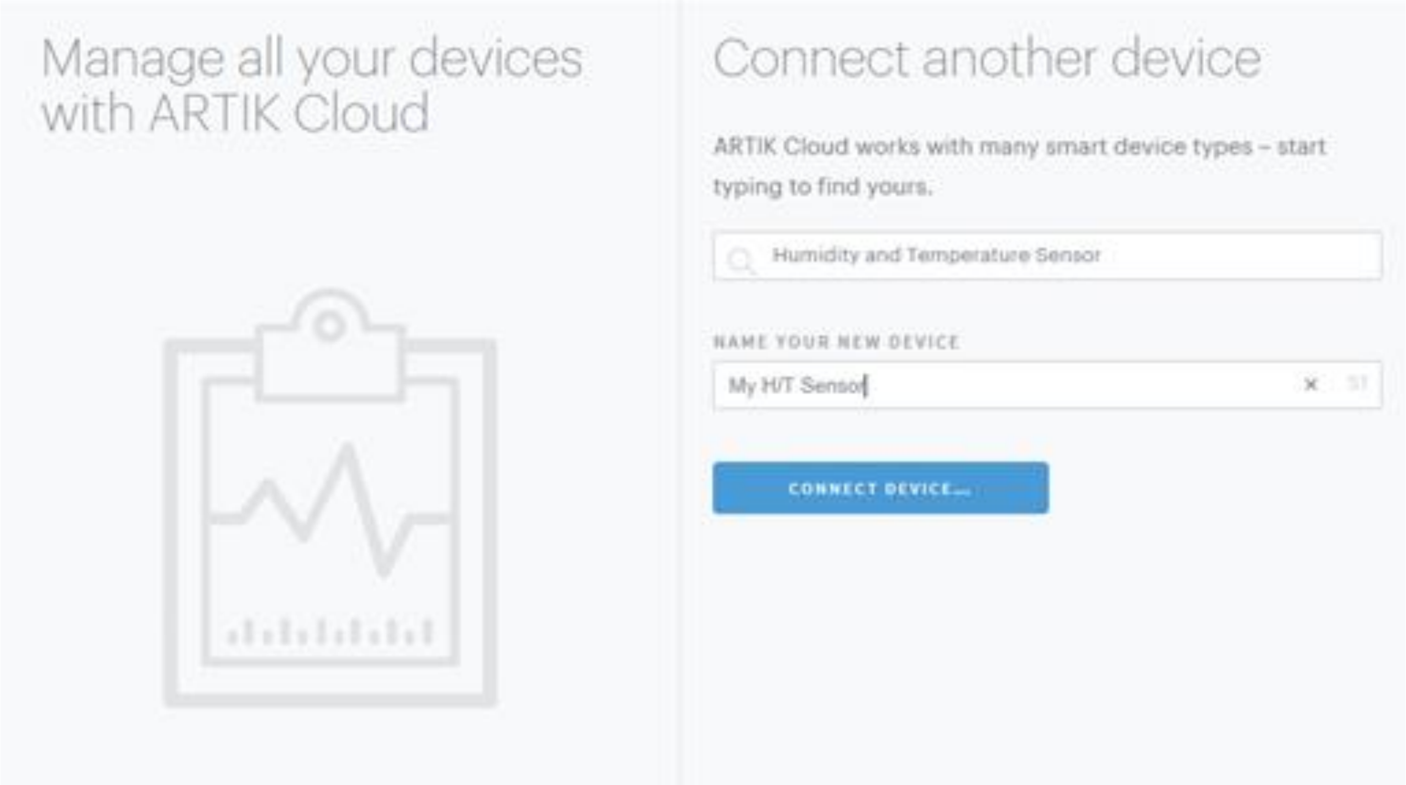
humidity DOUBLE

temperature DOUBLE

ARTIK Cloud and REST API Protocol

■ Using Node-RED, send data to ARTIK Cloud via REST protocol

- Step 3 : Connect a new device
  - Go to My ARTIK Cloud site, connect a new device. Configure "Humidity and Temperature Sensor" as a device type and name as "My H/T Senosr".
  - After connecting the device, get token from Device Info page.



Manage all your devices with ARTIK Cloud

Connect another device

ARTIK Cloud works with many smart device types – start typing to find yours.

Humidity and Temperature Sensor

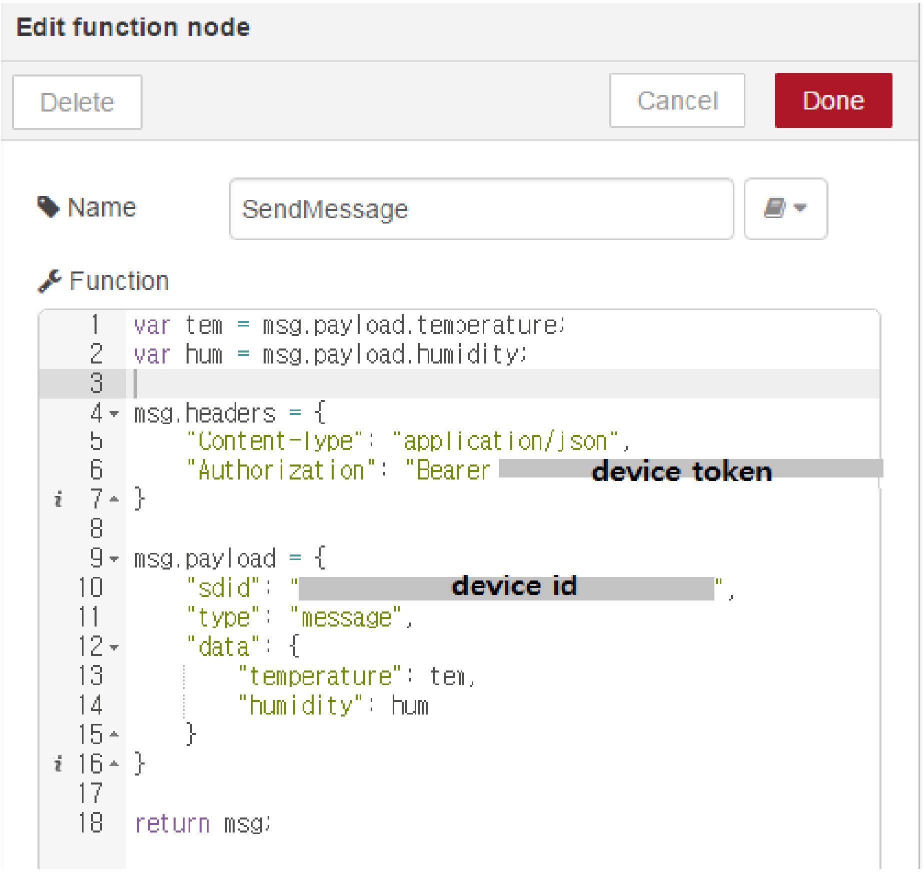
NAME YOUR NEW DEVICE

My H/T Sensor

ARTIK Cloud and REST API Protocol

■ Using Node-RED, send data to ARTIK Cloud via REST protocol

- Step 4 : Configure node setting.
  - For the "http request" node, make its method POST, and set its URL as : http://api.artik.cloud/v1.1/messages
  - Edit function node as following codes.



Name: SendMessage

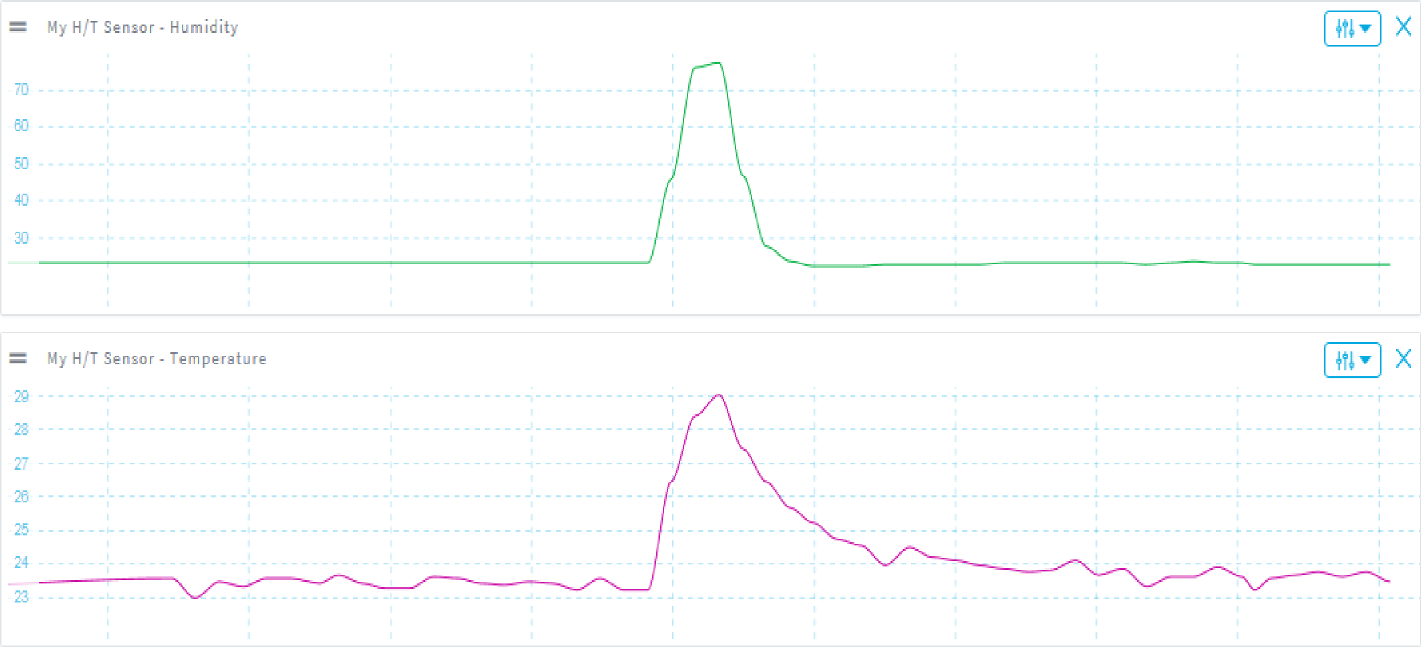
Function

```
var tem = msg.payload.temperature;
var hum = msg.payload.humidity;

msg.headers = {
    "Content-Type": "application/json",
    "Authorization": "Bearer device token
}

msg.payload = {
    "sdid": "device id",
    "type": "message",
    "data": {
        "temperature": tem,
        "humidity": hum
    }
}

return msg;
```

ARTIK Cloud and REST API Protocol

Using Node-RED, send data to ARTIK Cloud via REST protocol

- Step 5 : Click "Deploy" and go back to My ARTIK Cloud CHARTS.



ARTIK Cloud and MQTT Protocol

■ MQTT Protocol

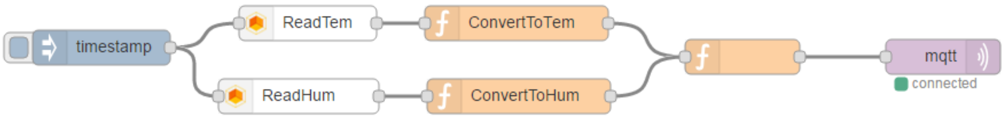
- MQTT is a lightweight messaging protocol.
- It is suitable for IoT, since it is bandwidth-efficient and uses little battery power.
- ARTIK Cloud devices can publish a data-only message to ARTIK Cloud or subscribe to receive Actions from ARTIK Cloud.

MQTT Components	Required Value
Security	SSL
Broker URL	api.artik.cloud
Broker port	8883
Username	Device ID
Password	Device token
Publish path (MQTT topic)	/v1.1/messages/<deviceID>
Subscription path (MQTT topic)	/v1.1/actions/<deviceID>
Error path (MQTT topic)	/v1.1/errors/<deviceID>

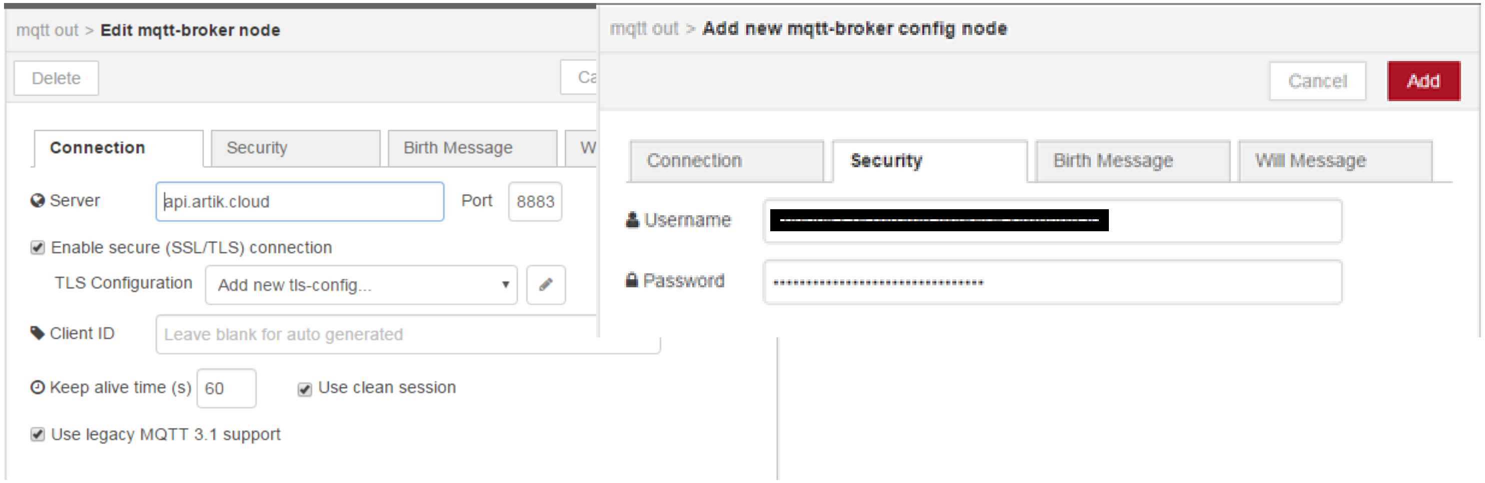
ARTIK Cloud and MQTT Protocol

■ Using Node-RED, send data to ARTIK Cloud via MQTT protocol

- Step 1 : Wire up node in Node-RED.
  - In previous flow, delete http request node and debug node, then add "mqtt" output node.



- Step 2 : Set up "mqtt" node.
  - Set the Server as api.artik.cloud on port 8883.
  - Under Security, set Username to Device ID and Password to Device token.
  - Check "Enable secure (SSL/TLS) connection" and "Verify server certificate".



mqtt out > **Edit mqtt-broker node**

Server: api.artik.cloud Port: 8883

☑ Enable secure (SSL/TLS) connection

TLS Configuration: Add new tls-config...

Client ID: Leave blank for auto generated

Keep alive time (s): 60 ☑ Use clean session

☑ Use legacy MQTT 3.1 support

mqtt out > **Add new mqtt-broker config node**

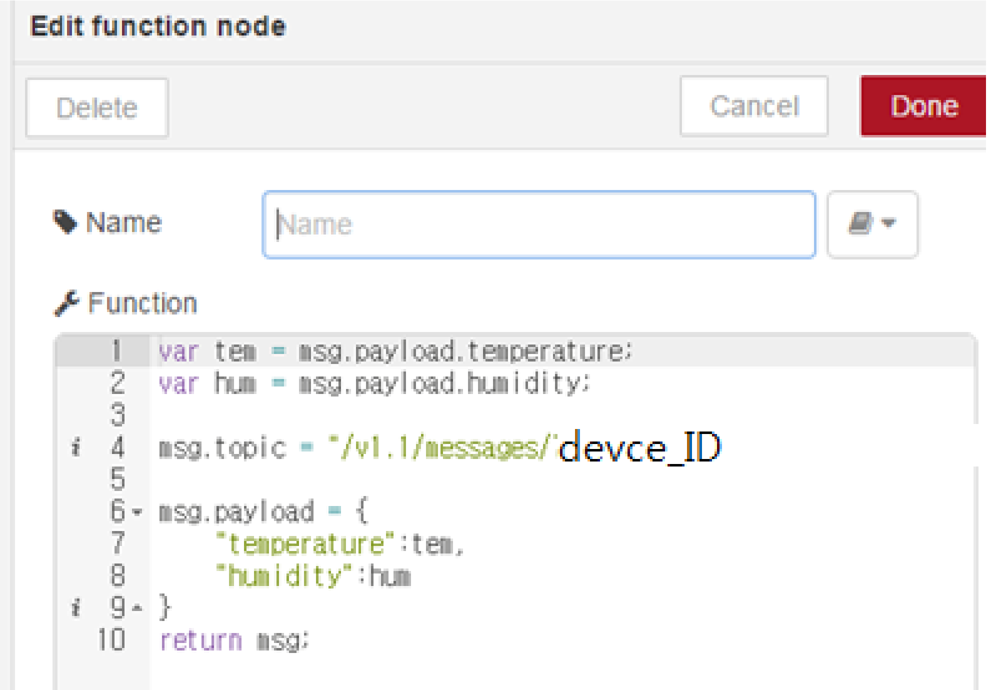
Username: [redacted]

Password: ••••••••••••••••••••••••••••••••

ARTIK Cloud and MQTT Protocol

■ Using Node-RED, send data to ARTIK Cloud via MQTT protocol

- Step 3 : Edit "function" node
  - You don’t need msg.headers since MQTT is a lightweight protocol.
  - Add a topic as shown with your Device ID inserted.



Name: Name

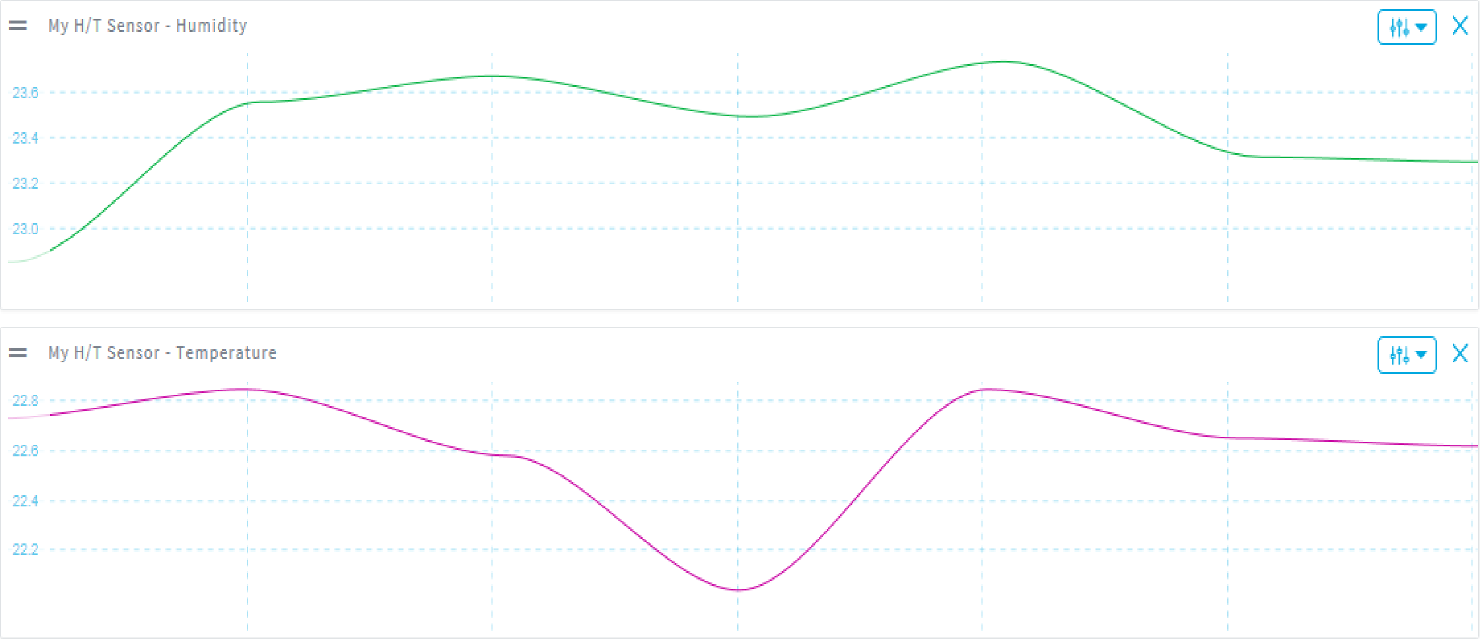
Function

```
var tem = msg.payload.temperature;
var hum = msg.payload.humidity;

msg.topic = "/v1.1/messages/"devce_ID

msg.payload = {
    "temperature":tem,
    "humidity":hum
}
return msg;
```

ARTIK Cloud and MQTT Protocol

■ Using Node-RED, send data to ARTIK Cloud via MQTT protocol

- Step 4 : Click "Deploy" and go to My ARTIK Cloud CHARTS



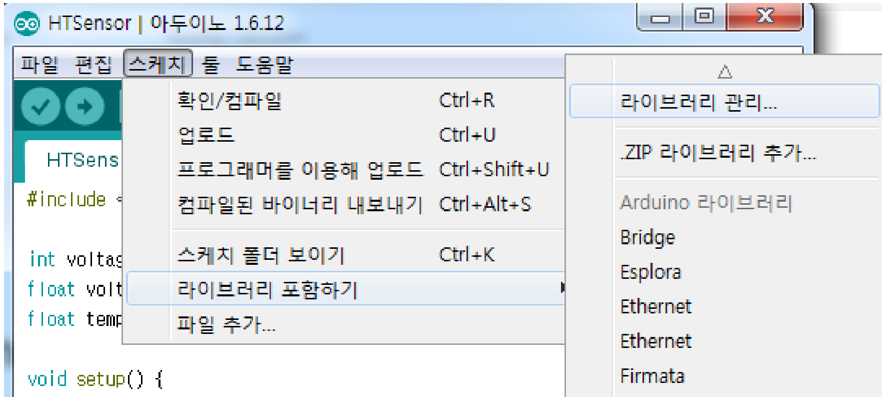
Ⅱ. ARTIK 기술 교육

10. ARTIK Cloud (3)

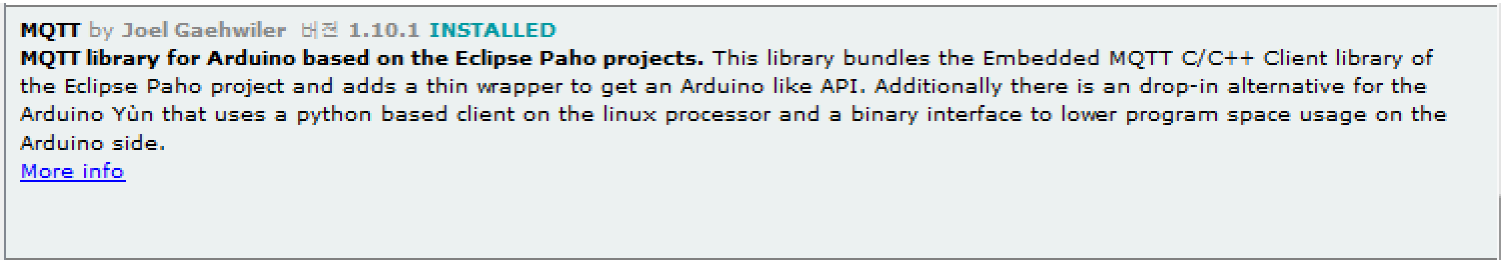
MQTT and Arduino IDE

Install Arduino IDE libraries

- Step 1 : Select [Sketch]>[Manage Libraries]



- Step 2 : Install MQTT library
  - Add #include <MQTTClient.h> to your sketch.

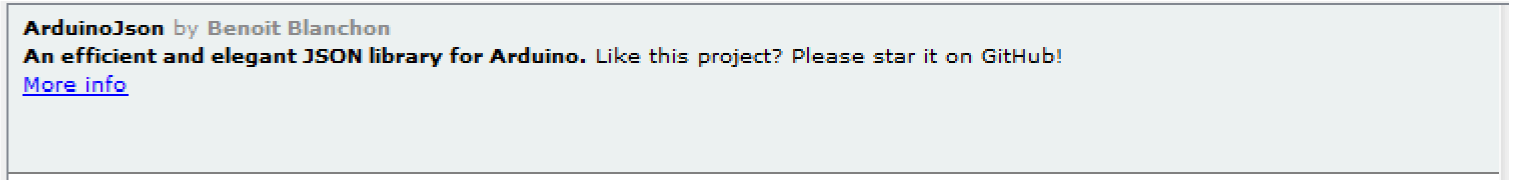


MQTT by **Joel Gaehwiler** 버전 **1.10.1** **INSTALLED**
MQTT library for Arduino based on the Eclipse Paho projects. This library bundles the Embedded MQTT C/C++ Client library of the Eclipse Paho project and adds a thin wrapper to get an Arduino like API. Additionally there is an drop-in alternative for the Arduino Yùn that uses a python based client on the linux processor and a binary interface to lower program space usage on the Arduino side.
More info

MQTT and Arduino IDE

Install Arduino IDE libraries

- Step 3 : Install the ArduinoJSON library
  - Add #include <ArduinoJson.h> to your sketch

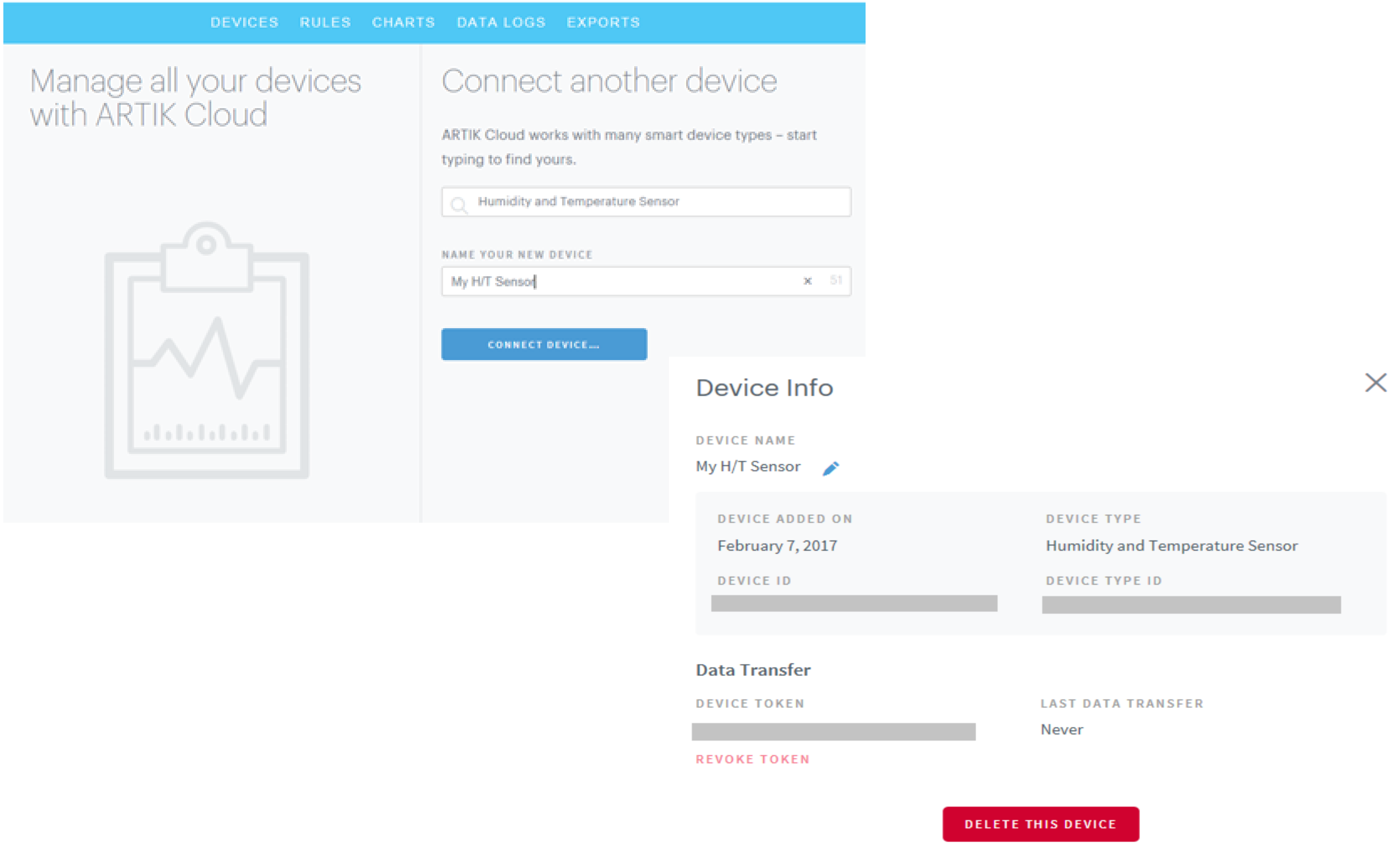


ArduinoJson by Benoit Blanchon
An efficient and elegant JSON library for Arduino. Like this project? Please star it on GitHub!
More info

- Step 4 : Include WiFiSSLClient library
  - If you installed Arduino IDE 1.6.9 or later, and update to ARTIK board version 0.3.6 or later, WiFiSSLClient library is included.

MQTT and Arduino IDE

■ Connect a new device and get a device token



Manage all your devices with ARTIK Cloud

Connect another device

ARTIK Cloud works with many smart device types – start typing to find yours.

Humidity and Temperature Sensor

NAME YOUR NEW DEVICE

My H/T Sensor

Device Info

DEVICE NAME
My H/T Sensor

DEVICE ADDED ON
February 7, 2017

DEVICE TYPE
Humidity and Temperature Sensor

DEVICE ID
[redacted]

DEVICE TYPE ID
[redacted]

Data Transfer

DEVICE TOKEN
[redacted]

LAST DATA TRANSFER
Never

MQTT Example (1)

■ Basic MQTT code with Arduino IDE

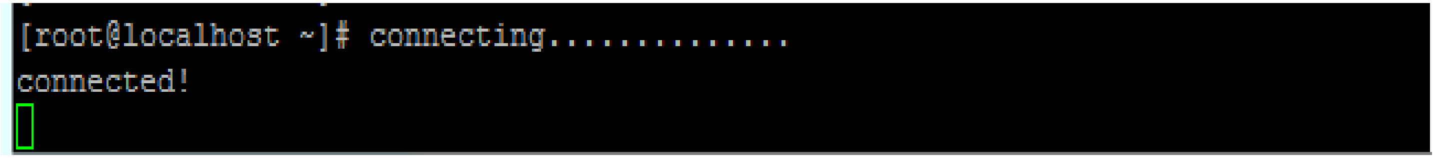
```
#include <WiFi.h>
#include <MQTTClient.h>
#include <ArduinoJson.h>
#include <DebugSerial.h>

// MQTT Parameters
char mqttCloudServer[] = "api.artik.cloud";
int mqttCloudPort = 8883;
char mqttCloudClientName[] = "My H/T Sensor"; // or
whatever you prefer
char mqttCloudUsername[] = "_device_ID_";
char mqttCloudPassword[] = "_device_token_";
char mqttCloudTopic[] = "/v1.1/messages/_device_ID_";
double temperature, humidity;
char buf[100] = "₩{₩"humidity₩":19.1,
₩"temperature₩":19.3 ₩}";

WiFiSSLClient ipCloudStack;
MQTTClient mqttCloudClient;

void setup() {
  DebugSerial.begin(9600);
  mqttCloudClient.begin(mqttCloudServer, mqttCloudPort,
ipCloudStack);
  connect();
}
```

```
void connect() {
  DebugSerial.print("connecting...");
  while (!mqttCloudClient.connect(mqttCloudClientName,
mqttCloudUsername, mqttCloudPassword)) {
    DebugSerial.print(".");
  }

  DebugSerial.println("₩nconnected!");
}

void loop() {
  mqttCloudClient.loop();

  if(!mqttCloudClient.connected()) {
    connect();
  }

  mqttCloudClient.publish(mqttCloudTopic, buf);
  exit(0);
}

void messageReceived(String topic, String payload, char *
bytes, unsigned int length) {
}
```

MQTT Example (1)

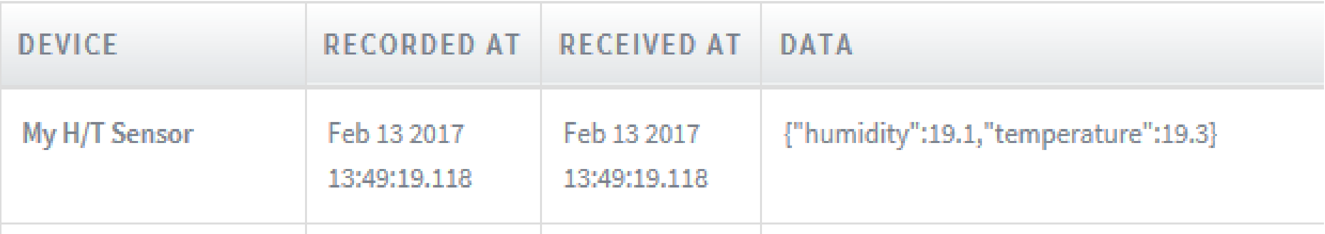
Execution Result

- Enter [Ctrl]+[u] or use menu to upload program to ARTIK.



```
[root@localhost ~]# connecting..............
connected!
```

- Go to My ARTIK Cloud DATA LOG, and you should see message received from ARTIK.



DEVICE	RECORDED AT	RECEIVED AT	DATA
My H/T Sensor	Feb 13 2017 13:49:19.118	Feb 13 2017 13:49:19.118	{"humidity":19.1,"temperature":19.3}

MQTT Example (2)

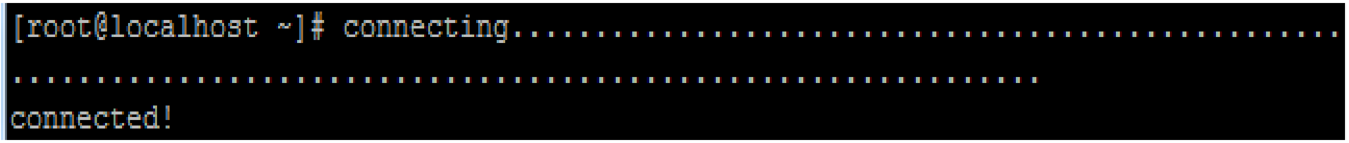
■ MQTT code automating the JSON formatting

```
#include <WiFi.h>
#include <MQTTClient.h>
#include <ArduinoJson.h>
#include <DebugSerial.h>

// MQTT Parameters
char mqttCloudServer[] = "api.artik.cloud";
int mqttCloudPort = 8883;
char mqttCloudClientName[] = "My H/T Sensor"; // or whatever
you prefer
char mqttCloudUsername[] = "_device_ID_";
char mqttCloudPassword[] = "_device_Token_";
char mqttCloudTopic[] = "/v1.1/messages/_device_ID_ ";
double temperature, humidity;
char buf[100];

WiFiSSLClient ipCloudStack;
MQTTClient mqttCloudClient;

void setup() {
  DebugSerial.begin(9600);
  mqttCloudClient.begin(mqttCloudServer, mqttCloudPort,
ipCloudStack);
  connect();
}

void connect() {
  DebugSerial.print("connecting...");
  while (!mqttCloudClient.connect(mqttCloudClientName,
mqttCloudUsername, mqttCloudPassword)) {
    DebugSerial.print(".");
  }
  DebugSerial.println("₩nconnected!");
}

void loop() {
  mqttCloudClient.loop();

  if(!mqttCloudClient.connected()) {
    connect();
  }

  loadBuffer(24.5, 29.5);
  mqttCloudClient.publish(mqttCloudTopic, buf);
  exit(0);
}

void messageReceived(String topic, String payload, char * bytes,
unsigned int length) {
}

int loadBuffer(double temp, double hum) {
  StaticJsonBuffer<200> jsonBuffer;

  JsonObject& dataPair = jsonBuffer.createObject();

  dataPair["temperature"] = temp;
  dataPair["humidity"] = hum;

  dataPair.printTo(buf, sizeof(buf));
}
```

MQTT Example (2)

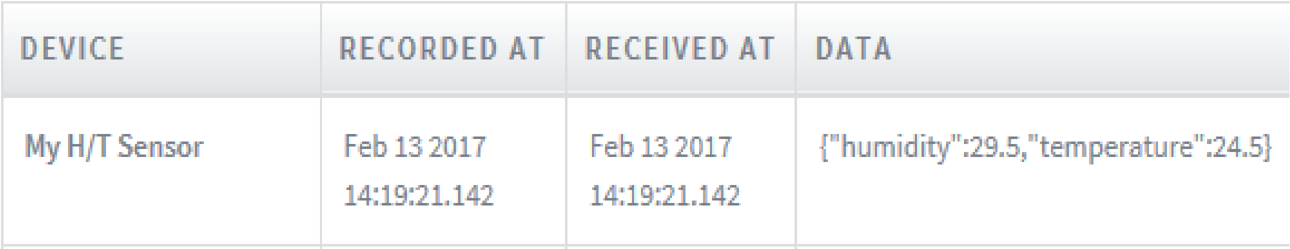
Execution Result

- Enter [Ctrl]+[u] or use menu to upload program to ARTIK.



```
[root@localhost ~]# connecting..................................................
.........................................................
connected!
```

- Go to My ARTIK Cloud DATA LOG, and you should see message received from ARTIK.



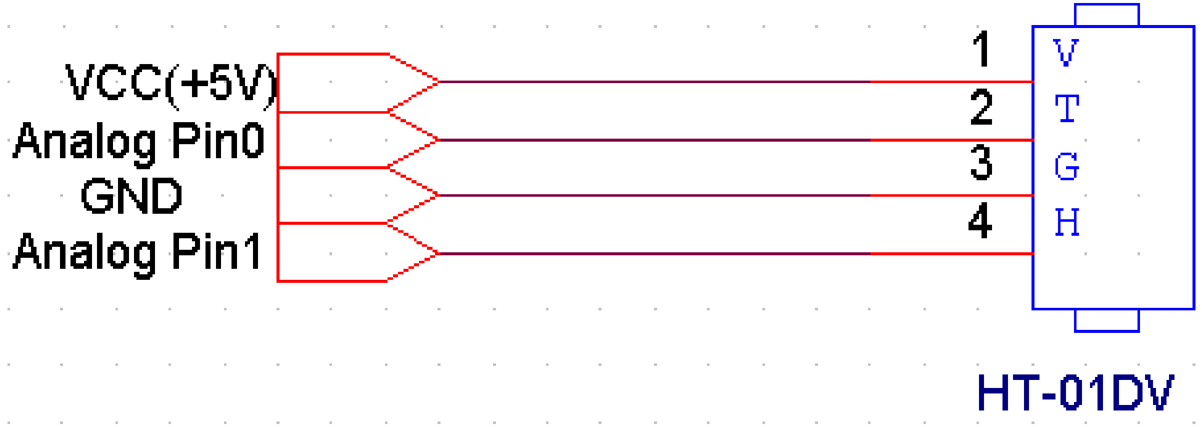
DEVICE	RECORDED AT	RECEIVED AT	DATA
My H/T Sensor	Feb 13 2017 14:19:21.142	Feb 13 2017 14:19:21.142	{"humidity":29.5,"temperature":24.5}

Send Data to ARTIK Cloud by MQTT

Required Hardware

- ARTIK 5 developer kit
- Humidity and Temperature Sensor (ETH-01DV)
- Breadboard
- Connector wires

Circuit Configuration



Send Data to ARTIK Cloud by MQTT

Source Code (1)

```
#include <WiFi.h>
#include <MQTTClient.h>
#include <ArduinoJson.h>
#include <DebugSerial.h>

// MQTT Parameters
char mqttCloudServer[] = "api.artik.cloud";
int mqttCloudPort = 8883;
char mqttCloudClientName[] = "My H/T Sensor"; // or
whatever you prefer
char mqttCloudUsername[] = "_device_ID_";
char mqttCloudPassword[] = "_device_Token_";
char mqttCloudTopic[] = "/v1.1/messages/_device_ID_ ";
double temperature, humidity;
char buf[100];

WiFiSSLClient ipCloudStack;
MQTTClient mqttCloudClient;
```

```
// Sensor Parameters
double temperature, humidity;
int voltage_raw0, voltage_raw1;
double voltage0, voltage1;

void setup() {
  DebugSerial.begin(9600);
mqttCloudClient.begin(mqttCloudServer, mqttCloudPort,
ipCloudStack);

  connect();
}

void loop() {
  mqttCloudClient.loop();

  if(!mqttCloudClient.connected()) {
    connect();
  }
```

Send Data to ARTIK Cloud by MQTT

Source Code (2)

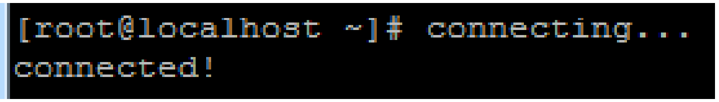
```
  readSensor();
  loadBuffer(temperature, humidity);
  mqttCloudClient.publish(mqttCloudTopic, buf);
  delay(5000);
}

void connect() {
  DebugSerial.print("connecting...");
  while (!mqttCloudClient.connect(mqttCloudClientName,
mqttCloudUsername, mqttCloudPassword)) {
    DebugSerial.print(".");
  }

  DebugSerial.println("₩nconnected!");
}

void messageReceived(String topic, String payload, char *
bytes, unsigned int length) {
}
```

```
void loadBuffer(double temp, double hum) {
  StaticJsonBuffer<200> jsonBuffer;

  JsonObject& dataPair = jsonBuffer.createObject();

  dataPair["temperature"] = temp;
  dataPair["humidity"] = hum;

  dataPair.printTo(buf, sizeof(buf));
}

void readSensor() {
  voltage_raw0 = analogRead(0);
  voltage_raw1 = analogRead(1);

  voltage0 = voltage_raw0*0.439453125*2;
  voltage1 = voltage_raw1*0.439453125*2;

  temperature = -66.875 + 217.75*(voltage0/5000);
  humidity = -12.5 + 125*(voltage1/5000);
}
```

Send Data to ARTIK Cloud by MQTT

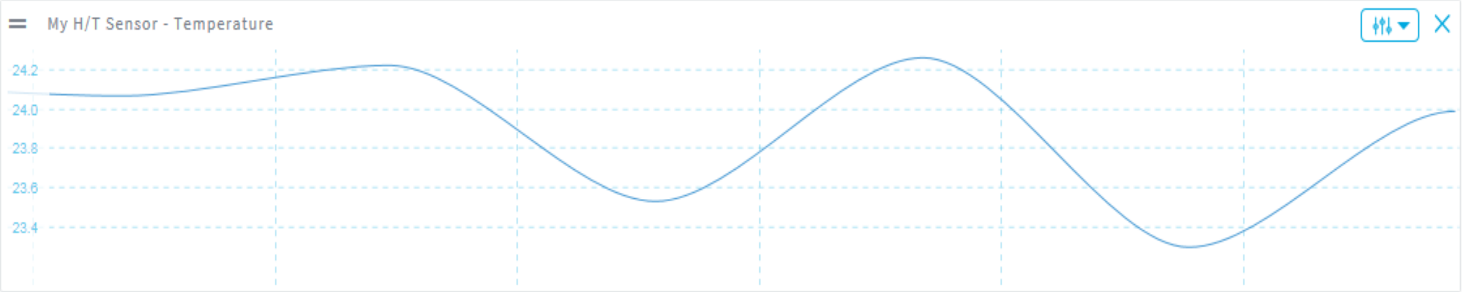
Execution Result

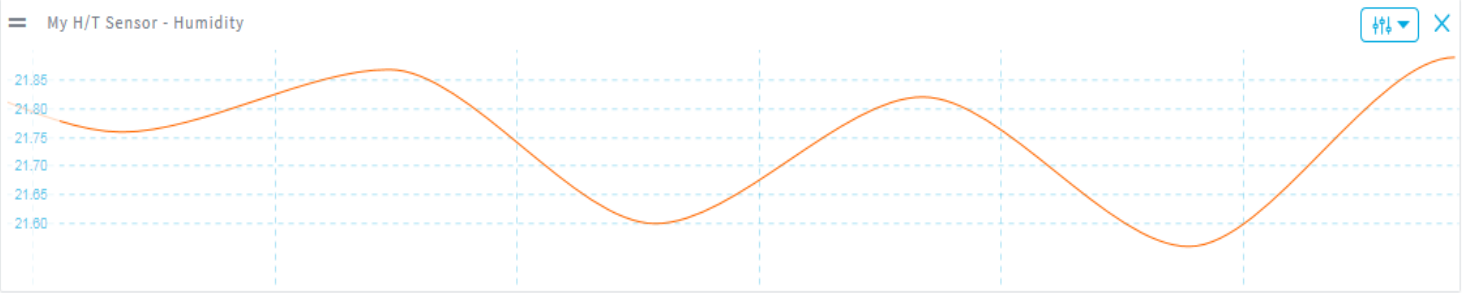
- Enter [Ctrl]+[u] or use menu to upload program to ARTIK.



```
[root@localhost ~]# connecting...
connected!
```

- Go to My ARTIK Cloud CHART, and you can see data graph.



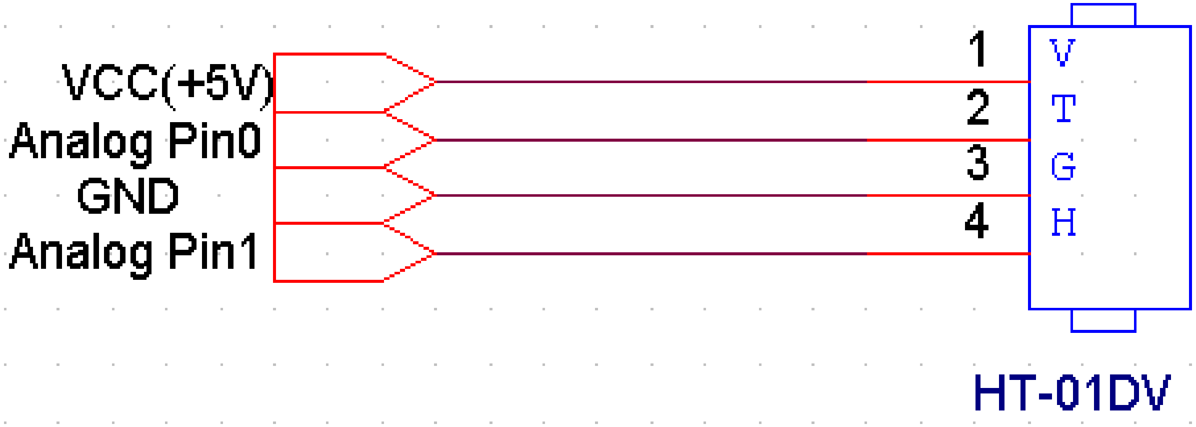


Receive Action from ARTIK Cloud

Required Hardware

- ARTIK 5 developer kit
- Humidity and Temperature Sensor (ETH-01DV)
- Breadboard
- Connector wires

Circuit Configuration

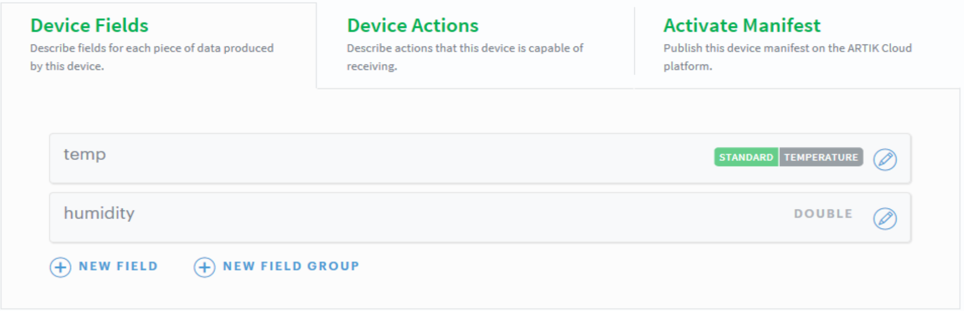


Receive Action from ARTIK Cloud

ARTIK Cloud setting

- Step 1 : On ARTIK Cloud Developer site, Create a new device type "temperature control and humidity regulator"

- Step 2 : Set up Manifest
  - Create two Device Fields
  - Create Device Actions



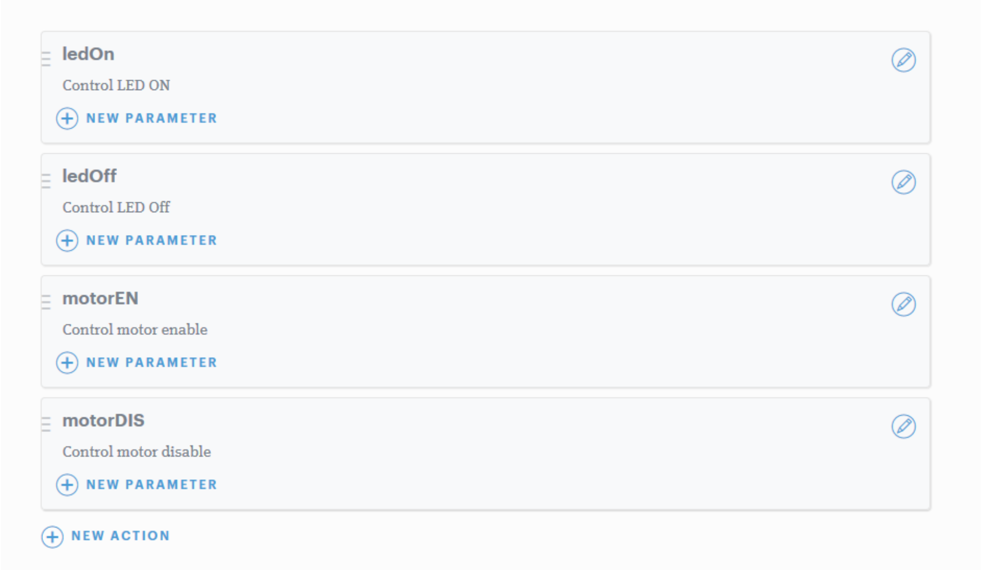
Device Fields
Describe fields for each piece of data produced by this device.

Device Actions
Describe actions that this device is capable of receiving.

Activate Manifest
Publish this device manifest on the ARTIK Cloud platform.

temp — STANDARD TEMPERATURE

humidity — DOUBLE



ledOn
Control LED ON

ledOff
Control LED Off

motorEN
Control motor enable

motorDIS
Control motor disable

Receive Action from ARTIK Cloud

■ ARTIK Cloud setting

- Step 3 : Go back to My ARTIK Cloud, write Rules

✓ IF Temperature control and humidity regulator temp is less than 25 THEN send to Temperature control and humidity regulator the action ledOn

✓ IF Temperature control and humidity regulator temp is more than or equal to 25 THEN send to Temperature control and humidity regulator the action ledOff

✓ IF Temperature control and humidity regulator humidity is less than 50 THEN send to Temperature control and humidity regulator the action motorDIS

✓ IF Temperature control and humidity regulator humidity is more than or equal to 50 THEN send to Temperature control and humidity regulator the action motorEN

Receive Action from ARTIK Cloud

■ Source Code (1)

```
#include <WiFi.h>
#include <MQTTClient.h>
#include <ArduinoJson.h>
#include <DebugSerial.h>

// MQTT Parameters
char mqttCloudServer[] = "api.artik.cloud";
int mqttCloudPort = 8883;
char mqttCloudClientName[] = "My H/T Sensor"; // or
whatever you prefer
char mqttCloudUsername[] = "_device_ID_";
char mqttCloudPassword[] = "_device_Token_";
char mqttCloudTopic1[] = "/v1.1/messages/_device_ID_ ";
char mqttCloudTopic2[] = "/v1.1/actions/_device_ID_ ";
double temperature, humidity;
char buf[100];

WiFiSSLClient ipCloudStack;
MQTTClient mqttCloudClient;
```

```
// Sensor Parameters
double temperature, humidity;
int voltage_raw0, voltage_raw1;
double voltage0, voltage1;

void setup() {
  DebugSerial.begin(9600);
  mqttCloudClient.begin(mqttCloudServer, mqttCloudPort,
ipCloudStack);

  connect();
}

void loop() {
  mqttCloudClient.loop();

  if(!mqttCloudClient.connected()) {
    connect();
  }

  readSensor();
  loadBuffer(temperature, humidity);
  mqttCloudClient.publish(mqttCloudTopic1, buf);
  mqttCloudClient.subscribe(mqttCloudTopic2);
```

Receive Action from ARTIK Cloud

■ Source Code (2)

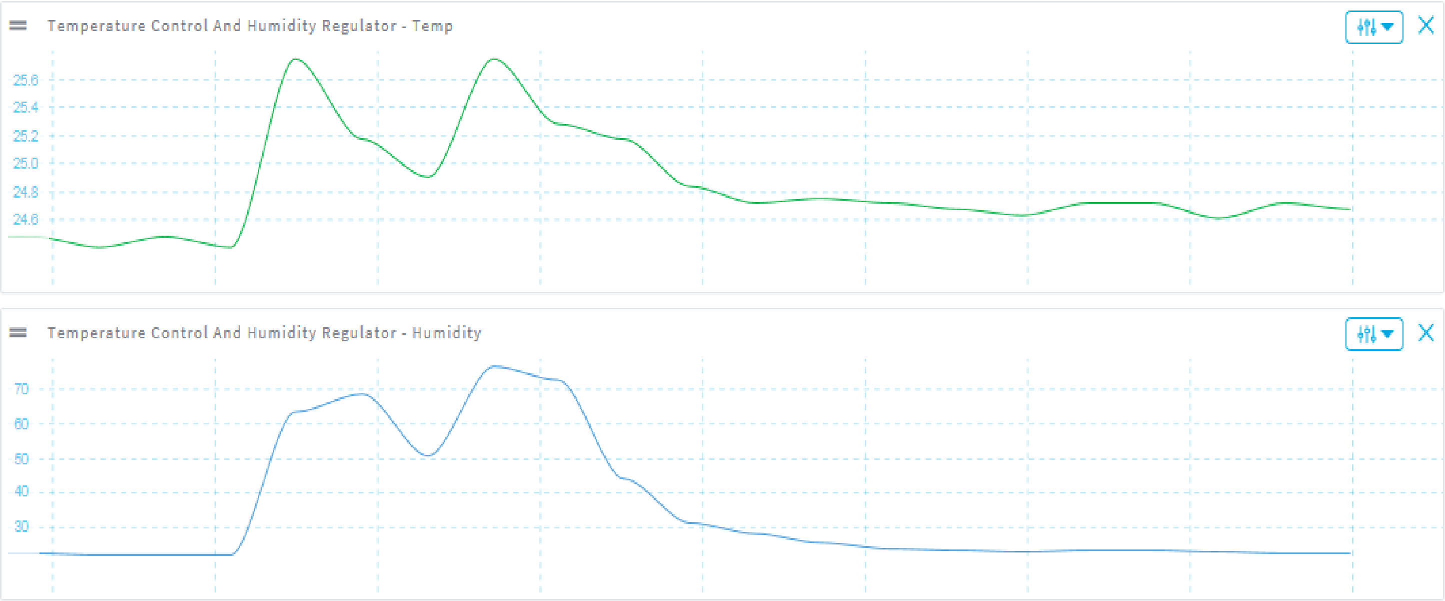
```
 delay(5000);
}

void connect() {
  DebugSerial.print("connecting...");
  while (!mqttCloudClient.connect(mqttCloudClientName,
mqttCloudUsername, mqttCloudPassword)) {
    DebugSerial.print(".");
  }

  DebugSerial.println("₩nconnected!");
}

void messageReceived(String topic, String payload, char *
bytes, unsigned int length) {
  DebugSerial.print("incoming");
  DebugSerial.print(topic);
  DebugSerial.print("–");
  DebugSerial.print(payload);
  DebugSerial.println();
}
```

```
void loadBuffer(double temp, double hum) {
  StaticJsonBuffer<200> jsonBuffer;

  JsonObject& dataPair = jsonBuffer.createObject();

  dataPair["temp"] = temp;
  dataPair["humidity"] = hum;

  dataPair.printTo(buf, sizeof(buf));
}

void readSensor() {
  voltage_raw0 = analogRead(0);
  voltage_raw1 = analogRead(1);

  voltage0 = voltage_raw0*0.439453125*2;
  voltage1 = voltage_raw1*0.439453125*2;

  temperature = −66.875 + 217.75*(voltage0/5000);
  humidity = −12.5 + 125*(voltage1/5000);
}
```

Receive Action from ARTIK Cloud

Execution Result

- Enter [Ctrl]+[u] or use menu to upload program to ARTIK.
- Go to My ARTIK Cloud CHART, and you can see data graph.



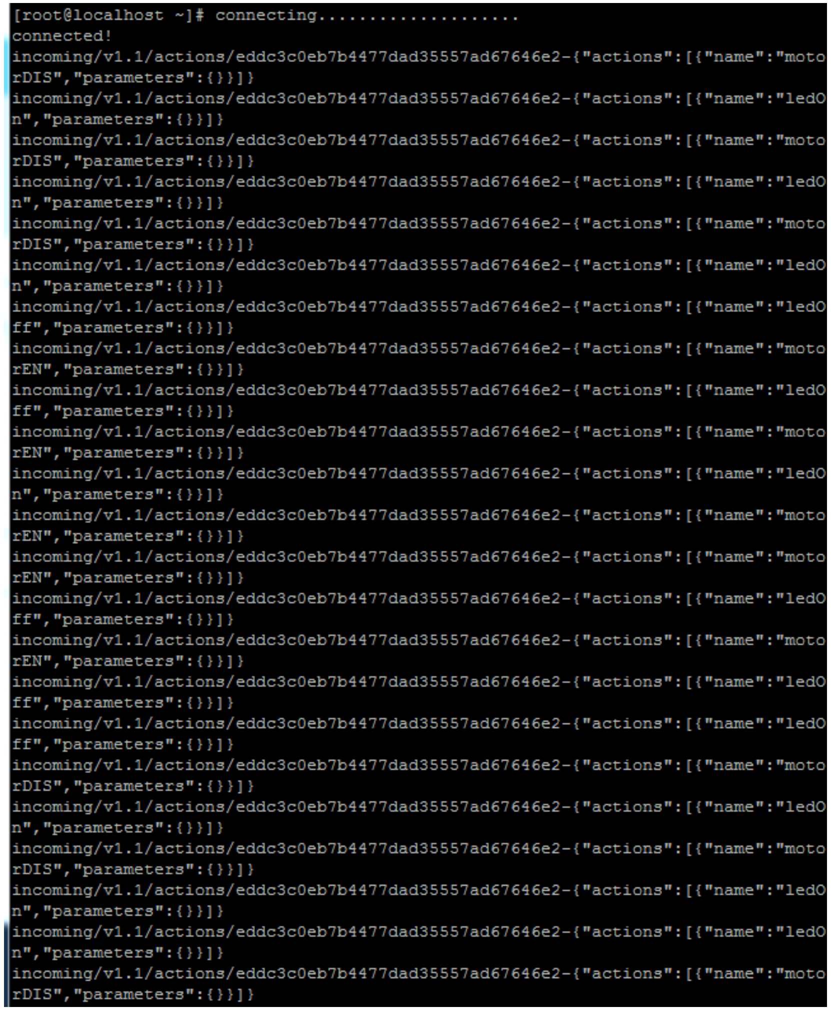
Temperature Control And Humidity Regulator - Temp

Temperature Control And Humidity Regulator - Humidity

Receive Action from ARTIK Cloud

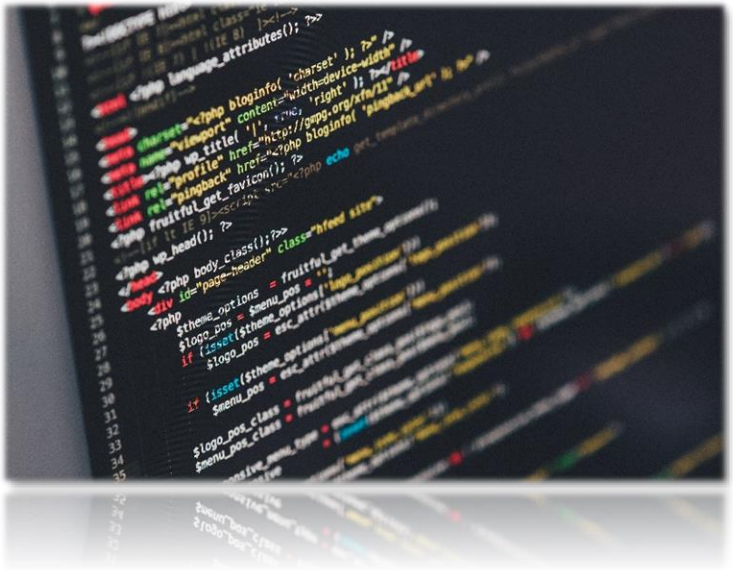
Execution Result

- ARTIK Cloud will use "temperature control and humidity regulator" input messages to generate Action messages.
- You can use Action messages to control LED or motor.



```
[root@localhost ~]# connecting....................
connected!
incoming/v1.1/actions/eddc3c0eb7b4477dad35557ad67646e2-{"actions":[{"name":"moto
rDIS","parameters":{}}]}
incoming/v1.1/actions/eddc3c0eb7b4477dad35557ad67646e2-{"actions":[{"name":"ledO
n","parameters":{}}]}
incoming/v1.1/actions/eddc3c0eb7b4477dad35557ad67646e2-{"actions":[{"name":"moto
rDIS","parameters":{}}]}
incoming/v1.1/actions/eddc3c0eb7b4477dad35557ad67646e2-{"actions":[{"name":"ledO
n","parameters":{}}]}
incoming/v1.1/actions/eddc3c0eb7b4477dad35557ad67646e2-{"actions":[{"name":"moto
rDIS","parameters":{}}]}
incoming/v1.1/actions/eddc3c0eb7b4477dad35557ad67646e2-{"actions":[{"name":"ledO
n","parameters":{}}]}
incoming/v1.1/actions/eddc3c0eb7b4477dad35557ad67646e2-{"actions":[{"name":"ledO
ff","parameters":{}}]}
incoming/v1.1/actions/eddc3c0eb7b4477dad35557ad67646e2-{"actions":[{"name":"moto
rEN","parameters":{}}]}
incoming/v1.1/actions/eddc3c0eb7b4477dad35557ad67646e2-{"actions":[{"name":"ledO
ff","parameters":{}}]}
incoming/v1.1/actions/eddc3c0eb7b4477dad35557ad67646e2-{"actions":[{"name":"moto
rEN","parameters":{}}]}
incoming/v1.1/actions/eddc3c0eb7b4477dad35557ad67646e2-{"actions":[{"name":"ledO
n","parameters":{}}]}
incoming/v1.1/actions/eddc3c0eb7b4477dad35557ad67646e2-{"actions":[{"name":"moto
rEN","parameters":{}}]}
incoming/v1.1/actions/eddc3c0eb7b4477dad35557ad67646e2-{"actions":[{"name":"moto
rEN","parameters":{}}]}
incoming/v1.1/actions/eddc3c0eb7b4477dad35557ad67646e2-{"actions":[{"name":"ledO
ff","parameters":{}}]}
incoming/v1.1/actions/eddc3c0eb7b4477dad35557ad67646e2-{"actions":[{"name":"moto
rEN","parameters":{}}]}
incoming/v1.1/actions/eddc3c0eb7b4477dad35557ad67646e2-{"actions":[{"name":"ledO
ff","parameters":{}}]}
incoming/v1.1/actions/eddc3c0eb7b4477dad35557ad67646e2-{"actions":[{"name":"ledO
ff","parameters":{}}]}
incoming/v1.1/actions/eddc3c0eb7b4477dad35557ad67646e2-{"actions":[{"name":"moto
rDIS","parameters":{}}]}
incoming/v1.1/actions/eddc3c0eb7b4477dad35557ad67646e2-{"actions":[{"name":"ledO
n","parameters":{}}]}
incoming/v1.1/actions/eddc3c0eb7b4477dad35557ad67646e2-{"actions":[{"name":"moto
rDIS","parameters":{}}]}
incoming/v1.1/actions/eddc3c0eb7b4477dad35557ad67646e2-{"actions":[{"name":"ledO
n","parameters":{}}]}
incoming/v1.1/actions/eddc3c0eb7b4477dad35557ad67646e2-{"actions":[{"name":"ledO
n","parameters":{}}]}
incoming/v1.1/actions/eddc3c0eb7b4477dad35557ad67646e2-{"actions":[{"name":"moto
rDIS","parameters":{}}]}
```

III. ARTIK 유닛 프로젝트 사례



유닛 프로젝트 종합

로봇 분야

- 밸런싱 로봇
- ✹ 볼핀타이어를 활용한 세그웨이
- 전기자전거
- 탐사용 RC카
- 손동작 인식 센서를 활용한 RC카
- 긴급제동 시스템
- StarWars BB-8 드로이드 설계

웨어러블 디바이스 분야

- 장애물 알림용 모자
- 거동이 불편한 사람을 위한 케어박스
- 독거노인을 위한 심장마비 감지 팔찌
- ✹ 움직임을 따라하는 로봇 손

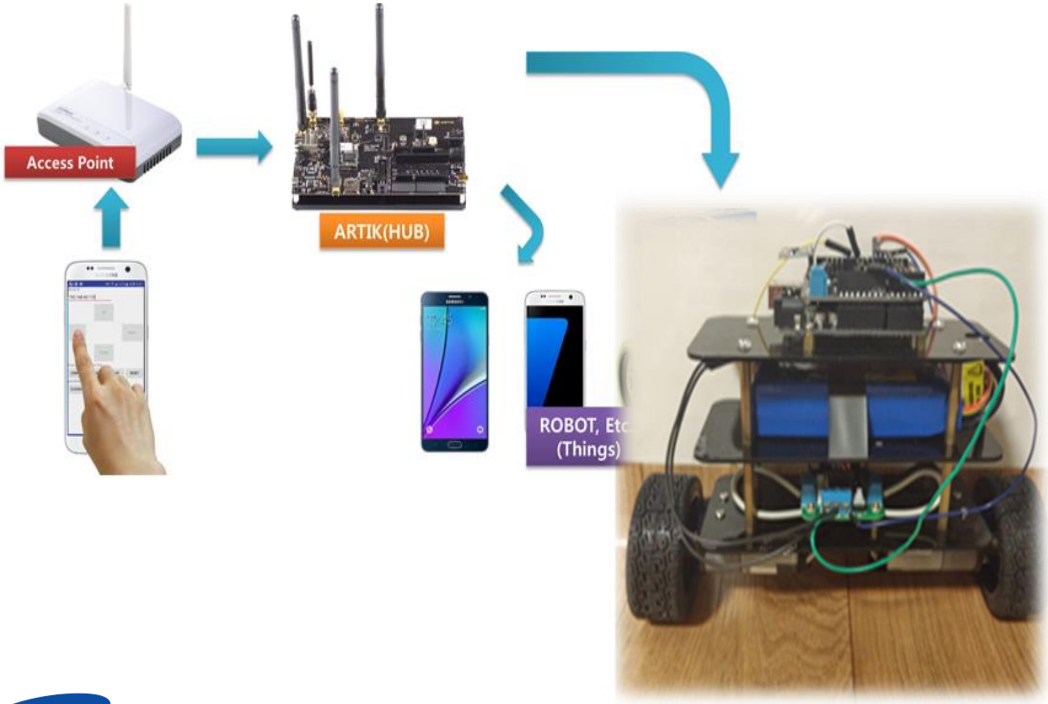
스마트 홈 분야

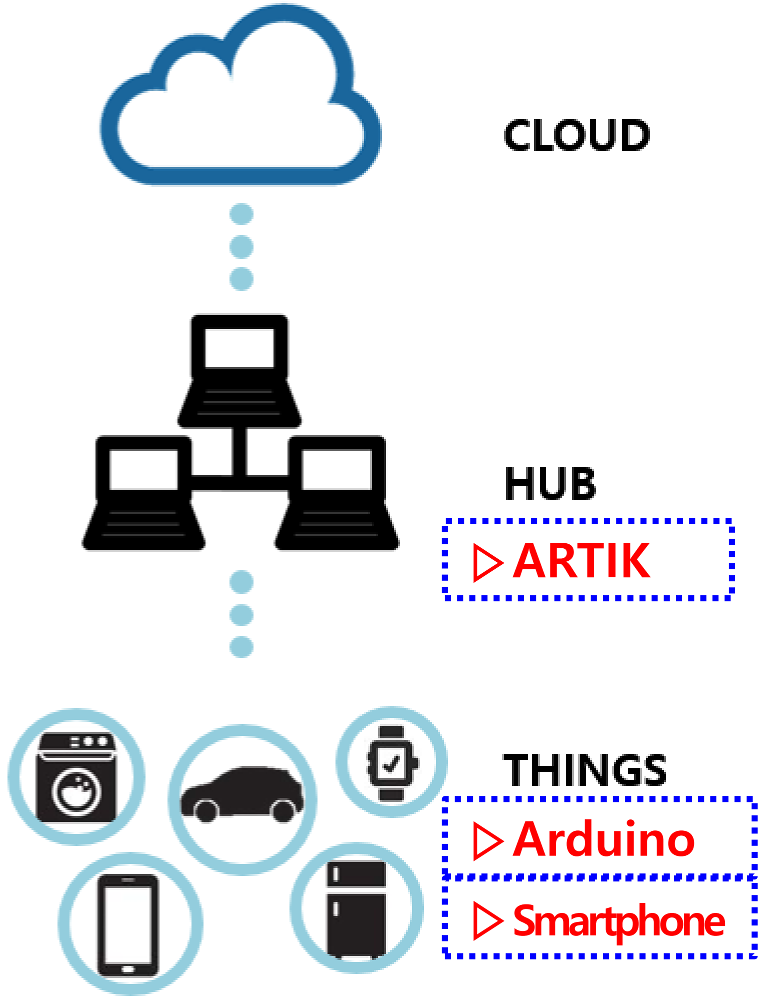
- 스마트 환기 시스템
- 화재대응 시스템
- 냉장고 IoT 시스템
- ✹ 스마트폰을 이용한 도어락
- 스마트폰을 이용한 무드등
- 클라우드를 활용한 원격 멀티탭
- 스마트 요람
- 스마트 욕조
- 스마트 화분

✹ 특허 출원 준비중

유닛 프로젝트 – 로봇분야

- 밸런싱 로봇
  - ✓2바퀴로 균형을 잡는 로봇
  - ✓자이로/가속도 센서를 이용
  - ✓로봇 본체(Thing)는 Arduino로 구현
  - ✓이동 명령 및 수행 임무는 스마트폰을 통해 ARTIK(Hub)를 거쳐 로봇 본체가 받음




CLOUD

HUB

▷ARTIK

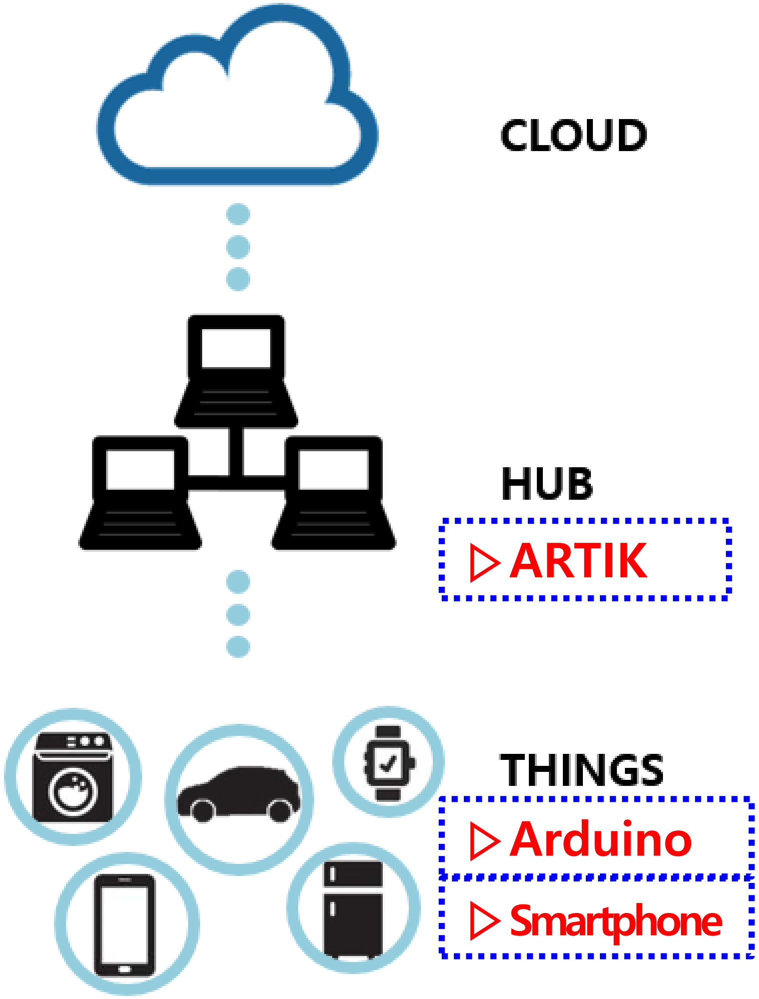
THINGS

▷Arduino

▷Smartphone

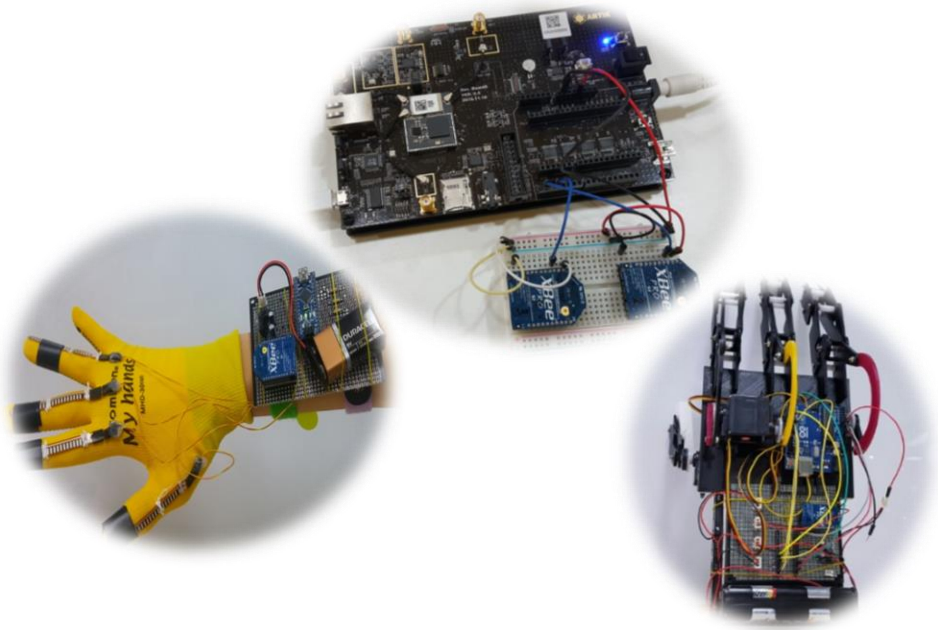
유닛 프로젝트 – 로봇분야

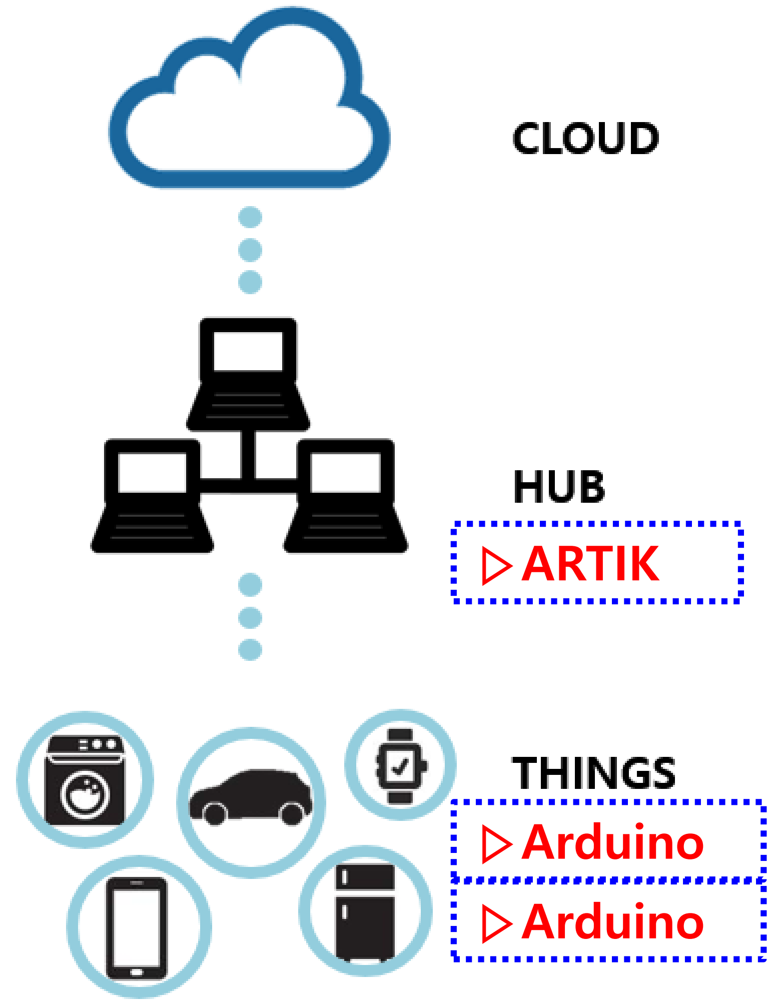
- 탐사용 RC카
  - ✓기존의 RC카를 응용
  - ✓차량 전복시 주행 가능
  - ✓주변 환경정보를 ARTIK(Hub)로 전송
  - ✓로봇 본체(Thing)는 Arduino로 구현



유닛 프로젝트 – 웨어러블 디바이스 분야

- 움직임을 따라하는 로봇 손
  - ✓확장 보드(Arduino)를 활용하여 서보모터제어
  - ✓휨 센서를 통한 사용자 모션 인식
  - ✓사용자 모션에 따른 서보모터 제어
  - ✓Arduino와 ARTIK 사이의 무선통신
    - ✓ ZigBee 모듈 활용





CLOUD

HUB

▷ARTIK

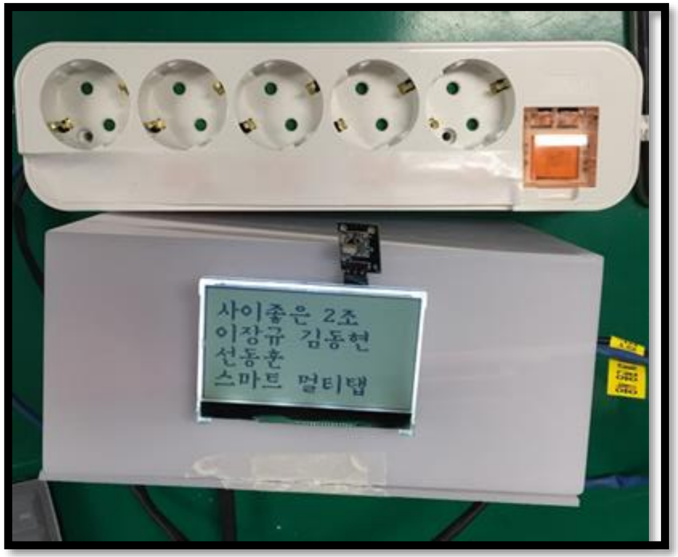
THINGS

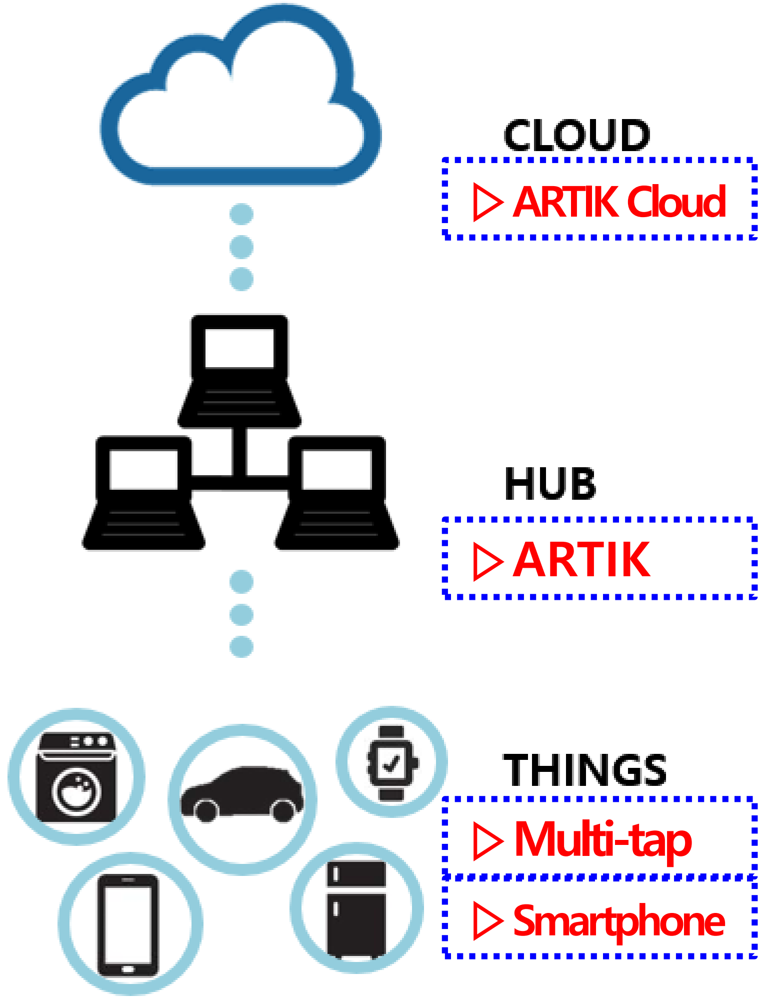
▷Arduino

▷Arduino

유닛 프로젝트 – 스마트홈 분야

- 스마트 멀티 탭
  - ✓스마트폰 어플리케이션을 통한 멀티탭 제어
  - ✓클라우드를 이용해 언제 어디서든 제어
    - – 2개의 팀이 ARTIK Cloud와 IoT Makers를 활용
  - ✓릴레이 소자를 이용해 각 구 마다 개별 제어





CLOUD
▷ ARTIK Cloud

HUB
▷ ARTIK

THINGS
▷ Multi-tap
▷ Smartphone